

# Руководство **FreeBSD**

Проект Документации **FreeBSD**

## Руководство FreeBSD

Проект Документации FreeBSD

Опубликовано Февраль 1999

Copyright © 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004 The FreeBSD Documentation Project

Copyright © 2000, 2001, 2002, 2003, 2004 Проект Русской Документации FreeBSD

Добро пожаловать в FreeBSD! Это Руководство охватывает процесс установки и ежедневного использования *FreeBSD 4.10-RELEASE* и *FreeBSD 5.3-RELEASE*. Это руководство находится в *процессе разработки* и являет собой результат работы множества людей. Многие из разделов до сих пор не существуют, а некоторые из существующих требуют обновления. Если вы заинтересованы в помощи этому проекту, отправьте письмо в Список рассылки Проекта Русской Документации FreeBSD <frdp@FreeBSD.org.ua>. Обновленная версия этого документа постоянно доступна с Основного Web сервера Проекта Русской Документации FreeBSD (<http://www.FreeBSD.org.ua>) и Основного Web сервера FreeBSD (<http://www.FreeBSD.org/ru/index.html>). Он также может быть загружен из Internet в одном из наиболее распространенных форматов с FTP сервера Проекта FreeBSD (<ftp://ftp.FreeBSD.org/pub/FreeBSD/doc/>) или с одного из многочисленных зеркал. Если вы предпочитаете иметь напечатанный (английский) вариант Руководства, то можете приобрести его на FreeBSD Mall (<http://www.freebsdmail.com/>). Вы также можете воспользоваться Поиском в Руководстве FreeBSD (<http://www.FreeBSD.org/ru/search/index.html>).

Распространение и использование исходных (SGML DocBook) и "скомпилированных" форм (SGML, HTML, PDF, PostScript, RTF и прочих) с модификацией или без оной, разрешены при соблюдении следующих соглашений:

1. Распространяемые копии исходного кода (SGML DocBook) должны сохранять вышеупомянутые объявления copyright, этот список положений и следующий отказ от ответственности в первых строках этого файла в неизменном виде.
2. Распространяемые копии скомпилированных форм (преобразованные в другие DTD, конвертированные в PDF, PostScript, RTF и другие форматы) должны повторять вышеупомянутые объявления copyright, этот список положений и следующий отказ от ответственности в документации и/или других материалах, поставляемых с дистрибуцией.

**Важно:** ЭТА ДОКУМЕНТАЦИЯ ПОСТАВЛЯЕТСЯ ПРОЕКТОМ ДОКУМЕНТАЦИИ FREEBSD "КАК ЕСТЬ" И ЛЮБЫЕ ЯВНЫЕ ИЛИ НЕЯВНЫЕ ГАРАНТИИ, ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ НЕЯВНЫМИ ГАРАНТИЯМИ, КОММЕРЧЕСКОЙ ЦЕННОСТИ И ПРИГОДНОСТИ ДЛЯ КОНКРЕТНОЙ ЦЕЛИ ОТРИЦАЮТСЯ. НИ ПРИ КАКИХ УСЛОВИЯХ ПРОЕКТ ДОКУМЕНТИРОВАНИЯ FREEBSD НЕ НЕСЕТ ОТВЕТСТВЕННОСТИ ЗА ЛЮБОЙ ПРЯМОЙ, КОСВЕННЫЙ, СЛУЧАЙНЫЙ, СПЕЦИАЛЬНЫЙ, ОБРАЗЦОВЫЙ ИЛИ ПОСЛЕДУЮЩИЙ УЩЕРБЫ (ВКЛЮЧАЯ, НО НЕ ОГРАНИЧИВАЯСЬ ПОСТАВКОЙ ТОВАРОВ ЗАМЕНЫ ИЛИ УСЛУГ; ПОТЕРЮ ДАННЫХ ИЛИ ИХ НЕПРАВИЛЬНУЮ ПЕРЕДАЧУ ИЛИ ПОТЕРИ; ПРИОСТАНОВЛЕНИЕ БИЗНЕСА), И ТЕМ НЕ МЕНЕЕ ВЫЗВАННЫЕ И В ЛЮБОЙ ТЕОРИИ ОТВЕТСТВЕННОСТИ, НЕЗАВИСИМО ОТ КОНТРАКТНОЙ, СТРОГОЙ ОТВЕТСТВЕННОСТИ, ИЛИ ПРАВОНАРУШЕНИИ (ВКЛЮЧАЯ ХАЛАТНОСТЬ ИЛИ ИНЫМ СПОСОБОМ), ВОЗНИКШЕМ ЛЮБЫМ ПУТЕМ ПРИ ИСПОЛЬЗОВАНИИ ЭТОЙ ДОКУМЕНТАЦИИ, ДАЖЕ ЕСЛИ БЫ БЫЛО СООБЩЕНО О ВОЗМОЖНОСТИ ТАКОГО УЩЕРБА.

FreeBSD это зарегистрированная торговая марка FreeBSD Foundation.

3Com и HomeConnect это зарегистрированные торговые марки 3Com Corporation.

3ware и Escalade это зарегистрированные торговые марки 3ware Inc.

ARM это зарегистрированная торговая марка ARM Limited.

Adaptec это зарегистрированная торговая марка Adaptec, Inc.

Adobe, Acrobat, Acrobat Reader и PostScript это или зарегистрированные торговые марки или торговые марки Adobe Systems Incorporated в Соединенных Штатах и/или других странах.

Apple, FireWire, Mac, Macintosh, Mac OS, Quicktime и TrueType это торговые марки Apple Computer, Inc., зарегистрированные в Соединенных Штатах и других странах.

Corel и WordPerfect это торговые марки или зарегистрированные торговые марки Corel Corporation и/или ее дочерних компаний в Канаде, Соединенных Штатах и/или других странах.

Sound Blaster это торговая марка Creative Technology Ltd. в Соединенных Штатах и/или других странах.

CVSup это зарегистрированная торговая марка John D. Polstra.

Heidelberg, Helvetica, Palatino и Times Roman это или зарегистрированные торговые марки или торговые марки Heidelberg Druckmaschinen AG в США и других странах.

IBM, AIX, EtherJet, Netfinity, OS/2, PowerPC, PS/2, S/390 и ThinkPad это торговые марки International Business Machines Corporation в Соединенных Штатах, других странах, или по всему миру.

IEEE, POSIX и 802 это зарегистрированные торговые марки Institute of Electrical and Electronics Engineers, Inc. в Соединенных Штатах.

Intel, Celeron, EtherExpress, i386, i486, Itanium, Pentium и Xeon это торговые марки или зарегистрированные торговые марки Intel Corporation или ее дочерних компаний в Соединенных Штатах и других странах.

Intuit и Quicken это зарегистрированные торговые марки и/или зарегистрированные сервис марки Intuit Inc., или одной из ее дочерних компаний в Соединенных Штатах и других странах.

Linux это зарегистрированная торговая марка Linus Torvalds.

LSI Logic, AcceleRAID, eXtremeRAID, MegaRAID и Mylex это торговые марки или зарегистрированные торговые марки LSI Logic Corp.

M-Systems и DiskOnChip это торговые марки или зарегистрированные торговые марки M-Systems Flash Disk Pioneers, Ltd.

Macromedia, Flash и Shockwave это торговые марки или зарегистрированные торговые марки Macromedia, Inc. в Соединенных Штатах и/или других странах.

Microsoft, FrontPage, IntelliMouse, MS-DOS, Outlook, Windows, Windows Media и Windows NT это или зарегистрированные торговые марки или торговые марки Microsoft Corporation в Соединенных Штатах и/или других странах.

Netscape и Netscape Navigator это зарегистрированные торговые марки Netscape Communications Corporation в США и других странах.

GateD и NextHop это зарегистрированные и незарегистрированные торговые марки NextHop в США и других странах.

Motif, OSF/1 и UNIX это зарегистрированные торговые марки, а IT DialTone и The Open Group это торговые марки Open Group в Соединенных Штатах и других странах.

Oracle это зарегистрированная торговая марка Oracle Corporation.

PowerQuest и PartitionMagic это зарегистрированные торговые марки PowerQuest Corporation в Соединенных Штатах и/или других странах.

RealNetworks, RealPlayer и RealAudio это зарегистрированные торговые марки RealNetworks, Inc.

Red Hat и RPM это торговые марки или зарегистрированные торговые марки Red Hat, Inc. в Соединенных Штатах и других странах.

SAP, R/3 и mySAP это торговые марки или зарегистрированные торговые марки SAP AG в Германии и нескольких других странах по всему миру.

Sun, Sun Microsystems, Java, Java Virtual Machine, JavaServer Pages, JDK, JSP, JVM, Netra, Solaris, StarOffice, Sun Blade, Sun Enterprise, Sun Fire, SunOS и Ultra это торговые марки или зарегистрированные торговые марки Sun Microsystems, Inc. в Соединенных Штатах и других странах.

Symantec и Ghost это зарегистрированные торговые марки Symantec Corporation в Соединенных Штатах и других странах.

MATLAB это зарегистрированная торговая марки MathWorks, Inc.

SpeedTouch это торговая марка Thomson.

U.S. Robotics и Sportster это зарегистрированные торговые марки U.S. Robotics Corporation.

VMware это торговая марка VMware, Inc.

Waterloo Maple и Maple это торговые марки или зарегистрированные торговые марки Waterloo Maple Inc.

Mathematica это зарегистрированная торговая марки Wolfram Research, Inc.

XFree86 это торговая марка XFree86 Project, Inc.

Ogg Vorbis и Xiph.Org это торговые марки Xiph.Org.

Многие из обозначений, используемые производителями и продавцами для обозначения своих продуктов, заявляются в качестве торговых марок. Когда такие обозначения появляются в этом документе, и Проекту FreeBSD известно о торговой марке, к обозначению добавляется знак "™" или "®".



# Содержание

Предисловие .....	i
I. В начале .....	i
1. Введение .....	1
1.1. Краткий обзор .....	1
1.2. Добро пожаловать во FreeBSD! .....	1
1.3. О Проекте FreeBSD .....	4
2. Установка FreeBSD .....	10
2.1. Краткий обзор .....	10
2.2. Перед установкой.....	10
2.3. Начало установки .....	16
2.4. Введение в Sysinstall.....	26
2.5. Выделение дискового пространства.....	31
2.6. Выбор устанавливаемых компонентов .....	43
2.7. Выбор источника для установки .....	46
2.8. Подтверждение установки.....	47
2.9. После установки.....	48
2.10. Поддерживаемое оборудование .....	90
2.11. Решение проблем .....	90
2.12. Расширенное руководство по установке .....	92
2.13. Подготовка собственного источника установки .....	93
3. Основы UNIX.....	100
3.1. Краткий обзор .....	100
3.2. Виртуальные консоли и терминалы.....	100
3.3. Права доступа.....	103
3.4. Структура каталогов .....	105
3.5. Организация дисков.....	108
3.6. Монтирование и размонтирование файловых систем .....	114
3.7. Процессы.....	117
3.8. Даемоны, сигналы, уничтожение процессов .....	119
3.9. Интерпретатор команд .....	121
3.10. Текстовые редакторы .....	123
3.11. Устройства и файлы устройств.....	123
3.12. Бинарные форматы .....	124
3.13. Дополнительная информация .....	126
4. Установка приложений: порты и пакеты .....	129
4.1. Обзор .....	129
4.2. Обзор установки программного обеспечения.....	129
4.3. Поиск нужного вам приложения .....	131
4.4. Использование системы пакетов.....	132
4.5. Использование Коллекции Портов .....	135
4.6. Действия после установки .....	142
4.7. Обработка нерабочих портов.....	143
5. X Window System.....	145
5.1. Обзор .....	145
5.2. Основы X .....	145

5.3. Установка X11 .....	148
5.4. Конфигурация X11 .....	149
5.5. Использование шрифтов в X11 .....	153
5.6. Менеджеры Экранов (Display Managers) X .....	157
5.7. Графические оболочки .....	160
<b>II. Общие задачи .....</b>	<b>167</b>
6. Приложения для настольного компьютера .....	168
6.1. Краткий обзор .....	168
6.2. Браузеры .....	168
6.3. Бизнес приложения .....	172
6.4. Программы просмотра документов .....	175
6.5. Финансовые программы .....	177
6.6. Итоги .....	178
7. Мультимедиа.....	180
7.1. Краткий обзор .....	180
7.2. Настройка звуковой карты .....	181
7.3. Звук MP3.....	185
7.4. Воспроизведение видео.....	187
7.5. Настройка ТВ тюнеров.....	196
7.6. Сканеры .....	198
8. Настройка ядра FreeBSD.....	203
8.1. Краткий обзор .....	203
8.2. Зачем собирать собственное ядро?.....	203
8.3. Сборка и установка собственного ядра .....	204
8.4. Конфигурационный файл .....	207
8.5. Создание файлов устройств.....	222
8.6. Если что-то пошло не так .....	223
9. Печать.....	226
9.1. Краткий обзор .....	226
9.2. Введение .....	226
9.3. Основная настройка .....	227
9.4. Расширенная настройка принтера .....	242
9.5. Использование принтеров .....	273
9.6. Альтернативы стандартному спулелу .....	281
9.7. Выявление проблем .....	281
10. Двоичная совместимость с Linux.....	286
10.1. Краткий обзор .....	286
10.2. Установка .....	286
10.3. Установка Mathematica® .....	290
10.4. Установка Maple™ .....	291
10.5. Установка MATLAB®.....	294
10.6. Установка Oracle® .....	296
10.7. Установка SAP® R/3®.....	300
10.8. Дополнительные сведения .....	321

<b>III. Системное Администрирование .....</b>	<b>324</b>
11. Настройка и оптимизация .....	325
11.1. Введение .....	325
11.2. Начальное конфигурирование .....	325
11.3. Основные настройки .....	327
11.4. Настройка приложений .....	327
11.5. Запуск сервисов .....	328
11.6. Настройка утилиты cron .....	331
11.7. Использование rc в FreeBSD 5.X .....	333
11.8. Настройка карт сетевых интерфейсов .....	334
11.9. Настройка виртуальных серверов .....	339
11.10. Файлы настройки .....	339
11.11. Настройка с помощью sysctl .....	343
11.12. Оптимизация дисков .....	344
11.13. Изменение ограничений, накладываемых ядром .....	348
11.14. Увеличение объема подкачки .....	350
11.15. Управление питанием и ресурсами .....	352
11.16. Использование и отладка FreeBSD ACPI .....	353
12. Процесс загрузки FreeBSD .....	360
12.1. Описание .....	360
12.2. Проблема загрузки .....	360
12.3. MBR и первый, второй и третий этапы загрузки .....	361
12.4. Взаимодействие с ядром во время загрузки .....	365
12.5. Хинты устройств .....	366
12.6. Init: инициализация управления процессами .....	367
12.7. Процесс остановки системы .....	368
13. Пользователи и основы управления учетными записями .....	369
13.1. Краткий обзор .....	369
13.2. Введение .....	369
13.3. Учетная запись суперпользователя .....	371
13.4. Системные учетные записи .....	371
13.5. Учетные записи пользователей .....	371
13.6. Изменение учетных записей .....	372
13.7. Ограничение пользователей .....	377
13.8. Персонализация пользователей .....	380
13.9. Группы .....	380
14. Безопасность .....	383
14.1. Краткое описание .....	383
14.2. Введение .....	383
14.3. Защита FreeBSD .....	385
14.4. DES, MD5, и шифрование .....	393
14.5. Одноразовые пароли .....	394
14.6. TCP Wrappers .....	399
14.7. <b>KerberosIV</b> .....	402
14.8. <b>Kerberos5</b> .....	409
14.9. Межсетевые экраны .....	417
14.10. OpenSSL .....	428
14.11. VPN через IPsec .....	430

14.12. OpenSSH .....	442
14.13. Списки контроля доступа файловой системы (ACL) .....	447
14.14. Сообщения безопасности FreeBSD .....	449
15. Принудительный контроль доступа (MAC) .....	452
15.1. Краткий обзор .....	452
15.2. Ключевые термины этой главы .....	453
15.3. Описание MAC .....	454
15.4. Метки MAC .....	455
15.5. Настройка модулей .....	460
15.6. Модуль MAC bsdextended .....	461
15.7. Модуль MAC ifoff .....	462
15.8. Модуль MAC portacl .....	462
15.9. Политики MAC, использующие метки .....	464
15.10. Модуль MAC partition .....	465
15.11. Модуль многоуровневой безопасности MAC (MLS) .....	466
15.12. Модуль MAC Biba .....	468
15.13. Модуль MAC LOMAC .....	469
15.14. Реализация защищенной среды с MAC .....	469
15.15. Другой пример: Использование MAC для защиты веб сервера .....	474
15.16. Решение проблем с инфраструктурой MAC .....	476
16. Устройства хранения .....	479
16.1. Краткий обзор .....	479
16.2. Имена устройств .....	479
16.3. Добавление дисков .....	480
16.4. RAID .....	482
16.5. USB устройства хранения .....	488
16.6. Запись и использование оптических носителей (CD) .....	490
16.7. Создание и использование оптических носителей (DVD) .....	496
16.8. Дискеты .....	500
16.9. Создание и использование архивных копий на магнитной ленте .....	503
16.10. Создание резервных копий на дискетах .....	506
16.11. Основы технологии резервного копирования .....	507
16.12. Сетевые файловые системы, файловые системы в памяти и с отображением в файл .....	514
16.13. Мгновенные копии файловых систем .....	518
16.14. Квотирование файловых систем .....	520
16.15. Шифрование дисковых разделов .....	523
17. Менеджер дискового пространства Vinum .....	529
17.1. Краткая аннотация .....	529
17.2. Диски слишком малы .....	529
17.3. Ограниченная пропускная способность .....	529
17.4. Целостность данных .....	532
17.5. Объекты Vinum .....	533
17.6. Несколько примеров .....	535
17.7. Правила именования объектов .....	541
17.8. Создание конфигурации Vinum .....	544
17.9. Vinum для корневой файловой системы .....	545
18. Локализация - I18N/L10N использование и настройка .....	552
18.1. Краткий обзор .....	552

18.2. Основы .....	552
18.3. Использование локализации.....	553
18.4. Компиляция I18N программ .....	559
18.5. Локализация FreeBSD для поддержки определенных языков .....	559
19. На переднем крае разработок .....	564
19.1. Краткий обзор .....	564
19.2. FreeBSD-CURRENT против FreeBSD-STABLE.....	564
19.3. Синхронизация ваших исходных текстов.....	568
19.4. Пересборка "world" .....	569
19.5. Отслеживание исходных текстов для нескольких машин.....	585
<b>IV. Сетевые коммуникации.....</b>	<b>587</b>
20. Последовательные соединения .....	588
20.1. Краткое описание.....	588
20.2. Введение .....	588
20.3. Терминалы .....	593
20.4. Входящие соединения по модему .....	597
20.5. Исходящие соединения по модему .....	605
20.6. Настройка последовательной консоли .....	609
21. PPP и SLIP.....	618
21.1. Краткий обзор .....	618
21.2. PPP уровня пользователя.....	618
21.3. PPP уровня ядра .....	631
21.4. Решение проблем с соединениями PPP.....	638
21.5. Использование PPP через Ethernet (PPPoE) .....	642
21.6. Использование <b>PPP</b> через ATM (PPPoA) .....	643
21.7. Использование SLIP.....	647
22. Электронная почта.....	657
22.1. Краткий обзор .....	657
22.2. Использование электронной почты.....	657
22.3. Настройка <b>sendmail</b> .....	660
22.4. Установка другой почтовой программы .....	662
22.5. Поиск и устранение неисправностей .....	665
22.6. Расширенное руководство.....	668
22.7. SMTP через UUCP .....	671
22.8. Настройка почты только для отправки .....	673
22.9. Использование почты с коммутируемым соединением .....	674
22.10. SMTP аутентификация .....	675
22.11. Почтовые программы пользователей .....	676
22.12. Использование fetchmail.....	683
22.13. Использование procmail.....	684
23. Сетевые серверы .....	686
23.1. Краткий обзор .....	686
23.2. "Супер-сервер" <b>inetd</b> .....	686
23.3. Network File System (NFS) .....	691
23.4. Network Information System (NIS/YP) .....	697
23.5. Автоматическая настройка сети (DHCP) .....	714
23.6. Domain Name System (DNS) .....	719

23.7. BIND9 и FreeBSD .....	732
23.8. Apache HTTP сервер .....	734
23.9. Файл сервер и печать для Microsoft® Windows клиентов (Samba).....	737
23.10. Протокол передачи файлов (FTP).....	740
23.11. Синхронизация часов через NTP .....	741
24. Сложные вопросы работы в сети.....	745
24.1. Краткий обзор .....	745
24.2. Сетевые плюзы и маршруты.....	745
24.3. Беспроводные сети.....	752
24.4. Bluetooth.....	760
24.5. Мосты .....	768
24.6. Работа с бездисковыми станциями.....	771
24.7. ISDN.....	780
24.8. Демон преобразования сетевых адресов (natd).....	784
24.9. IP по параллельному порту (PLIP) .....	788
24.10. IPv6.....	790
24.11. Асинхронный режим передачи (ATM) в FreeBSD 5.X .....	795
<b>V. Приложения.....</b>	<b>798</b>
A. Получение FreeBSD .....	799
A.1. Издатели CDRom и DVD.....	799
A.2. FTP сайты.....	802
A.3. Анонимный CVS .....	812
A.4. Использование CTM .....	815
A.5. Использование CVSup .....	819
A.6. Теги CVS.....	840
A.7. AFS сайты .....	845
A.8. rsync сайты.....	845
B. Библиография .....	847
B.1. Книги и журналы, специализирующиеся на FreeBSD.....	847
B.2. Руководства для пользователей.....	848
B.3. Руководства для администраторов .....	849
B.4. Руководства для программистов .....	849
B.5. Внутренности операционной системы.....	850
B.6. Безопасность .....	850
B.7. Оборудование.....	851
B.8. История UNIX .....	851
B.9. Прочие издания .....	852
C. Ресурсы в интернет.....	853
C.1. Списки рассылки.....	853
C.2. Новостные группы Usenet .....	869
C.3. Серверы World Wide Web .....	871
C.4. Адреса Email .....	879
C.5. Shell доступ .....	879
D. Ключи PGP.....	880
D.1. Офицеры.....	880
D.2. Члены Core .....	880
D.3. Разработчики .....	881

Глоссарий FreeBSD .....	909
Предметный указатель .....	911
Colophon.....	922

# Список таблиц

2-1. Пример сведений об оборудовании.....	11
2-2. Планирование разделов для первого диска.....	37
2-3. Разметка разделов для остальных дисков.....	38
2-4. Возможные профили безопасности.....	56
2-5. Названия FreeBSD ISO образов и их значения .....	94
3-1. Коды дисковых устройств .....	112
16-1. Соглашения по именованию физических дисков .....	479
17-1. Методы организации наборов Vinum .....	534
24-1. Распайка кабеля для параллельного порта для сетевой работы .....	788
24-2. Зарезервированные адреса IPv6.....	791

# Список иллюстраций

2-1. Меню конфигурации ядра .....	20
2-2. Визуальный интерфейс конфигурации ядра .....	21
2-3. Развернутый список драйверов .....	21
2-4. Конфигурация драйверов без конфликтов .....	23
2-5. Главное меню Sysinstall .....	23
2-6. Типичный вывод Device Probe .....	24
2-7. Выбор Sysinstall Exit .....	25
2-8. Выбор Usage в главном меню Sysinstall .....	26
2-9. Выбор меню документации.....	27
2-10. Меню документации Sysinstall.....	27
2-11. Выбор меню раскладки клавиатуры.....	28
2-12. Меню раскладки клавиатуры.....	29
2-13. Выбор параметров установки.....	29
2-14. Параметры Sysinstall.....	29
2-15. Начало стандартной установки .....	30
2-16. Выберите диск для FDisk.....	33
2-17. Типичные разделы fdisk перед редактированием .....	34
2-18. Разбиение в Fdisk с использованием всего диска .....	35
2-19. Меню менеджера загрузки Sysinstall.....	35
2-20. Выход из выбора диска.....	36
2-21. Редактор Sysinstall Disklabel .....	39
2-22. Редактор Sysinstall Disklabel с установками по умолчанию .....	40
2-23. Свободное место для корневого раздела .....	41
2-24. Редактирование размера корневого раздела .....	41
2-25. Выбор типа корневого раздела .....	42
2-26. Выбор точки монтирования корневой файловой системы .....	42
2-27. Редактор Sysinstall Disklabel .....	43
2-28. Выбор дистрибутивных наборов.....	44
2-29. Подтверждение выбора дистрибутивного набора.....	45
2-30. Выбор источника установки.....	46
2-31. Выбор Ethernet устройства.....	48



2-32. Настройка сети для ed0 .....	49
2-33. Редактирование <code>inetd.conf</code> .....	52
2-34. Настройка по анонимного FTP по умолчанию .....	53
2-35. Редактирование FTP Welcome Message .....	54
2-36. Редактирование <code>exports</code> .....	55
2-37. Параметры профиля безопасности .....	56
2-38. Параметры настройки системной консоли .....	58
2-39. Параметры хранителя экрана .....	58
2-40. Временной интервал хранителя экрана .....	59
2-41. Выход из меню конфигурации консоли .....	59
2-42. Выбор региона .....	60
2-43. Выбор страны .....	61
2-44. Выбор часового пояса .....	61
2-45. Выбор протокола мыши .....	63
2-46. Установка протокола мыши .....	63
2-47. Настройка порта мыши .....	63
2-48. Установка порта мыши .....	64
2-49. Запуск мышинного демона .....	64
2-50. Проверка мышинного демона .....	65
2-51. Верхняя часть меню настройки сети (Network Configuration) .....	66
2-52. Выбор MTA по умолчанию .....	67
2-53. Настройка Ntpdate .....	68
2-54. Нижняя часть меню настройки сети .....	68
2-55. Меню выбора способа конфигурации .....	70
2-56. Выбор десктопа по умолчанию .....	79
2-57. Выбор категории пакетов .....	80
2-58. Выбор пакетов .....	80
2-59. Установка пакетов .....	81
2-60. Подтверждение установки пакетов .....	81
2-61. Выбор User (пользователь) .....	82
2-62. Ввод информации о пользователе .....	83
2-63. Выход из меню управления пользователями и группами .....	84
2-64. Выход из установки .....	85
17-1. Организация сцепленных дисков .....	530
17-2. Организация с перемежением .....	531
17-3. Организация RAID-5 .....	532
17-4. Простой том <code>Vinum</code> .....	535
17-5. Зеркалированный том <code>Vinum</code> .....	537
17-6. Том с перемежением .....	539
17-7. Зеркалированный том с перемежением .....	541

## Список примеров

2-1. Использование существующего раздела без изменения .....	12
2-2. Сжатие существующих разделов .....	12
3-1. Пример имен диска, слайса, и раздела .....	112
3-2. Концептуальная модель диска .....	113

4-1. Загрузка пакета вручную и его локальная установка .....	132
11-1. Создание файла подкачки в FreeBSD 4.X .....	351
11-2. Создание файла подкачки в FreeBSD 5.X .....	351
12-1. Образец экрана boot0.....	361
12-2. Образец экрана boot2.....	362
12-3. Незащищённая консоль в /etc/ttys.....	367
13-1. Настройка adduser и добавление пользователя в FreeBSD 4.X.....	372
13-2. Добавление пользователя в FreeBSD 5.X.....	374
13-3. Интерактивное удаление учетной записи с помощью rmuser .....	375
13-4. Интерактивная работа с chpass суперпользователя.....	376
13-5. Интерактивная работа с chpass обычного пользователя.....	376
13-6. Изменение пароля.....	376
13-7. Изменение пароля другого пользователя суперпользователем.....	377
13-8. Добавление группы с использованием pw(8).....	381
13-9. Добавление пользователя в группу с использованием pw(8) .....	381
13-10. Использование id(1) для определения принадлежности к группам.....	381
14-1. Использование SSH для создания защищенного туннеля на SMTP.....	446
16-1. Использование dump через ssh .....	508
16-2. Использование dump при работе через ssh с заданием RSH .....	508
16-3. Скрипт для создания загрузочной дискеты.....	511
16-4. Использование vnconfi g для монтирования имеющегося образа файловой системы во FreeBSD 4.X 515	
16-5. Создание нового диска в файле с помощью vnconfig .....	515
16-6. Использование mdconfig для монтирования файла с образом существующей файловой системы во FreeBSD 5.X 516	
16-7. Создание нового диска, отображаемого в файл, при помощи mdconfig .....	516
16-8. Настройка и монтирование диска, отображаемого в файл, при помощи команды mdmfs .....	517
16-9. Диск md в памяти во FreeBSD 4.X.....	517
16-10. Создание нового диска с отображением в память при помощи mdconfig .....	518
16-11. Создание нового диска с отображением в память при помощи mdmfs .....	518
20-1. Добавление записей терминалов в /etc/ttys .....	595
22-1. Настройка базы данных доступа <b>sendmail</b> .....	660
22-2. Mail Aliases .....	661
22-3. Пример таблицы виртуального домена.....	662
23-1. Посылка сигнала HangUP процессу <b>inetd</b> .....	688
23-2. Монтирование ресурса при помощи <b>amd</b> .....	695
24-1. Офис подразделения или домашняя сеть .....	783
24-2. Центральный офис или другая локальная сеть .....	783
A-1. Извлечение кода из -CURRENT (ls(1)) с последующим удалением: .....	813
A-2. Использование SSH для извлечения дерева src/:.....	814
A-3. Извлечение ls(1) из ветви для 3.X-STABLE: .....	814
A-4. Создание списка изменений ls(1) (в виде unifi ed diff).....	814
A-5. Поиск доступных имен модулей: .....	814

# Предисловие

## Целевая аудитория

Люди, которые используют FreeBSD с недавнего времени, найдут, что первая часть этой книги проводит читателя через процесс установки FreeBSD и кратко освещает идеи и традиции, на которых базируется UNIX®. Работа с этой частью требует несколько большего, чем просто желание исследовать – необходима возможность принимать к сведению новые идеи.

Вторая, много большая часть Руководства, является всеобъемлющим справочником о всех темах, которые могут интересовать администраторов FreeBSD. Некоторые из глав этой части могут рекомендовать вам предварительное чтение других документов, о чём упоминается в кратком обзоре в начале каждой главы.

Список рекомендуемой дополнительной литературы вы можете найти в Прил. В.

## Изменения по сравнению со второй редакцией

Третья редакция является кульминацией более чем двух лет работы отдельных членов проекта документации FreeBSD. Вот основные изменения в новой редакции:

- Гл. 11, Настройка и оптимизация, была расширена новой информацией о ACPI управлении питанием и ресурсами, системной утилите `scop` и дополнительных параметрах оптимизации ядра.
- Гл. 14, Безопасность, была расширена новой информацией о виртуальных частных сетях (VPN), списках контроля доступа файловой системы (ACL), и сообщениях безопасности.
- Гл. 15, Принудительный контроль доступа (MAC), новая глава этой редакции. Она описывает, что такое MAC и как этот механизм может быть использован для защиты системы FreeBSD.
- Гл. 16, Устройства хранения, была расширена новой информацией о устройствах хранения USB, образах файловой системы, квотах файловой системы, файловых системах в файлах и в сети, зашифрованных дисковых разделах.
- Гл. 17, Менеджер дискового пространства `Vinum`, новая глава этой редакции. В ней описано как использовать `Vinum`, менеджер логических томов, который предоставляет независимые от устройств логические диски и программные уровни RAID-0, RAID-1 и RAID-5.
- К Гл. 21, PPP и SLIP, был добавлен раздел о решении проблем.
- Гл. 22, Электронная почта, была расширена новой информацией об использовании альтернативных транспортных агентов, SMTP аутентификации, UUCP, fetchmail, procmail, и другими разделами повышенной сложности.
- Гл. 23, Сетевые серверы, появилась в этой редакции. Эта главы включает информацию о установке Apache HTTP Server, FTPd, и настройке Samba сервера для клиентов Microsoft Windows. Некоторые разделы были перемещены сюда из Гл. 24, Сложные вопросы работы в сети.
- Гл. 24, Сложные вопросы работы в сети, была расширена новой информацией об использовании устройств Bluetooth в FreeBSD, настройке беспроводных сетей, и сетях Asynchronous Transfer Mode (ATM).

- Был добавлен глоссарий, объединяющий информацию о технических терминах, используемых в книге.
- Множество эстетических улучшений были внесены в таблицы и иллюстрации этой книги.

## Изменения во второй редакции

Вторая редакция является кульминацией более чем двухлетней работы членов Проекта документации FreeBSD. Нижеследующий список перечисляет все значительные изменения, внесенные в эту редакцию:

- Был добавлен полный указатель тем.
- Все ASCII-иллюстрации были заменены на графические.
- Был добавлен стандартный краткий обзор к каждому разделу для того, чтобы читатель мог получить представление о содержании раздела и о том, что необходимо знать для его изучения.
- Содержимое было логически реорганизовано на три части: "В Начале", "Системное администрирование" и "Приложения".
- Гл. 2 ("Установка FreeBSD") была полностью переписана, добавлено большое количество снимков экрана, чтобы облегчить понимание текста для новых пользователей.
- Гл. 3 ("Основы UNIX") была расширена и содержит дополнительную информацию о процессах, даемонах и сигналах.
- Гл. 4 ("Установка приложений: порты и пакеты") была расширена и содержит дополнительную информации об управлении бинарными пакетами.
- Гл. 5 ("X Window System") была полностью переписана и обращает больше внимания на современные технологии для рабочего стола, такие, как **KDE** и **GNOME** на XFree86™ 4.X.
- Гл. 12 ("Процесс загрузки FreeBSD") была расширена.
- Гл. 16 ("Устройства хранения") была составлена из того, что раньше было двумя различными главами: "Диски" и "Резервное копирование". Мы считаем, что данные темы будут проще и более полно описаны как одна глава. Был добавлен раздел о программном и аппаратном RAID.
- Гл. 20 ("Последовательные коммуникации") была полностью реорганизована и обновлена для FreeBSD 4.X/5.X.
- Гл. 21 ("PPP и SLIP") была существенно обновлена.
- Было добавлено множество новых разделов в Гл. 24 ("Сложные вопросы работы в сети").
- Гл. 22 ("Электронная почта") была расширена, теперь она включает больше информации о настройке **sendmail**.
- Гл. 10 ("Работа с приложениями, написанными для Linux®") была дополнена включением информации об установке **Oracle®** и **SAP® R/3®**.
- Следующие новые темы были рассмотрены в этой, второй, редакции:
  - Настройка и оптимизация (Гл. 11).
  - Мультимедиа (Гл. 7)

# Структура этой книги

Эта книга разбита на пять частей. В первой части, *В начале*, рассматривается установка и основные навыки использования FreeBSD. Предполагается, что читатель освоит эти разделы последовательно, возможно пропуская разделы, в которых обсуждаются уже знакомые для него темы. Вторая часть, *Общие задачи*, рассказывает о некоторых наиболее часто используемых возможностях FreeBSD. Этот раздел и все последующие могут быть прочитаны не по порядку. Каждая глава начинается с краткого обзора, который описывает, о чём говорится в ней и что читатель должен будет знать для прочтения этой главы. Это сделано для того, чтобы случайно встретивший этот материал читатель мог найти разделы, которые его интересуют. В третьей части, *Системное администрирование*, рассмотрены вопросы администрирования. В четвертой части, *Сетевые коммуникации*, охвачены темы, связанные с серверами и сетью. Пятая часть содержит приложения и справочную информацию.

## Гл. 1, Введение

Знакомит пользователя с FreeBSD. Рассказывает об истории проекта FreeBSD, его задачах и модели разработки.

## Гл. 2, Установка

Проводит пользователя через весь процесс установки. Обсуждаются также некоторые более сложные вопросы установки, такие как установка по последовательной консоли.

## Гл. 3, Основы UNIX

Рассказывает об основных командах и функциональности операционной системы FreeBSD. Если вы знакомы с Linux или другой UNIX-подобной операционной системой, возможно, вы пропустите эту главу.

## Гл. 4, Установка приложений: порты и пакеты

Рассказывает о процессе установки программного обеспечения сторонних производителей с использованием "Коллекции Портов FreeBSD" и стандартных бинарных пакетов.

## Гл. 5, X Window System

Описывает X Window System вообще и использование **XFree86** под управлением FreeBSD в частности. Также описывает популярные окружения рабочего стола, такие как **KDE** и **GNOME**.

## Гл. 6, Приложения для настольного компьютера

Перечисляет некоторые популярные приложения для рабочей станции, такие как веб-браузеры и офисные пакеты и описывает процесс их установки на FreeBSD.

## Гл. 7, Мультимедиа

Показывает, как настроить поддержку воспроизведения звука и видео на вашей системе. Также описывает некоторые примеры приложений для воспроизведения звука и видео.

## Гл. 8, Настройка ядра FreeBSD

Объясняет, почему вам может понадобиться перенастроить ядро и детально описывает процесс настройки, сборки и установки нового ядра.

### *Гл. 9, Печать*

Рассказывает об управлении принтерами в FreeBSD, включая информацию об титульных страницах, учёте использования принтеров и первоначальной настройке.

### *Гл. 10, Двоичная совместимость с Linux*

Описывает возможности Linux-совместимости в FreeBSD. Также предоставляет детальные инструкции по установке многих популярных приложений для Linux, таких как: **Oracle**, **SAP R/3** и **Mathematica®**.

### *Гл. 11, Настройка и оптимизация*

Описывает всевозможные параметры настройки FreeBSD, которые может использовать системный администратор для оптимальной настройки системы. Также описывает различные конфигурационные файлы, используемые в FreeBSD и расположение этих файлов на диске.

### *Гл. 12, Процесс загрузки FreeBSD*

Рассказывает о процессе загрузки FreeBSD и объясняет, как управлять этим процессом при помощи различных настроек.

### *Гл. 13, Пользователи и основы управления учётными записями*

Рассказывает о создании и управлении пользовательскими учётными записями. Также обсуждает установку ограничений ресурсов для пользователей и другие задачи управления пользователями.

### *Гл. 14, Безопасность*

Описывает множество различных утилит, которые помогут вам поддерживать FreeBSD в безопасном, надёжном состоянии, включая Kerberos, IPsec, OpenSSH и брандмауэры.

### *Гл. 15, Принудительный контроль доступа (MAC)*

Описывает что такое принудительный контроль доступа (Mandatory Access Control, MAC) и как этот механизм может быть использован для защиты системы FreeBSD.

### *Гл. 16, Устройства хранения*

Описывает как управлять накопителями информации и файловыми системами в FreeBSD, включая физические диски, массивы RAID, оптические и ленточные носители, диски в оперативной памяти и сетевые файловые системы.

### *Гл. 17, Менеджер дискового пространства Vinum*

Рассказывает как использовать Vinum, менеджер логических разделов, при помощи которого можно создавать и использовать независимые от устройств хранения логические диски и программно реализовывать RAID-0, RAID-1 и RAID-5.

### *Гл. 18, Локализация — I18N/L10N использование и настройка*

Описывает использование FreeBSD с языками, отличными от английского. Рассказывает о локализации на уровне системы и отдельных приложений.

### *Гл. 19, На переднем крае разработок*

Объясняет различия между FreeBSD-STABLE, FreeBSD-CURRENT и FreeBSD-RELEASE.

Рассказывает, кому из пользователей будет полезно отслеживать версию системы в разработке и вкратце описывает этот процесс.

### *Гл. 20, Последовательные соединения*

Объясняет, как подключать терминалы и модемы к вашей системе FreeBSD, как в серверном, так и в клиентском режиме.

### *Гл. 21, PPP и SLIP*

Описывает использование PPP, SLIP или PPP через Ethernet для соединения с удалёнными системами при помощи FreeBSD.

### *Гл. 22, Электронная почта*

Описывает использование различных компонентов почтового сервера и более углублённо рассматривает простые вопросы конфигурации для наиболее популярного программного обеспечения почтовых серверов: **sendmail**.

### *Гл. 23, Сетевые серверы*

Предоставляет детальные инструкции и примеры файлов настройки для использования компьютера с FreeBSD в качестве файлового сервера (NFS), сервера доменных имен (DNS), сервера сетевой информационной системы (NIS), или сервера точного времени (ntpd).

### *Гл. 24, Сложные вопросы работы в сети*

Рассматривает множество вопросов работы с сетью, включая совместный доступ компьютеров вашей локальной сети к интернет, расширенные вопросы маршрутизации, беспроводные соединения, bluetooth, ATM, IPv6 и многое другое.

### *Прил. А, Получение FreeBSD*

Перечисляет различные источники, из которых можно получить FreeBSD на CDROM или DVD, равно как и различные сайты в интернет, с которых можно скачать и установить FreeBSD.

### *Прил. В, Библиография*

Эта книга касается многих различных тем, которые могут сподвигнуть вас на более детальное изучение. Библиография перечисляет множество отличных книг, упоминаемых в тексте.

### *Прил. С, Ресурсы в интернет*

Описывает множество форумов, доступных для пользователей FreeBSD, где можно задать вопросы и поучаствовать в технических обсуждениях FreeBSD.

### *Прил. D, Ключи PGP*

Содержит ключи PGP некоторых разработчиков FreeBSD.

# Договоренности, используемые в этой книге

Для того чтобы обеспечить целостность и простоту чтения текста в данной книге, мы применяем некоторые договорённости.

## Типографические договорённости

### *Наклонный шрифт*

*Наклонный шрифт* используется для имен файлов, адресов в интернет (URL), выделенного текста и первого применения технических терминов.

### Моноширинный шрифт

**Моноширинных шрифт** используется для сообщений об ошибках, команд, имен пользователей, названий групп, названий устройств, переменных и фрагментов кода.

### Полужирный шрифт

**Полужирный шрифт** используется для обозначения приложений, команд и параметров.

## Пользовательский ввод

Клавиши представляются в виде **полужирного текста** для того, чтобы выделяться среди остального текста. Комбинации клавиш, которые должны вводиться одновременно, разделяются символом '+', например:

### **Ctrl+Alt+Del**

Это будет означать, что пользователь должен нажать клавиши **Ctrl**, **Alt** и **Del** одновременно.

Комбинации клавиш, которые должны вводиться последовательно, разделяются запятыми, например:

### **Ctrl+X, Ctrl+S**

Это будет означать, что пользователь должен нажать **Ctrl** и **X** одновременно, после чего одновременно нажать **Ctrl** и **S**.

## Примеры

Примеры, которые начинаются с `E:\>` обозначают команды MS-DOS®. Если не указано обратного, эти команды могут вводиться из окна "Сеанс MS-DOS" в современных системах Microsoft® Windows®.

```
E:\> tools\fdimage floppies\kern.flp A:
```

Примеры, которые начинаются с `#` обозначают команды, которые должны быть запущены с правами суперпользователя в FreeBSD. Вы можете войти в систему как пользователь `root` для того, чтобы ввести эти команды или войти в систему обычным пользователем и использовать `su(1)` для того, чтобы получить привилегии суперпользователя.

```
# dd if=kern.flp of=/dev/fd0
```



Примеры, начинающиеся с %, указывают, что команда должна быть исполнена с правами обычного пользователя. Если не указано обратного, используется синтаксис C-shell для установки переменных окружения и других команд.

```
% top
```

## Благодарности

Книга, которую вы держите в руках является собой результат труда многих сотен людей по всему миру. Не имеет значения, присылали ли они исправления опечаток или предоставляли целые главы, их труд был полезен.

Несколько компаний поддерживали разработку этого документа, оплачивая авторам их труд, оплачивая публикацию и т.д. В частности, BSDi (в последствии приобретённая компанией Wind River Systems (<http://www.windriver.com>)) оплачивала труд по улучшению этой книги участникам Проекта Документации FreeBSD, что в итоге сделало возможным выпуск первой печатной версии в марте 2000 года (ISBN 1-57176-241-8). Впоследствии компания Wind River Systems оплатила работу нескольких авторов по улучшению генерации книги в удобном для печати виде и добавлению нескольких глав. Кульминация этой работы являла собой публикацию второй печатной версии в ноябре 2001 года (ISBN 1-57176-303-1). В 2003-2004 годах FreeBSD Mall, Inc (<http://www.freebsdmall.com>) заплатила нескольким контрибьюторам за улучшение Handbook при подготовке к третьей редакции.

# I. В начале

Эта часть Руководства Пользователя FreeBSD предназначена для пользователей и администраторов - новичков в FreeBSD. Эти главы:

- Введут вас в FreeBSD.
- Проведут вас по процессу установки FreeBSD.
- Обучат вас некоторым основам UNIX.
- Покажут вам как устанавливать программные пакеты не входящие в стандартную поставку FreeBSD.
- Введут вас в X Window, оконную систему для UNIX, и опишут как настроить графическое окружение и сделать вашу работу более продуктивной.

Мы попытались сократить множество ссылок в тексте до минимума для того, чтоб вы могли прочитать эту секцию Руководства с начала до конца с минимумом перелистываний страниц.

# Глава 1. Введение

*Исправил, реорганизовал и частично переписал Jim Mock. Перевод на русский язык: Алексей Зелькин, Денис Пеплин.*

## 1.1. Краткий обзор

Мы благодарим вас за интерес к FreeBSD! Следующая глава расскажет о некоторых аспектах проекта FreeBSD, таких как история, цели, модель разработки, и прочее.

Из этой главы вы узнаете:

- Какое отношение имеет FreeBSD к другим операционным системам.
- Историю проекта FreeBSD.
- Цели проекта FreeBSD.
- Основы модели разработки FreeBSD с открытыми исходными текстами.
- И конечно, откуда появилось имя "FreeBSD".

## 1.2. Добро пожаловать во FreeBSD!

FreeBSD это основанная на 4.BSD-Lite операционная система для компьютеров Intel (x86 и Itanium®), AMD64, Alpha™ и Sun UltraSPARC®. Ведется работа по портированию и на другие архитектуры. Вы можете также прочесть о истории FreeBSD, или о текущем релизе. Если вы заинтересованы в помощи проекту (кодом, аппаратным обеспечением, деньгами), прочтите статью [Помощь FreeBSD](http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/contributing/index.html) ([http://www.FreeBSD.org/doc/en\\_US.ISO8859-1/articles/contributing/index.html](http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/contributing/index.html)).

### 1.2.1. Что может FreeBSD?

FreeBSD имеет заслуживающие внимания особенности. Некоторые из них:

- *Приоритетная многозадачность* с динамическим регулированием приоритетов плавно и правильно приспособится и справедливо распределит ресурсы компьютера между приложениями и пользователями, даже при тяжелейших нагрузках.
- *Многопользовательская доступность* которая позволяет множеству людей использовать FreeBSD совместно для ряда задач. Это значит, например, что системная периферия, такая как принтеры и ленточные устройства, правильно распределена между пользователями в системе или сети, и что лимиты каждого ресурса могут быть установлены пользователям или группам пользователей, защищая критические системные ресурсы от перегрузок.
- Мощный *TCP/IP стек* с поддержкой промышленных стандартов, таких как SLIP, PPP, NFS, DHCP и NIS. Это говорит о том, что FreeBSD может легко функционировать с другими системами предприятия, предоставляя жизненно важные функции, такие как NFS (удалённый доступ к файлам) и услуги электронной почты или представить вашу организацию в Internet с сервисами WWW, FTP, маршрутизации и брандмауэра (сетевая безопасность).

- *Защита памяти* гарантирует, что приложения (или пользователи) не смогут чинить препятствия друг другу. Фатальная ошибка в выполнении одного приложения не скажется на работоспособности всей системы.
- FreeBSD 32-х битная операционная система (64 бита на Alpha, Itanium, AMD64, и UltraSPARC) и была создана именно такой.
- Промышленный стандарт X Window System (X11R6) предоставляет графический интерфейс пользователя (GUI) для большинства VGA карт и мониторов и поставляется с полными исходными текстами.
- *Двоичная совместимость* с большинством программ, созданных для Linux, SCO, SVR4, BSDI и NetBSD.
- Тысячи *готовых к использованию* приложений доступны из FreeBSD коллекций *портов* и *пакетов*. Зачем искать что-то в сети, когда вы можете найти всё прямо здесь?
- Тысячи других *легко-адаптируемых* приложений доступны в Internet. FreeBSD в исходных текстах совместима с большинством популярных коммерческих UNIX систем и, таким образом, большинство приложений требуют несколько достаточно мелких (если требуют вообще) изменений для сборки.
- *Виртуальная память* с поддержкой сброса неиспользуемых страниц по требованию и "объединенные VM/буферный кэш" спроектированы так, чтобы максимально эффективно удовлетворить приложения с огромными аппетитами к памяти и, в то же время, сохранить интерактивность остальных пользователей.
- Поддержка *симметричной многопроцессорности* (SMP) для машин с несколькими процессорами.
- Полный комплект инструментов для разработчика: C, C++, Fortran и Perl. Множество дополнительных языков для прочих исследований и разработки также доступны из коллекций портов и пакетов.
- *Исходные тексты* всей системы означают, что вы имеете великолепную возможность контролировать ваше окружение. Зачем выбирать закрытые решения и уповать на милость производителя, когда вы можете иметь по-настоящему открытую систему?
- Обширная *online документация*.
- *И многое-многое другое!*

FreeBSD основана на 4.4BSD-Lite от Computer Systems Research Group (CSRG) Калифорнийского Университета, Беркли, и имеет в себе характерные черты модели развития BSD-систем. В дополнении к прекрасной работе предоставленной CSRG, Проект FreeBSD тратит многие тысячи часов для тонкой настройки системы для максимальной производительности и надёжности в условиях максимально приближенным к "боевым". Когда большинство коммерческих гигантов только пытаются достичь такого уровня возможностей, производительности и надежности операционных систем для PC, FreeBSD может предложить все это прямо *сейчас*!

Применение FreeBSD в действительности ограничено только вашими фантазиями. От разработки программного обеспечения до автоматизации производства, от контроля за качеством товара до коррекции азимутов удалённых спутниковых антенн; эти задачи можно решить с помощью коммерческих UNIX систем, но скорее всего они решаемы и с помощью FreeBSD! FreeBSD столь же многогранно полезна, как и буквально тысячи высококачественных приложений, разработанных исследовательскими центрами и университетами во всём мире, доступные за минимальную цену или даже бесплатно. Коммерческие приложения также доступны и появляются ежедневно в больших количествах.

Исходя из того, что исходные тексты FreeBSD всегда доступны, система может быть оптимизирована в почти невероятной степени для специальных приложений или проектов, а это невозможно при использовании операционных систем от большинства коммерческих производителей. Вот несколько примеров того, как люди используют FreeBSD:

- *Internet сервисы:* мощнейший TCP/IP стек делает FreeBSD идеальной платформой для большинства Internet приложений таких как:
  - FTP серверы
  - World Wide Web серверы (как стандартные, так и безопасные [SSL])
  - Серверы фильтрации пакетов и шлюзы NAT ('IP маскарading')
  - Серверы электронной почты
  - Серверы новостей USENET и BBS
  - и многое другое...

Вы можете начать своё знакомство с FreeBSD, используя недорогую машину класса 386 PC, а впоследствии увеличить её мощь до сервера масштаба предприятия с четырьмя процессорами Xeon и RAID контроллером.

- *Образование:* Вы студент и ваше образование связано с компьютерами или другими инженерными расчётами? Нет лучшего пути начать изучение операционных систем, архитектуры компьютера и работы в сети, чем покорить вершины FreeBSD. Количество доступных CAD, математических и графических пакетов также делают её чрезвычайно полезной для выполнения *другой* не менее важной работы.
- *Исследования:* С исходными текстами, которые доступны для всей системы, FreeBSD это превосходная платформа как для исследования операционных систем, так и исследования других областей компьютерных наук. Свободная природа FreeBSD предоставляет возможность удалённым группам сотрудничать, обмениваться идеями и совместными разработками, не беспокоясь о наличии специальных лицензий или ограничений на то, что может обсуждаться в открытых форумах.
- *Работа в сети:* Нужен новый маршрутизатор? Сервер имён (DNS)? Брандмауэр, защищающий от проникновения извне в вашу сеть? FreeBSD может превратить давно списанный и пылящийся в углу 386й или 486й PC в мощный маршрутизатор с возможностью фильтрации пакетов.
- *Рабочая станция X Window:* FreeBSD прекрасный выбор, если вам нужен недорогой X-терминал. Можно использовать свободно распространяемый сервер X11 или один из коммерческих серверов, поставляемых Xi Graphics (<http://www.xig.com>). В отличие от X терминала, на FreeBSD можно запускать множество приложений локально, если требуется, таким образом перенеся часть нагрузки с центрального сервера. FreeBSD может быть загружена "на бездисковой станции", что делает рабочую станцию более дешёвой и простой в администрировании.
- *Разработка программного обеспечения:* Базовая поставка FreeBSD распространяется с полным набором инструментов для разработки, включая знаменитые компилятор GNU C/C++ и отладчик.

FreeBSD доступна как в исходных текстах, так и в бинарном виде на CDROM, DVD и через анонимный доступ к FTP. Смотрите Прил. А для уточнений.

## 1.2.2. Кто использует FreeBSD?

FreeBSD используется в качестве платформы на некоторых крупнейших сайтах в интернет, включая:

- Yahoo! (<http://www.yahoo.com/>)
- Apache (<http://www.apache.org/>)
- Blue Mountain Arts (<http://www.bluemountain.com/>)
- Pair Networks (<http://www.pair.com/>)
- Sony Japan (<http://www.sony.co.jp/>)
- Netcraft (<http://www.netcraft.com/>)
- Weathernews (<http://www.wni.com/>)
- Supervalu (<http://www.supervalu.com/>)
- TELEHOUSE America (<http://www.telehouse.com/>)
- Sophos Anti-Virus (<http://www.sophos.com/>)
- JMA Wired (<http://www.jmawired.com/>)

и на многих других.

## 1.3. О Проекте FreeBSD

В следующей части рассказывается о том, что из себя представляет проект, включая краткую историю, цели проекта и модель разработки проекта.

### 1.3.1. Краткая история FreeBSD

*Предоставил Jordan Hubbard.*

Проект FreeBSD возник в первой половине 1993 года, частично как результат развития 'Неофициального комплекта исправлений к 386BSD (patchkit)', последними 3-мя координаторами этого проекта: Nate Williams, Rod Grimes и мною.

Нашей главной задачей было привести промежуточный снимок 386BSD в порядок, исправив множество проблем, которые механизм patchkit не мог решить. Некоторое из вас возможно помнят раннее название этого проекта: "386BSD 0.5" или "386BSD Interim".

386BSD была операционной системой Билла Джоилца, который на тот момент находился строго говоря, в состоянии полного пренебрежения к ней. Так как patchkit разрастался, его поддержание становилось более неудобным день от дня, мы пришли к единодушному соглашению что что-то нужно делать и решили помочь Биллу путем предоставления промежуточных "очистных" снимков. Эти планы были грубо оборваны, когда Билл внезапно решил прекратить поддержку проекта без всяких ясных комментариев, что должно быть сделано.

Нам потребовалось немного времени, чтобы прийти к решению продолжать следовать той же цели, даже без поддержки Билла, и мы приняли имя 'FreeBSD' приобретенное Дэвидом Гринмэном. Наши начальные цели были определены после консультаций с пользователями существовавшей системы, и

как только стало понятно, что проект на пути к тому, чтобы стать реальностью, я связался с Walnut Creek CDROM с идеями о путях последующего улучшения каналов распространения FreeBSD для множества пользователей без доступа к Internet. Walnut Creek CDROM не только поддержал идею распространения FreeBSD на CD, но также пошел далеко вперед и предоставил проекту компьютер для работы и быстрый доступ к Internet. Без почти беспрецедентной веры Walnut Creek CDROM в этот в то время полностью неизвестный проект, вряд ли FreeBSD зашел бы так далеко и так быстро, как сегодня.

Первым дистрибутивом, распространяемым как на CDROM, так и в сети, стал FreeBSD 1.0, выпущенный в декабре 1993 года. Эта версия была выполнена на основе ленты 4.3BSD-Lite ("Net/2") из Калифорнийского Университета в Беркли, с многочисленными добавлениями из проекта 386BSD и Фонда Свободного Программного Обеспечения. Это был довольно внушительный успех для первого предложения, и мы закрепили его с выходом FreeBSD 1.1 RELEASE в мае 1994 года.

В это же время, на горизонте сгустились тучи в связи с назревающим скандалом между Novell и Калифорнийским Университетом, Беркли. Это был вялотекущий судебный процесс о легальности версии Net/2 из Беркли. Обстоятельства тяжбы с Калифорнийским Университетом заключались в том, что большие куски Net/2 были "загромождены" кодом, права на который принадлежат Novell, которая в свою очередь, получила их (права на код) ранее от AT&T. Чтоб вернуть "благословение" Novell, Беркли выпустил версию 4.4BSD-Lite, которая была объявлена полностью "свободной" и всем пользователям Net/2 было рекомендовано переключиться на ее использование. Это также касалось FreeBSD, и проекту было дано время до конца Июля 1994 года для прекращения распространения его продукта базирующегося на Net/2. На этих условиях проекту было разрешено выпустить последний релиз до окончания срока, это был FreeBSD 1.1.5.1.

Тогда FreeBSD приступил к сложной задаче буквально полному изобретению себя из абсолютно новой и довольно неполной системы 4.4BSD-Lite. "Lite" был в прямом смысле light (легким) потому, что CSRG Berkeley удалил большие куски кода, необходимого для создания реально загружающейся системы (по причине различных лицензионных требований), и фактически порт 4.4BSD для платформы Intel был очень неполным. Проекту потребовалось время почти до ноября 1994 года для того, чтобы выполнить этот переход и на этом этапе FreeBSD 2.0 была опубликована в сети и на CDROM (в конце декабря). Несмотря на множество "острых углов" в этой версии, она пользовалась значительным успехом и была продолжена более устойчивой и простой в установке FreeBSD 2.0.5, выпущенной в июне 1995 года.

Мы выпустили FreeBSD 2.1.5 в августе 1996, и она стала достаточно популярной среди большого количества ISP и коммерческих производителей, чтобы выпустить еще один релиз из ветви 2.1-STABLE. Это была FreeBSD 2.1.7.1, вышедшая в феврале 1997 и завершившая главную ветвь разработки 2.1-STABLE. Сейчас в режиме поддержки, в эту ветвь (RELENG\_2\_1\_0) вносятся только расширения безопасности и другие критически важные исправления.

FreeBSD 2.2 была ответвлена от основной линии разработки ("-CURRENT") в ноябре 1996 как ветвь RELENG\_2\_2, а первая полная версия (2.2.1) появилась в апреле 1997. Последующие версии ветви 2.2 появлялись летом и в конце 97 года, а последняя версия (2.2.8) вышла в ноябре 1998. Первая официальная версия 3.0 была подготовлена к выходу в октябре 1998, завершив развитие ветви 2.2

Третье ветвление произошло 20 января 1999 года, появились ветви 4.0-CURRENT и 3.X-STABLE. Из ветви 3.X-STABLE были получены: 3.1 - 15 февраля 1999, 3.2 - 15 мая 1999, 3.3 - 16 сентября 1999, 3.4 - 20 декабря 1999, 3.5 - 24 июня 2000, за которым последовал через несколько дней немного обновленный 3.5.1, который содержал несколько исправлений в области безопасности Kerberos. Это был последний релиз из ветви 3.X.

Другое ветвление было выполнено 13 марта 2000 года, в результате чего появилась ветвь 4.X-STABLE. Из этой ветви было выпущено несколько релизов: 4.0-RELEASE был представлен в марте 2000 года,

самый свежий 4.10-RELEASE был выпущен May 2004. Из ветви 4.X-stable (RELENG\_4) будут выпущены и следующие релизы.

Долгожданный 5.0-RELEASE был анонсирован 19 января 2003 года. Он стал кульминацией приблизительно трех лет работы, с этого релиза начался курс FreeBSD на расширенную поддержку мультипроцессорности и потоков в приложениях, была также представлена поддержка платформ UltraSPARC и ia64. За этим релизом последовал релиз 5.1 в июне 2003 года. Последним релизом 5.X из ветви -CURRENT стал 5.2.1-RELEASE, представленный в феврале 2004.

Ветвь RELENG\_5 была создана в августе 2004, после чего был выпущен релиз 5.3-RELEASE, который открыл серию релизов из ветви 5-STABLE.

На данный момент, долговременные разработки и проекты продолжаются в ветке 6.X-CURRENT, и по ходу разработки будут доступны снэпшот-релизы 6.X на CDROM (и, конечно же, в сети), постоянно выкладываемые на сервер снэпшотов (<ftp://current.FreeBSD.org/pub/FreeBSD/snapshots/>) как промежуточные результаты.

### 1.3.2. Цели Проекта FreeBSD

*Предоставил Jordan Hubbard.*

Целью Проекта FreeBSD является предоставление программного обеспечения, которое может быть использовано для любых целей и без дополнительных ограничений. Многие из нас вносят значительный вклад в код (и проект) и совершенно не требуют за это какой-либо финансовой компенсации сейчас и в последствии, хотя мы определенно не собираемся отказываться от нее. Мы верим, что первая и основная наша "миссия" это предоставление кода для всех, кому он необходим, и для любых целей, поэтому этот код становится все более и более распространен и предоставляет самые широкие возможности. Это, я верю, является одной из основных целей Свободного Программного Обеспечения и мы с энтузиазмом поддерживаем ее.

Тот код в нашем дереве исходных текстов, который попадает под Общую Публичную Лицензию GNU (GPL) или Общую Публичную Лицензию Библиотек GNU (LGPL), предоставляется с дополнительными условиями, хотя они обеспечивают только возможность доступа а не его ограничение. По причине дополнительных сложностей, которые могут появиться при коммерческом использовании GPL продуктов, мы предпочитаем ПО, предоставленное под более свободной лицензией BSD, когда это возможно.

### 1.3.3. Модель Разработки FreeBSD

*Предоставил Satoshi Asami.*

Разработка FreeBSD это очень открытый и гибкий процесс — FreeBSD, буквально выражаясь, создан из кода, предоставленного сотнями людей со всего мира, в чем вы можете убедиться взглянув на список этих людей ([http://www.FreeBSD.org/doc/en\\_US.ISO8859-1/articles/contributors/article.html](http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/contributors/article.html)). Мы постоянно ищем новых разработчиков и новые идеи, и тот, кто заинтересован в более тесном взаимодействии и хочет принять участие в проекте, должен просто связаться с нами в рассылке `freebsd-hackers` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers>). Для тех кто желает использовать FreeBSD в различных областях для работы, доступен Список рассылки анонсов FreeBSD (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce>).



Для участия в процессе разработки FreeBSD независимо или тесно сотрудничая с нами, полезно знать следующее:

## CVS Репозиторий

Главное дерево исходных текстов FreeBSD поддерживается с помощью CVS (<http://www.cvshome.org/>) (Concurrent Versions System), свободно доступной системой контроля исходных текстов, которая поставляется вместе с FreeBSD. Основной CVS репозиторий (<http://www.FreeBSD.org/cgi/cvsweb.cgi>) располагается на компьютере, находящемся в городе Санта Клара, Калифорния (США), откуда и распространяется на множество зеркал по всему миру. Дерево CVS, содержащее ветви -CURRENT и -STABLE, может быть легко скопировано на ваш локальный компьютер. Обратитесь к разделу Синхронизация дерева исходных текстов за информацией об этом процессе.

## Список коммиттеров

*Коммиттеры* это люди, которые имеют доступ на *запись* к главному дереву CVS, и имеют право вносить изменения в главное дерево исходных текстов FreeBSD (термин "коммиттер" появился от названия команды `cvsv(1) commit`, которая используется для внесения изменений в CVS репозиторий). Лучший способ для предоставления ваших изменений на рассмотрение коммиттеров - это использование команды `send-pr(1)`. Если что-то произошло с системой, вы можете достучаться до них посылкой письма по адресу `cvs-committers`.

## CORE группа FreeBSD

*CORE группа FreeBSD* могла бы быть эквивалентом Совета Директоров, если бы Проект FreeBSD был компанией. Главная задача CORE группы — убедиться, что проект в целом в хорошем состоянии и движется в правильном направлении. Приглашение постоянных и ответственных разработчиков присоединится к группе коммиттеров одна из функций CORE группы, также как приглашение новых членов в CORE группу по мере того, как другие уходят. Нынешний состав группы был выбран из рядов коммиттеров путем общего голосования в июле 2004 года. Выборы проходят каждые 2 года.

Некоторые члены CORE группы имеют особые области ответственности, это означает, что они являются ответственными за работу отдельной большой части системы. За полным списком разработчиков FreeBSD и областей их ответственности обращайтесь к Списку участников ([http://www.FreeBSD.org/doc/en\\_US.ISO8859-1/articles/contributors/article.html](http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/contributors/article.html)).

**Замечание:** Большинство членов CORE группы являются волонтерами и не получают никакой финансовой выгоды от участия в проекте, поэтому вы не должны рассматривать их работу как "гарантированную поддержку". Аналогия с "советом директоров" не очень точна и вероятно гораздо правильнее будет сказать, что это люди, которые посвятили себя FreeBSD, пожертвовав личной жизнью!

## Внешняя помощь

Последней, но однозначно не менее значимой, наибольшей группой разработчиков являются пользователи, которые предоставляют комментарии и исправления ошибок нам на почти постоянной основе. Основной путь участвовать в не централизованной разработке это подписка на

Список рассылки FreeBSD, посвящённый техническим дискуссиям (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers>), где обсуждаются подобные вещи. Обратитесь к Прил. С за дальнейшей информацией о различных списках рассылки FreeBSD.

#### *Список контрибьюторов FreeBSD*

([http://www.FreeBSD.org/doc/en\\_US.ISO8859-1/articles/contributors/article.html](http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/contributors/article.html)) очень длинный и постоянно растёт, поэтому почему бы вам не присоединится к нему предоставив что-нибудь FreeBSD сегодня?

Предоставление кода не единственный способ помочь проекту; для более-менее полного списка вещей, которые необходимо сделать пожалуйста обратитесь к веб сайту проекта FreeBSD (<http://www.FreeBSD.org/ru/index.html>).

Вообще говоря, наша модель разработки организована как "нечеткий набор концентрированных колец". Централизованная модель разработана для удобства *пользователей* FreeBSD, которые получают простую систему контроля за одной центральной базой кода, и позволяет не оставить за бортом проекта потенциальных помощников! Мы желаем предоставить стабильную операционную систему с большим количеством когерентных прикладных программ, которые пользователи смогут легко установить и использовать — наша модель работает очень хорошо для этой задачи.

Мы все приглашаем тех, кто хотел бы присоединится к нам как разработчик FreeBSD, выделить немного времени и привнести свой взнос в постоянный успех FreeBSD!

### 1.3.4. Текущая Версия FreeBSD

FreeBSD - это свободно доступный, основанный на 4.4BSD-Lite релиз с полными исходными текстами для компьютерных систем, основанных на Intel i386™, i486™, Pentium®, Pentium Pro, Celeron®, Pentium II, Pentium III, Pentium 4 (или совместимыми), Xeon™, DEC Alpha и Sun UltraSPARC. В основном он основан на программном обеспечении от группы CSRG, U.C. Berkley, с некоторым дополнениями из NetBSD, OpenBSD, 386BSD и Free Software Foundation.

Со времен FreeBSD версии 2.0 в конце 94, производительность, возможности, и стабильность FreeBSD выросла многократно. Самое большое изменение это полное обновление системы виртуальной памяти с объединённым VM/файловым буферизованным кэшем, который не только увеличивает производительность, но и уменьшает количество используемой памяти, делая 5 MB конфигурацию более доступным минимумом. Другие улучшения включают в себя полную поддержку NIS (клиент и сервер), поддержка транзакций TCP, поддержка "звонка-по-запросу" в PPP, встроенная поддержка DHCP, улучшенная подсистема SCSI, поддержка ISDN, ATM, FDDI, Fast и Gigabit Ethernet (1000 Mbit) адаптеров, улучшенная поддержка последних версий контролеров Adaptec и многие тысячи исправленных ошибок.

В дополнение к базовой системе, FreeBSD предоставляет коллекцию портированного ПО, включающую в себя тысячи популярных программ. На время подготовки этого документа она включала в себя более 11,800 портов! В коллекцию входят множество программ от http (WWW) серверов, до игр, языков программирования, текстовых редакторов и всего прочего. Полная коллекция портов требует приблизительно 300 MB дискового пространства, потому что порт представляет собой "изменения" оригинальных исходных текстов. Это сильно упрощает нам процесс обновления портов, и очень сильно уменьшает количество занимаемого дискового места по сравнению со старой (1.0) коллекцией портов. Для того, чтобы скомпилировать и установить программу, необходимо всего лишь войти в каталог

порта программы, набрать `make install` и дать системе сделать все остальное. Исходные тексты для каждого порта, который вы устанавливаете, загружаются автоматически с CDROM или локального FTP сервера, поэтому вам нужно только дисковое пространство для сборки тех портов, которые необходимы. Почти каждый порт также предоставляется как пред-компилированный "пакет", который может быть установлен с помощью простой команды (`pkg_add`) теми, кто предпочитает не компилировать порты из исходных текстов. Дополнительная информация о пакетах и портах находится в Гл. 4.

Множество дополнительных документов, которые вы можете найти очень полезными в процессе установки и использования FreeBSD, находятся в каталоге `/usr/share/doc` на любой машине, работающей под управлением современной версии FreeBSD. Вы можете просматривать локально установленные документы с помощью любого браузера, поддерживающего HTML, используя следующие ссылки:

#### Руководство FreeBSD

[/usr/share/doc/ru\\_RU.KOI8-R/books/handbook/index.html](/usr/share/doc/ru_RU.KOI8-R/books/handbook/index.html)

#### FreeBSD FAQ (Часто задаваемые вопросы)

[/usr/share/doc/ru\\_RU.KOI8-R/books/faq/index.html](/usr/share/doc/ru_RU.KOI8-R/books/faq/index.html)

Вы также можете просмотреть основные (и наиболее часто обновляемые) копии на <http://www.FreeBSD.org/ru/> (<http://www.FreeBSD.org/ru/index.html>).

# Глава 2. Установка FreeBSD

*Реструктурировал, исправил и частично переписал Jim Mock. Обзор sysinstall, скриншоты и общее руководство Randy Pratt. Перевод на русский язык: Денис Пеплин.*

## 2.1. Краткий обзор

FreeBSD поставляется простой в использовании текстовой программой установки **sysinstall**. Это основная программа установки FreeBSD, хотя поставщики могут предлагать свои программы. В этой главе описывается использование **sysinstall** для установки FreeBSD.

Прочтя эту главу, вы узнаете:

- Как создать дискеты для установки FreeBSD.
- Как FreeBSD видит и делит на разделы жесткие диски.
- Как запустить **sysinstall**.
- Вопросы, которые **sysinstall** задаст вам, что имеется ввиду, и как ответить на эти вопросы.

Перед прочтением этой главы вам потребуется:

- Прочитать информацию о поддерживаемом оборудовании, поставляемую с устанавливаемой версией FreeBSD, и убедиться, что ваше оборудование поддерживается.

**Замечание:** Как правило, эти инструкции по установке написаны для i386 ("PC совместимых") компьютеров. Когда это возможно, приводятся инструкции, специфичные для других платформ (например, Alpha). Хотя это руководство поддерживается в актуальном состоянии настолько, насколько это возможно, вы можете обнаружить небольшие различия между программой установки и тем, что показано здесь. Предполагается, что вы будете использовать эту главу в качестве общего руководства, а не как пошаговую инструкцию по установке.

## 2.2. Перед установкой

### 2.2.1. Соберите информацию о компьютере

Перед установкой FreeBSD попытайтесь собрать информацию об устройствах компьютера. Во время установки FreeBSD покажет информацию об устройствах (жестких дисках, сетевых картах, CDROM и т.д.) с номером модели и производителем. FreeBSD также попытается определить правильную конфигурацию для этих устройств, включая информацию об IRQ и портах ввода-вывода. Из-за возможных проблем с оборудованием этот процесс не всегда завершается успешно, и возможно вам придется исправлять определенную FreeBSD конфигурацию.

Если у вас уже есть установленная операционная система, например Windows или Linux, неплохо будет использовать ее возможности для просмотра настроек оборудования. Если вы не уверены, какие настройки карты расширения использовать, можете найти их на самой карте. Часто используемые

номера прерываний 3, 5 и 7, порты ввода- вывода обычно пишутся в шестнадцатеричном виде, например 0x330.

Мы рекомендуем распечатать эту информацию перед установкой FreeBSD. Вам может помочь использование таблицы вроде этой:

**Таблица 2-1. Пример сведений об оборудовании**

Название устройства	IRQ	Порт ввода-вывода	Примечания
Первый жесткий диск	нет	нет	40 GB, Seagate, первый IDE master
CDROM	нет	нет	Первый IDE slave
Второй жесткий диск	нет	нет	20 GB, IBM, второй IDE master
Первый IDE контроллер	14	0x1f0	
Сетевая карта	нет	нет	Intel® 10/100
Модем	нет	нет	3Com® 56K факс-модем, COM1
...			

### 2.2.2. Сделайте резервное копирование данных

Если компьютер, на который вы устанавливаете FreeBSD, содержит важные данные, убедитесь в наличии резервных копий и проверьте их сохранность перед установкой FreeBSD. Во время установки FreeBSD запросит подтверждение перед тем, как записать данные на диск, но, если процесс запущен, изменения нельзя отменить.

### 2.2.3. Решите куда установить FreeBSD

Если вы хотите, чтобы FreeBSD использовала весь жесткий диск, не о чем беспокоиться — можете пропустить этот раздел.

Однако, если нужно совмещать FreeBSD с другими операционными системами, необходимо иметь представление как данные размещаются на диске и как это касается вас.

#### 2.2.3.1. Разделы диска для i386™

Диск PC может быть поделен на отдельные части. Эти части называют *разделами*. Первоначально PC поддерживал только четыре раздела на диск. Эти разделы называются *главными разделами*. Чтобы обойти это ограничение и дать возможность создавать более чем четыре раздела, был создан новый тип раздела, *расширенный раздел*. Диск может содержать только один расширенный раздел. Специальные разделы, называемые *логическими разделами*, могут быть созданы внутри расширенного раздела.

Каждый раздел имеет *ID раздела* — номер, который используется для определения типа данных на разделе. FreeBSD использует ID раздела 165.

Как правило, каждая операционная система, которую вы используете, определяет разделы своим способом. Например, DOS и ее потомки, такие как Windows, присваивают каждому главному и логическому разделу *букву диска*, начиная с c:.

FreeBSD нужно устанавливать в главный раздел. FreeBSD может хранить все свои данные, включая создаваемые вами файлы, на этом одном разделе. Тем не менее, если дисков много, вы можете создать разделы FreeBSD на всех дисках или на некоторых из них. При установке FreeBSD должен быть доступен по крайней мере один раздел. Это может быть чистый, подготовленный для установки раздел, или раздел с данными, которые больше не нужны.

Если все разделы на диске уже используются, вы должны освободить один из них для FreeBSD, используя программы, поставляемые с имеющейся операционной системой (например, `fdisk` для DOS или Windows).

Если есть резервный раздел, используйте его. Однако, возможно сначала придется ужать один или несколько существующих разделов.

FreeBSD для установки нужен диск не менее 100 MB. Однако, это *очень* минимальная установка, при которой не останется места для ваших личных файлов. Более реальный объем — 250 MB без графической оболочки, и более 350 MB с графической оболочкой. Если вы собираетесь устанавливать большое количество дополнительного ПО, понадобится еще больше дискового пространства.

Вы можете использовать коммерческие программы, такие как **PartitionMagic®**, для изменения размера разделов и освобождения места под FreeBSD. Каталог `tools` на CDRом содержит две свободно распространяемые утилиты, которые могут быть использованы для этой задачи. Они называются **FIPS** и **PResizer**. Документация на обе эти утилиты доступна из того же каталога. **FIPS**, **PResizer**, и **PartitionMagic** могут изменять размер разделов FAT16 и FAT32, используемых системами от MS-DOS до Windows ME. **PartitionMagic** это единственная известная программа, способная изменять размер NTFS.

**Внимание:** Неправильное использование этих утилит может привести к уничтожению данных на диске. Удостоверьтесь в наличии свежих и исправных резервных копий данных перед их использованием.

### Пример 2-1. Использование существующего раздела без изменения

Представьте что в компьютере один 4 GB диск, на котором уже установлена Windows, и диск разбит на два логических диска C: и D:, каждый по 2 GB. 1 GB данных на C:, и 0.5 GB данных на D:.

Это означает, что диск состоит из двух разделов, по одному на каждую букву. Вы можете скопировать все данные с D: на C:, это освободит второй раздел для FreeBSD.

### Пример 2-2. Сжатие существующих разделов

Представьте, что в компьютере один 4 GB диск, на котором уже установлена Windows. При установке Windows вы создали один большой раздел, получив при этом диск C: размером 4 GB. Вы используете 1.5 GB, и хотите выделить 2 GB для FreeBSD.

Для установки FreeBSD нужно выбрать:

1. Сделать резервную копию Windows, затем переустановить Windows, выделив 2 GB под ее раздел.
2. Использовать одну из утилит, таких как **PartitionMagic**, описанную выше, для сжатия раздела Windows.

### 2.2.3.2. Разделы диска для Alpha

Вам потребуется выделенный диск для FreeBSD на Alpha. В настоящее время невозможно установить две операционные системы на диск. В зависимости от конкретного компьютера Alpha, диск может быть SCSI или IDE, главное, чтобы ваш компьютер мог с него загрузиться.

В соответствии с соглашениями, принятыми в руководствах Digital / Compaq, весь SRM ввод показан в верхнем регистре. SRM нечувствителен к регистру.

Для поиска имен и типов дисков на компьютере используйте команду `SHOW DEVICE SRM` консоли:

```
>>>SHOW DEVICE
dka0.0.0.4.0          DKA0          TOSHIBA CD-ROM XM-57   3476
dkc0.0.0.1009.0       DKC0          RZ1BB-BS   0658
dkc100.1.0.1009.0     DKC100        SEAGATE ST34501W   0015
dva0.0.0.0.1          DVA0
ewa0.0.0.3.0          EWA0          00-00-F8-75-6D-01
pkc0.7.0.1009.0       PKC0          SCSI Bus ID 7   5.27
pqa0.0.0.4.0          PQA0          PCI EIDE
pqb0.0.1.4.0          PQB0          PCI EIDE
```

Этот пример с Digital Personal Workstation 433au показывает три подключенных диска. Первый — устройство CDROM с именем DKA0, другие два называются DKC0 и DKC100 соответственно.

Диски с именами DKx — SCSI диски. Например, DKA100 означает SCSI диск со SCSI целевым ID 1 на первой SCSI шине (A), в то время как DKC300 означает SCSI диск с SCSI ID 3 на третьей SCSI шине (C). Имя устройства PKx означает SCSI адаптер. Как показано в выводе `SHOW DEVICE`, SCSI устройство CDROM определяется как любой другой жесткий диск SCSI.

Имена IDE дисков выглядят как DQx, в то время как PQx — IDE контроллер.

### 2.2.4. Соберите информацию о конфигурации сети

Если вы хотите подключиться к сети в процессе установки FreeBSD (например, при установке с FTP или NFS сервера), нужно знать конфигурацию сети. Вам будет предложено ввести эту информацию, чтобы FreeBSD смогла подключиться к сети для продолжения установки.

#### 2.2.4.1. Подключение к сети Ethernet, или через кабельный/DSL модем

Если вы подключаетесь к сети Ethernet, или соединение с интернет подключено к Ethernet через кабельный или DSL модем, понадобится следующая информация:

1. IP адрес
2. IP адрес шлюза по умолчанию
3. Имя хоста
4. IP адрес DNS сервера

## 5. Маска подсети

Если у вас нет этой информации, спросите системного администратора или провайдера интернет. Они могут сказать, что данные присваиваются автоматически, с использованием *DHCP*. Если это так, запомните это.

### 2.2.4.2. Подсоединение с помощью модема

Если вы дозваниваетесь до провайдера с помощью обычного модема, вы все же сможете установить FreeBSD через интернет, но это займет очень много времени.

Вам нужно знать:

1. Номер телефона провайдера
2. COM порт, к которому подключен модем
3. Имя пользователя и пароль учетной записи для доступа в интернет

### 2.2.5. Проверьте сведения об обнаруженных ошибках FreeBSD

Хотя проект FreeBSD борется за то, чтобы каждый релиз FreeBSD был настолько стабильным, насколько это возможно, ошибки порой вкрадываются в процесс разработки. В очень редких случаях эти ошибки влияют на процесс установки. Как только эти проблемы обнаруживаются и исправляются, они попадают в сообщения об ошибках FreeBSD (<http://www.FreeBSD.org/ru/releases/5.3-R/errata.html>), находящиеся на сайте FreeBSD. Вы можете проверить сообщения об ошибках перед установкой, чтобы убедиться, что не существует проблем, о которых стоит беспокоиться.

Информация о релизах, включая сообщения об ошибках каждого релиза, находится на странице информации о релизах (<http://www.FreeBSD.org/ru/releases/index.html>) сайта FreeBSD (<http://www.FreeBSD.org/ru/index.html>).

### 2.2.6. Получение установочных файлов FreeBSD

Программа установки FreeBSD может установить FreeBSD из файлов, расположенных в одном из следующих мест:

#### Локальный диск

- CDROM или DVD
- Раздел DOS на вашем компьютере
- Лента SCSI или QIC
- Гибкие диски

#### Сеть

- FTP сервер (через файрволл или HTTP прокси, если потребуется)
- NFS сервер



- Соединение через параллельный или последовательный порт

Если вы купили FreeBSD на CD или DVD, у вас уже есть все, что нужно, переходите к следующему разделу (Разд. 2.2.7).

Если у вас нет установочных файлов FreeBSD, перейдите к Разд. 2.13, который описывает, как подготовиться к установке FreeBSD любым указанным выше способом. После прочтения этого раздела, вернитесь сюда и прочтите Разд. 2.2.7.

## 2.2.7. Подготовка загрузочных дисков

Процесс установки FreeBSD начинается с загрузки в ваш компьютер программы установки FreeBSD — эта программа не запускается из других операционных систем. Компьютер обычно загружает операционную систему, установленную на жестком диске, но также может быть настроен для использования "загрузочной" дискеты. Большинство современных компьютеров могут также загрузиться с компакт-диска в приводе CDROM.

**Подсказка:** Если у вас есть FreeBSD на CDROM или DVD (купленный или записанный самостоятельно), и компьютер позволяет загрузку с CDROM или DVD (обычно этот пункт в BIOS называется "Boot Order" или что-то вроде), можете пропустить этот раздел. Образы FreeBSD CDROM и DVD являются загрузочными и могут быть использованы для установки FreeBSD без какой-либо специальной подготовки.

Для создания загрузочных дисков сделайте следующее:

### 1. Получение образов загрузочных дисков

Загрузочные диски есть на установочном диске в каталоге `floppies/`, и могут быть также закачаны из каталога `floppies`,

`ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/<arch>/<version>-RELEASE/floppies/`.

Замените `<arch>` и `<version>` архитектурой компьютера и номером версии, которую вы хотите установить, соответственно. Например, загрузочные дискеты FreeBSD 5.3-RELEASE для i386 находятся в `ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/5.3-RELEASE/floppies/`.

Расширение файла образа диска `.flp`. Каталог `floppies/` содержит множество разных образов, ваш выбор будет зависеть от устанавливаемой версии FreeBSD и, в некоторых случаях, от конфигурации компьютера. Если вы устанавливаете FreeBSD 4.X, в большинстве случаев понадобятся всего лишь два файла, `kern.flp` и `mfsroot.flp`. При установке FreeBSD 5.X в большинстве случаев потребуются три дискеты, `boot.flp`, `kern1.flp`, и `kern2.flp`. Обратитесь к файлу `README.TXT` в том же каталоге за свежей информацией об этих образах дисков.

**Замечание:** Дополнительные драйверы устройств могут потребоваться в некоторых системах 5.X, старше чем FreeBSD 5.3. Эти драйверы находятся в образе `drivers.flp`.

**Важно:** FTP клиент должен использовать *бинарный режим* для загрузки образов дисков. Некоторые веб браузеры используют *текстовый (ASCII)* режим, который точно не позволит загрузиться с этих дисков.

## 2. Подготовка дискет

Необходимо подготовить по одной дискете на каждый загруженный образ. Эти дискеты должны быть без дефектов. Лучший способ проверить это — отформатировать дискеты самостоятельно. Не доверяйте заводскому форматированию дискет. Утилита форматирования в Windows не сообщит о наличии плохих секторов, она просто пометит их как "плохие" и проигнорирует. Советуем использовать новые дискеты если вы выбрали этот способ установки.

**Важно:** Если при попытке установки FreeBSD программа установки рушится, зависает, или делает что-то не так, сразу проверьте дискеты. Попробуйте записать образы на новые дискеты и попытайтесь еще раз.

## 3. Запись образов на дискеты

Файлы с расширением `.flp` это *не* обычные файлы, которые можно записать на диск. Это образы всего содержимого диска. Это означает, что вы *не можете* просто скопировать их с одного диска на другой. Вместо этого, нужно использовать специальные утилиты для записи образов на диск.

Если вы записываете дискеты на компьютере под MS-DOS/Windows, используйте утилиту `fdimage`.

Если вы используете образы с CDROM, и буква вашего CDROM `E:`, запустите ее так:

```
E:\> tools\fdimage floppies\kern.flp A:
```

Повторите эту команду для каждого файла `.flp`, вставляя новую дискету каждый раз, пометьте каждый диск именем файла, который вы скопировали на него. Измените команду если потребуется, в зависимости от места, куда вы поместили файлы `.flp`. Если у вас нет CDROM, `fdimage` может быть загружена из каталога `tools` (<ftp://ftp.FreeBSD.org/pub/FreeBSD/tools/>) FTP сервера FreeBSD.

Если вы записываете дискеты под UNIX (например, под другой системой FreeBSD), используйте утилиту `dd(1)` для записи образов непосредственно на дискеты. Под FreeBSD запустите:

```
# dd if=kern.flp of=/dev/fd0
```

Под FreeBSD, `/dev/fd0` означает первый гибкий диск (диск A:). `/dev/fd1` будет диском B:, и так далее. Другие UNIX системы могут по-другому именовать устройства гибких дисков, вам возможно понадобится прочитать документацию по соответствующей системе.

Теперь вы готовы к установке FreeBSD.

## 2.3. Начало установки

**Важно:** Как правило, программа установки не будет производить никаких изменений на дисках, пока не выдаст следующее сообщение:

```
Last Chance: Are you SURE you want continue the installation?
```

```
If you're running this on a disk with data you wish to save then WE
STRONGLY ENCOURAGE YOU TO MAKE PROPER BACKUPS before proceeding!
```

```
We can take no responsibility for lost disk contents!
```

Установка может быть прервана в любой момент до этого предупреждения без каких-либо изменений на жестком диске. Если вы считаете, что что-то настроили неправильно, можете просто выключить компьютер без риска что-либо повредить.

## 2.3.1. Загрузка

### 2.3.1.1. Загрузка i386

1. Компьютер выключен.
2. Включите компьютер. После включения он должен показать способ входа в меню установки BIOS, как правило это клавиши **F2**, **F10**, **Del**, или **Alt+S**. Используйте те клавиши, которые показаны на экране. В некоторых случаях компьютер может показывать картинку после запуска. Как правило, нажатие **Esc** уберет картинку и позволит вам увидеть необходимую информацию.
3. Найдите установки системы, указывающие ей с какого устройства загружаться. Обычно они обозначаются как "Boot Order", и там как правило отображен список устройств, таких как Floppy, CDROM, First Hard Disk, и так далее.

Если вы подготовили дискеты, убедитесь, что выбран Floppy. Если вы загружаетесь с CDROM, убедитесь, что выбран он. Если вы не уверены, посмотрите руководство к компьютеру и/или к его материнской плате.

Сделайте изменения, затем сохраните их и выйдите. Компьютер должен перезагрузиться.

4. Если вы подготовили дискеты, как описано в Разд. 2.2.7, одна из них будет первым загрузочным диском; как правило, это дискета с kern.flp. Вставьте эту дискету в дисковод.

Если вы загружаетесь с CDROM, потребуется сначала включить компьютер и вставить компакт-диск, как только это станет возможно.

Если компьютер запускается как обычно, и загружает существующую операционную систему, возможны следующие причины:

1. Диск был вставлен недостаточно рано в процессе загрузки. Оставьте его внутри и перегрузите компьютер.
  2. Установки BIOS, измененные ранее, действуют неправильно. Надо изменять их, пока они не заработают.
  3. BIOS вашего компьютера не поддерживает загрузку с выбранного типа носителя.
5. FreeBSD начнет загрузку. Если загрузка происходит с CDROM, вы увидите что-то вроде этого (информация о версии удалена):

```
Verifying DMI Pool Data .....
Boot from ATAPI CD-ROM :
  1. FD 2.88MB System Type-(00)
Uncompressing ... done

BTX loader 1.00 BTX version is 1.01
Console: internal video/keyboard
BIOS drive A: is disk0
```

```

BIOS drive B: is disk1
BIOS drive C: is disk2
BIOS drive D: is disk3
BIOS 639kB/261120kB available memory

```

```
FreeBSD/i386 bootstrap loader, Revision 0.8
```

```
/kernel text=0x277391 data=0x3268c+0x332a8 |
```

```

|
Hit [Enter] to boot immediately, or any other key for command prompt.
Booting [kernel] in 9 seconds... _

```

Если происходит загрузка с дискеты, вы увидите что-то вроде этого (информация о версии удалена):

```
Verifying DMI Pool Data .....
```

```

BTX loader 1.00  BTX version is 1.01
Console: internal video/keyboard
BIOS drive A: is disk0
BIOS drive C: is disk1
BIOS 639kB/261120kB available memory

```

```
FreeBSD/i386 bootstrap loader, Revision 0.8
```

```
/kernel text=0x277391 data=0x3268c+0x332a8 |
```

```
Please insert MFS root floppy and press enter:
```

Следуя инструкциям, уберите дискету с `kern.flp`, вставьте дискету с `mfsroot.flp` и нажмите **Enter**. В FreeBSD 5.3 и выше набор дисков другой, он подробно описан в предыдущем разделе. Загрузитесь с первой дискеты; последовательно вставляйте остальные диски при появлении соответствующего приглашения.

- Идет ли загрузка с дискет или с CDROM, в процессе загрузки появится:

```

Hit [Enter] to boot immediately, or any other key for command prompt.
Booting [kernel] in 9 seconds... _

```

Подождите десять секунд или нажмите **Enter**. (в FreeBSD 4.X будет запущено меню конфигурации ядра).

### 2.3.1.2. Загрузка Alpha

- Компьютер выключен.
- Включите компьютер и дождитесь приглашения к загрузке на мониторе.
- Если вы подготовили загрузочные дискеты, как описано в Разд. 2.2.7, одна из них будет первым загрузочным диском; как правило, это дискета с `kern.flp`. Вставьте эту дискету в дисковод и наберите следующую команду для загрузки (замените на имя вашего дисковода если потребуется):

```
>>>BOOT DVA0 -FLAGS " -FILE "
```

Если вы загружаетесь с CDROM, вставьте компакт-диск в привод и наберите для начала установки следующую команду (замените на имя соответствующего устройства CDROM если потребуется):

```
>>>BOOT DKA0 -FLAGS " -FILE "
```

4. FreeBSD начнет загружаться. Если загрузка идет с дискета, через некоторое время вы увидите сообщение:

```
Please insert MFS root floppy and press enter:
```

Следуя инструкциям, удалите дискету с `kern.flp`, вставьте дискету с `mfsroot.flp` и нажмите **Enter**.

5. Идет ли загрузка с дискет или с CDROM, в процессе загрузки появится:

```
hit [Enter] to boot immediately, or any other key for command prompt.  
Booting [kernel] in 9 seconds... _
```

Подождите десять секунд или нажмите **Enter**. Будет запущено меню конфигурации ядра.

### 2.3.2. Конфигурация ядра (kernel)

**Замечание:** Начиная с FreeBSD версии 5.0 и более поздних, `userconfig` был удален и заменен методом `device.hints(5)`. За более подробной информацией о `device.hints(5)` обращайтесь к Разд. 12.5

*kernel* это ядро операционной системы. В нем реализовано множество функций, включая доступ ко всем устройствам, которые могут быть в системе, таким как жесткие диски, сетевые карты, звуковые карты и т.д. Каждое устройство, поддерживаемое ядром FreeBSD имеет свой драйвер. У каждого драйвера есть имя из двух или трех букв, например `sa` для драйвера последовательного доступа SCSI, или `sio` для драйвера последовательного ввода-вывода (который управляет COM портами).

Когда ядро загружено, каждый драйвер проверяет систему на предмет наличия поддерживаемого оборудования. Если оно найдено, драйвер конфигурирует оборудование и делает его доступным остальной части ядра.

Такая проверка обычно называется *тестированием устройства*. К сожалению, это не всегда безопасно. Некоторые драйвера конфликтуют между собой, и проверка части оборудования иногда может привести другую часть в неработоспособное состояние. Это основное ограничение архитектуры PC.

Многие старые устройства называются ISA — в противоположность PCI. Стандарт ISA требует, чтобы в каждом устройстве была прописана некоторая информация, обычно номер Interrupt Request Line (IRQ) и адрес порта ввода-вывода, которые используются драйвером. Эта информация обычно устанавливается с использованием перемычек на карте, или с помощью DOS утилиты.

Часто это вызывало проблемы, поскольку было невозможно использовать два устройства с одинаковыми IRQ или адресами портов.

Новые устройства поддерживают PCI стандарт, который не требует установки параметров вручную, так как устройства способны общаться с BIOS и договариваться, какие IRQ и адреса портов ввода-вывода использовать.

Если в вашем компьютере есть устройства ISA, FreeBSD драйвер для этого устройства должен быть настроен с IRQ и адресом порта, установленными на карте. Для этого может понадобиться собрать сведения об оборудовании (Разд. 2.2.1).

К сожалению, номера IRQ и адреса портов, используемые некоторыми драйверами, конфликтуют. Это потому, что некоторые устройства ISA поставляются с конфликтующими номерами IRQ или адресами портов. По умолчанию драйвера FreeBSD предусмотрительно настроены для использования заводских установок, поэтому будут работать столько устройств, сколько возможно.

Это как правило не проблема при каждодневном использовании FreeBSD. В компьютере обычно нет двух конфликтующих устройств, поскольку в противном случае одно из них не будет работать (независимо от используемой операционной системы).

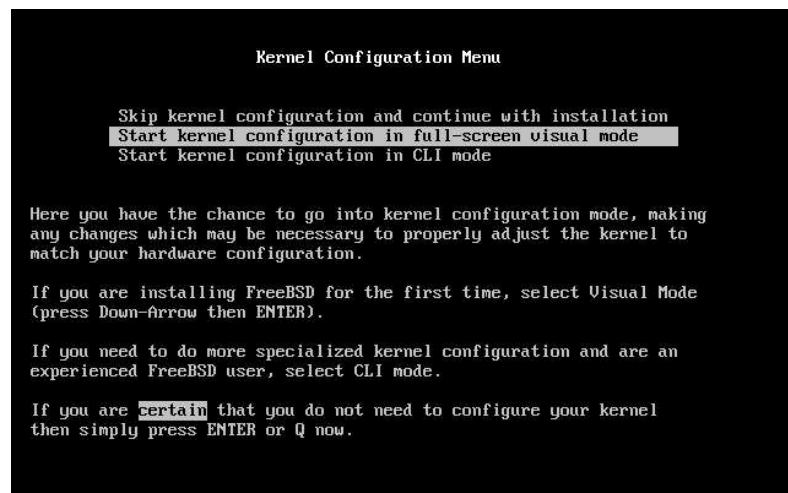
Это может стать проблемой при установке FreeBSD в первый раз, потому что ядро, используемое для установки, собрано с максимальным количеством драйверов для поддержки множества различных конфигураций оборудования. Это означает, что некоторые из драйверов имеют конфликтующие настройки. Устройства тестируются в строго определенном порядке, и если у вас есть устройство, тестирующееся позже, но конфликтующее с ранее протестированным устройством, оборудование может не работать или быть некорректно протестировано в процессе установки FreeBSD.

Поэтому первое, что вы должны сделать при установке FreeBSD — проверить список устройств, настроенных в ядре. Отключите некоторые из них, если таких устройств нет. Подтвердите (и измените) конфигурацию, если это устройство присутствует, но установки по умолчанию неверны.

Возможно, это звучит гораздо сложнее чем есть на самом деле.

На Рис. 2-1 показано главное меню конфигурации ядра. Мы рекомендуем выбрать опцию **Start kernel configuration in full-screen visual mode**, так как это самый простой интерфейс для нового пользователя.

Рисунок 2-1. Меню конфигурации ядра



Экран конфигурации ядра (Рис. 2-2) поделен на четыре части:

1. Свернутый список устройств, помеченных как "active", поделен на группы, такие как Storage, и Network. Каждый драйвер показан в виде описания, имени из двух или трех букв, используемых IRQ и порта. В дополнение, если какой-либо активный драйвер конфликтует с другим активным

драйвером, после имени драйвера пишется CONF. В этом разделе также показано общее число конфликтующих активных драйверов.

2. Драйверы, помеченные как неактивные. Они остаются в ядре, но не будут тестировать свои устройства при загрузке ядра. Они поделены на группы так же как и активные драйверы.
3. Дополнительная информация о выбранном в данный момент драйвере, включая IRQ и адрес порта.
4. Информация об управляющих клавишах, доступных в данный момент.

**Рисунок 2-2. Визуальный интерфейс конфигурации ядра**



Не беспокойтесь, если появятся сообщения о конфликтах, так и должно быть; все драйверы включены и как уже объяснялось, некоторые из них будут конфликтовать с другими.

Сейчас вам нужно просмотреть список драйверов и разрешить конфликты.

### Разрешение конфликтов драйверов

1. Нажмите **X**. Это полностью раскроет список драйверов и вы сможете увидеть их все. Вам понадобятся клавиши навигации для прокрутки назад и вперед в списке активных драйверов.

Рис. 2-3 показывает результат действия **X**.

Рисунок 2-3. Развернутый список драйверов

Active Drivers		7 conflicts	Dev	IRQ	Port
Storage :					
AdvanSys SCSI narrow controller			adu0		
Adaptec 154x SCSI controller			aha0		
Adaptec 152x SCSI and compatible sound cards			aic0		
ATA/ATAPI compatible disk controller			ata0	14	0x1f0
ATA/ATAPI compatible disk controller			ata1	15	0x170
Buslogic SCSI controller			bt0		
Floppy disk controller			fdc0	6	0x3f0
Inactive Drivers					
Storage :					
Network :					
Communications : (Collapsed)					
Input :					
Multimedia :					
Miscellaneous : (Collapsed)					
[Enter] Collapse device list [C] Collapse all lists					
[TAB] Change fields [Q] Save and Exit [?] Help					

- Отключите все драйвера устройств, которых у вас нет. Для отключения драйвера, переместитесь к нему с помощью клавиш навигации и нажмите **Del**. Драйвер будет помещен в список Inactive Drivers.

Если вы случайно отключили нужное устройство, нажмите **Tab** для переключения на Inactive Drivers, выберите отключенный драйвер, и нажмите **Enter**, чтобы вернуть его обратно в список активных драйверов.

**Внимание:** Не отключайте `sc0`. Он управляет экраном, и он нужен, если только вы не производите установку через последовательный порт.

**Внимание:** Отключайте `atkbd0` только если вы используете USB клавиатуру. Если клавиатура обычная, оставьте `atkbd0`.

- Если конфликтов нет, можете пропустить этот этап. В другом случае оставшиеся конфликты должны быть проверены. Если они не помечены как "allowed conflict" в окне сообщений, должны быть изменены или IRQ/адрес порта для тестирования устройства, или IRQ/адрес порта физического устройства.

Для изменения IRQ и адреса порта ввода-вывода драйвера, выберите устройство и нажмите **Enter**. Курсор переместится в третье окно и вы сможете изменить значения. Вам нужно ввести значения IRQ и адреса порта, определенные в процессе сбора информации об оборудовании. Нажмите **Q** для завершения редактирования конфигурации устройства и возврата в список активных устройств.

Если вы не уверены, какими должны быть эти значения, попробуйте использовать `-1`. Некоторые драйвера FreeBSD могут безопасно протестировать оборудование для определения правильных значений, и значение `-1` указывает им сделать это.

Процедура изменения адреса физического устройства различна для разных устройств. Для некоторых устройств вам может понадобиться извлечь карту из компьютера и установить



перемычки или DIP переключатели. Другие устройства могут поставляться с DOS дискетой, содержащей программы, используемые для перенастройки карты. В любом случае, обращайтесь к документации, поставляемой с устройством. Изменение адреса, несомненно, потребует перезагрузки компьютера, так что вам потребуется опять вернуться к процессу установки FreeBSD после перенастройки карты.

4. Когда все конфликты будут разрешены, экран будет выглядеть как Рис. 2-4.

**Рисунок 2-4. Конфигурация драйверов без конфликтов**

Active Drivers		Dev	IRQ	Port
Storage :				
ATA/ATAPI compatible disk controller		ata0	14	0x1f0
ATA/ATAPI compatible disk controller		ata1	15	0x170
Floppy disk controller		fdc0	6	0x3f0
Network :				
NE1000,NE2000,3C503,WD/SMC80xx Ethernet adapters		ed0	10	0x280
Communications :				
Parallel Port chipset		ppc0	7	
Inactive Drivers		Dev		
Storage :		(Collapsed)		
Network :		(Collapsed)		
Communications :		(Collapsed)		
Input :				
Multimedia :				
Miscellaneous :		(Collapsed)		
-----				
Port address : 0x1f0				
IRQ number :		14		
Flags :		0x0		
-----				
[Enter] Edit device parameters		[DEL] Disable device		
[TAB] Change fields		[Q] Save and Exit		[?] Help

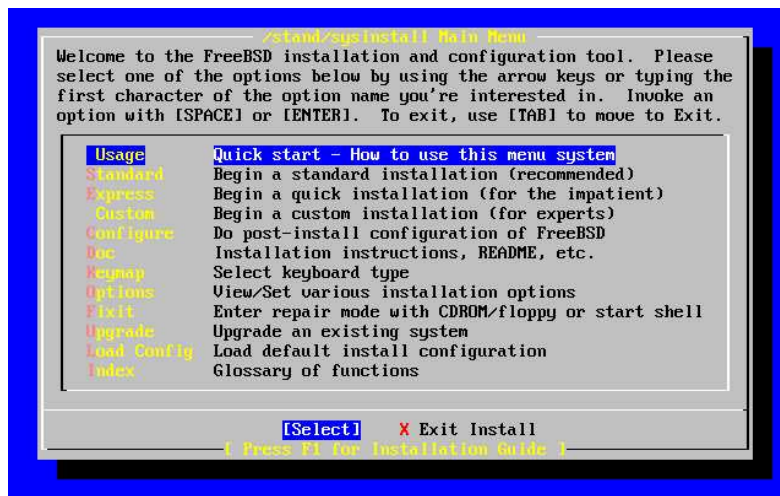
Как видите, список активных драйверов стал гораздо меньше, поскольку в нем только драйвера реально установленных устройств.

Теперь вы можете сохранить изменения и перейти к следующему пункту установки. Нажмите **Q** для выхода из меню конфигурации. Появится сообщение:

Save these parameters before exiting? ([Y]es/[N]o/[C]ancel)

Нажмите **Y** для сохранения параметров в памяти (они будут сохранены на диск после завершения установки) и тестирование начнется. После отображения белым по черному результатов тестирования запустится **sysinstall** и отобразится главное меню (Рис. 2-5).

Рисунок 2-5. Главное меню Sysinstall



### 2.3.3. Просмотр результатов тестирования устройств

Последние несколько сотен линий, отображенные на экране, сохраняются и могут быть просмотрены.

Для просмотра буфера нажмите **Scroll Lock**. Это включит прокрутку экрана. Вы можете использовать клавиши навигации или **PageUp** и **PageDown** для просмотра результатов. Нажмите **Scroll Lock** еще раз для отключения прокрутки.

Сделайте это сейчас для просмотра текста, ушедшего за экран, когда ядро закончило тестирование устройств. Вы увидите текст вроде Рис. 2-6, хотя в деталях он будет отличаться в зависимости от устройств, имеющихся в вашем компьютере.

Рисунок 2-6. Типичный вывод Device Probe

```

avail memory = 253050880 (247120K bytes)
Preloaded elf kernel "kernel" at 0xc0817000.
Preloaded mfs_root "/mfsroot" at 0xc0817084.
md0: Preloaded image </mfsroot> 4423680 bytes at 0xc03ddcd4

md1: Malloc disk
Using $PIR table, 4 entries at 0xc00fde60
npx0: <math processor> on motherboard
npx0: INT 16 interface
pcib0: <Host to PCI bridge> on motherboard
pci0: <PCI bus> on pcib0
pci1: <VIA 82C598MVP (Apollo MVP3) PCI-PCI (AGP) bridge> at device 1.0 on pci0
pci1: <PCI bus> on pci1
pci1: <Matrox MGA G200 AGP graphics accelerator> at 0.0 irq 11
isab0: <VIA 82C586 PCI-ISA bridge> at device 7.0 on pci0
isa0: <ISA bus> on isab0
atapci0: <VIA 82C586 ATA33 controller> port 0xe000-0xe00f at device 7.1 on pci0
ata0: at 0x1f0 irq 14 on atapci0

```

```

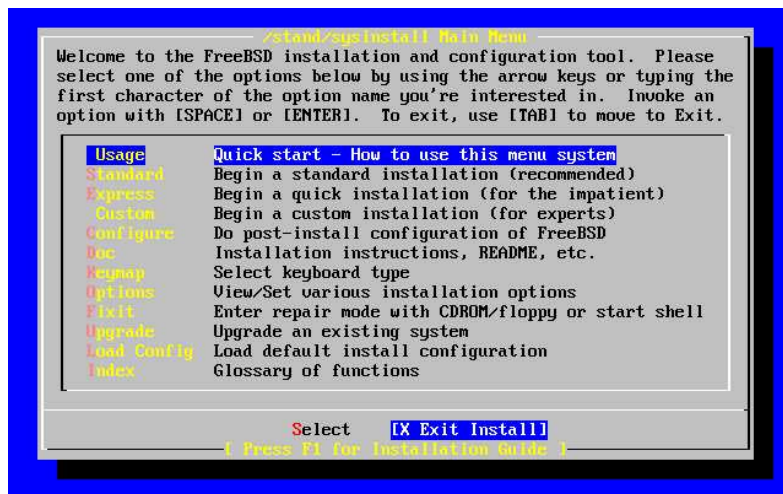
atal: at 0x170 irq 15 on atapci0
uhci0 <VIA 83C572 USB controller> port 0xe400-0xe41f irq 10 at device 7.2 on pci
0
usb0: <VIA 83572 USB controller> on uhci0
usb0: USB revision 1.0
uhub0: VIA UHCI root hub, class 9/0, rev 1.00/1.00, addr1
uhub0: 2 ports with 2 removable, self powered
pci0: <unknown card> (vendor=0x1106, dev=0x3040) at 7.3
dc0: <ADMtek AN985 10/100BaseTX> port 0xe800-0xe8ff mem 0xdb000000-0xeb0003ff ir
q 11 at device 8.0 on pci0
dc0: Ethernet address: 00:04:5a:74:6b:b5
miibus0: <MII bus> on dc0
ukphy0: <Generic IEEE 802.3u media interface> on miibus0
ukphy0: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
ed0: <NE2000 PCI Ethernet (RealTek 8029)> port 0xec00-0xec1f irq 9 at device 10.
0 on pci0
ed0 address 52:54:05:de:73:1b, type NE2000 (16 bit)
isa0: too many dependant configs (8)
isa0: unexpected small tag 14
orm0: <Option ROM> at iomem 0xc0000-0xc7fff on isa0
fdc0: <NEC 72065B or clone> at port 0x3f0-0x3f5,0x3f7 irq 6 drq2 on isa0
fdc0: FIFO enabled, 8 bytes threshold
fd0: <1440-KB 3.5" drive> on fdc0 drive 0
atkbd0: <Keyboard controller (i8042)> at port 0x60,0x64 on isa0
atkbd0: <AT Keyboard> flags 0x1 irq1 on atkbd0
kbd0 at atkbd0
psm0: <PS/2 Mouse> irq 12 on atkbd0
psm0: model Generic PS/@ mouse, device ID 0
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff on isa0
sc0: <System console> at flags 0x100 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
sio0 at port 0x3f8-0x3ff irq 4 flags 0x10 on isa0
sio0: type 16550A
sio1 at port 0x2f8-0x2ff irq 3 on isa0
sio1: type 16550A
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
pppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/15 bytes threshold
plip0: <PLIP network interface> on ppbus0
ad0: 8063MB <IBM-DHEA-38451> [16383/16/63] at ata0-master UDMA33
acd0: CD-RW <LITE-ON LTR-1210B> at atal-slave PIO4
Mounting root from ufs:/dev/md0c
/stand/sysinstall running as init on vty0

```

Внимательно проверьте результаты тестирования устройств и убедитесь, что FreeBSD обнаружила все устройства, какие нужно. Если устройство не найдено, его не будет в списке. Если драйвер устройства нуждается в настройке IRQ и адреса порта, проверьте, введены ли они правильно.

Если вам нужно внести изменения в UserConfig, потребуется выйти из **sysinstall** и начать сначала. Это также хороший путь познакомиться поближе с процессом конфигурации.

## Рисунок 2-7. Выбор Sysinstall Exit



Используйте клавиши навигации для выбора **Exit Install** из главного меню. Отобразится сообщение:

```

User Confirmation Requested
Are you sure you wish to exit? The system will reboot
(be sure to remove any floppies from the drives).

[ Yes ]    No

```

Программа установки запустится опять, если компакт-диск остался в устройстве и было выбрано **[ Yes ]**.

Если вы загружаетесь с дискет, необходимо будет извлечь дискету с `mfsroot.flp` и заменить ее дискетой с `kern.flp` перед перезагрузкой.

## 2.4. Введение в Sysinstall

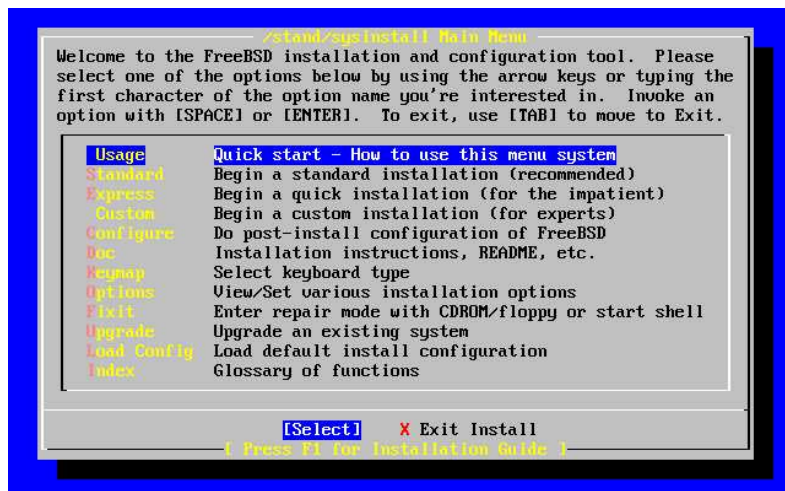
Утилита **sysinstall** это программа установки, предоставляемая проектом FreeBSD. Это консольное приложение, разделенное на несколько меню и экранов, которые вы можете использовать для настройки и управления процессом установки.

Меню **sysinstall** управляется клавишами навигации, **Enter**, пробелом, и другими. Подробное описание клавиш и их функций содержится в информации по использованию **sysinstall**.

Для просмотра этой информации убедитесь, что выбраны пункт **Usage** и кнопка **[Select]**, как показано на Рис. 2-8, затем нажмите **Enter**.

Будут показаны инструкции по использованию меню. После просмотра инструкций, нажмите **Enter** для возврата в главное меню.

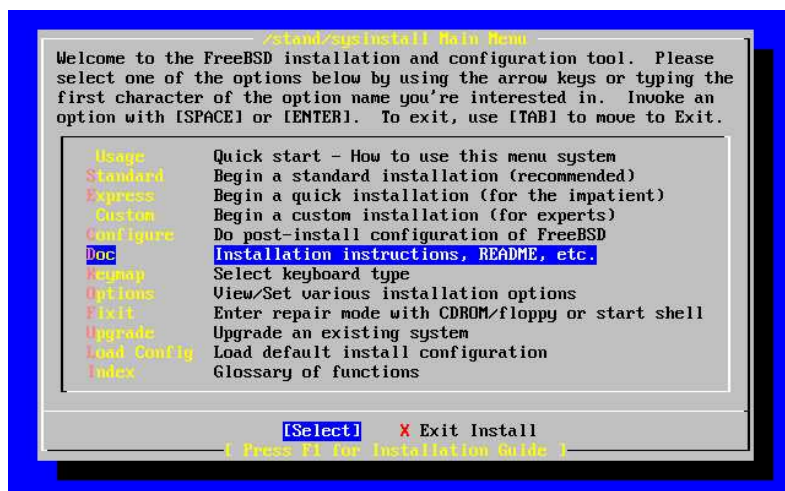
Рисунок 2-8. Выбор Usage в главном меню Sysinstall



### 2.4.1. Выбор меню документации (Doc)

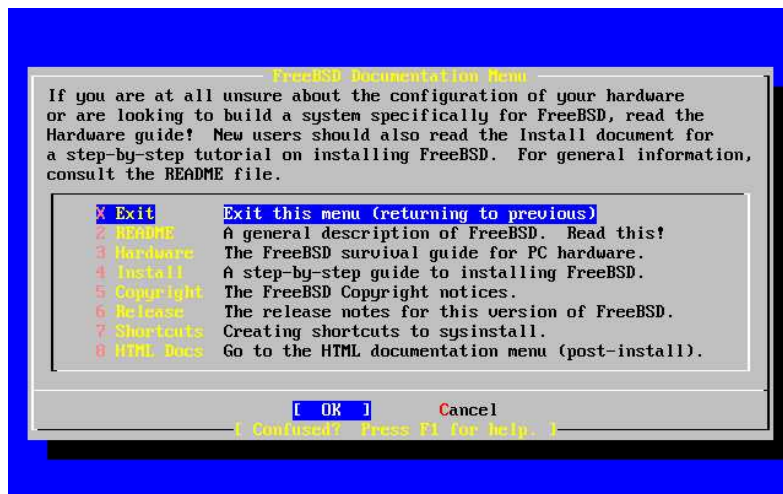
Из главного меню выберите клавишами навигации **Doc** и нажмите **Enter**.

Рисунок 2-9. Выбор меню документации



Будет отображено меню документации.

Рисунок 2-10. Меню документации Sysinstall



Рекомендуется прочитать предоставляемую документацию.

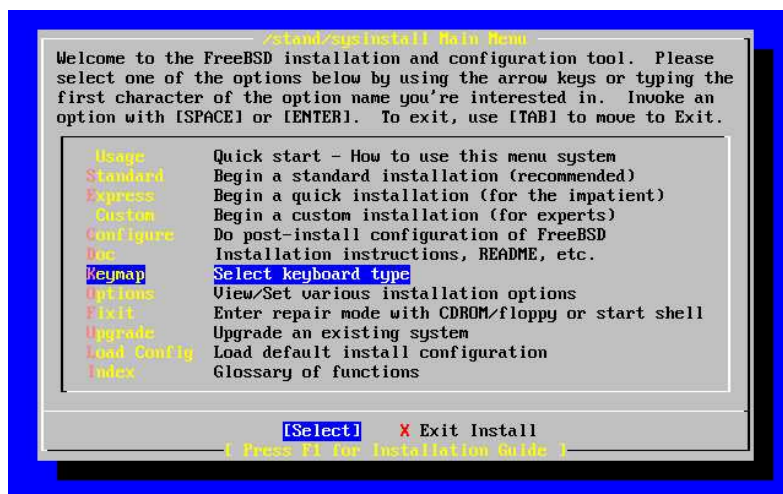
Для просмотра документа выберите его с помощью клавиш навигации и нажмите **Enter**. После прочтения документа нажмите **Enter** для возврата в меню документации.

Для возврата в главное меню выберите **Exit** с помощью клавиш навигации и нажмите **Enter**.

## 2.4.2. Выбор меню раскладки клавиатуры (Keymap)

Для изменения раскладки клавиатуры выберите из меню с помощью клавиш навигации **Keymap** и нажмите **Enter**. Это потребуется только при использовании нестандартной или не-US клавиатуры.

Рисунок 2-11. Выбор меню раскладки клавиатуры

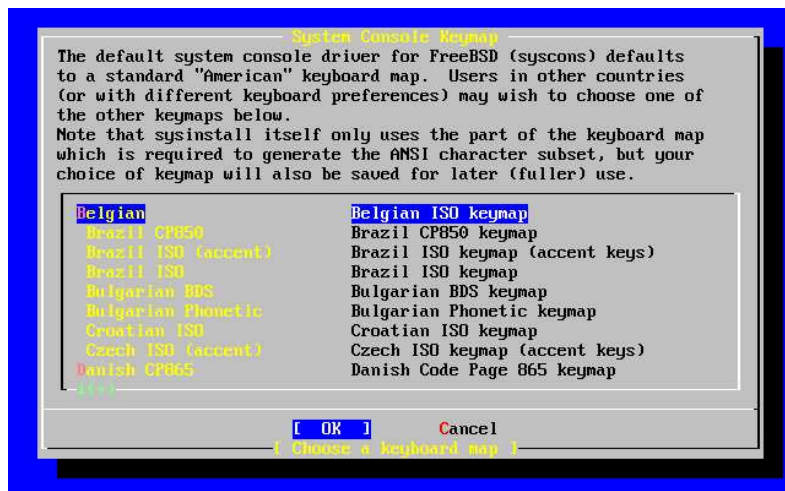


Различные раскладки клавиатуры могут быть выбраны из меню с использованием клавиш навигации, затем следует нажать **Space**. Нажатие **Space** еще раз приведет к отмене выбора. Когда необходимые

раскладки будут выбраны, перейдите на [ OK ] с помощью клавиш навигации и нажмите **Enter**.

На экран выведена только часть списка. Нажав **Tab**, можно выбрать [ Cancel ], вернуться к раскладке по умолчанию и перейти к главному меню.

Рисунок 2-12. Меню раскладки клавиатуры



### 2.4.3. Параметры установки (Options)

Выберите пункт Options и нажмите **Enter**.

Рисунок 2-13. Выбор параметров установки

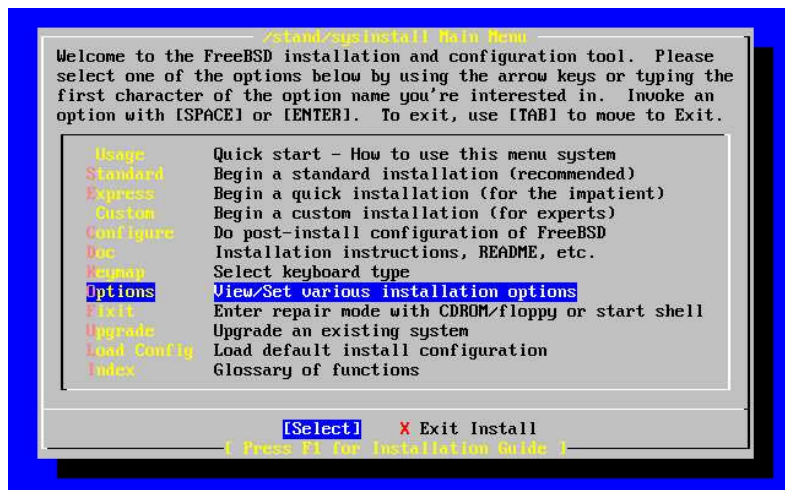
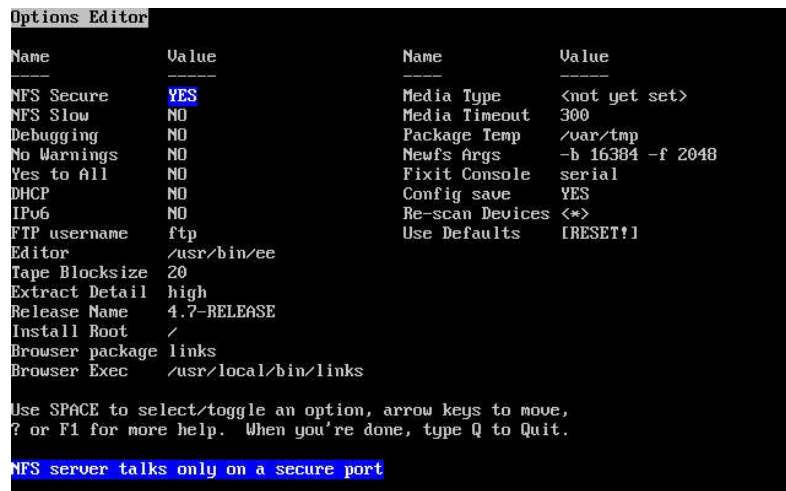




Рисунок 2-14. Параметры Sysinstall



Параметры по умолчанию обычно устраивают большинство пользователей и не нуждаются в изменении. Имя релиза зависит от устанавливаемой версии.

Описание выбранного пункта будет появляться внизу экрана с синей подсветкой. Обратите внимание, что один из параметров — **Use Defaults**, означает сброс всех параметров к значениям по умолчанию.

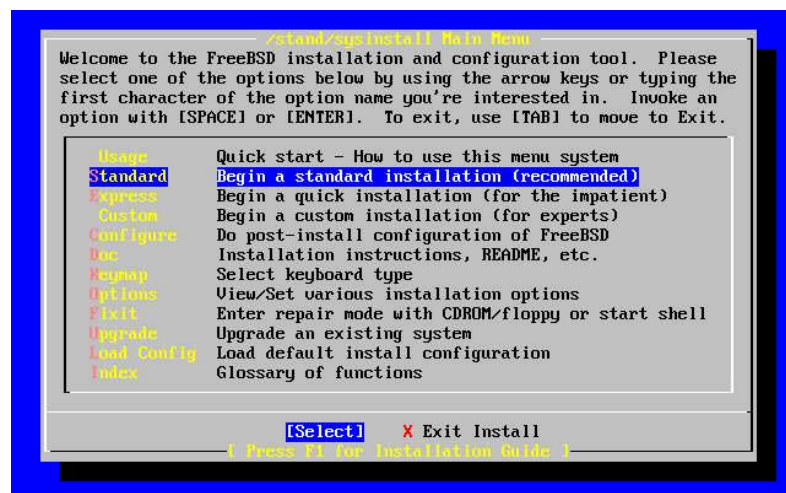
Нажатие **F1** отобразит справку по различным параметрам.

Нажатием **Q** можно перейти к главному меню.

## 2.4.4. Начало стандартной установки (Standard)

Пункт **Standard** рекомендуется для новых пользователей UNIX или FreeBSD. Используйте клавиши навигации для выбора пункта **Standard**, а затем нажмите **Enter** для запуска установки.

Рисунок 2-15. Начало стандартной установки





## 2.5. Выделение дискового пространства

Ваша первая задача — выделить дисковое пространство под FreeBSD и разметить его, чтобы **sysinstall** могла его подготовить. Для этого вам нужно знать, как FreeBSD ищет информацию на диске.

### 2.5.1. Нумерация дисков в BIOS

Перед установкой и настройкой FreeBSD нужно позаботиться кое о чем, особенно если жестких дисков несколько.

В PC, работающем под BIOS-зависимой операционной системой, такой как MS-DOS или Microsoft Windows, BIOS может отходить от обычного порядка нумерации дисков. Это позволяет пользователю загружаться не только с так называемого "primary master" диска. Это особенно удобно для тех пользователей, кто обнаружил, что простейший и самый дешевый путь делать резервную копию системы — купить второй идентичный первому жесткий диск и регулярно делать копии первого диска на второй, используя **Ghost®** или **XCOPY**. Затем, если первый диск выйдет из строя, будет заражен вирусом или поврежден из-за сбоя операционной системы, он может быть легко восстановлен путем логической перестановки дисков в BIOS. Это все равно что переключить кабели дисков, но без вскрытия корпуса.

Более дорогостоящие системы со SCSI контроллерами зачастую имеют расширения BIOS, позволяющие сходным путем менять порядок до семи SCSI дисков.

Пользователи, привыкшие пользоваться этими полезными функциями, могут быть удивлены, что во FreeBSD результаты не совпадают с ожидаемыми. FreeBSD не использует BIOS, и не знает о "логическом отображении дисков в BIOS". Это может привести к очень сложным ситуациям, особенно когда диски имеют одинаковую геометрию и содержат точную копию данных друг друга.

При использовании FreeBSD всегда восстанавливайте настройки BIOS к первоначальной нумерации перед установкой системы и оставляйте их в таком виде. Если вам понадобится переключить диски, сделайте это, но путем физического переконфигурирования, вскрыв корпус, переключив перемычки и кабели.

**Рассказ о необыкновенных приключениях файлов Билла и Фреда:**

Билл разобрал старый Wintel компьютер, чтобы сделать еще один компьютер под FreeBSD для Фреда. Билл установил один SCSI диск как нулевое устройство SCSI и поставил на него FreeBSD.

Фред начал использовать систему, но через несколько дней обнаружил, что старый SCSI диск сообщает о множестве сбоев и сказал об этом Биллу.

Еще через несколько дней Билл решил, что настало время решить проблему, и достал такой же SCSI диск из "зачетки" в кладовке. Первая проверка поверхности показала, что диск работает нормально; Билл установил этот диск как четвертое устройство SCSI и скопировал образ диска с нулевого устройства на четвертое. Теперь, когда новый диск был установлен и отлично работал, Билл решил что неплохо бы начать использовать его, и с помощью функции SCSI BIOS поменял порядок дисков, чтобы система могла грузиться с четвертого устройства SCSI. FreeBSD загрузилась и работала без проблем.

Фред поработал еще несколько дней, и скоро они с Биллом решили, что настало время для нового приключения — время обновить версию FreeBSD. Билл удалил нулевое устройство SCSI, потому что оно "подключивало", и установил на его место такой же диск из "зачетки". Затем Билл установил новую версию FreeBSD на новое нулевое устройство SCSI используя дискеты Фреда с интернет сервера FTP. Установка прошла отлично.

Фред использовал новую версию FreeBSD несколько дней и удостоверился, что она вполне подходит для работы в инженерном отделе. Настало время скопировать все архивы со старого диска. Фред смонтировал четвертое устройство SCSI (последнюю копию старой версии FreeBSD) и обнаружил, что ни одного из его драгоценных файлов на четвертом устройстве SCSI нет.

Куда делись данные?

Когда Билл сделал копию с нулевого устройства SCSI на четвертое устройство SCSI, оно стало "клоном". Когда Билл поменял настройки SCSI BIOS, чтобы загрузиться с четвертого устройства SCSI, он всего лишь обманул сам себя. FreeBSD все еще работала с нулевого устройства SCSI. Изменение этих настроек BIOS привело к загрузке части кода Boot и Loader с выбранного в BIOS диска, но после загрузки драйверов FreeBSD настройки BIOS были проигнорированы, и FreeBSD вернулась к нормальной нумерации. Как показано "на пальцах", система продолжила работать с нулевым устройством SCSI, и все данные Фреда остались там, а не на четвертом устройстве SCSI. То, что система грузилась с четвертого устройства SCSI, было всего лишь обманутыми ожиданиями.

Мы рады упомянуть, что данные не были уничтожены или повреждены при нашем исследовании этого феномена. Старое нулевое устройство SCSI было вытащено из груды железа, и все файлы Фреда вернулись к нему.

Хотя в этом рассказе были использованы SCSI диски, с IDE дисками все точно так же.

## 2.5.2. Создание слайсов с использованием FDisk

**Замечание:** Внесенные вами изменения не будут записываться на диск сразу. Если вы думаете, что сделали ошибку, и хотите начать сначала, можете использовать меню для выхода из **sysinstall** и попробовать еще раз или нажатием **U** вызвать опцию Undo (отмена). Если вы запутались и не можете выйти, просто выключите компьютер.

После начала стандартной установки в **sysinstall** будет показано это сообщение:

```

Message

In the next menu, you will need to set up a DOS-style ("fdisk")
partitioning scheme for your hard disk. If you simply wish to devote
all disk space to FreeBSD (overwriting anything else that might be on
the disk(s) selected) then use the (A)ll command to select the default
partitioning scheme followed by a (Q)uit. If you wish to allocate only
free space to FreeBSD, move to a partition marked "unused" and use the
(C)reate command.

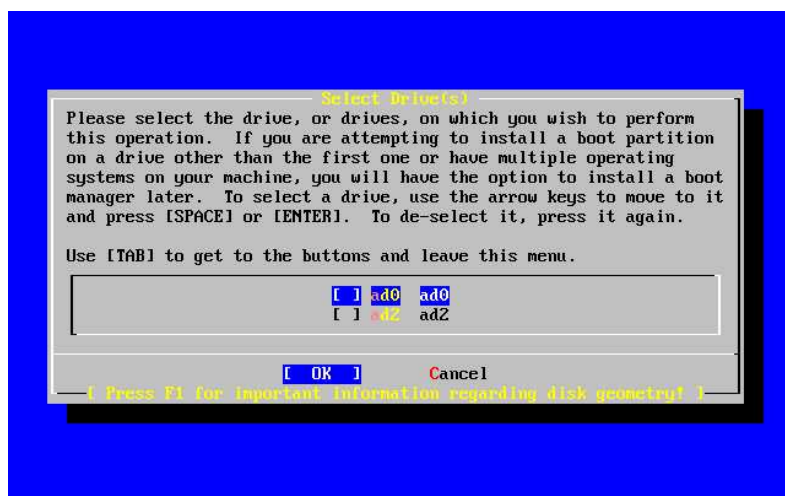
```

```
[ OK ]
```

```
[ Press enter or space ]
```

Нажмите **Enter** как предлагается. Будет показан список всех жестких дисков, обнаруженных ядром во время тестирования устройств. Рис. 2-16 показывает пример системы с двумя IDE дисками. Они были названы `ad0` и `ad2`.

**Рисунок 2-16. Выберите диск для FDisk**



Вы можете быть удивлены, почему устройства `ad1` здесь нет. Почему оно было пропущено?

Предположим, что у вас есть два жестких диска IDE, один master на первом контроллере IDE, а второй master на втором контроллере IDE. Если FreeBSD пронумерует их в том порядке, в котором нашла, `ad0` и `ad1`, все будет работать.

Но если вы добавите третий диск, как slave устройство на первый контроллер IDE, он станет `ad1`, а предыдущий `ad1` станет `ad2`. Поскольку имена устройств (таких как `ad1s1a`) используются для обращения к файловым системам, вы можете вдруг обнаружить, что некоторые из ваших файловых систем больше не отображаются правильно и вам потребуется изменить конфигурацию FreeBSD.

Для обхода этой проблемы, ядро может быть настроено так, чтобы именовать IDE диски на основе их местоположения, а не порядка, в котором они были найдены. С этой схемой master диск на втором контроллере IDE будет *всегда* устройством `ad2`, если даже нет устройств `ad0` или `ad1`.

Это конфигурация ядра FreeBSD по умолчанию, поэтому на экране показаны `ad0` и `ad2`. У компьютера, с которого был взят этот снимок экрана, есть по одному IDE диску на обоих master каналах IDE контроллеров и ни одного диска на каналах slave.

Вы должны выбрать диск, на который хотите установить FreeBSD, и нажать [ OK ]. Запустившийся **FDisk** будет выглядеть примерно как Рис. 2-17.

Экран **FDisk** разбит на три раздела.

Первый раздел, занимающая первые две линии экрана, показывает подробную информацию о выбранном в данный момент диске, включая его имя во FreeBSD, геометрию и общий размер диска.

Второй раздел показывает имеющиеся в данный момент на диске слайсы, где они начинаются и заканчиваются, их размер, имя, которое им дала FreeBSD, описание и подтип. На этом примере показаны два маленьких неиспользованных слайса, которые являются артефактами схемы разметки диска на PC. Также показан один большой FAT слайс, который почти всегда является диском `C:` в MS-DOS / Windows, и дополнительный слайс, который может содержать диски с другими буквами для MS-DOS / Windows.

Третий раздел показывает команды, доступные в **FDisk**.

Рисунок 2-17. Типичные разделы fdisk перед редактированием

```

Disk name:      ad0      FDISK Partition Editor
DISK Geometry:  16383 cyls/16 heads/63 sectors = 16514064 sectors (8063MB)

Offset      Size(ST)      End      Name  PType      Desc  Subtype  Flags
-----
0           63           62      -      6      unused    0
63      4193217      4193279  ad0s1  2      fat       14      >
4193280      1008      4194287  -      6      unused    0      >
4194288      12319776      16514063  ad0s2  4      extended  15      >

The following commands are supported (in upper or lower case):

A = Use Entire Disk      G = set Drive Geometry      C = Create Slice      F = 'DD' mode
D = Delete Slice         Z = Toggle Size Units      S = Set Bootable      I = Wizard m.
T = Change Type          U = Undo All Changes      Q = Finish

Use F1 or ? to get more help, arrow keys to select.

```

Ваши действия теперь будут зависеть от того, как вы хотите разбить диск на слайсы.

Если вы хотите использовать для FreeBSD весь диск (это приведет к удалению всех других данных на этом диске когда вы подтвердите **sysinstall** продолжение процесса установки), нажмите **A**, что соответствует опции Использовать весь диск (Use Entire Disk). Существующие слайсы будут удалены, и заменены на небольшую область, помеченную как *неиспользуемая* (`unused`) (это опять же артефакт разметки диска PC), и один большой слайс для FreeBSD. Когда вы сделаете это, нужно выбрать вновь созданный слайс FreeBSD используя клавиши навигации, а затем нажать **S**, чтобы сделать слайс загрузочным. Экран будет похож на Рис. 2-18. Обратите внимание, что **A** в колонке `Flags` означает, что слайс *активен* и с него будет происходить загрузка.

Если вы будете удалять существующий слайс для освобождения места под FreeBSD, выберите слайс, используя клавиши навигации, и нажмите **D**. Затем можете нажать **C**, и получить приглашение на ввод размера слайса, который вы хотите создать. Введите соответствующее значение и нажмите **Enter**.

Значение по умолчанию в этом поле означает наибольший размер слайса, который может быть выбран; это может быть наибольший непрерывный блок неразмеченного пространства или размер всего жесткого диска.

Если вы уже освободили место для FreeBSD (возможно, используя утилиту вроде **PartitionMagic**), можете нажать **C** для создания нового слайса. Будет также предложено ввести размер слайса, который вы хотите создать.

**Рисунок 2-18. Разбиение в Fdisk с использованием всего диска**

```

Disk name:      ad0                      FDISK Partition Editor
DISK Geometry: 16383 cyls/16 heads/63 sectors = 16514064 sectors (8063MB)

Offset      Size(ST)      End      Name  PType      Desc  Subtype  Flags
-----
0           63           62      -      6      unused     0
63      16514001    16514063    ad0s1   3      freebsd    165      Ca

The following commands are supported (in upper or lower case):

A = Use Entire Disk      G = set Drive Geometry    C = Create Slice      F = 'DD' mode
D = Delete Slice        Z = Toggle Size Units    S = Set Bootable     I = Wizard n.
T = Change Type         U = Undo All Changes    Q = Finish

Use F1 or ? to get more help, arrow keys to select.

```

Когда закончите, нажмите **Q**. Изменения будут сохранены в **sysinstall**, но еще не записаны на диск.

### 2.5.3. Установка менеджера загрузки (Boot Manager)

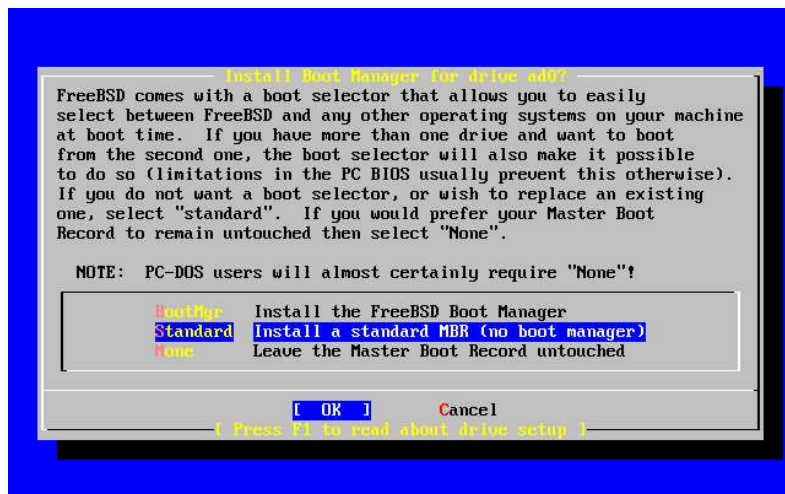
Теперь вам предлагается установить менеджер загрузки. Как правило, нужно выбрать установку менеджера загрузки если:

- У вас больше чем один диск и вы устанавливаете FreeBSD не на первый диск.
- Вы устанавливаете FreeBSD вместе с другой операционной на один и тот же диск, и хотите выбирать при загрузке FreeBSD или другую операционную систему.

Если FreeBSD единственная операционная система, установленная на этом компьютере, и находится на первом жестком диске, подойдет менеджер загрузки **Standard**. Выберите **None** если вы используете менеджер загрузки сторонних разработчиков, способный загрузить FreeBSD.

Сделайте выбор и нажмите **Enter**.

Рисунок 2-19. Меню менеджера загрузки Sysinstall



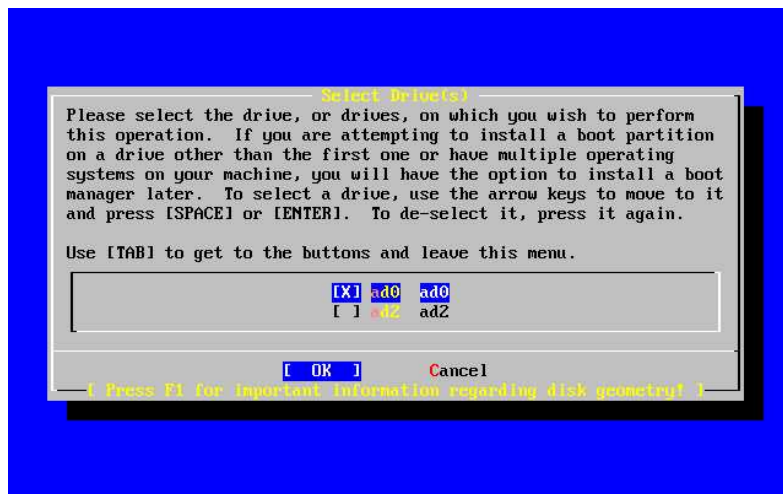
Экран подсказки, вызываемый по нажатию **F1**, описывает проблемы, которые могут быть встречены при попытке совместного использования диска операционными системами.

### 2.5.4. Создание слайсов на другом диске

Если дисков больше чем один, вернитесь к экрану выбора дисков (Select Drives) после выбора менеджера загрузки. Если вы собираетесь устанавливать FreeBSD более чем на один диск, можете выбрать другой диск и повторить процесс разбиения на слайсы с использованием **FDisk**.

**Важно:** Если вы устанавливаете FreeBSD не на первый жесткий диск, потребуется установить менеджер загрузки FreeBSD на оба диска.

Рисунок 2-20. Выход из выбора диска



Клавиша **Tab** переключает между последним выбранным диском, [ OK ], и [ Cancel ].

Нажмите **Tab** один раз для выбора [ OK ], затем нажмите **Enter** для продолжения установки.

### 2.5.5. Создание разделов с помощью Disklabel

Теперь вы должны создать несколько разделов внутри каждого только что созданного слайса. Запомните, что у каждого раздела есть буква с a до h, а разделы b, c, и d имеют соглашения, которых вы должны придерживаться.

Некоторые приложения могут выигрывать от определенных схем разделов, особенно если у вас разделы на более чем одном диске. Тем не менее, для вашей первой установки FreeBSD не нужно слишком углубляться в принципы разбиения диска. Более важно установить FreeBSD и начать ее использовать. Вы всегда можете переустановить FreeBSD для изменения схемы разделов, когда поближе познакомитесь с операционной системой.

Эта схема показывает четыре раздела — один для подкачки и три для файловых систем.

Таблица 2-2. Планирование разделов для первого диска

Раздел	Файловая система	Размер	Описание
a	/	100 MB	Корневая файловая система. Любая другая файловая система будет смонтирована на эту. 100 MB это подходящий размер для этой файловой системы. Вы не будете хранить на ней слишком много данных, а обычная установка FreeBSD разместит здесь около 40 MB данных. Оставшееся пространство используется для временных файлов, а также оставляет возможность расширения для будущих версий FreeBSD, которым может понадобится больше места в /.

Раздел	Файловая система	Размер	Описание
b	N/A	2-3 x RAM	Здесь находится раздел подкачки. Выбор правильного размера раздела подкачки это немного искусство. Хороший практический способ выбрать размер подкачки это установить его равным двум или трем размерам доступной физической памяти (RAM). Должно быть хотя бы 64 MB подкачки; если в компьютере меньше чем 32 MB памяти — установите размер подкачки равным 64 MB. Если у вас больше одного диска, можно расположить подкачку на каждом диске. FreeBSD будет использовать каждый диск, что серьезно увеличит скорость подкачки. В этом случае, определите общий размер подкачки, который вам нужен (например, 128 MB), и поделите его на число имеющихся дисков (например, два) для определения размера разделов подкачки, которые нужно разместить на каждом вашем диске, в этом примере 64 MB на диск.
e	/var	50 MB	Каталог /var содержит файлы, которые постоянно меняются; логи и другие административные файлы. Многие из этих файлов интенсивно читаются и записываются в процессе ежедневной работы FreeBSD. Размещение их на отдельной файловой системе позволяет FreeBSD оптимизировать доступ к этим файлам без затрагивания других каталогов, не имеющих такой же модели доступа.
f	/usr	Остальная часть диска	Все другие файлы как правило хранятся в каталоге /usr и его подкаталогах.

Если вы устанавливаете FreeBSD более чем на один диск, вы должны также создать разделы в других слайсах, которые настроили. Простейший путь сделать это — создать два раздела на каждом диске, один для подкачки, а другой для файловой системы.

**Таблица 2-3. Разметка разделов для остальных дисков**

Раздел	Файловая система	Размер	Описание
b	N/A	Смотрите описание	Как уже обсуждалось, вы можете распространить подкачку на каждый диск. Даже если раздел a свободен, соглашение говорит о том, что подкачка находится на разделе b.



Раздел	Файловая система	Размер	Описание
е	/diskn	Остальная часть диска	Остальная часть диска занята одним большим разделом. Он легко может быть помещен на раздел а вместо раздела е. Однако, соглашение говорит, что раздел а зарезервирован на слайсе для корневой (/) файловой системы. вы можете не следовать этому соглашению, но программа <b>sysinstall</b> будет ему следовать, поэтому приняв его, вы сделаете установку несколько проще. Вы можете монтировать эти файловые системы к любой точке; в этом примере предлагается смонтировать их как каталоги /diskn, где n это номер, который уникален для каждого диска. Но вы можете использовать другую схему, если захотите.

Теперь, выбрав разметку разделов, можете приступить к их созданию в **sysinstall**. Вы увидите это сообщение:

```

                                Message
Now, you need to create BSD partitions inside of the fdisk
partition(s) just created. If you have a reasonable amount of disk
space (200 MB or more) and don't have any special requirements, simply
use the (A)uto command to allocate space automatically. If you have
more specific needs or just don't care for the layout chosen by
(A)uto, press F1 for more information on manual layout.

[ OK ]
[ Press enter or space ]

```

Нажмите **Enter** для запуска редактора разделов FreeBSD, называемого **Disklabel**.

Рис. 2-21 показывает экран только что запущенного **Disklabel**. Экран поделен на три раздела.

Первые несколько линий показывают имя диска, с которым вы сейчас работаете и слайс, содержащий раздел, который вы создаете (здесь **Disklabel** называет это **именем раздела** (Partition name) вместо имени слайса). Этот экран также показывает объем свободного пространства на слайсе, т.е. пространство, выделенное под слайс, но еще не отданное под раздел.

В центре экрана показаны уже созданные разделы, имена файловых систем, содержащихся в разделах, их размер и некоторые опции, применяемые при создания файловых систем.

Нижняя треть экрана показывает управляющие клавиши, работающие в **Disklabel**.

Рисунок 2-21. Редактор Sysinstall Disklabel

```

FreeBSD Disklabel Editor
Disk: ad0 Partition name: ad0s1 Free: 16514001 blocks (8063MB)

Part      Mount      Size Newfs  Part      Mount      Size Newfs
-----
-----

The following commands are valid here (upper or lower case):
C = Create      D = Delete    M = Mount pt.
N = Newfs Opts  Q = Finish   S = Toggle SoftUpdates
T = Toggle Newfs U = Undo      A = Auto Defaults  R = Delete+Merge

Use F1 or ? to get more help, arrow keys to select.

```

**Disklabel** может автоматически создать разделы и присвоить им размеры по умолчанию. Попробуйте это, нажав **A**. Вы увидите экран как на Рис. 2-22. В зависимости от размера диска, значения по умолчанию могут подходить или не подходить вам. Это не имеет значения, если вы не принимаете их.

**Замечание:** Начиная с FreeBSD 4.5, по умолчанию под каталог `/tmp` выделяется собственный раздел вместо использования части раздела `/`. Это помогает избежать заполнения раздела `/` временными файлами.

Рисунок 2-22. Редактор Sysinstall Disklabel с установками по умолчанию

```

FreeBSD Disklabel Editor
Disk: ad0 Partition name: ad0s1 Free: 0 blocks (0MB)

Part      Mount      Size Newfs  Part      Mount      Size Newfs
-----
-----
ad0s1a    /           128MB UFS    Y
ad0s1b    swap        503MB SWAP
ad0s1e    /var        256MB UFS+S Y
ad0s1f    /tmp        256MB UFS+S Y
ad0s1g    /usr        6919MB UFS+S Y

The following commands are valid here (upper or lower case):
C = Create      D = Delete    M = Mount pt.  W = Write
N = Newfs Opts  Q = Finish   S = Toggle SoftUpdates
T = Toggle Newfs U = Undo      A = Auto Defaults  R = Delete+Merge

Use F1 or ? to get more help, arrow keys to select.

```

Если вы решили не использовать разделы по умолчанию и заменить их на свои, используйте клавиши навигации для выбора первого раздела, затем нажмите **D** для его удаления. Повторите это для удаления всех предложенных разделов.

Для создания первого раздела (а, монтируемого как / — root), убедитесь, что выбран соответствующий слайс вверху экрана и нажмите **C**. Появится диалог, предлагающий выбрать размер нового раздела (как показано на Рис. 2-23). Вы можете ввести количество блоков диска, или количество мегабайт с **M** после номера, или гигабайт с **G**, или цилиндров с **C**.

**Замечание:** Начиная с FreeBSD 5.X, пользователи могут: выбирать UFS2 (который используется по умолчанию в FreeBSD 5.1 и выше), используя опцию Custom Newfs (**Z**), использовать опцию Auto Defaults и изменять параметры с помощью опции Custom Newfs, или добавлять опцию `-O 2` в процессе создания раздела. Не забудьте добавить `-U` для включения SoftUpdates если вы используете опцию Custom Newfs!

Рисунок 2-23. Свободное место для корневого раздела



Размер по умолчанию задан для создания корневого раздела на весь слайс. Если вы используете размеры разделов, описанные ранее в примере, удалите это значение используя **Backspace**, а затем введите **64M**, как показано на Рис. 2-24. Затем нажмите **[ OK ]**.

Рисунок 2-24. Редактирование размера корневого раздела



После указания размера раздела вам будет задан вопрос, должен ли этот раздел содержать файловую систему или раздел подкачки. Диалог показан на Рис. 2-25. Первый раздел будет содержать файловую систему, поэтому проверьте, что выбрана **FS** и нажмите **Enter**.

Рисунок 2-25. Выбор типа корневого раздела



Наконец, поскольку вы создаете файловую систему, нужно сказать **Disklabel** где файловая система будет смонтирована. Диалог показан на Рис. 2-26. Точка монтирования корневой файловой системы **/**, поэтому введите **/**, и нажмите **Enter**.

Рисунок 2-26. Выбор точки монтирования корневой файловой системы



На экране будет показан вновь созданный раздел. Вам нужно повторить эту процедуру для других разделов. При создании раздела подкачки вопроса про точку монтирования не будет, поскольку раздел подкачки никогда не монтируется. Когда будете создавать последний раздел, /usr, можете оставить предложенный размер как есть, чтобы использовать весь остаток слайса.

Последний экран FreeBSD редактора DiskLabel будет похож на Рис. 2-27, хотя ваш выбор значений может быть другим. Нажмите **Q**, чтобы выйти.

Рисунок 2-27. Редактор Sysinstall Disklabel



## 2.6. Выбор устанавливаемых компонентов

### 2.6.1. Выбор дистрибутивного набора (Distribution Set)

Выбор дистрибутивного набора зависит в основном от направления будущего использования системы и от доступного дискового пространства. Предустановленные опции варьируются от наименьшей возможной конфигурации до полной установки. Для новичков в UNIX и/или FreeBSD лучшим выбором будет одна из этих предустановленных опций. Настройка дистрибутивного набора как правило нужна более опытным пользователям.

Нажмите **F1** для получения информации о дистрибутивных наборах и их содержимом. После просмотра помощи нажмите **Enter** для возврата к меню выбора дистрибутивного набора.

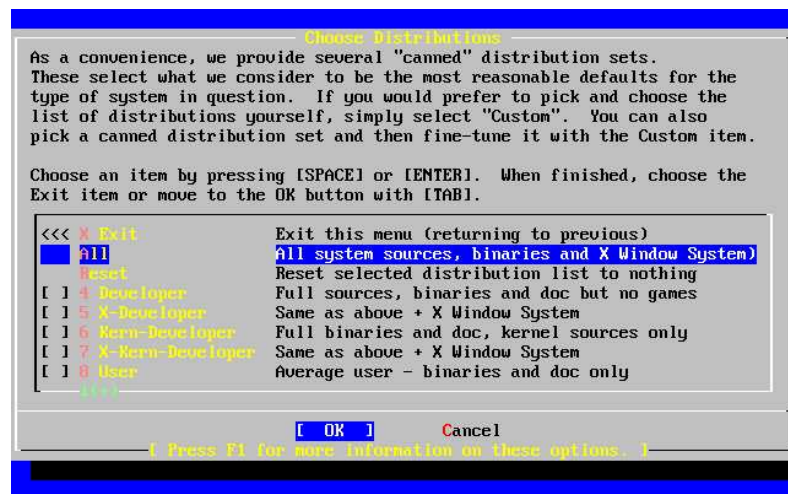
Если желательно наличие графического интерфейса пользователя, должен быть выбран дистрибутивный набор с названием, начинающимся на **x**. Задачи настройки **X** сервера и выбора десктопа по умолчанию должны быть выполнены после установки FreeBSD. Более подробная информация по настройке **X** сервера находится в Гл. 5.

Устанавливаемая по умолчанию версия **X11** зависит от устанавливаемой версии FreeBSD. Для FreeBSD версий до 5.3, устанавливается **XFree86 4.X**. Для FreeBSD 5.3 и более поздних, по умолчанию устанавливается **Xorg**.

Если планируется пересборка ядра, выберите опцию, включающую исходные тексты. Информация о том, зачем пересобирать ядро и как это сделать, находится на Гл. 8.

Ясно, что наиболее универсальная система включает все. Если места на диске достаточно, выберите **All**, как показано на Рис. 2-28 и нажмите **Enter**. Если есть сомнения относительно того, хватит ли диска, используйте наиболее подходящую опцию. Не беспокойтесь о том, какой выбор будет наилучшим, другие части дистрибутива могут быть добавлены после установки.

Рисунок 2-28. Выбор дистрибутивных наборов



## 2.6.2. Установка коллекции портов

После выбора подходящего дистрибутива можно будет выбрать установку коллекции портов FreeBSD. Коллекция портов — легкий и удобный путь установки программ. Коллекция портов не содержит исходных кодов программ. Это набор файлов, который автоматизирует загрузку, компилирование и установку пакетов программного обеспечения сторонних разработчиков. Гл. 4 показывает, как использовать коллекцию портов.

Программа установки не проверяет, есть ли достаточно места. Выберите эту опцию только в том случае, если его достаточно. В FreeBSD 5.3, коллекция портов занимает около 300 MB. В более современных релизах это значение всегда больше.

```

User Confirmation Requested
Would you like to install the FreeBSD ports collection?

This will give you ready access to over 11,800 ported software packages,
at a cost of around 300 MB of disk space when "clean" and possibly much
more than that if a lot of the distribution tarballs are loaded
(unless you have the extra CDs from a FreeBSD CD/DVD distribution
available and can mount it on /cdrom, in which case this is far less
of a problem).

The ports collection is a very valuable resource and well worth having
on your /usr partition, so it is advisable to say Yes to this option.

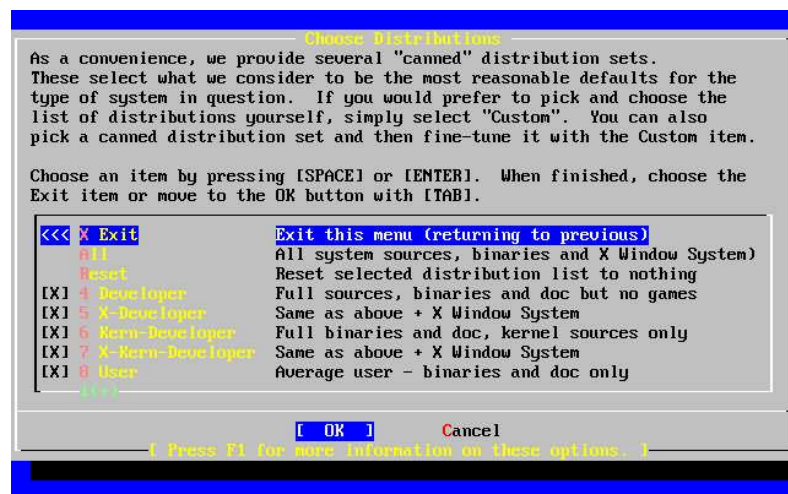
For more information on the ports collection & the latest ports,
visit:
    http://www.FreeBSD.org/ports

[ Yes ]      No

```

Выберите [ Yes ] для установки коллекции портов, или [ No ], чтобы пропустить установку. Нажмите **Enter**, чтобы продолжить. Меню выбора дистрибутивных наборов появится опять.

**Рисунок 2-29. Подтверждение выбора дистрибутивного набора**



Если вы согласны с выбранными опциями, переместитесь на **Exit**, убедитесь, что выбран [ OK ] и нажмите **Enter**, чтобы продолжить.

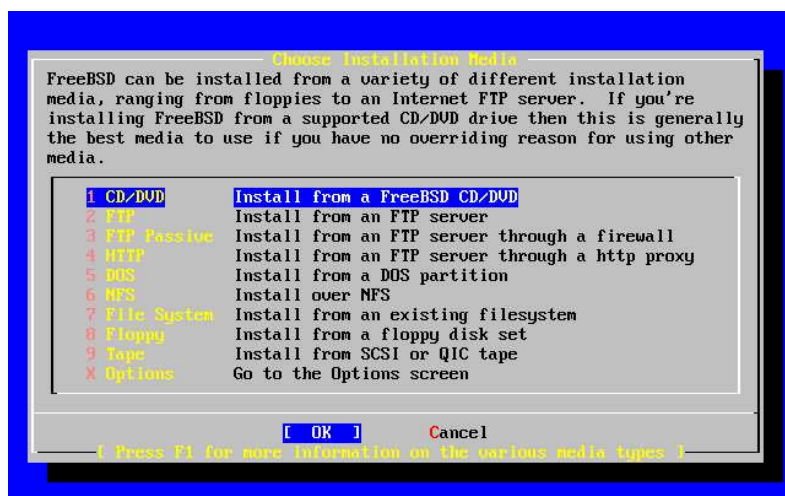
## 2.7. Выбор источника для установки

При установке с CDROM или DVD используйте клавиши навигации, для перехода к пункту **Install from a FreeBSD CD/DVD**. Убедитесь, что выбран [ OK ], и нажмите **Enter** для запуска установки.

При других методах установки выберите соответствующую опцию и следуйте инструкциям.

Нажмите **F1** для просмотра справки по источникам установки. Нажмите **Enter** для возврата к меню выбора источника установки.

Рисунок 2-30. Выбор источника установки



**Режимы установки с FTP:** Есть три режима установки через FTP, которые вы можете выбрать: активный FTP, пассивный FTP, или через HTTP прокси.

Активный FTP: Install from an FTP server

С этой опцией все загрузки по FTP будут выполнены в "активном" режиме. Этот режим не позволяет работать через файрволл, но зачастую позволяет работать со старыми серверами FTP, не поддерживающими пассивный режим. Если соединение прерывается в пассивном режиме (по умолчанию), попробуйте активный!

Пассивный FTP: Install from an FTP server through a firewall

Эта опция говорит **sysinstall** использовать "пассивный" режим для работы с FTP. Он позволяет работать через файрволл, не разрешающий входящие соединения на случайных TCP портах.



FTP через HTTP прокси: Install from an FTP server through a http proxy

Эта опция говорит **sysinstall** использовать HTTP протокол (как веб браузер) для работы с FTP через прокси. Прокси будет транслировать все запросы и посылать их на FTP сервер. Это позволяет проходить через файрволл, на котором FTP запрещен, но есть HTTP прокси. В этом случае потребуется указать прокси и FTP сервер.

Для работы с FTP через прокси, необходимо поместить имя сервера как часть имени пользователя после знака "@". Прокси сервер "обманет" настоящий сервер. Например, предположим что вы хотите провести установку с ftp.FreeBSD.org, используя FTP через прокси foo.example.com, прослушивающем порт 1024.

В этом случае, войдите в меню параметров, установите имя пользователя FTP ftp@ftp.FreeBSD.org, а вместо пароля введите свой адрес email. В качестве источника установки выберите FTP (или пассивный FTP, если прокси его поддерживает), и URL ftp://foo.example.com:1234/pub/FreeBSD.

Так как /pub/FreeBSD с сервера ftp.FreeBSD.org идет через прокси foo.example.com, вы сможете провести установку с этого компьютера (файлы будут загружены с ftp.FreeBSD.org как требуется для установки).

## 2.8. Подтверждение установки

Теперь можно начинать установку. Это последний шанс отменить установку, и таким образом избежать изменений на жестком диске.

```
User Confirmation Requested
Last Chance! Are you SURE you want to continue the installation?

If you're running this on a disk with data you wish to save then WE
STRONGLY ENCOURAGE YOU TO MAKE PROPER BACKUPS before proceeding!

We can take no responsibility for lost disk contents!

[ Yes ]      No
```

Выберите [ Yes ] и нажмите **Enter**, чтобы начать.

Время установки сильно зависит от выбранного дистрибутивного набора, источника установки и скорости компьютера. Появится несколько сообщений о статусе процесса установки.

Установка будет завершена, когда отобразится следующее сообщение:

```
Message

Congratulations! You now have FreeBSD installed on your system.

We will now move on to the final configuration questions.
For any option you do not wish to configure, simply select No.

If you wish to re-enter this utility after the system is up, you may
do so by typing: /stand/sysinstall .
```

[ OK ]

[ Press enter to continue ]

Нажмите **Enter** для начала послеустановочной настройки.

Выбор [ No ] и нажатие **Enter** прервет процесс установки, изменения в систему внесены не будут.

Появится следующее сообщение:

Message

Installation complete with some errors. You may wish to scroll through the debugging messages on VTY1 with the scroll-lock feature. You can also choose "No" at the next prompt and go back into the installation menus to retry whichever operations have failed.

[ OK ]

Это сообщение появилось, поскольку ничего не было установлено. Нажатие **Enter** вернет вас в главное меню установки, чтобы выйти из нее.

## 2.9. После установки

После успешной установки необходимо настроить множество параметров. Некоторые параметры могут быть заданы из меню параметров после установки, перед загрузкой установленной FreeBSD, или после нее с использованием `sysinstall` (`/stand/sysinstall` в FreeBSD версии 5.2 и более ранних), где надо выбрать пункт **Configure**.

### 2.9.1. Настройка сетевых устройств (Network Device Configuration)

Если вы настраивали PPP для установки через FTP, этот экран не появится, и настройку можно будет произвести позже как описано выше.

Чтобы лучше узнать о локальных сетях и настройке FreeBSD в качестве шлюза/маршрутизатора, обратитесь к главе Сложные вопросы работы в сети.

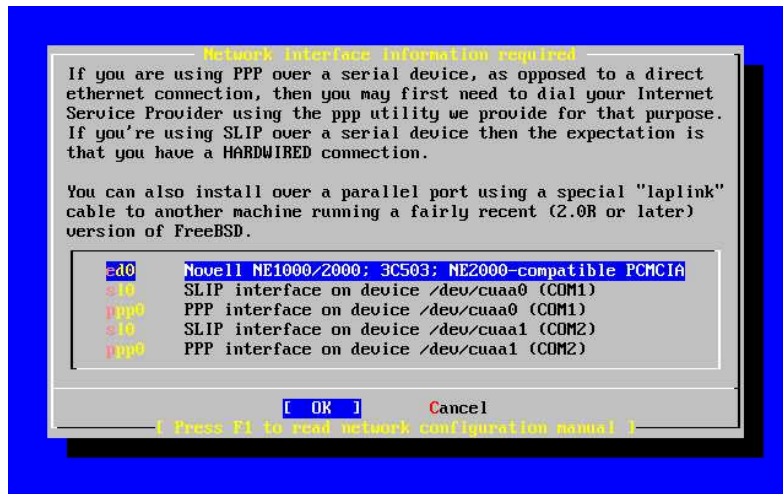
User Confirmation Requested

Would you like to configure any Ethernet or SLIP/PPP network devices?

[ Yes ] No

Для настройки сетевого устройства выберите [ Yes ] и нажмите **Enter**. Или нажмите [ No ], чтобы продолжить.

## Рисунок 2-31. Выбор Ethernet устройства



Выберите интерфейс для настройки с помощью клавиш навигации и нажмите **Enter**.

```

User Confirmation Requested
Do you want to try IPv6 configuration of the interface?

Yes    [ No ]

```

Для частной локальной сети обычный протокол интернет (IPv4) вполне достаточен, поэтому выбрана кнопка [ No ] и нажат **Enter**.

Если вы хотите подключиться к существующей сети IPv6 через сервер RA, выберите [ Yes ] и нажмите **Enter**. Поиск RA серверов займет несколько секунд.

```

User Confirmation Requested
Do you want to try DHCP configuration of the interface?

Yes    [ No ]

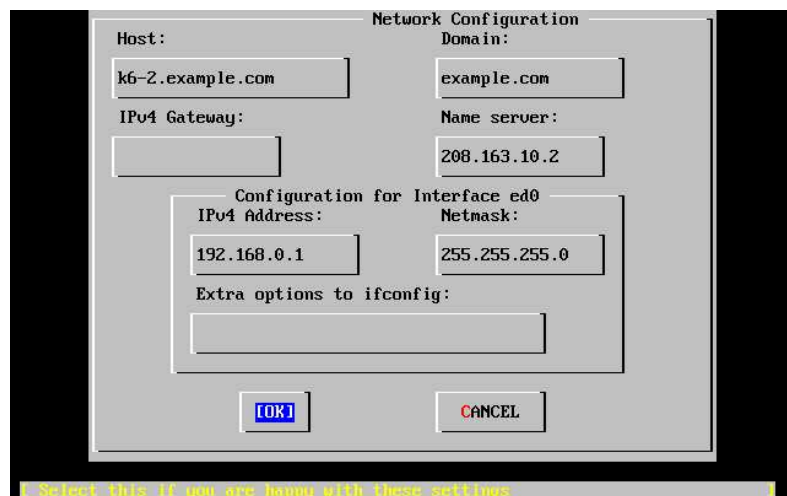
```

Если DHCP (Dynamic Host Configuration Protocol) не нужен, выберите [ No ] с помощью клавиш навигации и нажмите **Enter**.

Выбор [ Yes ] запустит **dhclient**, и, если все пройдет нормально, заполнит информацию о конфигурации сети автоматически. Обратитесь к Разд. 23.5 за более подробными сведениями.

Следующий экран конфигурации сети показывает настройку устройства Ethernet системы, которая будет работать шлюзом для локальной сети.

Рисунок 2-32. Настройка сети для ed0



Используйте **Tab** для выбора полей и заполнения их соответствующими данными:

#### Host

Полное имя хоста, в этом примере `k6-2.example.com`.

#### Domain

Имя домена, в котором находится ваш компьютер, в этом примере `example.com`.

#### IPv4 Gateway

IP хоста, пересылающего пакеты наружу локальной сети. Вам потребуется заполнить его, если это компьютер, подключенный к сети. *Оставьте это поле пустым*, если компьютер является шлюзом в интернет для сети. Шлюз IPv4 известен также как шлюз по умолчанию или маршрут по умолчанию.

#### Name server

IP адрес местного сервера DNS. В этой локальной сети нет DNS сервера, поэтому использован IP адрес DNS сервера провайдера (`208.163.10.2`).

#### IPv4 address

IP адрес, использованный для этого интерфейса, `192.168.0.1`

#### Netmask

Адрес блока, использованного для этой локальной сети, это блок класса C (`192.168.0.0 - 192.168.255.255`). Маска по умолчанию дана для сети класса C (`255.255.255.0`).

#### Дополнительные параметры для ifconfig

Любые специфичные для интерфейса опции к `ifconfig`, которые вы хотите добавить. В данном случае ничего.

Используйте **Tab** для выбора [ OK ] после окончания настройки и нажмите **Enter**.

```
User Confirmation Requested
Would you like to Bring Up the ed0 interface right now?
```

```
[ Yes ]    No
```

Выбор [ Yes ] и нажатие **Enter** введет компьютер в сеть. Тем не менее, компьютеру все еще требуется перезагрузка.

### 2.9.2. Настройка шлюза (Configure Gateway)

```
User Confirmation Requested
Do you want this machine to function as a network gateway?
```

```
[ Yes ]    No
```

Если компьютер будет шлюзом для локальной сети, пересылая пакеты между другими компьютерами, выберите [ Yes ] и нажмите **Enter**. Если это обычный компьютер, выберите [ No ] и нажмите **Enter** для продолжения.

### 2.9.3. Настройка сервисов интернет (Configure Internet Services)

```
User Confirmation Requested
Do you want to configure inetd and the network services that it provides?
```

```
Yes      [ No ]
```

Если выбрана [ No ], различные сервисы, такие как **telnetd** не будут запущены. Это означает, что удаленные пользователи не смогут зайти по **telnet** на этот компьютер. Локальные пользователи все же смогут заходить на удаленные компьютеры по **telnet**.

Эти сервисы могут быть включены после установки путем редактирования `/etc/inetd.conf` с помощью вашего любимого текстового редактора. Обращайтесь к Разд. 23.2.1 за более подробной информацией.

Выберите [ Yes ] если хотите настроить эти сервисы во время установки. Появится дополнительный запрос подтверждения:

```
User Confirmation Requested
The Internet Super Server (inetd) allows a number of simple Internet
services to be enabled, including finger, ftp and telnetd.  Enabling
these services may increase risk of security problems by increasing
the exposure of your system.
```

```
With this in mind, do you wish to enable inetd?
```

```
[ Yes ]    No
```

Нажмите [ Yes ], чтобы продолжить.

```
User Confirmation Requested
inetd(8) relies on its configuration file, /etc/inetd.conf, to determine
```

which of its Internet services will be available. The default FreeBSD `inetd.conf(5)` leaves all services disabled by default, so they must be specifically enabled in the configuration file before they will function, even once `inetd(8)` is enabled. Note that services for IPv6 must be separately enabled from IPv4 services.

Select [Yes] now to invoke an editor on `/etc/inetd.conf`, or [No] to use the current settings.

[ Yes ] No

Выбор [ Yes ] позволит добавить сервисы путем удаления # перед началом строки.

### Рисунок 2-33. Редактирование `inetd.conf`

```

^I (escape) menu    ^g search prompt  ^k delete line    ^p prev li    ^g prev page
^o ascii code      ^x search         ^l undelete line  ^n next li    ^o next page
^u end of file      ^a begin of line  ^w delete word    ^b back 1 char
^t begin of file    ^e end of line    ^r restore word   ^f forward 1 char
^c command          ^d delete char    ^j undelete char  ^z next word
L: 1 C: 1 =====
Sep 22 23:42:04 k6-2 login: ROOT LOGIN (root) ON ttyv121:13:33 obrien Exp $
#
# Internet server configuration database
#
# Define *both* IPv4 and IPv6 entries for dual-stack support.
# To disable a service, comment it out by prefixing the line with '#'.
# To enable a service, remove the '#' at the beginning of the line.
#
#ftp    stream  tcp    nowait  root    /usr/libexec/lukemftpd  ftpd -l -r
#ftp    stream  tcp    nowait  root    /usr/libexec/ftpd       ftpd -l
#ftp    stream  tcp6   nowait  root    /usr/libexec/ftpd       ftpd -l
#telnet stream  tcp    nowait  root    /usr/libexec/telnetd    telnetd
#telnet stream  tcp6   nowait  root    /usr/libexec/telnetd    telnetd
#shell  stream  tcp    nowait  root    /usr/libexec/rshd       rshd
#shell  stream  tcp6   nowait  root    /usr/libexec/rshd       rshd
#login  stream  tcp    nowait  root    /usr/libexec/rlogind    rlogind
#login  stream  tcp6   nowait  root    /usr/libexec/rlogind    rlogind
#finger stream  tcp    nowait/3/10 nobody /usr/libexec/fingerd    fingerd -s
file "/etc/inetd.conf", 119 lines

```

После добавления нужных сервисов нажатие **Esc** отобразит меню, позволяющее выйти с сохранением изменений.

## 2.9.4. Анонимный (Anonymous) FTP

User Confirmation Requested

Do you want to have anonymous FTP access to this machine?

Yes [ No ]

### 2.9.4.1. Запрещение анонимного FTP

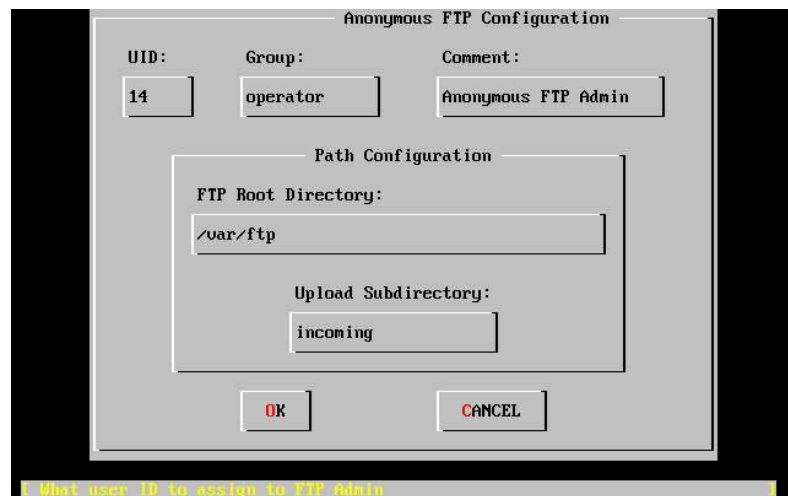
Выбор кнопки по умолчанию [ No ] и нажатие **Enter** все же позволит пользователям, имеющим учетные записи с паролями, использовать FTP для доступа к компьютеру.

### 2.9.4.2. Разрешение анонимного FTP

Кто угодно сможет получить доступ к компьютеру, если вы разрешите анонимные соединения FTP. Предварительно должны быть рассмотрены возможные проблемы с безопасностью. Более подробная информация о безопасности находится в Гл. 14.

Чтобы разрешить анонимный FTP, выберите [ Yes ], используя клавиши навигации, и нажмите **Enter**. Появится такой (или похожий) экран:

**Рисунок 2-34. Настройка по анонимного FTP по умолчанию**



Нажатие **F1** отобразит справку:

This screen allows you to configure the anonymous FTP user.

The following configuration values are editable:

**UID:** The user ID you wish to assign to the anonymous FTP user.  
All files uploaded will be owned by this ID.

**Group:** Which group you wish the anonymous FTP user to be in.

**Comment:** String describing this user in /etc/passwd

**FTP Root Directory:**

Where files available for anonymous FTP will be kept.

**Upload subdirectory:**

Where files uploaded by anonymous FTP users will go.

Корневой каталог ftp по умолчанию будет размещен в /var. Если в нем предположительно не хватает места для нужд FTP, можно использовать каталог /usr, выбрав корневой каталог FTP (FTP Root Directory) /usr/ftp.

Когда будут выбраны подходящие значения, нажмите **Enter**, чтобы продолжить.

```

User Confirmation Requested
Create a welcome message file for anonymous FTP users?

[ Yes ]      No

```

Если вы выберете [ Yes ] и нажмете **Enter**, запустится редактор, позволяющий отредактировать сообщение FTP.

### Рисунок 2-35. Редактирование FTP Welcome Message

```

^I (escape) menu ^y search prompt ^k delete line ^p prev line ^g prev page
^O ascii code ^x search ^l undelete line ^n next line ^o next page
^u end of file ^a begin of line ^w delete word ^b back char ^z next word
^t begin of file ^e end of line ^r restore word ^f forward char
^c command ^d delete char ^j undelete char ESC-Enter: exit
=====
Your welcome message here.

file "/var/ftp/etc/ftpmotd", 1 lines, read only

```

Этот текстовый редактор называется ee. Используйте инструкции, чтобы изменить сообщение, или измените сообщение позже, используя выбранный вами редактор. Обратите внимание, что имя/расположение файла показаны внизу окна редактора.

Нажмите **Esc** и появится меню с пунктом по умолчанию a) leave editor (выйти из редактора). Нажмите **Enter**, чтобы выйти и продолжить. Нажмите **Enter** еще раз, чтобы сохранить изменения, если они были сделаны.

## 2.9.5. Настройка сетевой файловой системы (Configure Network File System)

Сетевая файловая система (Network File System, NFS) позволяет совместно использовать файлы в сети. Компьютер может быть настроен как сервер, клиент, или как то и другое. Обратитесь к Разд. 23.3 за более подробной информацией.

### 2.9.5.1. Сервер NFS (NFS Server)

```

User Confirmation Requested
Do you want to configure this machine as an NFS server?

Yes      [ No ]

```



Если вам не нужен NFS сервер, выберите [ No ] и нажмите **Enter**.

Если выбран пункт [ Yes ], появится сообщение, говорящее о том, что должен быть создан файл exports.

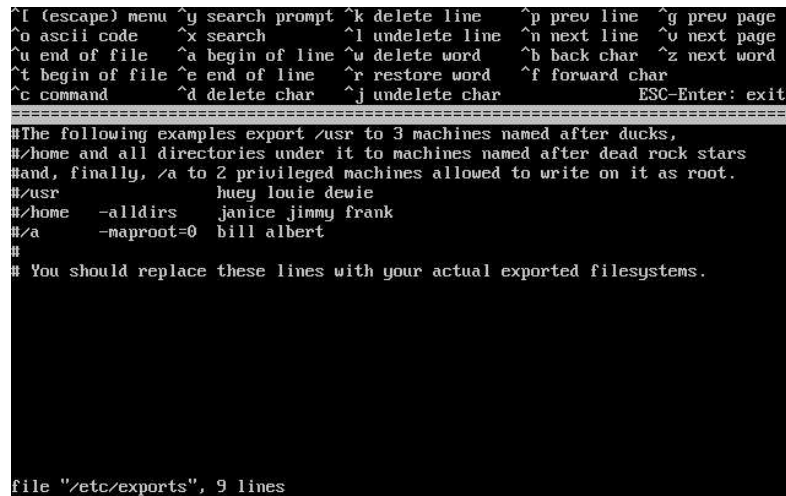
```

                                Message
Operating as an NFS server means that you must first configure an
/etc/exports file to indicate which hosts are allowed certain kinds of
access to your local filesystems.
Press [Enter] now to invoke an editor on /etc/exports
                                [ OK ]

```

Нажмите **Enter**, чтобы продолжить. Запустится текстовый редактор, позволяющий создать и отредактировать файл exports.

**Рисунок 2-36. Редактирование exports**



```

^I (escape) menu ^y search prompt ^k delete line ^p prev line ^g prev page
^o ascii code ^x search ^l undelete line ^n next line ^u next page
^u end of file ^a begin of line ^w delete word ^b back char ^z next word
^t begin of file ^e end of line ^r restore word ^f forward char
^c command ^d delete char ^j undelete char ESC-Enter: exit
=====
#The following examples export /usr to 3 machines named after ducks,
#/home and all directories under it to machines named after dead rock stars
#and, finally, /a to 2 privileged machines allowed to write on it as root.
#/usr      huey louie dewie
#/home     -alldirs  janice jimmy frank
#/a        -maproot=0 bill albert
#
# You should replace these lines with your actual exported filesystems.

file "/etc/exports", 9 lines

```

Используйте инструкции для добавления экспортируемых файловых систем сейчас, или позднее с помощью выбранного вами текстового редактора. Обратите внимание, что имя/расположение файла показаны внизу окна редактора.

Нажмите **Esc** и появится меню с пунктом по умолчанию a) leave editor. Нажмите **Enter**, чтобы выйти и продолжить.

### 2.9.5.2. Клиент NFS (NFS Client)

NFS клиент позволяет организовать доступ к серверам NFS.

```

                                User Confirmation Requested
Do you want to configure this machine as an NFS client?

                                Yes    [ No ]

```

С помощью клавиш навигации выберите [ Yes ] или [ No ], как потребуется, и нажмите **Enter**.

## 2.9.6. Профиль безопасности (Security Profile)

“Профиль безопасности” это набор параметров настройки, с помощью которого делается попытка достичь требуемого уровня безопасности и удобства путем разрешения или запрещения запуска определенных программ и другими настройками. Чем строже профиль безопасности, тем меньше программ будет запущено по умолчанию. Это один из основных принципов безопасности: не запускайте ничего кроме того, что необходимо.

Имейте ввиду, что профиль безопасности — это лишь установка по умолчанию. Любая программа может быть запущена или остановлена после установки FreeBSD путем редактирования или добавления соответствующих строк в `/etc/rc.conf`. Для получения более подробной информации обратитесь к странице справочника `rc.conf(5)`.

Следующая таблица описывает действие каждого профиля безопасности. Колонки показывают выбранный вами профиль безопасности, а ряды — программы или функции, которые профиль включает или выключает.

**Таблица 2-4. Возможные профили безопасности**

	<b>Extreme (Максимальный)</b>	<b>Moderate (Умеренный)</b>
sendmail(8)	НЕТ	ДА
sshd(8)	НЕТ	ДА
portmap(8)	НЕТ	МОЖЕТ БЫТЬ <sup>а</sup>
NFS server	НЕТ	ДА
securelevel(8)	ДА <sup>б</sup>	НЕТ

Примечания: а. Portmapper включен, если компьютер был настроен в процессе установки как NFS клиент или сервер. б.

```

User Confirmation Requested
Do you want to select a default security profile for this host (select
No for "medium" security)?

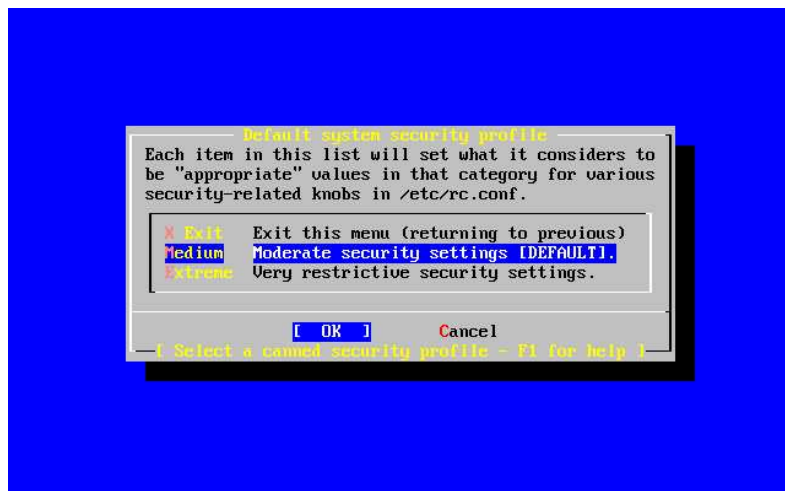
[ Yes ]      No

```

Выбор [ **No** ] и нажатие **Enter** установит профиль безопасности к среднему значению.

Выбор [ **Yes** ] и нажатие **Enter** позволит выбрать другой профиль безопасности.

Рисунок 2-37. Параметры профиля безопасности



Нажмите **F1**, чтобы получить помощь. Нажмите **Enter** для возврата к меню выбора.

Используйте клавиши навигации для выбора **Medium**, если вы не уверены, что необходим другой уровень. С подсвеченным **[ OK ]** нажмите **Enter**.

Будет отображен запрос на подтверждение, соответствующий выбранным настройкам безопасности.

Message

Moderate security settings have been selected.

Sendmail and SSHd have been enabled, securelevels are disabled, and NFS server setting have been left intact. PLEASE NOTE that this still does not save you from having to properly secure your system in other ways or exercise due diligence in your administration, this simply picks a standard set of out-of-box defaults to start with.

To change any of these settings later, edit /etc/rc.conf

[OK]

Message

Extreme security settings have been selected.

Sendmail, SSHd, and NFS services have been disabled, and securelevels have been enabled. PLEASE NOTE that this still does not save you from having to properly secure your system in other ways or exercise due diligence in your administration, this simply picks a more secure set of out-of-box defaults to start with.

To change any of these settings later, edit /etc/rc.conf

[OK]

Нажмите **Enter**, чтобы продолжить послеустановочные настройки.

**Внимание:** Профиль безопасности не панацея! Если даже вы выберете экстремальные установки, понадобится поддерживать безопасность, читая соответствующий список рассылки (Разд. С.1), используя хорошие пароли и идентификационные фразы, и придерживаясь хорошей практики безопасности. Профиль всего лишь повышает желаемый уровень безопасности свежеставленной системы до приемлемого значения.

## 2.9.7. Настройки системной консоли (System Console Settings)

Есть несколько параметров для настройки системной консоли.

```

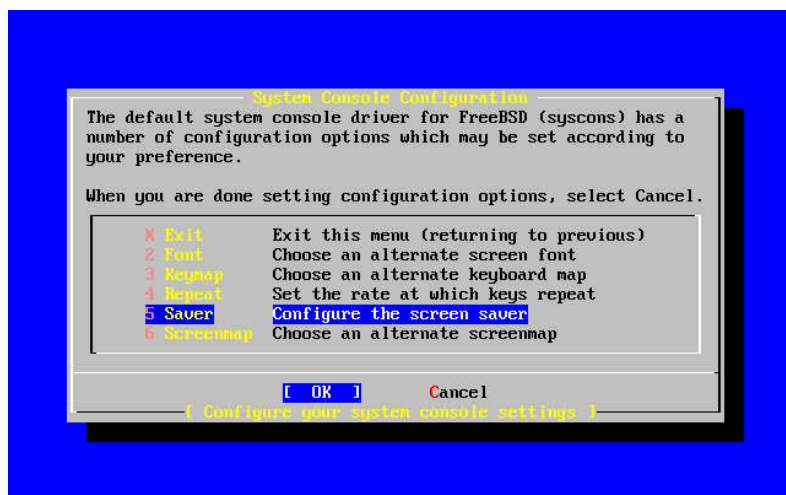
User Confirmation Requested
Would you like to customize your system console settings?

[ Yes ] No

```

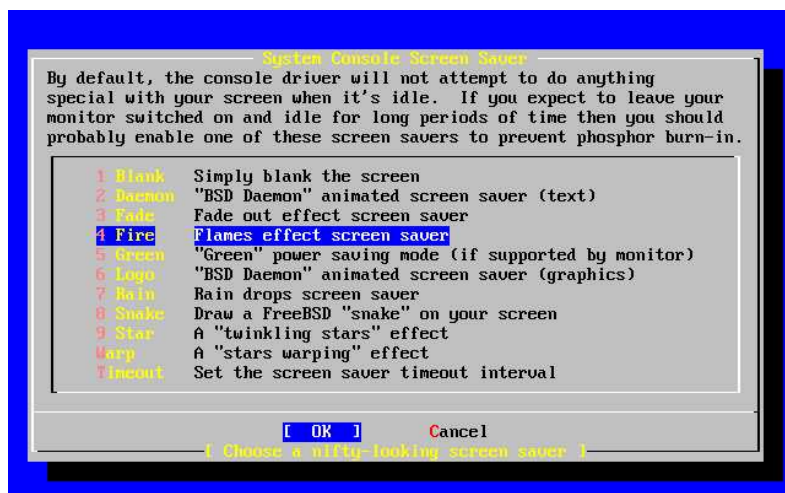
Для просмотра и настройки параметров выберите [ Yes ] и нажмите **Enter**.

Рисунок 2-38. Параметры настройки системной консоли



Часто используемая опция это хранитель экрана (screen saver). Используйте клавиши навигации для выбора **Saver** и нажмите **Enter**.

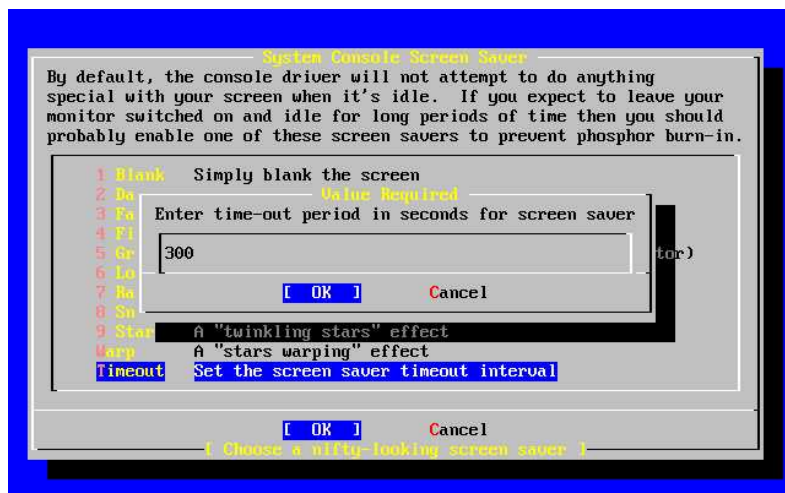
Рисунок 2-39. Параметры хранителя экрана



Выберите подходящий хранитель экрана с помощью клавиш навигации и нажмите **Enter**. Опять появится меню настройки системной консоли.

Время по умолчанию 300 секунд. Для изменения временного интервала выберите **Saver** еще раз. В меню настроек хранителя экрана выберите **Timeout** с помощью клавиш навигации и нажмите **Enter**. Появится меню:

Рисунок 2-40. Временной интервал хранителя экрана



Значение может быть изменено, затем выберите [ OK ] и нажмите **Enter** для возврата в меню настройки системной консоли.

Рисунок 2-41. Выход из меню конфигурации консоли



Выбор **Exit** и нажатие **Enter** вернет вас к послеустановочной настройке.

### 2.9.8. Установка часового пояса (Setting The Time Zone)

Установка часового пояса на компьютере позволит ему автоматически вносить поправки к местному времени и правильно выполнять другие, связанные с часовым поясом функции.

Пример приведен для компьютера, расположенного в восточном часовом поясе Соединенных Штатов. Ваш выбор будет зависеть от вашего географического положения.

```

User Confirmation Requested
Would you like to set this machine's time zone now?

[ Yes ]   No

```

Выберите **[ Yes ]** и нажмите **Enter** для установки часового пояса.

```

User Confirmation Requested
Is this machine's CMOS clock set to UTC? If it is set to local time
or you don't know, please choose NO here!

Yes    [ No ]

```

Выберите **[ Yes ]** или **[ No ]** в зависимости от настроек часов компьютера и нажмите **Enter**.

Рисунок 2-42. Выбор региона



Соответствующий регион выбран с помощью клавиш навигации и нажат **Enter**.

Рисунок 2-43. Выбор страны



Выберите соответствующую страну с помощью клавиш навигации и нажмите **Enter**.

Рисунок 2-44. Выбор часового пояса



Выбран соответствующий часовой пояс с помощью клавиш навигации и нажат **Enter**.

```
Confirmation
Does the abbreviation 'EDT' look reasonable?

[ Yes ]   No
```

Правильно будет согласиться с назначением аббревиатуры временного пояса. Если она подходит, нажмите **Enter**, чтобы продолжить послеустановочную настройку.

## 2.9.9. Совместимость с Linux (Linux Compatibility)

```
User Confirmation Requested
Would you like to enable Linux binary compatibility?

[ Yes ]   No
```

Выбор [ Yes ] и нажатие **Enter** позволит запускать программы Linux под FreeBSD. Программа установки добавит соответствующие пакеты для совместимости с Linux.

При установке по FTP, компьютеру потребуется соединиться с интернет. Иногда на сервере ftp нет всех необходимых компонент, например для бинарной совместимости с Linux. Эти компоненты могут быть установлены позже, если потребуется.

## 2.9.10. Настройка мыши (Mouse Settings)

Эти настройки позволят вырезать и вставлять текст в консоли и пользовательских программах с помощью трехкнопочной мыши. Если используется двухкнопочная мышь, обратитесь к странице справочника moused(8) после установки, чтобы узнать подробности об эмуляции трехкнопочной мыши. Этот пример приведен для настройки не-USB мыши (например мыши для порта PS/2 или COM):

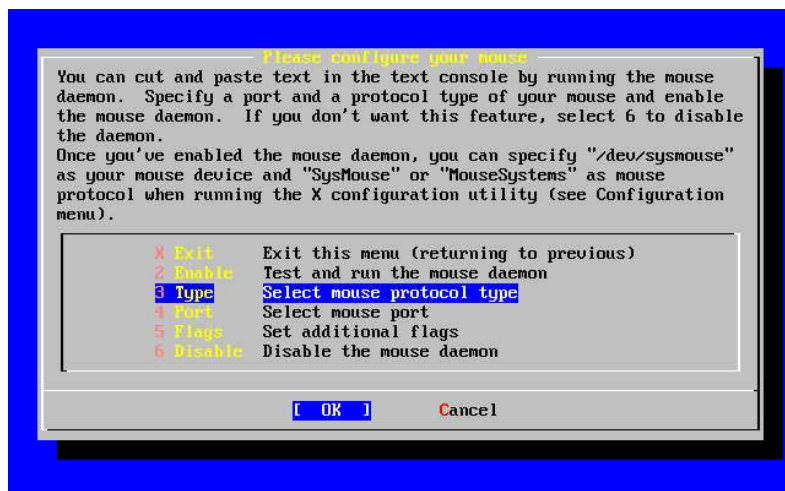


User Confirmation Requested  
Does this system have a non-USB mouse attached to it?

[ Yes ]      No

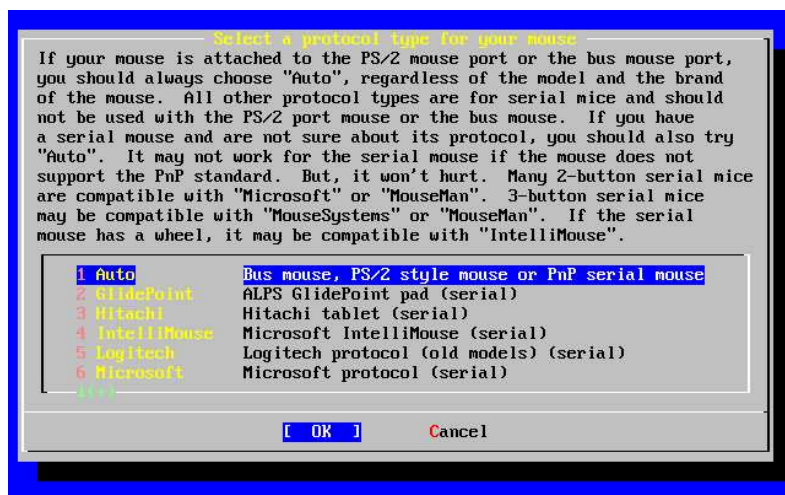
Выберите [ Yes ] для не-USB мыши или [ No ] для USB мыши и нажмите **Enter**.

Рисунок 2-45. Выбор протокола мыши



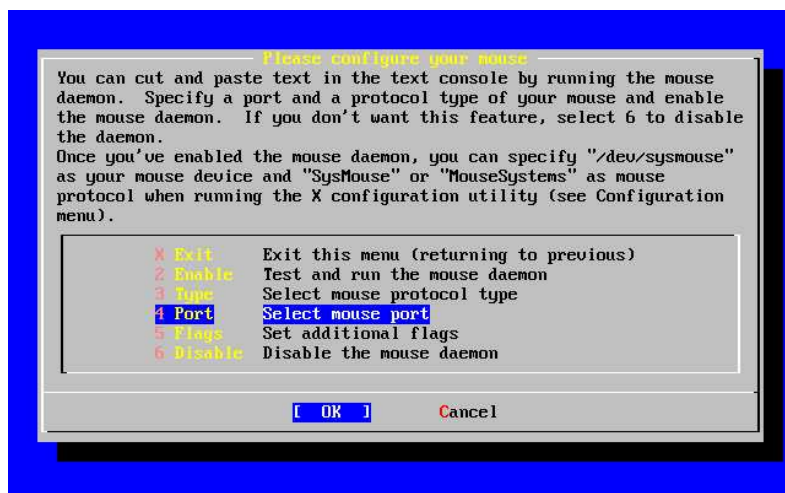
Используйте клавиши навигации для выбора **Type** и нажмите **Enter**.

Рисунок 2-46. Установка протокола мыши



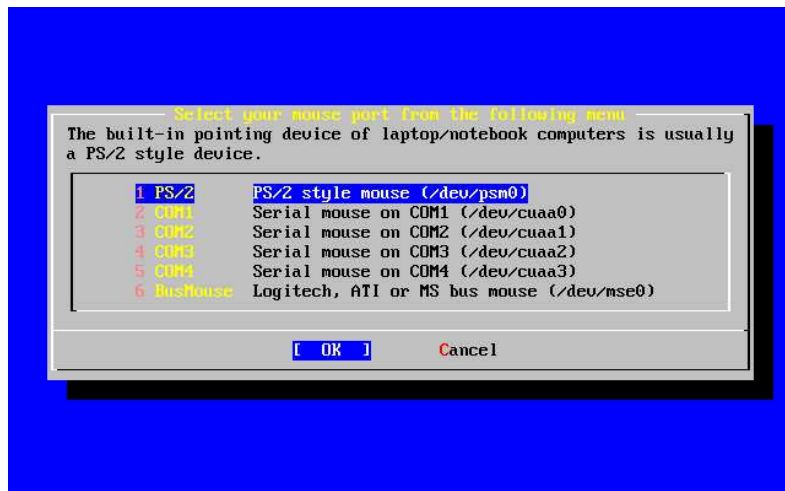
В этом примере использована PS/2 мышь, поэтому подойдет протокол по умолчанию **Auto**. Чтобы изменить протокол, используйте клавиши навигации для выбора другого пункта. Убедитесь, что выбран [ OK ], и нажмите **Enter** для выхода из меню.

Рисунок 2-47. Настройка порта мыши



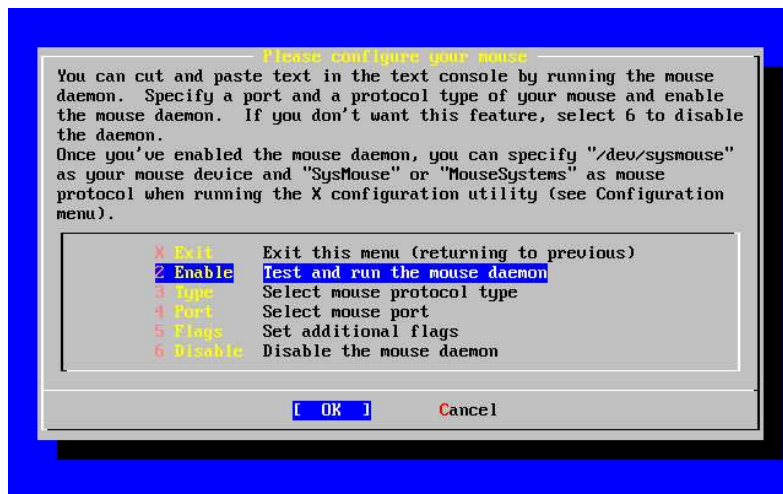
Используйте клавиши навигации для выбора Port и нажмите **Enter**.

Рисунок 2-48. Установка порта мыши



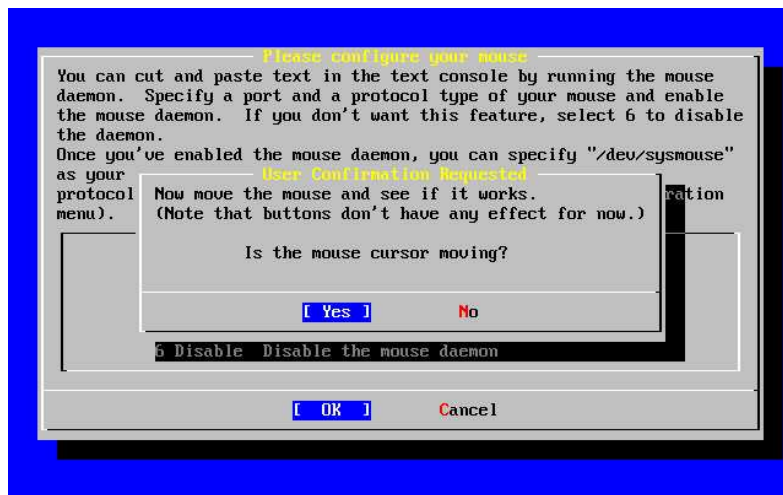
К этой системе подключена мышь PS/2, поэтому подходит значение по умолчанию PS/2. Чтобы изменить порт, используйте клавиши навигации и нажмите **Enter**.

Рисунок 2-49. Запуск мышиного демона



Наконец, используйте клавиши навигации для выбора **Enable**, затем нажмите **Enter** для запуска и тестирования демона мыши.

Рисунок 2-50. Проверка мышиного демона



Подвигайте курсор по экрану и убедитесь, что он движется правильно. Если это так, выберите [ **Yes** ] и нажмите **Enter**. Если нет, мышь не была правильно настроена — выберите [ **No** ] и попробуйте использовать другие опции настройки.

Выберите **Exit** с помощью клавиш навигации и нажмите **Enter** для возврата к послеустановочной настройке.

## 2.9.11. Настройка дополнительных сетевых сервисов

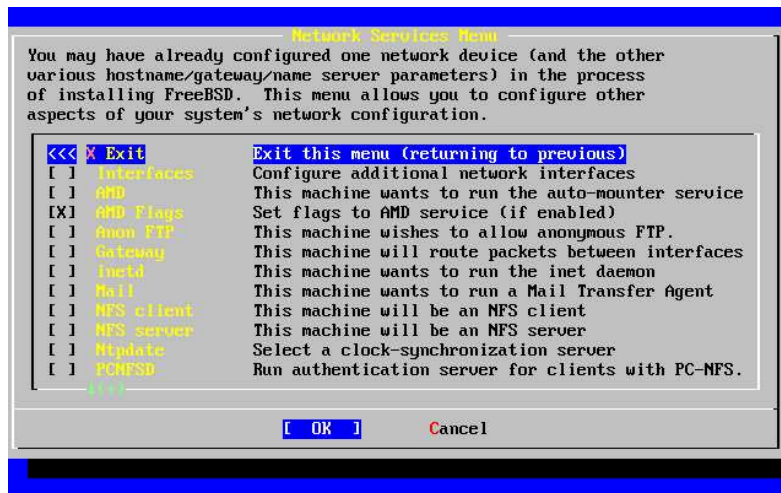
Настройка сетевых сервисов может быть пугающей задачей для новых пользователей, если у них нет

опыта в этой области. Работа в сети, включая интернет, критична для многих современных операционных систем, включая FreeBSD; очень важно иметь некоторые знания о дополнительных сетевых возможностях FreeBSD. Настраивая их во время установки, пользователь получает некоторую информацию о различных доступных сервисах.

Сетевые сервисы — это программы, принимающие данные из любой точки сети. Чтобы убедиться, что эти программы не делают ничего “вредного”, требуется особое внимание. К сожалению, программисты не совершенны и время от времени допускают ошибки в сетевых сервисах; эти ошибки позволяют атакующим взломать их и использовать в собственных целях. Важно, чтобы вы включали сетевые сервисы только в том случае, если знаете, какие сервисы нужны. В случае сомнений лучше не включать сервис, пока он не понадобится. Вы всегда можете включить его позже, запустив **sysinstall** еще раз, или используя возможности, предоставляемые файлом `/etc/rc.conf`.

Выбор опции **Networking** отобразит меню, похожее на это:

**Рисунок 2-51. Верхняя часть меню настройки сети (Network Configuration)**



Первая опция, **Interfaces** (интерфейсы), была ранее описана в Разд. 2.9.1, эту опцию можно проигнорировать.

Выбор опции **AMD** добавит поддержку BSD утилиты автоматического монтирования. Она обычно используется вместе с протоколом NFS (смотрите ниже) для автоматического монтирования удаленных файловых систем. Здесь не требуется специальной конфигурации.

Следующая линия это опция флаги **AMD** (flags). Когда эта опция выбрана, появится меню для ввода специфичных флагов AMD. Меню уже содержит набор различных опций:

```
-a /.amd_mnt -l syslog /host /etc/amd.map /net /etc/amd.map
```

Опция `-a` указывает месторасположение монтирования систем по умолчанию, в этом примере `/.amd_mnt`. Опция `-l` указывает `log` файл по умолчанию; тем не менее, если для ведения логов используется `syslogd`, все логи отправляются даемону системных логов. Каталог `/host` используется для монтирования экспортированных файловых систем с удаленного хоста, а `/net` для монтирования экспортированных файловых систем с IP адреса. Файл `/etc/amd.map` определяет опции по умолчанию для монтируемых AMD каталогов.

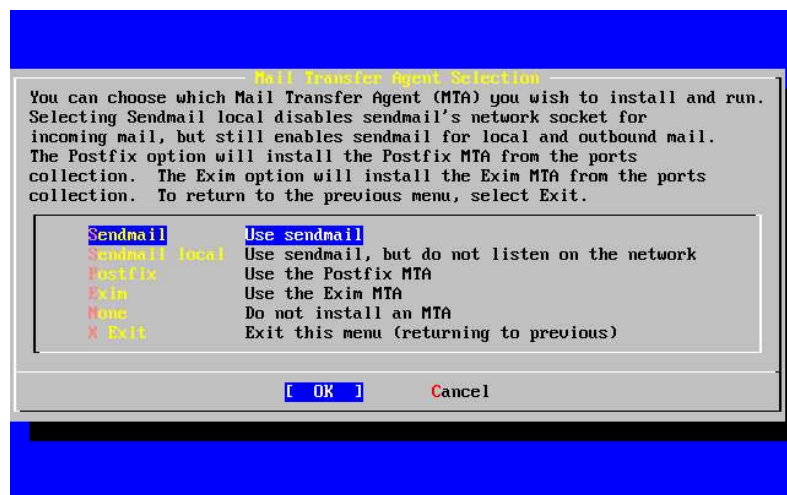
Опция **Anon FTP** позволяет получить анонимный доступ к FTP. Выберите ее, чтобы сделать этот компьютер анонимным FTP сервером. Помните о риске безопасности, создаваемом этой опцией. Будет отображено еще одно меню для разъяснения риска безопасности и для более тонкой настройки.

Опция **Gateway** сделает компьютер шлюзом, как было объяснено ранее. Этот пункт может быть использован для отмены опции **Gateway**, если она была ошибочно установлена в процессе установки.

Опцию **Inetd** можно использовать для настройки или полного отключения демона `inetd(8)`, как было описано выше.

Опция **Mail** используется для настройки системного MTA по умолчанию (Mail Transfer Agent, агент передачи почты). Выбор этой опции приведет к появлению следующего меню:

**Рисунок 2-52. Выбор MTA по умолчанию**



Здесь предоставляется выбор MTA для установки по умолчанию. MTA это почтовый сервер, доставляющий почту пользователям системы или интернет.

Выбор **Sendmail** приведет к установке популярного сервера **sendmail**, MTA по умолчанию для FreeBSD. Опция **Sendmail local** настроит **sendmail** в качестве MTA по умолчанию, но отключит возможность получения входящей почты из интернет. Другие MTA, представленные здесь, **Postfix** и **Exim**, действуют подобно **Sendmail**. Оба они доставляют почту; тем не менее, некоторые пользователи предпочитают эти альтернативы **sendmail**.

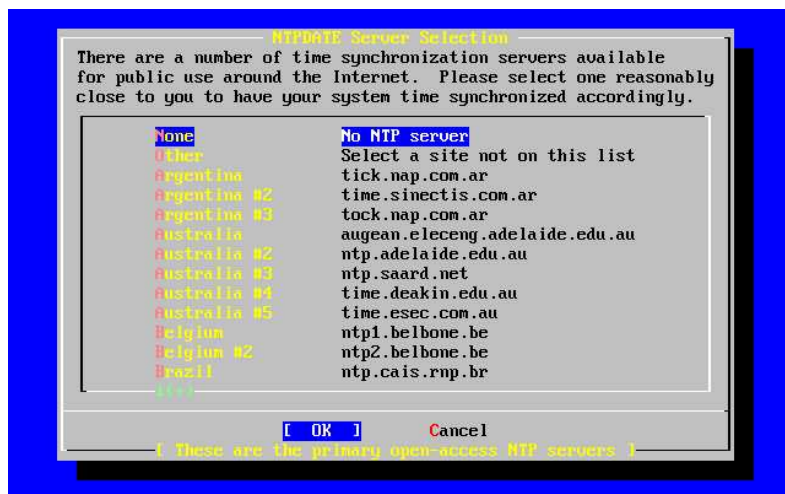
После выбора MTA или пропуска этого пункта появится меню настройки сети со следующей опцией, клиент **NFS**.

Клиент **NFS** позволяет настроить систему для соединения с сервером через NFS. Сервер NFS дает другим машинам доступ к файловой системе через протокол NFS. Если это отдельно стоящий компьютер, опцию можно не выбирать. NFS может потребовать дополнительной настройки позже; обращайтесь к Разд. 23.3 за более подробной информацией о настройке сервера и клиента.

Ниже находится опция **NFS сервер**, позволяющая вам настроить систему для работы в качестве NFS сервера. Она добавляет требуемую информацию для запуска процедуры вызова удаленных сервисов RPC (remote procedure call services). RPC используется для организации соединения между хостами и программами.

Следующая строка это **Ntpdate**, которая отвечает за синхронизацию времени. Когда эта опция выбрана, появится приблизительно такое меню:

Рисунок 2-53. Настройка Ntpdate

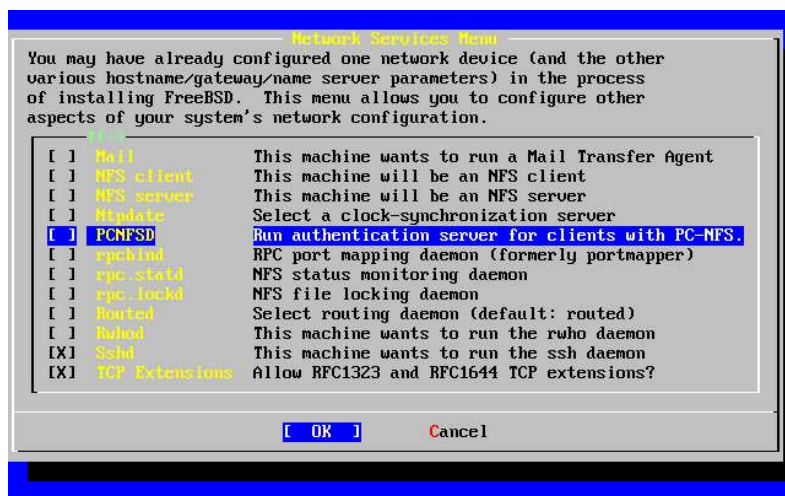


Выберите из этого меню ближайший к вашему местонахождению сервер. Выбор ближайшего сервера сделает синхронизацию времени более точной, поскольку у более дальнего сервера может быть более длинная задержка соединения.

Следующая опция это выбор PCNFSD. Эта опция устанавливает пакет net/pcnfsd из коллекции портов. Это полезная утилита, предоставляющая сервисы авторизации NFS для систем, в которых собственная отсутствует, таких как Microsoft MS-DOS.

Теперь вы можете прокрутить меню немного вниз, чтобы увидеть другие опции:

Рисунок 2-54. Нижняя часть меню настройки сети



Утилиты `rpcbind(8)`, `rpc.statd(8)` и `rpc.lockd(8)` для удаленного вызова процедур (Remote Procedure Calls, RPC). Утилита `rpcbind` управляет соединением между NFS серверами и клиентами, она требуется серверу NFS для корректной работы. Демон **`rpc.statd`** взаимодействует с демонами **`rpc.statd`** на других хостах для обеспечения мониторинга статуса. Полученный статус обычно хранится в файле `/var/db/statd.status`. Последняя опция это **`rpc.lockd`**, которая, будучи выбранной, предоставляет сервисы блокировки файлов. Она обычно используется с **`rpc.statd`** для отслеживания хостов, запрашивающих блокировки и частоты этих запросов. Хотя две последние опции прекрасно подходят для отладки, они не требуются серверам и клиентам NFS для корректной работы.

Ниже в списке опций находится **`Routed`**, демон маршрутизации. Утилита `routed(8)` управляет сетевыми таблицами маршрутизации, обнаруживает ширококестельные маршрутизаторы, и предоставляет копию таблиц маршрутизации любому физически подключенному к сети хосту по запросу. Это обычно используется компьютерами, являющимися шлюзом для локальной сети. Когда эта опция выбрана, появится меню выбора местоположения утилиты по умолчанию. Значение по умолчанию уже выбрано, чтобы подтвердить его нажмите **Enter**. Появится следующее меню, запрашивающее флаги для **`routed`**. По умолчанию это флаг `-q`, он должен уже быть на экране.

На следующей линии находится опция **`Rwhod`**, выбор которой приведет к запуску демона `rwod(8)` при старте системы. Утилита `rwod` периодически рассылает ширококестельные системные сообщения по сети или собирает их в режиме "потребителя". Дополнительную информацию можно найти на страницах справочника `ruptime(1)` и `rwod(1)`.

Следующая перед последней опцией в списке это демон `sshd(8)`. Это сервер безопасной оболочки (secure shell) из **OpenSSH** и он настоятельно рекомендуется для использования вместо стандартных серверов `telnet` и `FTP`. Сервер `sshd` используется для создания безопасных соединений от одного хоста к другому с использованием шифрования.

И наконец, опция **TCP Extensions**. Она включает расширения TCP, определенные в RFC 1323 и RFC 1644. Хотя для многих хостов они могут повысить скорость соединения, с другими соединения может быть оборвано. Для серверов эти расширения не рекомендуются, но клиентским компьютерам могут дать некоторые преимущества.

Теперь, после настройки сетевых сервисов, вы можете прокрутить меню к верхнему пункту, **Exit**, и перейти к следующему разделу настройки.

## 2.9.12. Настройка X Сервера (Configure X Server)

**Замечание:** В FreeBSD 5.3-RELEASE настройка X сервера была удалена из **`sysinstall`**, X сервер нужно будет настроить после установки FreeBSD. Более подробная информация, относящаяся к установке и настройке X сервера, находится в Гл. 5. Вы можете пропустить этот раздел, если только не устанавливаете версию FreeBSD до 5.3-RELEASE.

Чтобы использовать графический интерфейс пользователя **KDE**, **GNOME**, или другие, нужно настроить X сервер.

**Замечание:** Чтобы запустить **XFree86** не из под `root`, вам потребуется установленный `x11/wrapper`. Он устанавливается по умолчанию начиная с FreeBSD 4.7. В более ранних версиях он может быть добавлен из меню выбора пакетов (Package Selection menu).



Чтобы узнать, поддерживается ли ваша видеокарта, обратитесь к сайту XFree86 (<http://www.xfree86.org/>).

```

User Confirmation Requested
Would you like to configure your X server at this time?

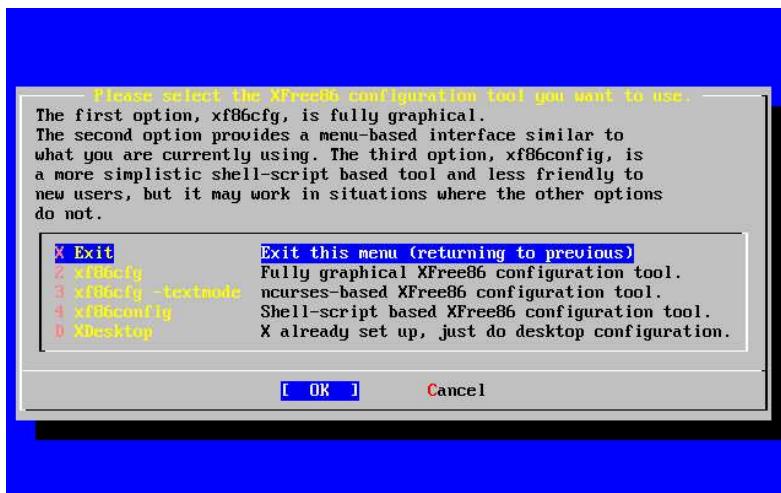
[ Yes ]   No

```

**Внимание:** Необходимо узнать параметры монитора и информацию о видеокарте. Установкой неправильных параметров можно повредить оборудование. Если у вас нет этой информации, выберите [ No ] и выполните конфигурацию после установки, когда необходимая информация будет собрана. Используя `sysinstall` (`/stand/sysinstall` в FreeBSD версии 5.2 и более ранних), выберите **Configure**, а затем **XFree86**. Неправильная настройка X сервера может "подвесить" компьютер. Зачастую настройку X сервера рекомендуют производить после завершения установки.

Если у вас есть информация о графической карте и мониторе, выберите [ Yes ] и нажмите **Enter** для начала настройки X сервера.

Рисунок 2-55. Меню выбора способа конфигурации



Есть несколько путей настройки X сервера. Используйте клавиши навигации для выбора одного из методов и нажмите **Enter**. Внимательно прочтите все инструкции.

Команды **xf86cfg** и **xf86cfg -textmode** могут сделать экран темным и запуститься через несколько секунд. Будьте терпеливы.

Далее описывается использование программы настройки **xf86config**. Выбор настроек будет зависеть от оборудования системы, так что ваши настройки возможно будут отличаться от тех, что приведены:

```

Message
You have configured and been running the mouse daemon.
Choose "/dev/sysmouse" as the mouse port and "SysMouse" or

```



"MouseSystems" as the mouse protocol in the X configuration utility.

[ OK ]

[ Press enter to continue ]

Это говорит о том, что предварительно настроенный демон мыши был обнаружен. Нажмите **Enter**, чтобы продолжить.

При запуске **xf86config** отобразит краткое введение:

This program will create a basic XF86Config file, based on menu selections you make.

The XF86Config file usually resides in /usr/X11R6/etc/X11 or /etc/X11. A sample XF86Config file is supplied with XFree86; it is configured for a standard VGA card and monitor with 640x480 resolution. This program will ask for a pathname when it is ready to write the file.

You can either take the sample XF86Config as a base and edit it for your configuration, or let this program produce a base XF86Config file for your configuration and fine-tune it.

Before continuing with this program, make sure you know what video card you have, and preferably also the chipset it uses and the amount of video memory on your video card. SuperProbe may be able to help with this.

Press enter to continue, or ctrl-c to abort.

Нажатие **Enter** запустит настройку мыши. Следуя инструкциям, выберите "Mouse Systems" в качестве протокола мыши и /dev/sysmouse в качестве порта мыши, даже если вы используете PS/2, как показано в примере.

First specify a mouse protocol type. Choose one from the following list:

1. Microsoft compatible (2-button protocol)
2. Mouse Systems (3-button protocol) & FreeBSD moused protocol
3. Bus Mouse
4. PS/2 Mouse
5. Logitech Mouse (serial, old type, Logitech protocol)
6. Logitech MouseMan (Microsoft compatible)
7. MM Series
8. MM HitTablet
9. Microsoft IntelliMouse

If you have a two-button mouse, it is most likely of type 1, and if you have a three-button mouse, it can probably support both protocol 1 and 2. There are two main varieties of the latter type: mice with a switch to select the protocol, and mice that default to 1 and require a button to be held at boot-time to select protocol 2. Some mice can be convinced to do 2 by sending a special sequence to the serial port (see the ClearDTR/ClearRTS options).

Enter a protocol number: 2

You have selected a Mouse Systems protocol mouse. If your mouse is normally in Microsoft-compatible mode, enabling the ClearDTR and ClearRTS options may cause it to switch to Mouse Systems mode when the server starts.

Please answer the following question with either 'y' or 'n'.

Do you want to enable ClearDTR and ClearRTS? n

You have selected a three-button mouse protocol. It is recommended that you do not enable Emulate3Buttons, unless the third button doesn't work.

Please answer the following question with either 'y' or 'n'.

Do you want to enable Emulate3Buttons? y

Now give the full device name that the mouse is connected to, for example /dev/tty00. Just pressing enter will use the default, /dev/mouse.

On FreeBSD, the default is /dev/sysmouse.

Mouse device: /dev/sysmouse

Следующий пункт настройки это клавиатура. В примере показана модель generic 101-key. В качестве варианта можно использовать любое имя, или просто нажать **Enter**, чтобы принять значение по умолчанию.

Please select one of the following keyboard types that is the better description of your keyboard. If nothing really matches, choose 1 (Generic 101-key PC)

- 1 Generic 101-key PC
- 2 Generic 102-key (Intl) PC
- 3 Generic 104-key PC
- 4 Generic 105-key (Intl) PC
- 5 Dell 101-key PC
- 6 Everex STEPnote
- 7 Keytronic FlexPro
- 8 Microsoft Natural
- 9 Northgate OmniKey 101
- 10 Winbook Model XP5
- 11 Japanese 106-key
- 12 PC-98xx Series
- 13 Brazilian ABNT2
- 14 HP Internet
- 15 Logitech iTouch
- 16 Logitech Cordless Desktop Pro
- 17 Logitech Internet Keyboard
- 18 Logitech Internet Navigator Keyboard
- 19 Compaq Internet
- 20 Microsoft Natural Pro
- 21 Genius Comfy KB-16M
- 22 IBM Rapid Access
- 23 IBM Rapid Access II
- 24 Chicony Internet Keyboard
- 25 Dell Internet Keyboard

Enter a number to choose the keyboard.

1

Please select the layout corresponding to your keyboard

- 1 U.S. English
- 2 U.S. English w/ ISO9995-3
- 3 U.S. English w/ deadkeys
- 4 Albanian
- 5 Arabic
- 6 Armenian
- 7 Azerbaidjani
- 8 Belarusian
- 9 Belgian
- 10 Bengali
- 11 Brazilian
- 12 Bulgarian
- 13 Burmese
- 14 Canadian
- 15 Croatian
- 16 Czech
- 17 Czech (qwerty)
- 18 Danish

Enter a number to choose the country.

Press enter for the next page

1

Please enter a variant name for 'us' layout. Or just press enter for default variant

us

Please answer the following question with either 'y' or 'n'.

Do you want to select additional XKB options (group switcher, group indicator, etc.)? n

Теперь переходим к настройке монитора. Не превышайте значений для вашего монитора. Он может быть поврежден. Если у вас есть какие-то сомнения, проводите настройку после того, как получите информацию.

Now we want to set the specifications of the monitor. The two critical parameters are the vertical refresh rate, which is the rate at which the whole screen is refreshed, and most importantly the horizontal sync rate, which is the rate at which scanlines are displayed.

The valid range for horizontal sync and vertical sync should be documented in the manual of your monitor. If in doubt, check the monitor database /usr/X11R6/lib/X11/doc/Monitors to see if your monitor is there.

Press enter to continue, or ctrl-c to abort.

You must indicate the horizontal sync range of your monitor. You can either select one of the predefined ranges below that correspond to industry-standard monitor types, or give a specific range.

It is VERY IMPORTANT that you do not specify a monitor type with a horizontal sync range that is beyond the capabilities of your monitor. If in doubt, choose a conservative setting.

```

    hsync in kHz; monitor type with characteristic modes
1  31.5; Standard VGA, 640x480 @ 60 Hz
2  31.5 - 35.1; Super VGA, 800x600 @ 56 Hz
3  31.5, 35.5; 8514 Compatible, 1024x768 @ 87 Hz interlaced (no 800x600)
4  31.5, 35.15, 35.5; Super VGA, 1024x768 @ 87 Hz interlaced, 800x600 @ 56 Hz
5  31.5 - 37.9; Extended Super VGA, 800x600 @ 60 Hz, 640x480 @ 72 Hz
6  31.5 - 48.5; Non-Interlaced SVGA, 1024x768 @ 60 Hz, 800x600 @ 72 Hz
7  31.5 - 57.0; High Frequency SVGA, 1024x768 @ 70 Hz
8  31.5 - 64.3; Monitor that can do 1280x1024 @ 60 Hz
9  31.5 - 79.0; Monitor that can do 1280x1024 @ 74 Hz
10 31.5 - 82.0; Monitor that can do 1280x1024 @ 76 Hz
11 Enter your own horizontal sync range

```

Enter your choice (1-11): 6

You must indicate the vertical sync range of your monitor. You can either select one of the predefined ranges below that correspond to industry-standard monitor types, or give a specific range. For interlaced modes, the number that counts is the high one (e.g. 87 Hz rather than 43 Hz).

```

1  50-70
2  50-90
3  50-100
4  40-150
5  Enter your own vertical sync range

```

Enter your choice: 2

You must now enter a few identification/description strings, namely an identifier, a vendor name, and a model name. Just pressing enter will fill in default names.

The strings are free-form, spaces are allowed.

Enter an identifier for your monitor definition: Hitachi

Далее производится выбор видеокарты из списка. Если вы пропустили карту в списке, продолжайте нажимать **Enter**, и список повторится. Показан только отрывок:

Now we must configure video card specific settings. At this point you can choose to make a selection out of a database of video card definitions. Because there can be variation in Ramdacs and clock generators even between cards of the same model, it is not sensible to blindly copy the settings (e.g. a Device section). For this reason, after you make a selection, you will still be asked about the components of the card, with the settings from the chosen database entry presented as a strong hint.

The database entries include information about the chipset, what driver to run, the Ramdac and ClockChip, and comments that will be included in the Device section. However, a lot of definitions only hint about what driver to run (based on the chipset the card uses) and are untested.

If you can't find your card in the database, there's nothing to worry about. You should only choose a database entry that is exactly the same model as your card; choosing one that looks similar is just a bad idea (e.g. a GemStone Snail 64 may be as different from a GemStone Snail 64+ in terms of hardware as can be).

Do you want to look at the card database? y

288	Matrox Millennium G200 8MB	mgag200
289	Matrox Millennium G200 SD 16MB	mgag200
290	Matrox Millennium G200 SD 4MB	mgag200
291	Matrox Millennium G200 SD 8MB	mgag200
292	Matrox Millennium G400	mgag400
293	Matrox Millennium II 16MB	mga2164w
294	Matrox Millennium II 4MB	mga2164w
295	Matrox Millennium II 8MB	mga2164w
296	Matrox Mystique	mga1064sg
297	Matrox Mystique G200 16MB	mgag200
298	Matrox Mystique G200 4MB	mgag200
299	Matrox Mystique G200 8MB	mgag200
300	Matrox Productiva G100 4MB	mgag100
301	Matrox Productiva G100 8MB	mgag100
302	MediaGX	mediagx
303	MediaVision Proaxcel 128	ET6000
304	Mirage Z-128	ET6000
305	Miro CRYSTAL VRX	Verite 1000

Enter a number to choose the corresponding card definition.  
Press enter for the next page, q to continue configuration.

288

Your selected card definition:

Identifier: Matrox Millennium G200 8MB  
Chipset: mgag200  
Driver: mga  
Do NOT probe clocks or use any Clocks line.

Press enter to continue, or ctrl-c to abort.

Now you must give information about your video card. This will be used for the "Device" section of your video card in XF86Config.

You must indicate how much video memory you have. It is probably a good idea to use the same approximate amount as that detected by the server you intend to use. If you encounter problems that are due to the used server not supporting the amount memory you have (e.g. ATI Mach64 is limited to 1024K with the SVGA server), specify the maximum amount supported by the server.

How much video memory do you have on your video card:

- 1 256K
- 2 512K
- 3 1024K
- 4 2048K
- 5 4096K
- 6 Other

Enter your choice: 6

Amount of video memory in Kbytes: 8192

You must now enter a few identification/description strings, namely an identifier, a vendor name, and a model name. Just pressing enter will fill in default names (possibly from a card definition).

Your card definition is Matrox Millennium G200 8MB.

The strings are free-form, spaces are allowed.  
Enter an identifier for your video card definition:

Далее, производится установка видеорежимов для требуемого разрешения. Обычно используемые значения 640x480, 800x600 и 1024x768, но они зависят от возможностей видеокарты, размера монитора и зрительного комфорта. Выбирая глубину цвета, устанавливайте самое большое поддерживаемое видеокартой значение.

For each depth, a list of modes (resolutions) is defined. The default resolution that the server will start-up with will be the first listed mode that can be supported by the monitor and card.  
Currently it is set to:

```
"640x480" "800x600" "1024x768" "1280x1024" for 8-bit
"640x480" "800x600" "1024x768" "1280x1024" for 16-bit
"640x480" "800x600" "1024x768" "1280x1024" for 24-bit
```

Modes that cannot be supported due to monitor or clock constraints will be automatically skipped by the server.

- 1 Change the modes for 8-bit (256 colors)
- 2 Change the modes for 16-bit (32K/64K colors)
- 3 Change the modes for 24-bit (24-bit color)
- 4 The modes are OK, continue.

Enter your choice: 2

Select modes from the following list:

- 1 "640x400"
- 2 "640x480"
- 3 "800x600"
- 4 "1024x768"
- 5 "1280x1024"
- 6 "320x200"
- 7 "320x240"
- 8 "400x300"
- 9 "1152x864"
- a "1600x1200"
- b "1800x1400"
- c "512x384"

Please type the digits corresponding to the modes that you want to select.  
For example, 432 selects "1024x768" "800x600" "640x480", with a  
default mode of 1024x768.

Which modes? 432

You can have a virtual screen (desktop), which is screen area that is larger than the physical screen and which is panned by moving the mouse to the edge of the screen. If you don't want virtual desktop at a certain resolution, you cannot have modes listed that are larger. Each color depth can have a differently-sized virtual screen

Please answer the following question with either 'y' or 'n'.  
Do you want a virtual screen that is larger than the physical screen? n

For each depth, a list of modes (resolutions) is defined. The default resolution that the server will start-up with will be the first listed mode that can be supported by the monitor and card.  
Currently it is set to:

"640x480" "800x600" "1024x768" "1280x1024" for 8-bit  
"1024x768" "800x600" "640x480" for 16-bit  
"640x480" "800x600" "1024x768" "1280x1024" for 24-bit

Modes that cannot be supported due to monitor or clock constraints will be automatically skipped by the server.

- 1 Change the modes for 8-bit (256 colors)

- 2 Change the modes for 16-bit (32K/64K colors)
- 3 Change the modes for 24-bit (24-bit color)
- 4 The modes are OK, continue.

Enter your choice: 4

Please specify which color depth you want to use by default:

- 1 1 bit (monochrome)
- 2 4 bits (16 colors)
- 3 8 bits (256 colors)
- 4 16 bits (65536 colors)
- 5 24 bits (16 million colors)

Enter a number to choose the default depth.

4

Наконец, настройки нужно сохранить. Убедитесь, что вы ввели `/etc/XF86Config` в качестве места хранения настроек.

I am going to write the XF86Config file now. Make sure you don't accidentally overwrite a previously configured one.

Shall I write it to `/etc/X11/XF86Config`? y

Если настройка прервалась, вы можете запустить настройку опять, выбрав [ Yes ], когда появится следующее сообщение:

```

User Confirmation Requested
The XFree86 configuration process seems to have
failed. Would you like to try again?

```

```

[ Yes ]      No

```

Если у вас проблемы с настройкой **XFree86**, выберите [ No ], нажмите **Enter** и продолжайте процесс установки. После установки вы можете использовать `xf86cfg -textmode` или `xf86config` для получения доступа к утилитам настройки в режиме командной строки под `root`. Это дополнительный метод настройки **XFree86**, описанный в Гл. 5. Если вы решите не настраивать **XFree86** сейчас, следующим появится меню выбора пакетов.

Настройкой по умолчанию, позволяющей уничтожить процесс X сервера, является комбинация клавиш **Ctrl+Alt+Backspace**. Эта комбинация может быть применена, если что-то не так с установками сервера, она позволяет предотвратить повреждение оборудования.

Настройкой по умолчанию, позволяющей переключать видео режимы во время работы X, является комбинация клавиш **Ctrl+Alt++** or **Ctrl+Alt+-**.

После запуска **XFree86** вы можете настроить высоту, ширину и центровку изображения используя **xvidtune**.



Будут выданы предупреждения, что неверные установки могут повредить ваше оборудование. Прислушайтесь к ним. Если вы сомневаетесь в чем-то, не делайте этого. Вместо этого, используйте настройки монитора для подстройки его под X Window. Вид экрана может несколько отличаться при переключении обратно в текстовый режим, но это лучше, чем повреждение оборудования.

Прочитайте справочник `xvidtune(1)` перед тем, как производить какие-либо настройки.

После успешной настройки **XFree86** будет предложен выбор десктопа по умолчанию.

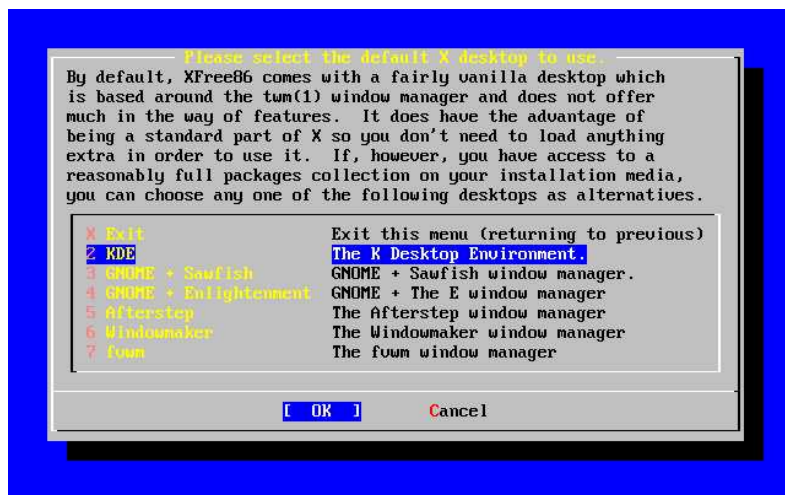
### 2.9.13. Выбор десктопа по умолчанию (Select Default X Desktop)

**Замечание:** В FreeBSD 5.3-RELEASE настройка X сервера была удалена из **sysinstall**, X сервер нужно будет настроить после установки FreeBSD. Более подробная информация, относящаяся к установке и настройке X сервера, находится в Гл. 5. Вы можете пропустить этот раздел, если только не устанавливаете версию FreeBSD до 5.3-RELEASE.

Доступны самые разные оконные менеджеры. Их функциональность варьируется от самых минимальных до полноценных десктопов с большим набором программ. Некоторые требуют минимума места на диске и небольшого количества памяти, другие, с большим набором функций, требуют гораздо больше. Лучший путь определить наиболее подходящий вам — попробовать несколько. Они доступны из коллекции портов и в виде пакетов, и могут быть добавлены после установки.

Вы можете выбрать один из популярных десктопов, он будет установлен и настроен в качестве десктопа по умолчанию. Это позволит запустить его сразу после установки.

Рисунок 2-56. Выбор десктопа по умолчанию



Используйте клавиши навигации, чтобы выбрать десктоп, и нажмите **Enter**. Будет запущена установка выбранного десктопа.

## 2.9.14. Установка пакетов (Install Packages)

Пакеты — это прекомпилированные бинарные файлы и это удобный способ установки программ.

В качестве примера показана установка одного пакета. Если потребуется, можно установить дополнительные пакеты. После установки для добавления пакетов может быть использована команда `sysinstall (/stand/sysinstall` в FreeBSD версии 5.2 и более ранних).

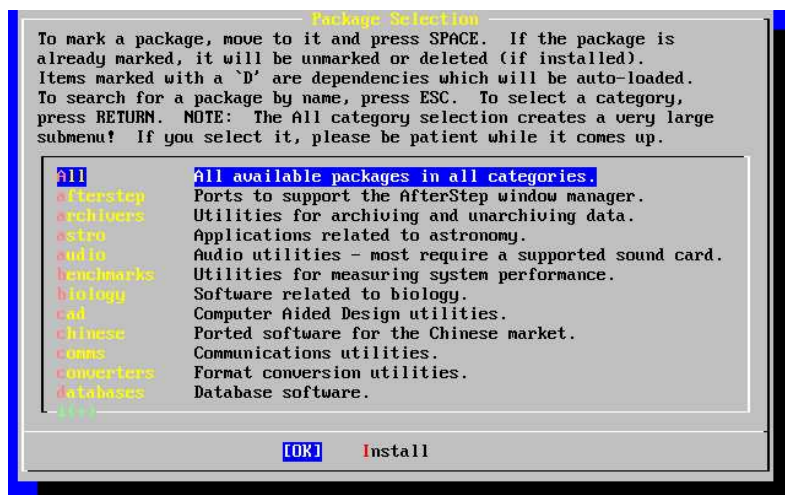
User Confirmation Requested

The FreeBSD package collection is a collection of hundreds of ready-to-run applications, from text editors to games to WEB servers and more. Would you like to browse the collection now?

[ Yes ] No

Выбор [ Yes ] и нажатие **Enter** приведет к появлению экрана выбора пакетов:

Рисунок 2-57. Выбор категории пакетов

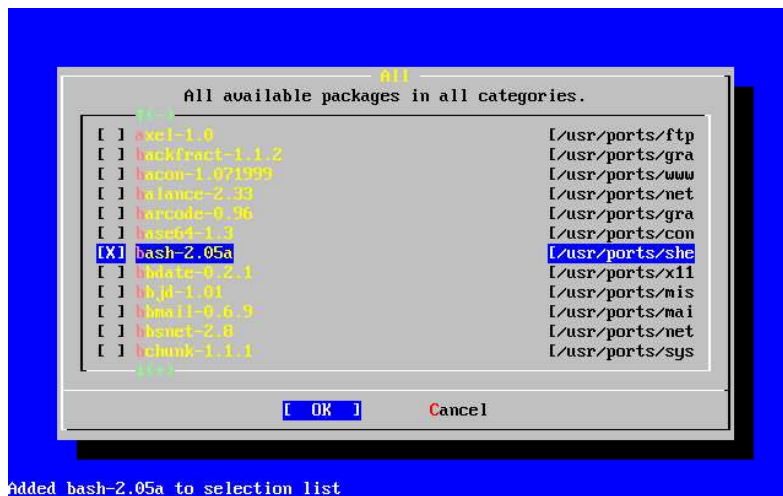


Только пакеты с текущего носителя доступны для установки в любое время.

Все доступные пакеты будут показаны если выбрать категорию **All**, можно также выбирать отдельные категории. Перейдите к выбранной категории с помощью клавиш навигации и нажмите **Enter**.

Появится меню, содержащее доступные в данной категории пакеты.

Рисунок 2-58. Выбор пакетов



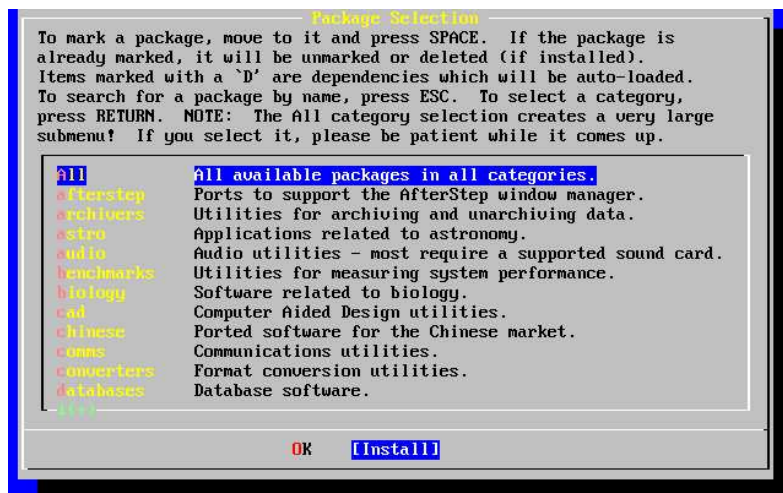
Выбрана оболочка **bash**. Выберите все необходимые пакеты, перемещаясь по меню и нажимая клавишу пробела на выбираемых пакетах. Краткое описание пакета будет появляться в нижней левой части экрана.

Нажатие **Tab** переключает между последним выбранным пакетом, [ OK ], и [ Cancel ].

После того, как будет закончена отметка пакетов для установки, нажмите **Tab** один раз для переключения на [ OK ] и нажмите **Enter** для переключения на меню выбора пакетов.

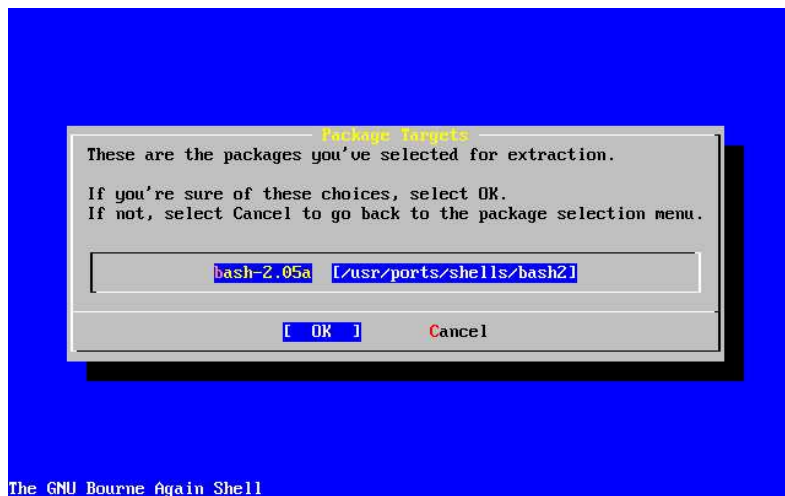
Нажимая клавиши навигации влево или вправо, можно переключаться между [ OK ] и [ Cancel ]. Этот метод может быть применен также для выбора [ OK ] и возврата к меню выбора пакетов нажатием **Enter**.

Рисунок 2-59. Установка пакетов



Используйте **Tab** и клавиши навигации для выбора [ Install ] и нажмите **Enter**. вам потребуется подтвердить установку пакетов:

Рисунок 2-60. Подтверждение установки пакетов



Выбор [ OK ] и нажатие **Enter** запустит установку пакетов. Во время установки будут выдаваться сообщения. Обратите внимание на возможные сообщения об ошибках.

После установки пакетов настройка продолжится. Если вы не выбрали ни один из пакетов и хотите вернуться к завершению настройки, выберите **Install** в любом случае.

### 2.9.15. Добавление пользователей/групп (Add Users/Groups)

В процессе установки нужно добавить хотя бы одного пользователя, чтобы использовать систему без входа под `root`. Корневой каталог обычно мал и запуск приложений под `root` быстро заполнит его. Ниже показано предупреждение:

```

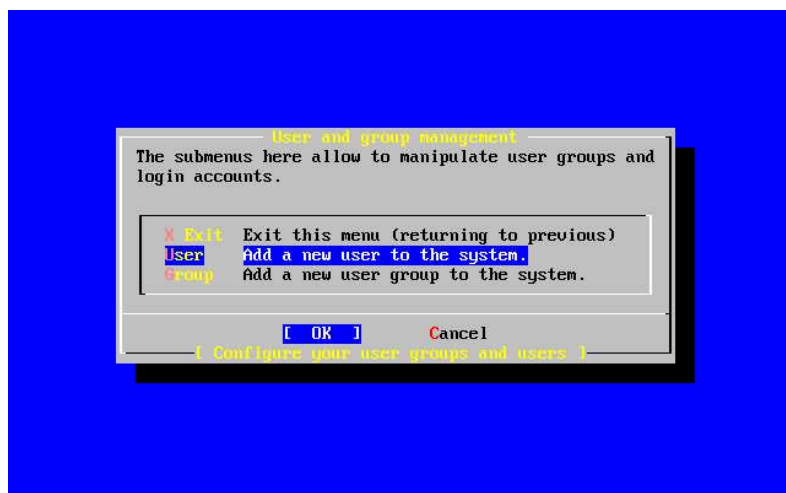
User Confirmation Requested
Would you like to add any initial user accounts to the system? Adding
at least one account for yourself at this stage is suggested since
working as the "root" user is dangerous (it is easy to do things which
adversely affect the entire system).

[ Yes ]   No

```

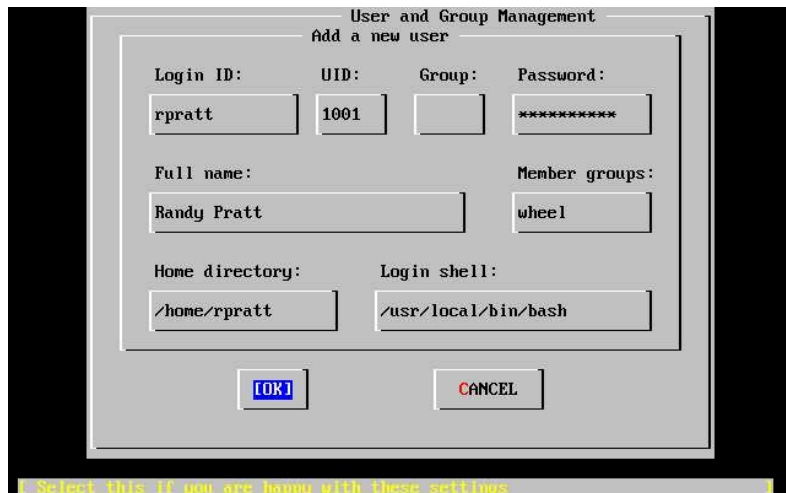
Выберите [ Yes ] и нажмите **Enter**, чтобы продолжить добавление пользователя.

Рисунок 2-61. Выбор User (пользователь)



Выберите **User** с помощью клавиш навигации и нажмите **Enter**.

Рисунок 2-62. Ввод информации о пользователе



При выборе полей с помощью **Tab** в нижней части экрана будет появляться описание, помогающее ввести необходимую информацию:

#### Логин (Login ID)

Имя нового пользователя (обязательно).

#### UID

Числовой ID (идентификатор) для этого пользователя (оставьте пустым для автоматического выбора).

**Группа (Group)**

Имя группы этого пользователя (оставьте пустым для автоматического выбора).

**Пароль (Password)**

Пароль этого пользователя (заполняйте это поле с осторожностью!).

**Полное имя**

Полное имя пользователя (комментарий).

**Член групп (Member groups)**

Группы, к которым принадлежит пользователь (т.е. имеет права доступа).

**Домашний каталог (Home directory)**

Домашний каталог пользователя (оставьте пустым для выбора по умолчанию).

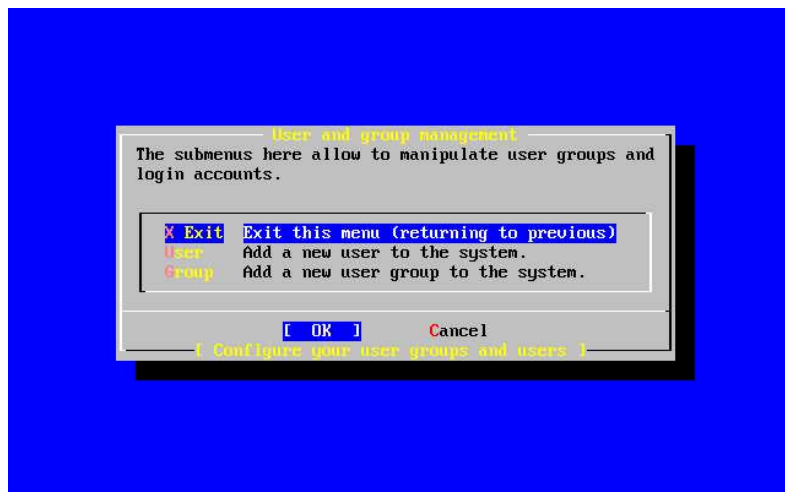
**Оболочка (Login shell)**

Оболочка пользователя, запускаемая при входе в систему (оставьте пустым для оболочки по умолчанию, например `/bin/sh`).

Оболочка была изменена с `/bin/sh` на `/usr/local/bin/bash` для использования **bash**, которая была перед этим установлена из пакета. Не пытайтесь использовать несуществующую оболочку, вы не сможете войти в систему. Наиболее часто используемая в мире BSD оболочка это C shell, которую можно обозначить как `/bin/tcsh`.

Пользователь был добавлен в группу `wheel`, чтобы иметь возможность стать суперпользователем с привилегиями `root`.

Когда все будет введено, нажмите [ OK ] и меню управления пользователями и группами (User and Group Management) появится снова:

**Рисунок 2-63. Выход из меню управления пользователями и группами**

Сейчас также можно добавить группы, если известно, для чего они потребуются. Иначе в это меню можно войти, запустив `sysinstall (/stand/sysinstall` в FreeBSD версии 5.2 и более ранних) после окончания установки.

После завершения добавления пользователей, выберите **Exit** с помощью клавиш навигации и нажмите **Enter** для продолжения установки.

## 2.9.16. Установка пароля root

```

                Message
Now you must set the system manager's password.
This is the password you'll use to log in as "root".

```

```

                [ OK ]

```

```

                [ Press enter to continue ]

```

Нажмите **Enter** для установки пароля `root`.

Необходимо два раза правильно ввести пароль. Излишне упоминать, что должна быть возможность восстановления пароля, если вы его забудете.

```

Changing local password for root.
New password :
Retype new password :

```

Установка продолжится после успешного ввода пароля.

## 2.9.17. Выход из установки (Exiting Install)

Если нужно настроить дополнительные сетевые устройства, или произвести другие настройки, вы можете сделать это сейчас или после установки с помощью `sysinstall (/stand/sysinstall` в FreeBSD версии 5.2 и более ранних).

```

                User Confirmation Requested
Visit the general configuration menu for a chance to set any last
options?

```

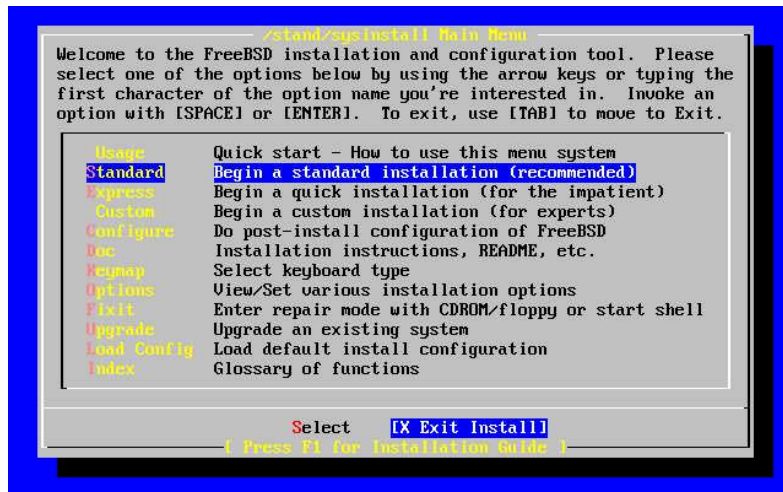
```

                Yes    [ No ]

```

Выберите **[ No ]** с помощью клавиш навигации и нажмите **Enter** для возврата к главному меню установки.

## Рисунок 2-64. Выход из установки



Выберите **[X Exit Install]** с помощью клавиш навигации и нажмите **Enter**. Будет задан вопрос о подтверждении выхода из установки:

```

User Confirmation Requested
Are you sure you wish to exit? The system will reboot (be sure to
remove any floppies from the drives).

[ Yes ]   No

```

Выберите **[ Yes ]** и удалите дискету, если загрузка была с нее. CDROM заблокирован, пока компьютер не начнет перегружаться. CDROM будет разблокирован и диск нужно будет извлечь из привода (быстро).

Система начнет перезагрузку, смотрите за сообщениями об ошибках, которые могут появиться.

## 2.9.18. Загрузочный процесс FreeBSD (FreeBSD Bootup)

### 2.9.18.1. Загрузка FreeBSD на i386

Если все пройдет нормально, вы увидите прокручивающиеся за экран сообщения, затем приглашение ко входу в систему. Вы можете просмотреть сообщения, нажав **Scroll-Lock** и используя **PgUp** и **PgDn**. Нажатие **Scroll-Lock** еще раз вернет вас к приглашению.

Все сообщение может не отобразиться (ограничение буфера), но может быть просмотрено путем ввода команды `dmesg` в командной строке.

Войдите, используя имя пользователя/пароль, добавленные во время установки (`root` в этом примере). Избегайте входа под `root`, если это не требуется.

Типичные сообщения при загрузке (информация о версии удалена):

```

Copyright (c) 1992-2002 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994
The Regents of the University of California. All rights reserved.

```



```

Timecounter "i8254" frequency 1193182 Hz
CPU: AMD-K6(tm) 3D processor (300.68-MHz 586-class CPU)
  Origin = "AuthenticAMD" Id = 0x580 Stepping = 0
  Features=0x8001bf<FPU,VME,DE,PSE,TSC,MSR,MCE,CX8,MMX>
  AMD Features=0x80000800<SYSCALL,3DNow!>
real memory = 268435456 (262144K bytes)
config> di sn0
config> di lnc0
config> di le0
config> di ie0
config> di fe0
config> di cs0
config> di bt0
config> di aic0
config> di aha0
config> di adv0
config> q
avail memory = 256311296 (250304K bytes)
Preloaded elf kernel "kernel" at 0xc0491000.
Preloaded userconfig_script "/boot/kernel.conf" at 0xc049109c.
md0: Malloc disk
Using $PIR table, 4 entries at 0xc00fde60
npx0: <math processor> on motherboard
npx0: INT 16 interface
pcib0: <Host to PCI bridge> on motherboard
pci0: <PCI bus> on pcib0
pcib1: <VIA 82C598MVP (Apollo MVP3) PCI-PCI (AGP) bridge> at device 1.0 on pci0
pci1: <PCI bus> on pcib1
pci1: <Matrox MGA G200 AGP graphics accelerator> at 0.0 irq 11
isab0: <VIA 82C586 PCI-ISA bridge> at device 7.0 on pci0
isa0: <ISA bus> on isab0
atapci0: <VIA 82C586 ATA33 controller> port 0xe000-0xe00f at device 7.1 on pci0
ata0: at 0x1f0 irq 14 on atapci0
ata1: at 0x170 irq 15 on atapci0
uhci0: <VIA 83C572 USB controller> port 0xe400-0xe41f irq 10 at device 7.2 on pci0
usb0: <VIA 83C572 USB controller> on uhci0
usb0: USB revision 1.0
uhub0: VIA UHCI root hub, class 9/0, rev 1.00/1.00, addr 1
uhub0: 2 ports with 2 removable, self powered
chip1: <VIA 82C586B ACPI interface> at device 7.3 on pci0
ed0: <NE2000 PCI Ethernet (RealTek 8029)> port 0xe800-0xe81f irq 9 at
device 10.0 on pci0
ed0: address 52:54:05:de:73:1b, type NE2000 (16 bit)
isa0: too many dependant configs (8)
isa0: unexpected small tag 14
fdc0: <NEC 72065B or clone> at port 0x3f0-0x3f5,0x3f7 irq 6 drq 2 on isa0
fdc0: FIFO enabled, 8 bytes threshold
fd0: <1440-KB 3.5" drive> on fdc0 drive 0
atkbd0: <keyboard controller (i8042)> at port 0x60-0x64 on isa0
atkbd0: <AT Keyboard> flags 0x1 irq 1 on atkbd0
kbd0 at atkbd0
psm0: <PS/2 Mouse> irq 12 on atkbd0

```

```

psm0: model Generic PS/2 mouse, device ID 0
vga0: <Generic ISA VGA> at port 0x3c0-0x3df iomem 0xa0000-0xbffff on isa0
sc0: <System console> at flags 0x1 on isa0
sc0: VGA <16 virtual consoles, flags=0x300>
sio0 at port 0x3f8-0x3ff irq 4 flags 0x10 on isa0
sio0: type 16550A
siol at port 0x2f8-0x2ff irq 3 on isa0
siol: type 16550A
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
ppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/15 bytes threshold
ppbus0: IEEE1284 device found /NIBBLE
Probing for PnP devices on ppbus0:
plip0: <PLIP network interface> on ppbus0
lpt0: <Printer> on ppbus0
lpt0: Interrupt-driven port
ppi0: <Parallel I/O> on ppbus0
ad0: 8063MB <IBM-DHEA-38451> [16383/16/63] at ata0-master using UDMA33
ad2: 8063MB <IBM-DHEA-38451> [16383/16/63] at atal-master using UDMA33
acd0: CDROM <DELTA OTC-H101/ST3 F/W by OIPD> at ata0-slave using PIO4
Mounting root from ufs:/dev/ad0sla
swapon: adding /dev/ad0slb as swap device
Automatic boot in progress...
/dev/ad0sla: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0sla: clean, 48752 free (552 frags, 6025 blocks, 0.9% fragmentation)
/dev/ad0slf: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0slf: clean, 128997 free (21 frags, 16122 blocks, 0.0% fragmentation)
/dev/ad0slg: FILESYSTEM CLEAN; SKIPPING CHECKS
/dev/ad0slg: clean, 3036299 free (43175 frags, 374073 blocks, 1.3% fragmentation)
/dev/ad0sle: filesystem CLEAN; SKIPPING CHECKS
/dev/ad0sle: clean, 128193 free (17 frags, 16022 blocks, 0.0% fragmentation)
Doing initial network setup: hostname.
ed0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet 192.168.0.1 netmask 0xfffff00 broadcast 192.168.0.255
    inet6 fe80::5054::5ff::fede:731b%ed0 prefixlen 64 tentative scopeid 0x1
    ether 52:54:05:de:73:1b
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x8
    inet6 ::1 prefixlen 128
    inet 127.0.0.1 netmask 0xff000000
Additional routing options: IP gateway=YES TCP keepalive=YES
routing daemons:.
additional daemons: syslogd.
Doing additional network setup:.
Starting final network daemons: creating ssh RSA host key
Generating public/private rsal key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
cd:76:89:16:69:0e:d0:6e:f8:66:d0:07:26:3c:7e:2d root@k6-2.example.com
    creating ssh DSA host key
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.

```

```

Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
f9:a1:a9:47:c4:ad:f9:8d:52:b8:b8:ff:8c:ad:2d:e6 root@k6-2.example.com.
setting ELF ldconfig path: /usr/lib /usr/lib/compat /usr/X11R6/lib
/usr/local/lib
a.out ldconfig path: /usr/lib/aout /usr/lib/compat/aout /usr/X11R6/lib/aout
starting standard daemons: inetd cron sshd usbd sendmail.
Initial rc.i386 initialization:.
rc.i386 configuring syscons: blank_time screensaver moused.
Additional ABI support: linux.
Local package initialization:.
Additional TCP options:.

```

```
FreeBSD/i386 (k6-2.example.com) (ttyv0)
```

```
login: rpratt
Password:
```

Генерирование ключей RSA и DSA может занять некоторое время на медленных компьютерах. Это может случиться только при первой загрузке свежеставленной системы. Последующие загрузки будут быстрее.

Если X сервер был настроен и был выбран десктоп по умолчанию, он может быть запущен из командной строки командой `startx`.

### 2.9.18.2. Загрузка FreeBSD на Alpha

Когда процедура установки будет завершена, вы сможете запустить FreeBSD в приглашении SRM примерно такой командой:

```
>>>BOOT DKC0
```

Это команда дает указание на загрузку с заданного диска. Чтобы FreeBSD в будущем загружалась автоматически, используйте эти команды:

```

>>> SET BOOT_OSFLAGS A
>>> SET BOOT_FILE "
>>> SET BOOTDEF_DEV DKC0
>>> SET AUTO_ACTION BOOT

```

Загрузочные сообщения будут похожи (но не идентичны) сообщениям, выводимым FreeBSD при загрузке на i386.

### 2.9.19. Завершение работы FreeBSD (FreeBSD Shutdown)

Важно правильно завершать работу операционной системы. Нельзя просто выключать питание. Сначала нужно стать суперпользователем, введя в командной строке `su` и пароль `root`. Это работает только если пользователь является членом группы `wheel`. Или зайдите под `root` и используйте команду `shutdown -h now`.

```
The operating system has halted.
Please press any key to reboot.
```

Можно безопасно выключать питание после того, как будет выполнена команда `shutdown` и появится сообщение "Please press any key to reboot". Если вместо выключения питания будет нажата любая клавиша, система перезагрузится.

Вы также можете использовать комбинацию клавиш **Ctrl+Alt+Del** для перезагрузки системы, хотя это не рекомендуется в обычной ситуации.

## 2.10. Поддерживаемое оборудование

В настоящее время FreeBSD работает на разнообразных PC, с шинами ISA, VLB, EISA, и PCI с процессорами "x86" от Intel, AMD, Cyrix, или NexGen, так же как и с множеством компьютеров на процессорах Compaq Alpha. Поддерживаются также IDE или ESDI диски, различные SCSI контроллеры, карты PCMCIA, устройства USB, сетевые карты и карты последовательных портов. FreeBSD также поддерживает шину MCA от IBM.

Список поддерживаемого оборудования поставляется с каждым релизом FreeBSD в информации о поддерживаемом FreeBSD оборудовании. Этот документ обычно может быть найден в файле `HARDWARE.TXT` в корневом каталоге CDROM, дистрибутива на FTP или в меню документации **sysinstall**. В нем находится список аппаратных устройств, которые поддерживаются этим релизом FreeBSD для данной архитектуры. Копии списка поддерживаемого оборудования для различных релизов и архитектур также можно найти на странице Информации о релизах (<http://www.FreeBSD.org/ru/releases/index.html>) сайта FreeBSD.

## 2.11. Решение проблем

Следующий раздел описывает основные проблемы при установке, о которых обычно сообщают пользователи. Здесь также несколько вопросов и ответов от тех, кто хочет настроить двойную загрузку FreeBSD и MS-DOS.

### 2.11.1. Что делать, если что-то идет не так

По причине различных ограничений архитектуры PC, невозможно протестировать устройства достоверно на 100%, тем не менее, есть несколько вещей, которые вы можете сделать, если тестирование завершится неудачно.

Сверьтесь со списком поддерживаемого оборудованию для вашей версии FreeBSD, чтобы убедиться, что ваше оборудование поддерживается.

Если оборудование поддерживается, но все же происходят зависания или вы встретились с другими проблемами, перегрузите компьютер, и когда появится опция визуальной настройки ядра, выберите ее. Это позволит вам просмотреть список оборудования и дать системе информацию о настройке. Ядро на загрузочном диске настроено в предположении, что у большинства аппаратных устройств IRQ, адреса ввода-вывода и каналы DMA установлены производителем к значениям по умолчанию. Если оборудование было перенастроено, вам потребуется использовать редактор настроек, чтобы сообщить FreeBSD, где что искать.

Возможно также, что тестирование устройства, которого нет, приведет позже к ошибке тестирования существующего устройства. В этом случае тестирование конфликтующего драйвера (драйверов) должно быть запрещено.

**Замечание:** Некоторые проблемы установки можно обойти или уменьшить, обновив встроенное программное обеспечение различных компонент оборудования, особенно материнской платы. Встроенное программное обеспечение также называют BIOS и у большинства производителей компьютеров или материнских плат есть вебсайты, где можно найти обновления и информацию об обновлениях.

Большинство производителей настоятельно советуют не обновлять BIOS пока не возникнет серьезной необходимости, например появление критически важных обновлений. Процесс обновления может пойти неверно, что приведет к неустранимому повреждению микросхемы BIOS.

**Внимание:** Не выключайте драйвера, которые понадобятся вам во время установки, такие как экран (`sc0`). Если после выхода из редактора конфигурации установка загадочно зависает или прерывается, возможно вы удалили или изменили что-то, что не надо было трогать. Перегрузитесь и попробуйте еще раз.

В режиме настройки вы можете:

- Просматривать список устройств, включенных в ядро.
- Отключать драйвера устройств для оборудования, которого нет в системе.
- Менять IRQ, DRQ, адреса портов, используемые драйвером устройства.

После настройки ядра под оборудование нажмите Q для загрузки с новыми настройками. После завершения установки все изменения, внесенные в режиме настройки, будут действовать постоянно, так что вам не придется делать настройку при каждой загрузке. Все же очень неплохо, если вы со временем соберете новое ядро.

### 2.11.2. Работа с существующими разделами MS-DOS®

Многие пользователи устанавливают FreeBSD на PC с установленной операционной системой Microsoft. Для этих целей в FreeBSD есть утилита, называемая **FIPS**. Ее можно найти в каталоге `tools` установочного CD-ROM или загрузить с одного из множества зеркал FreeBSD.

Утилита **FIPS** позволит вам разделить существующий раздел MS-DOS на две части, сохранив имеющиеся данные, и сделать установку на второй свободный раздел. Вам потребуется сначала дефрагментировать раздел MS-DOS с помощью Windows утилиты **Disk Defragmenter** (запустите Explorer, кликните правой клавишей мыши на жестком диске и выберите дефрагментацию диска), или используйте Norton Disk Tools. Затем запустите утилиту **FIPS**. Она запросит необходимую информацию, следуйте инструкциям на экране. Затем можете перезагрузиться и установить FreeBSD на новый свободный слайс. Смотрите меню **Distributions** для определения размера необходимого для выбранного типа установки свободного пространства.

Есть также очень полезная программа **PartitionMagic** от PowerQuest (<http://www.powerquest.com> (<http://www.powerquest.com/>)). Эта программа гораздо более функциональна, чем **FIPS**, и очень рекомендуется, если вы планируете часто добавлять/убирать операционные системы. Она стоит денег, и если вы планируете установить FreeBSD и оставить ее, вам возможно отлично подойдет **FIPS**.

### 2.11.3. Использование файловых систем MS-DOS и Windows®

В настоящее время, FreeBSD не поддерживает файловые системы, сжатые с помощью программы **Double Space™**. Поэтому файловая система должна быть разжата, чтобы FreeBSD смогла получить доступ к данным. Это может быть сделано с помощью приложения **Compression Agent**, находящегося в меню Пуск > Программы > Системные.

FreeBSD поддерживает файловые системы MS-DOS. Для этого потребуется использовать команду `mount_msdos(8)` (в FreeBSD 5.X, это `mount_msdosfs(8)`) с необходимыми параметрами. Наиболее распространенный способ использования:

```
# mount_msdos /dev/ad0s1 /mnt
```

В этом примере файловая система MS-DOS расположена на первом разделе первого жесткого диска. Ваша конфигурация может быть другой, проверьте вывод команд `dmesg` и `mount`. Они должны предоставить достаточно информации для определения названий разделов.

**Замечание:** Расширенные разделы MS-DOS обычно располагаются после разделов FreeBSD. Другими словами, номер слайса может быть больше, чем используемый FreeBSD. Например, первый раздел MS-DOS может быть `/dev/ad0s1`, раздел FreeBSD `/dev/ad0s2`, а расширенный раздел MS-DOS `/dev/ad0s3`. Некоторых это иногда запутывает на первых порах.

Разделы NTFS также можно смонтировать похожим способом, используя команду `mount_ntfs(8)`.

### 2.11.4. Вопросы и ответы пользователей Alpha

Этот раздел отвечает на некоторые обычно задаваемые вопросы об установке FreeBSD на Alpha.

**1. Могу я загрузиться с ARC или с консоли Alpha BIOS?**

Нет. FreeBSD, как и Compaq Tru64 и VMS, может загружаться только с консоли SRM.

**2. Помогите, у меня нет места! Должен ли я сначала все удалить?**

К сожалению, да.

**3. Могу ли я смонтировать файловые системы Compaq Tru64 или VMS?**

В настоящее время нет.

## 2.12. Расширенное руководство по установке

*Предоставлено Valentino Vaschetto.*

Этот раздел описывает особые случаи установки FreeBSD.

### 2.12.1. Установка FreeBSD на систему без монитора или клавиатуры

Этот тип установки называется "установка без монитора", потому что на компьютере, на который вы пробуете установить FreeBSD или не подсоединен монитор, или даже нет VGA выхода. Вы спросите, как это возможно? Используя последовательную консоль. Последовательная консоль в своей основе имеет другой компьютер, который служит дисплеем и клавиатурой для системы. Чтобы сделать это, создайте инсталляционные дискеты, как описано в Разд. 2.2.7.

Чтобы модифицировать дискеты для загрузки с последовательной консоли, следуйте этой инструкции:

#### 1. Включение загрузки с последовательной консоли

Если вы произведете загрузку с только что полученных дискет, FreeBSD загрузится в нормальный режим установки. Мы хотим, чтобы FreeBSD загрузилась на последовательную консоль для нашей установки. Чтобы сделать это, смонтируйте `kern.flp` на вашу систему FreeBSD используя команду `mount(8)`.

```
# mount /dev/fd0 /mnt
```

Теперь, когда дискета смонтирована, нужно перейти в каталог `/mnt`:

```
# cd /mnt
```

Сейчас вам нужно настроить дискету для загрузки в последовательную консоль. Создайте файл `boot.config`, содержащий `/boot/loader -h`. Это нужно для того, чтобы передать загрузчику флаг загрузки с последовательной консоли.

```
# echo "/boot/loader -h" > boot.config
```

Теперь, когда ваша дискета правильно настроена, размонтируйте ее с помощью команды `umount(8)`:

```
# cd /
# umount /mnt
```

Теперь удалите дискету из дисковода.

#### 2. Подсоединение нуль-модемного кабеля

Теперь нужно соединить два компьютера нуль-модемным кабелем. Просто подсоедините этот кабель к последовательным портам двух компьютеров. *Обычный последовательный кабель не будет работать*, вам потребуется нуль-модемный кабель, поскольку в нем некоторые соединения перевернуты.

#### 3. Загрузка для установки

Настало время начать установку. Вставьте диск `kern.flp` в дисковод компьютера, на который вы будете производить установку без монитора и включите его.

#### 4. Подключение к компьютеру без монитора

Теперь вы можете подключиться к этому компьютеру с помощью `cu(1)`:

```
# cu -l /dev/cuaa0
```

Это все! Теперь вы можете управлять компьютером без монитора через сессию `cu`. Будет запрошен диск `mfsroot.flp`, затем появится выбор типа терминала. Выберите цветную консоль FreeBSD и проводите установку!

## 2.13. Подготовка собственного источника установки

**Замечание:** Чтобы не повторяться, "диск FreeBSD" в данном случае означает FreeBSD CDROM или DVD, который вы купили или подготовили самостоятельно.

Может быть несколько ситуаций, в которых вам потребуется подготовить собственный диск и/или источник для установки FreeBSD. Это может быть физический носитель, такой как лента, или источник, с которого **sysinstall** сможет взять файлы, такой как локальный сервер FTP, или раздел MS-DOS.

Например:

- У вас есть много компьютеров, подсоединенных к локальной сети, и один диск с FreeBSD. Вы хотите создать локальный сервер FTP, используя содержимое диска FreeBSD, а затем использовать с ваших компьютеров этот локальный сервер FTP вместо подсоединения к интернет.
- У вас есть диск с FreeBSD, и FreeBSD не распознает ваш CD/DVD привод, а MS-DOS/Windows распознает. Вы хотите скопировать файлы установки FreeBSD на раздел DOS этого же компьютера и установить FreeBSD, используя эти файлы.
- На компьютере, на который вы хотите провести установку, нет ни привода CD/DVD, ни сетевой карты, но вы можете подсоединиться в "Laplink-стиле" последовательным или параллельным кабелем к другому компьютеру, на котором они есть.
- Вы хотите создать ленту, которую можно использовать для установки FreeBSD.

### 2.13.1. Создание установочного CDROM

В составе каждого релиза проект FreeBSD предоставляет доступ к двум образам CDROM ("ISO images"). Эти образы могут быть записаны ("прожжены") на CD если у вас есть CD-рекордер, а затем использованы для установки FreeBSD. Если у вас есть CD-рекордер и дешевый канал интернет, это простейший путь установить FreeBSD.

#### 1. Скачайте соответствующие ISO образы

ISO образы для каждого релиза могут быть загружены с

`ftp://ftp.FreeBSD.org/pub/FreeBSD/ISO-IMAGES-архитектура/версия` или с ближайшего зеркала. Замените *архитектура* и *версия* в соответствии архитектурой (Alpha или i386) и номером релиза соответственно.

Этот каталог обычно содержит следующие образы:

**Таблица 2-5. Названия FreeBSD ISO образов и их значения**

Имя файла	Содержимое
<i>версия</i> -mini.iso	Все, что вам нужно для установки FreeBSD.
<i>версия</i> -disc1.iso	Все, что вам нужно для установки FreeBSD, и столько дополнительных пакетов сторонних производителей, сколько уместилось на диске.



**Имя файла***версия* -disc2.iso**Содержимое**

"live fi lesystem", используемая в сочетании со средством "Repair" программы **sysinstall**. Копия дерева CVS FreeBSD. Столько дополнительных пакетов сторонних производителей, сколько уместилось на диске.

Вам *необходимо* загрузить или образ mini ISO, или образ первого диска. Не загружайте их оба, так как образ первого диска содержит все, что есть на образе mini ISO.

Используйте mini ISO, если доступ в интернет затруднен. Он позволит вам установить FreeBSD, и вы сможете установить впоследствии программы сторонних производителей используя систему портов/пакетов (смотрите Гл. 4) если необходимо.

Используйте образ первого диска, если хотите выбрать программы сторонних производителей с этого диска.

Дополнительные образы дисков полезны, но не необходимы, особенно если у вас есть высокоскоростной доступ к интернет.

## 2. Запись CD

Затем вам нужно записать образы CD на диски. Если вы делаете это из другой системы FreeBSD, обратитесь к Разд. 16.6 за более подробной информацией (в частности, Разд. 16.6.3 и Разд. 16.6.4).

Если вы делаете это в другой системе, потребуется использовать те утилиты для управления CD-рекордером, которые есть в этой системе. Образы дисков предоставляются в стандартном формате ISO, который поддерживается многими программами.

### 2.13.2. Создание локального сервера FTP с FreeBSD

Диск FreeBSD сформирован так же, как и сервер FTP. Это сильно упрощает создание локального сервера FTP, который может быть использован другими компьютерами вашей сети для установки FreeBSD.

1. Убедитесь, что на компьютере FreeBSD, на котором будет установлен сервер FTP, CDROM находится в приводе и смонтируйте /cdrom.  
# **mount /cdrom**
2. Создайте учетную запись для anonymous FTP в /etc/passwd. Сделайте это, отредактировав /etc/passwd с помощью vipw(8) и добавив эту строку:  
ftp:\*:99:99::0:0:FTP:/cdrom:/nonexistent
3. Убедитесь что сервис FTP включен в /etc/inetd.conf.

Всякий, кто может подсоединиться по сети к вашему компьютеру, может выбрать тип носителя FTP и набрать **ftp://ваш компьютер** после выбора "Other" в меню серверов FTP во время установки.

**Внимание:** Этот подход хорош для компьютера в вашей локальной сети, защищенного с помощью файрволла. Предоставление сервиса FTP другим компьютерам через интернет (а не через локальную сеть)

привлекает к вашему компьютеру внимание кракеров и других лиц, чье внимание нежелательно. Мы настоятельно рекомендуем придерживаться правильной политики безопасности, если вы делаете это.

### 2.13.3. Создание дискет для установки

Если вам требуется выполнить установку с дискет (чего мы *не* рекомендуем делать) или по причине не поддерживаемого оборудования, или просто потому, что вы не любите простых путей, потребуется сначала подготовить несколько дискет для установки.

Как минимум, вам потребуется столько 1.44 MB или 1.2 MB дискет, сколько нужно, чтобы вместить все файлы из каталога `bin` (binary distribution). Если вы подготавливаете эти дискеты из DOS, они *должны* быть отформатированы с помощью команды `MS-DOS FORMAT`. Если вы используете Windows, используйте Explorer для форматирования дисков (кликните правой кнопкой мыши на диске A: и выберите 'Format').

*Не* доверяйте заводскому форматированию дискет. Отформатируйте их еще раз самостоятельно, просто для уверенности. Множество проблем, о которых сообщали наши пользователи, были результатом использования неправильно отформатированных дисков, поэтому мы сейчас обращаем на это внимание.

Если вы создаете образы на другом компьютере FreeBSD, форматирование все еще не лишне, хотя вам не потребуется создавать файловую систему DOS на каждой дискете. Вы можете использовать `disklabel` и `newfs` для создания на них файловых систем UFS, в следующей последовательности (для 3.5" 1.44 MB дискет):

```
# fdformat -f 1440 fd0.1440
# disklabel -w -r fd0.1440 floppy3
# newfs -t 2 -u 18 -l 1 -i 65536 /dev/fd0
```

**Замечание:** Используйте `fd0.1200` и `floppy5` для 5.25" 1.2 MB дискет.

Затем вы можете смонтировать их и писать на них как на любую другую файловую систему.

После форматирования дискет вам потребуется скопировать на них файлы. Файлы дистрибутива разделены на части, размер которых позволяет легко разместить пять частей на обычной 1.44 MB дискете. Заполните все дискеты файлами, помещая столько файлов на одну дискету, сколько уместится, пока у вас не будет всех необходимых компонент дистрибутива. Каждый компонент должен располагаться в подкаталоге на дискете, например `a:\bin\bin.aa`, `a:\bin\bin.ab`, и так далее.

Как только в процессе установки появится экран **Media**, выберите **Floppy** и установка продолжится.

### 2.13.4. Установка с раздела MS-DOS

Чтобы подготовиться к установке с раздела MS-DOS, скопируйте файлы с дистрибутива в каталог `freebsd` корневого каталога раздела. Например, `c:\freebsd`. Структура каталога CDROM или сервера FTP должна быть в точности воспроизведена в этом каталоге, мы предлагаем использовать

команду DOS `xcopy` если вы копируете дистрибутив с CD. Например, чтобы подготовить минимальную установку FreeBSD:

```
C:\> md c:\freebsd
C:\> xcopy e:\bin c:\freebsd\bin\ /s
C:\> xcopy e:\manpages c:\freebsd\manpages\ /s
```

Предполагается, что на C: есть свободное место, а на E: смонтирован CDROM.

Если у вас нет CDROM, можно загрузить дистрибутив с [ftp.FreeBSD.org](http://ftp.FreeBSD.org) ([ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/5.3-RELEASE/](http://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/5.3-RELEASE/)). Каждый компонент находится в своем собственном каталоге, например компонент *base* можно найти в каталоге *5.3/base/* ([ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/5.3-RELEASE/base/](http://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/5.3-RELEASE/base/)).

**Замечание:** В 4.X и более поздних релизах FreeBSD компонент "base" называется "bin". Поправьте команды и ссылки соответственно, если вы используете эти версии.

Все компоненты, которые вы будете устанавливать с раздела MS-DOS (и для которых у вас есть место), нужно расположить в `c:\freebsd` — для минимальной установки нужна только компонента BIN.

### 2.13.5. Создание ленты для установки

Установка с ленты это возможно самый простой метод, уступающий только установке с FTP или CDROM. Программа установки может найти на ленте файлы, которые записаны на нее с помощью `tar`. После получения всех файлов дистрибутива, которые вам нужны, запишите их на ленту с помощью `tar`:

```
# cd /freebsd/distdir
# tar cvf /dev/rwt0 dist1 ... dist2
```

Когда вы приступите к установке, нужно убедиться в наличии достаточного места во временном каталоге (который будет предложено выбрать), чтобы поместилось *все* содержимое записанной ленты. Поскольку лента — устройство не-произвольного доступа, этот метод установки требует некоторого места во временном хранилище.

**Замечание:** При начале установки лента должна быть в приводе до загрузки с дискеты. В ином случае она может быть не найдена при тестировании устройств.

### 2.13.6. Перед установкой по сети

Есть три возможных типа установки по сети. Через последовательный порт (SLIP или PPP), параллельный порт (PLIP (laplink кабель)), или Ethernet (стандартный Ethernet контроллер (включая некоторые PCMCIA)).

Поддержка SLIP немного примитивна, и ограничена в основном постоянными соединениями, такими как последовательный кабель между ноутбуком и другим компьютером. Соединение должно быть постоянным, так как устанавливаемый SLIP в настоящее время не имеет возможности дозвона; эта

возможность предоставляется программой PPP, которая должна быть использована вместо SLIP когда это возможно.

Если вы используете модем, PPP это конечно почти единственный выбор. Убедитесь что у вас есть информация о провайдере, так как она понадобится на довольно ранней стадии процесса установки.

Если вы используете PAP или CHAP для соединения с провайдером (другими словами, вы можете соединяться с провайдером из Windows без использования скрипта), все что вам понадобится это ввести `dial` в приглашении **ppp**. Иначе вам нужно знать, как дозвониться до провайдера с помощью "АТ команд", специфичных для вашего модема, так как программа дозвона PPP предоставляет только очень простой эмулятор терминала. Обратитесь к руководству `user-ppp` и FAQ ([http://www.FreeBSD.org/doc/ru\\_RU.KOI8-R/books/faq/ppp.html](http://www.FreeBSD.org/doc/ru_RU.KOI8-R/books/faq/ppp.html)) для получения более подробных сведений. Если у вас проблемы, логи могут быть выведены на экран с помощью команды `set log local ...`

Если доступно постоянное соединение с другой машиной FreeBSD (2.0-R или выше), вы можете также подумать об установке через параллельный порт "laplink" кабелем. Скорость соединения через параллельный порт значительно выше, чем обычно возможно через последовательную линию (до 50 кбайт/с), это приведет к ускорению установки.

Наконец, для самой быстрой установки по сети хорошим выбором будет Ethernet адаптер! FreeBSD поддерживает большинство PC Ethernet карт; список поддерживаемых карт (и требуемых для них установок) предоставляется в информации об оборудовании для каждого релиза FreeBSD. Если вы используете одну из поддерживаемых PCMCIA Ethernet карт, убедитесь также, что она подключена *перед* тем, как ноутбук будет включен! К сожалению, FreeBSD в настоящее время не поддерживает горячее подключение PCMCIA карт во время установки.

Вам также потребуется знать IP адрес в сети, сетевую маску для класса подсети, и имя компьютера. Если вы делаете установку через соединение PPP и у вас нет статического IP, не бойтесь, IP адрес будет динамически выделен вам провайдером. Системный администратор может подсказать вам, какие значения использовать для данной установки по сети. Если вы будете обращаться к другим хостам по имени, а не по IP адресу, понадобится также ДНС сервер и, возможно, адрес шлюза (если вы используете PPP, это IP адрес вашего провайдера), чтобы связаться с ним. Если вы хотите производить установку с FTP через HTTP прокси, потребуется также адрес прокси. Если вы не знаете ответов на все или большинство этих вопросов, свяжитесь с системным администратором или ISP *перед* тем, как начать этот тип установки.

### 2.13.6.1. Перед установкой через NFS

Установка через NFS очень проста. Просто скопируйте компоненты FreeBSD, которые вам нужны, на NFS сервер, а затем укажите на него при выборе источника установки NFS.

Если этот сервер поддерживает только "привилегированные порты" (это как правило так для рабочих станций Sun), потребуется установить параметр `NFS Secure` в меню **Options** перед тем, как начать установку.

Если у вас Ethernet карта плохого качества с низкой скоростью передачи данных, вы возможно захотите переключить флаг `NFS Slow` в **Options**.

Чтобы установка по NFS работала, сервер должен поддерживать монтирование подкаталогов, например, если установочный каталог дистрибутива FreeBSD 5.3 находится на:

`ziggy:/usr/archive/stuff/FreeBSD`, `ziggy` должен позволять непосредственное монтирование `/usr/archive/stuff/FreeBSD`, а не только `/usr` или `/usr/archive/stuff`.

В файле FreeBSD `/etc/exports` это управляется параметром `-alldirs`. Другие NFS серверы могут иметь другие соглашения. Если вы получаете сообщения сервера `permission denied`, это может означать, что соответствующая опция не включена.

# Глава 3. Основы UNIX

*Переписал Chris Shumway. Перевод на русский язык: Алексей Зелькин, Алексей Докучаев, Денис Пеплин.*

## 3.1. Краткий обзор

В этой главе мы попытаемся раскрыть основные принципы и команды операционной системы FreeBSD. Большая часть нижеизложенного материала в более или менее равной степени применима к любой UNIX-подобной операционной системе. Если вы уверены, что не найдете здесь ничего нового для себя, можете смело пропустить эту главу. Если же вы новичок, мы настоятельно рекомендуем внимательно прочесть эту главу.

Прочитав эту главу, вы узнаете:

- Как использовать "виртуальные консоли" FreeBSD.
- Как работают права доступа на файлы в UNIX.
- Иерархия каталогов FreeBSD.
- Организация дисков FreeBSD.
- Как монтировать и размонтировать файловые системы.
- Что такое процессы, демоны и сигналы.
- Что такое командная оболочка (или интерпретатор команд) и как настроить личное рабочее окружение.
- Как пользоваться стандартными текстовыми редакторами.
- Что такое устройства и файлы устройств.
- Какие бинарные форматы используются в FreeBSD.
- Как пользоваться справочным руководством для получения дополнительной информации.

## 3.2. Виртуальные консоли и терминалы

FreeBSD можно использовать разными способами. Один из них — набор команд в текстовом терминале. Используйте этот способ, и вся гибкость и мощь систем UNIX будет в ваших руках. Этот раздел рассказывает о "терминалах", "консолях" и их использовании в FreeBSD.

### 3.2.1. Консоль

Если во время установки вы не настроили FreeBSD для автоматического запуска графической среды при загрузке, система запросит ввод логина сразу после завершения стартовых скриптов. Вы увидите примерно следующее:

```
Additional ABI support:.  
Local package initialization:.
```

```
Additional TCP options:.
```

```
Fri Sep 20 13:01:06 EEST 2002
```

```
FreeBSD/i386 (pc3.example.org) (ttyv0)
```

```
login:
```

В вашей системе сообщение может быть другим, но очень похожим на это. В данный момент нас интересуют последние две строки. Вторая с конца строка:

```
FreeBSD/i386 (pc3.example.org) (ttyv0)
```

В этой строке содержится немного информации о только что загруженной системе. Это консоль "FreeBSD", работающей на Intel или совместимом процессоре x86 архитектуры <sup>1</sup>. Имя этого компьютера (у каждого компьютера UNIX есть имя) `pc3.example.org`, и в данный момент вы видите системную консоль — терминал `ttyv0`.

Наконец, последняя строка всегда:

```
login:
```

Здесь вам предлагается ввести "имя пользователя", чтобы войти в FreeBSD. Следующий раздел описывает способ, которым вы можете сделать это.

### 3.2.2. Вход в FreeBSD

FreeBSD это многопользовательская, многопроцессорная система. Это формальное описание системы, которая может быть использована множеством разных людей, одновременно запускающих большое количество программ на одном компьютере.

Любой многопользовательской системе нужен способ отличать каждого "пользователя" от остальных. В FreeBSD (и всех UNIX подобных операционных системах), эта задача решается путем "входа" пользователя в систему перед запуском каких-либо программ. У каждого пользователя есть уникальное имя ("имя пользователя") и персональный, секретный ключ ("пароль"). Перед тем, как разрешить пользователю выполнять какие-либо программы, FreeBSD запрашивает их оба.

Сразу после загрузки FreeBSD и завершения работы стартовых скриптов<sup>2</sup>, система предложит вам ввести имя пользователя:

```
login:
```

В этом примере, предположим что ваше имя пользователя `john`. Введите `john` в этом приглашении и нажмите **Enter**. Далее должно появиться приглашение ввести "пароль":

```
login: john
Password:
```

Введите соответствующий имени `john` пароль и нажмите **Enter**. Пароль *не виден*! Не беспокойтесь об этом. Это сделано по соображениям безопасности.

Если вы ввели пароль правильно, то сразу же войдете в FreeBSD и можете начать выполнять команды.

Вы увидите сообщение дня (MOTD, или message of the day) за которым последует командная строка (с символом #, \$, или %). Это означает, что вы успешно вошли в FreeBSD.

### 3.2.3. Множественные консоли

Запуск команд UNIX из консоли — это конечно хорошо, но FreeBSD может выполнять множество программ одновременно, поэтому наличие одной консоли может быть недостатком. В таком случае очень полезны "виртуальные консоли".

FreeBSD может быть настроена для работы с несколькими виртуальными консолями. Вы можете переключаться с одной виртуальной консоли на другую, нажимая соответствующие сочетания клавиш на клавиатуре. У каждой консоли есть свой канал вывода и FreeBSD заботится о том, чтобы правильно перенаправить ввод с клавиатуры и вывод на монитор, как только вы переключитесь с одной консоли на другую.

Для переключения между консолями зарезервированы специальные комбинации клавиш<sup>3</sup>. Вы можете использовать сочетания **Alt-F1**, **Alt-F2**, до **Alt-F8** чтобы переключаться между различными виртуальными консолями в FreeBSD.

При переключении от одной консоли к другой, FreeBSD заботится о сохранении и восстановлении вывода на экран. Результатом является "иллюзия" наличия множества "виртуальных" экранов и клавиатур, которые могут быть использованы для ввода и запуска команд FreeBSD. Программы, которые вы запускаете на одной виртуальной консоли, не прекращают выполнение, когда консоль становится невидимой. Они продолжают выполняться, когда вы переключаетесь на другую виртуальную консоль.

### 3.2.4. Файл /etc/ttys

В конфигурации по умолчанию FreeBSD запускает восемь виртуальных консолей. Тем не менее, это не ограничение оборудования, и вы можете легко настроить систему для загрузки большего или меньшего числа виртуальных консолей. Число и параметры виртуальных консолей задаются в файле /etc/ttys.

Вы можете использовать этот файл для настройки виртуальных консолей FreeBSD. Любая не закомментированная строка в этом файле (строка, не начинающаяся с символа #), содержит настройки для одного терминала или виртуальной консоли. Версия этого файла по умолчанию, поставляемая с FreeBSD, содержит настройки для девяти виртуальных консолей и включает восемь терминалов. Это строки, начинающиеся с `ttyv`:

#	name	getty	type	status	comments
#	ttyv0	"/usr/libexec/getty Pc"	cons25	on	secure
#	Virtual terminals				
	ttyv1	"/usr/libexec/getty Pc"	cons25	on	secure
	ttyv2	"/usr/libexec/getty Pc"	cons25	on	secure
	ttyv3	"/usr/libexec/getty Pc"	cons25	on	secure
	ttyv4	"/usr/libexec/getty Pc"	cons25	on	secure
	ttyv5	"/usr/libexec/getty Pc"	cons25	on	secure
	ttyv6	"/usr/libexec/getty Pc"	cons25	on	secure
	ttyv7	"/usr/libexec/getty Pc"	cons25	on	secure
	ttyv8	"/usr/X11R6/bin/xdm -nodaemon"	xterm	off	secure



За детальным описанием каждой колонки этого файла и всех опций, которые можно указать для настройки виртуальных консолей, обращайтесь к `ttys(5)`.

### 3.2.5. Консоль в однопользовательском режиме

Детальное описание "однопользовательского режима" можно найти в Разд. 12.6.2. Стоит отметить, что при работе FreeBSD в однопользовательском режиме есть только одна консоль. Виртуальных консолей нет. Установки консоли в однопользовательском режиме можно найти в файле `/etc/ttys`. Обратите внимание на строку, начинающуюся с `console`:

```
# name  getty                                type    status      comments
#
# If console is marked "insecure", then init will ask for the root password
# when going to single-user mode.
console none                                unknown off  secure
```

**Замечание:** Как сказано в комментариях выше строки `console`, можно отредактировать эту строку и изменить `secure` на `insecure`. Если вы сделаете это, FreeBSD даже при загрузке в однопользовательском режиме будет запрашивать пароль `root`.

*Будьте осторожны при изменении этого значения на `insecure`.* Если вы забудете пароль `root`, загрузка в однопользовательский режим сильно усложнится. Это все еще возможно, но несколько более сложно для тех, кто еще не очень освоился с процессом загрузки FreeBSD и вызова программ.

## 3.3. Права доступа

FreeBSD является прямым потомком BSD UNIX и основывается на некоторых ключевых концепциях UNIX. В первую очередь это, конечно, тот факт, что FreeBSD – многопользовательская операционная система. Это означает, что несколько пользователей могут работать одновременно, решая различные задачи и совершенно не мешая друг другу. На системе лежит ответственность за правильное разделение и управление такими ресурсами как память, процессорное время, периферийные устройства и прочее.

Многопользовательская среда предполагает наличие механизма регулирования прав доступа к любому ресурсу в системе. Существует три типа прав доступа: на чтение, запись и исполнение. Права сгруппированы три по три, соответственно чтение/запись/выполнение для владельца/группы/всех остальных. Численное представление:

Значение	Права доступа	Список файлов каталога
0	Ничего не разрешено	---
1	Нельзя читать и писать, разрешено исполнять	--x
2	Нельзя читать и исполнять, разрешено писать	-w-

Значение	Права доступа	Список файлов каталога
3	Нельзя читать, разрешено писать <code>-wx</code> и исполнять	
4	Разрешено читать, нельзя писать <code>r--</code> и исполнять	
5	Разрешено читать и исполнять, нельзя писать <code>r-x</code>	
6	Разрешено читать и писать, нельзя исполнять <code>rw-</code>	
7	Разрешено все <code>rwx</code>	

Вы можете использовать опцию `-l` команды `ls(1)` для получения подробного листинга каталога, включающего колонку с информацией о правах на файл для владельца, группы и всех остальных. Например, команда `ls -l` в произвольном каталоге может вывести следующее:

```
% ls -l
total 530
-rw-r--r--  1 root  wheel    512 Sep  5 12:31 myfile
-rw-r--r--  1 root  wheel    512 Sep  5 12:31 otherfile
-rw-r--r--  1 root  wheel   7680 Sep  5 12:31 email.txt
...
```

Вот как выглядит первая колонка вывода `ls -l`:

```
-rw-r--r--
```

Первый (считая слева) символ говорит обычный ли это файл, каталог, символьное устройство, сокет или любое другое псевдофайловое устройство. В нашем случае – указывает на обычный файл. Следующие три символа (в данном случае это `rw-`) задают права доступа владельца файла. Затем идут права группы, которой принадлежит файл (`r--`). Последняя тройка (`r--`) определяет права для всех остальных. Минус означает отсутствие каких-либо прав (т.е. нельзя ни читать, ни писать, ни выполнять). В данном случае права установлены таким образом, что владелец может читать и писать в файл, а группа и другие могут только читать. Таким образом, численное представление прав 644, где каждая цифра представляет три части прав на файл.

Права на устройства контролируются аналогичным образом. В FreeBSD все устройства представлены в виде файлов, которые можно открывать, читать и писать в них. Эти специальные файлы содержатся в каталоге `/dev`.

Каталоги также являются файлами. К ним применимы те же права на чтение, запись и выполнение. Правда, в данном случае "выполнение" имеет несколько другой смысл. Когда каталог помечен как "исполнимый", это означает, что можно "зайти" в него (с помощью команды `cd`, `change directory`). Это также означает, что в данном каталоге можно получить доступ к файлам, имена которых известны (конечно, если собственные права на файл разрешают такой доступ).

Если же требуется получить список файлов в некотором каталоге, права доступа на него должны включать доступ на чтение. Для того, чтобы удалить из каталога какой-либо файл, имя которого известно, на этот каталог должны быть даны права на запись и на исполнение.

Существуют и другие права доступа, но они как правило используются в особых случаях, например, `setuid`-бит на выполняемые файлы и `sticky`-бит на каталоги. За дополнительными сведениями по этому

вопросу обращайтесь к `chmod(1)`.

### 3.3.1. Символические обозначения прав

*Предоставил Tom Rhodes.*

Символические обозначения, иногда называемые символическими выражениями, используют буквы вместо восьмеричных значений для назначения прав на файлы и каталоги. Символические выражения используют синтаксис (кто) (действие) (права), где существуют следующие значения:

Опция	Буква	Значение
(кто)	u	Пользователь (User)
(кто)	g	Группа (Group)
(кто)	o	Другие (Other)
(кто)	a	Все (All, "world")
(действие)	+	Добавление прав
(действие)	-	Удаление прав
(действие)	=	Явная установка прав
(права)	r	Чтение (Read)
(права)	w	Запись (Write)
(права)	x	Выполнение (Execute)
(права)	t	Sticky бит
(права)	s	SUID или SGID

Эти значения используются командой `chmod(1)` так же как и раньше, но с буквами. Например, вы можете использовать следующую команду для запрета доступа других пользователей к *FILE*:

```
% chmod go= FILE
```

Для изменения более чем одного набора прав можно применить список, разделенный запятыми. Например, следующая команда удалит права группы и "всех остальных" на запись в *FILE*, а затем добавит права на выполнение для всех:

```
% chmod go-w,a+x FILE
```

## 3.4. Структура каталогов

Файловая система FreeBSD является ключевым моментом в понимании устройства всей системы.

Самым важным понятием является, несомненно, корневой каталог, обозначаемый символом `/`.

Корневой каталог монтируется самым первым на этапе загрузки и содержит все необходимое, чтобы подготовить систему к загрузке в многопользовательский режим. Корневой каталог также содержит точки монтирования всех других файловых систем.

Точкой монтирования называется каталог, который будет соответствовать корню примонтированной файловой системы. Стандартные точки монтирования включают `/usr`, `/var`, `/tmp`, `/mnt` и `/cdrom`. Эти каталоги обычно перечислены в файле `/etc/fstab`, в котором указаны файловые системы и их точки

монтирования. Большинство файловых систем, описанных в `/etc/fstab` монтируются автоматически из скрипта `rc(8)`, если только для них не указана опция `noauto`. Более детальная информация находится в Разд. 3.6.1.

Полное описание иерархии файловой системы есть в `hier(7)`. Здесь же мы упомянем лишь наиболее важные каталоги.

Каталог	Описание
<code>/</code>	Корневой каталог файловой системы.
<code>/bin/</code>	Основные утилиты, необходимые для работы как в однопользовательском, так и в многопользовательском режимах.
<code>/boot/</code>	Программы и конфигурационные файлы, необходимые для нормальной загрузки операционной системы.
<code>/boot/defaults/</code>	Конфигурационные файлы с настройками по умолчанию, используемые в процессе загрузки операционной системы (см. <code>loader.conf(5)</code> ).
<code>/dev/</code>	Файлы устройств (см. <code>intro(4)</code> ).
<code>/etc/</code>	Основные конфигурационные файлы системы и скрипты.
<code>/etc/defaults/</code>	Основные конфигурационные файлы системы с настройками по умолчанию (см. <code>rc(8)</code> ).
<code>/etc/mail/</code>	Конфигурационные файлы для систем обработки почты (например, <code>sendmail(8)</code> ).
<code>/etc/namedb/</code>	Конфигурационные файлы для утилиты <code>named</code> (см. <code>named(8)</code> ).
<code>/etc/periodic/</code>	Файлы сценариев, выполняемые ежедневно, еженедельно и ежемесячно (см. <code>cron(8)</code> и <code>periodic(8)</code> ).
<code>/etc/ppp/</code>	Конфигурационные файлы для утилиты <code>ppp</code> (см. <code>ppp(8)</code> ).
<code>/mnt/</code>	Пустой каталог, часто используемый системными администраторами как временная точка монтирования.
<code>/proc/</code>	Виртуальная файловая система, отображающая текущие процессы (см. <code>procfs(5)</code> , <code>mount_procfs(8)</code> ).
<code>/rescue/</code>	Статически слинкованные программы для восстановления после сбоев. Обратитесь к <code>rescue(8)</code> .
<code>/root/</code>	Домашний каталог пользователя <code>root</code> .
<code>/sbin/</code>	Системные утилиты и утилиты администрирования, необходимые для работы как в однопользовательском, так и в многопользовательском режимах.

**Каталог**`/stand/``/tmp/``/usr/``/usr/bin/``/usr/include/``/usr/lib/``/usr/libdata/``/usr/libexec/``/usr/local/``/usr/obj/``/usr/ports/``/usr/sbin/``/usr/share/``/usr/src/``/usr/X11R6/``/var/``/var/log/``/var/mail/``/var/spool/``/var/tmp/``/var/yp/`**Описание**

Программы, необходимые для работы в автономном режиме (например, при установке системы).

Временный каталог (часто монтируется как `mfs(8)`; обычно содержимое этого каталога теряется во время перезагрузки системы).

Большинство пользовательских утилит и приложений.

Пользовательские утилиты и приложения общего назначения.

Стандартные заголовочные файлы для языка C.

Файлы стандартных библиотек.

Файлы данных для различных утилит.

Системные демоны и утилиты (выполняемые другими программами).

Локальные пользовательские приложения, библиотеки, и т.д. Также используется по умолчанию коллекцией портов. Внутри `/usr/local` иерархия каталогов должна следовать `hier(7)` для `/usr`. Исключение составляют каталог `man`, который расположен непосредственно в `/usr/local`, а не в `/usr/local/share`, и документация портов, которая расположена в `share/doc/port`.

Архитектурно-зависимые файлы и каталоги, образующиеся в процессе сборки системы из исходных текстов в `/usr/src`.

Коллекция портов FreeBSD (опционально).

Системные утилиты и утилиты администрирования (исполняемые пользователем).

Архитектурно-независимые файлы.

Исходные тексты BSD и/или программ.

Утилиты, приложения и библиотеки X11R6 (X Window System; необязательно).

Файлы журналов общего назначения, временные, перемещаемые файлы и файлы очередей.

Различные файлы системных журналов.

Почтовые ящики пользователей.

Файлы очередей печати, почты, и пр.

Временные файлы, которые необходимо сохранять между перезагрузками операционной системы.

Карты (maps) NIS.

### 3.5. Организация дисков

Наименьшая единица, которую FreeBSD использует для обращения к файлам, это имя файла. Имена файлов чувствительны к регистру, поэтому `readme.txt` и `README.TXT` — два разных файла. FreeBSD не использует расширение файла (`.txt`) для определения программа это, документ или другой тип данных.

Файлы хранятся в каталогах. Каталоги могут не содержать файлов, или могут содержать много сотен файлов. Каталоги также могут содержать другие каталоги, что позволяет создавать иерархию каталогов один в другом. Это упрощает организацию данных.

Обращение к файлам происходит путем задания имени файла или каталога, дополняемого прямым слэшем `/`, за которым может следовать имя другого каталога. Если есть каталог `foo`, содержащий каталог `bar`, который содержит файл `readme.txt`, полное имя, или *путь* к файлу будет `foo/bar/readme.txt`.

Каталоги хранятся в файловых системах. Каждая файловая система содержит один каталог на верхнем уровне, называемый *корневым каталогом* этой файловой системы. Этот корневой каталог может содержать другие каталоги.

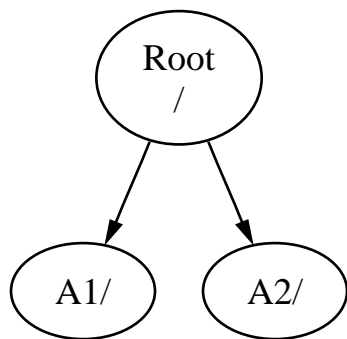
Внешне это может быть похоже на те операционные системы, которые вы возможно использовали. Есть несколько отличий: например, MS-DOS использует `\` для разделения имен файлов и каталогов, а Mac OS® использует `:`.

FreeBSD не использует букв дисков, или других имен дисков в пути. Вам не нужно писать `c:/foo/bar/readme.txt` в FreeBSD.

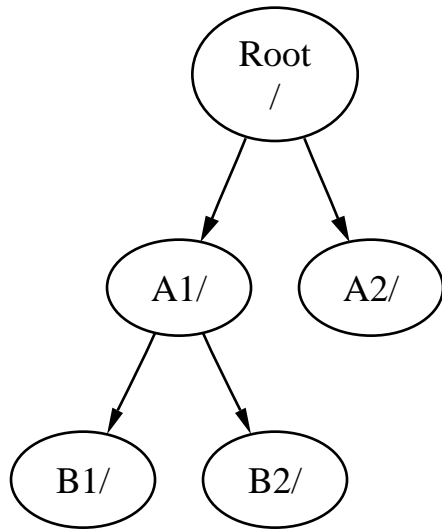
Вместо этого, одна файловая система назначается *корневой файловой системой*. Обращение к корневому каталогу корневой файловой системы происходит через `/`. Любая другая файловая система *монтируется* к корневой файловой системе. Неважно как много дисков есть в вашей системе FreeBSD, каждый каталог будет выглядеть как расположенный на том же диске.

Предположим, у вас есть три файловых системы: A, B, и C. Каждая файловая система имеет один корневой каталог, в котором содержатся другие каталоги, называемые A1, A2 (и аналогично B1, B2 и C1, C2).

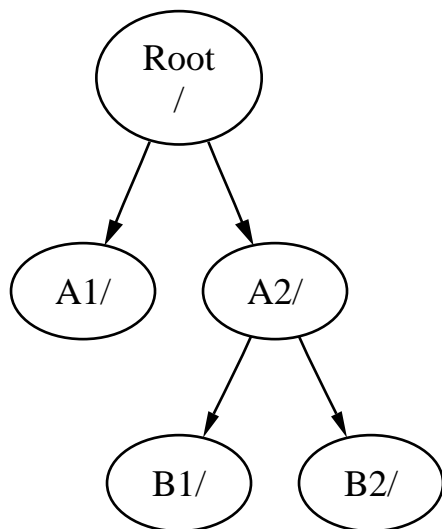
Назовем A корневой файловой системой. Если вы используете команду `ls` для просмотра содержимого каталога, вы увидите два подкаталога, A1 и A2. Дерево каталогов выглядит так:



Файловая система должна быть примонтирована к каталогу другой файловой системы. Предположим, что вы монтируете файловую систему в каталог A1. Корневой каталог заменяется A1, а каталоги в B отображаются соответственно:

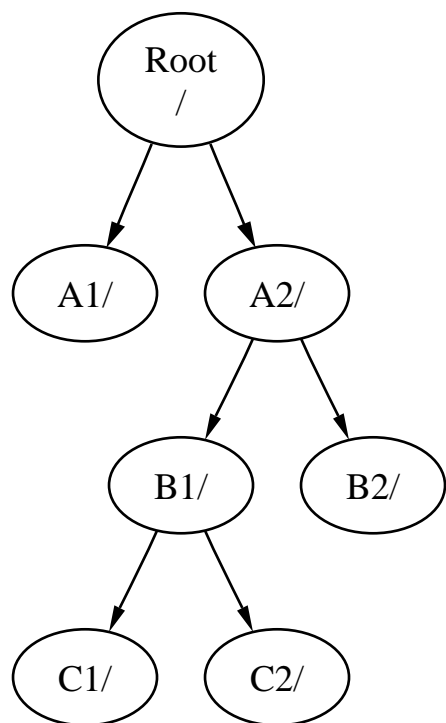


Если потребуется, любые файлы из каталогов B1 или B2 могут быть получены через путь /A1/B1 или /A1/B2. Все файлы, бывшие в /A1, временно скрыты. Они появятся, если B будет *размонтирована* с A. Если B была смонтирована на A2, диаграмма будет выглядеть так:

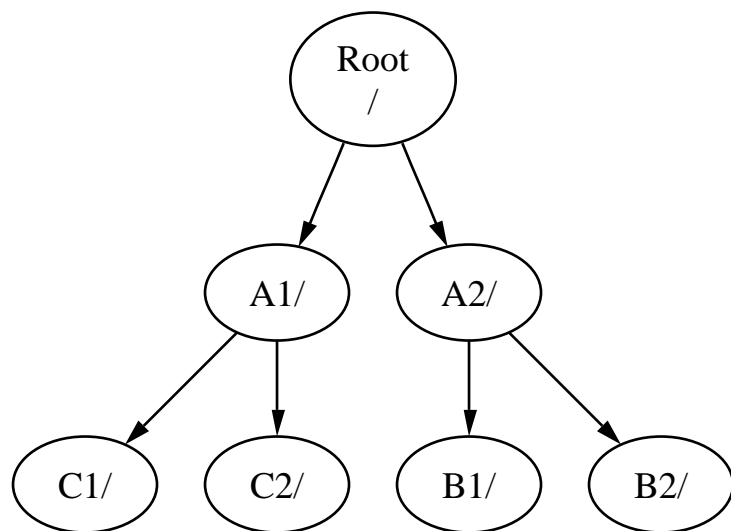


а пути будут /A2/B1 и /A2/B2 соответственно.

Файловые системы могут быть смонтированы одна на другую. Продолжая предыдущий пример, файловая система C может быть смонтирована на каталог B1 файловой системы B в таком порядке:



Или с может быть смонтирована прямо на файловую систему A, на каталог A1 :



Если вы знакомы с MS-DOS, это похоже, хотя и не идентично, команде `join`.

Как правило, это не должно вас интересовать. Обычно вы создаете файловые системы во время установки FreeBSD, решаете куда их монтировать, и ничего не меняете, пока не понадобится добавить новый диск.

Можно создать одну большую корневую файловую систему и не создавать других. У такого подхода



есть несколько недостатков и одно преимущество.

### Преимущества нескольких файловых систем

- Различные файловые системы могут иметь различные *опции монтирования*. Например, в целях безопасности корневая файловая система может быть смонтирована только для чтения, что делает невозможным случайное удаление или редактирование критически важного файла. Отделение файловых систем, используемых пользователями для записи, таких как /home, от других файловых систем позволяет также монтировать их с параметром *nosuid*; этот параметр отменяет действие битов *suid/ guid* на исполняемых файлах, в этой файловой системе, что потенциально повышает безопасность.
- FreeBSD автоматически оптимизирует расположение файлов на файловой системе в зависимости от того, как файловая система используется. Файловая система, содержащая множество мелких часто записываемых файлов, будет иметь оптимизацию, отличную от таковой для файловой системы, содержащей несколько больших файлов. На одной большой файловой системе эта оптимизация не работает.
- Файловые системы FreeBSD очень устойчивы к внезапному отключению. Тем не менее, потеря питания в критический момент все же может повредить структуру файловой системы. Разделение данных на несколько файловых систем повышает шансы, что система все-таки будет работать и делает более легким восстановление с резервной копии.

### Преимущество одной файловой системы

- Размер файловых систем фиксирован. Если вы создаете файловую систему при установке FreeBSD и задаете определенный размер, позднее вы можете обнаружить что нужен раздел большего размера. Это не так легко сделать без резервного копирования, создания файловых систем нового размера и последующего восстановления сохраненных данных.

**Важно:** В FreeBSD 4.4 и выше представлена новая команда, *growfs(8)*, которая позволяет увеличивать размер файловой системы на лету, устраняя это ограничение.

Файловые системы содержатся в разделах. Этот термин не имеет того же смысла, что и при более раннем его использовании в этой главе, из-за наследия UNIX в FreeBSD. Каждый раздел обозначается буквой от a до h. Каждый раздел может содержать только одну файловую систему, это значит что файловая система может быть описана ее точкой монтирования в файловой иерархии, или буквой раздела, в котором она содержится.

FreeBSD также использует дисковое пространство под *раздел подкачки (swap space)*. Подкачка позволяет FreeBSD работать с *виртуальной памятью*. Ваш компьютер может работать так, как если бы у него было больше памяти, чем есть на самом деле. Когда у FreeBSD кончается память, она перемещает часть данных, не используемых в данный момент, в раздел подкачки и возвращает их обратно (перемещая в подкачку что-то другое), когда они нужны.

По некоторым разделам есть определенные соглашения.

Раздел	Соглашение
a	Как правило, содержит корневую файловую систему

Раздел	Соглашение
b	Как правило, содержит раздел подкачки
c	Как правило, такого же размера, что и весь слайс (slice). Это позволяет утилитам, которым нужно работать над всем слайсом (например, сканер плохих блоков), работать с разделом c. В обычной ситуации не нужно создавать файловую систему на этом разделе.
d	Раздел d создавался для специальных целей, хотя сейчас не используется. На настоящий момент, некоторые утилиты могут работать некорректно при попытке обращения к d, поэтому <b>sysinstall</b> обычно не создает раздел d.

Каждый раздел-содержащий-файловую-систему хранится на том, что во FreeBSD называется *слайс* (slice). Слайс — это термин FreeBSD, то, что обычно называют разделом, и опять же это из-за UNIX основы FreeBSD. Слайсы нумеруются с 1 по 4.

Номера слайсов следуют за именем устройства, предваряемые строчной s, начиная с 1. Так "da0s1" это первый слайс первого SCSI устройства. Может быть только четыре физических слайса на диске, но могут быть логические слайсы нужного типа внутри физических слайсов. Эти дополнительные слайсы нумеруются начиная с 5, так что "ad0s5" это первый дополнительный слайс на первом IDE диске. Эти устройства используются файловыми системами, занимающими весь слайс.

Слайсы, "экслюзивно выделенные (dangerously dedicated)" физические устройства и другие устройства содержат *разделы*, представляемые буквами от a до h. Эти буквы добавляются к имени устройства. "da0a" это раздел a на первом устройстве da, который "экслюзивно выделен". "ad1s3e" это пятый раздел в третьем слайсе второго IDE диска.

Наконец, каждый диск идентифицирован. Имя диска начинается с кода, обозначающего тип диска, затем идет номер диска. В отличие от слайсов, нумерация дисков начинается с 0. Основные коды, которые вам могут встретиться, есть в Табл. 3-1.

В то время, как ссылка на раздел FreeBSD требует также указания слайса и диска, содержащего раздел, ссылка на слайс требует также указания имени диска. Делайте это указанием имени диска, s, номера слайса, и затем буквы раздела. Примеры показаны в Прим. 3-1.

Прим. 3-2 показывает концептуальную модель диска, которая должна помочь прояснить ситуацию.

Для установки FreeBSD вы должны сначала настроить слайсы дисков, затем создать разделы внутри слайсов, которые будут использованы для FreeBSD, а затем создать файловую систему (или подкачку) в каждом разделе и решить, куда файловая система будет смонтирована.

**Таблица 3-1. Коды дисковых устройств**

Код	Значение
ad	ATAPI (IDE) диск
da	SCSI direct access диск
acd	ATAPI (IDE) CDROM
cd	SCSI CDROM
fd	Floppy disk

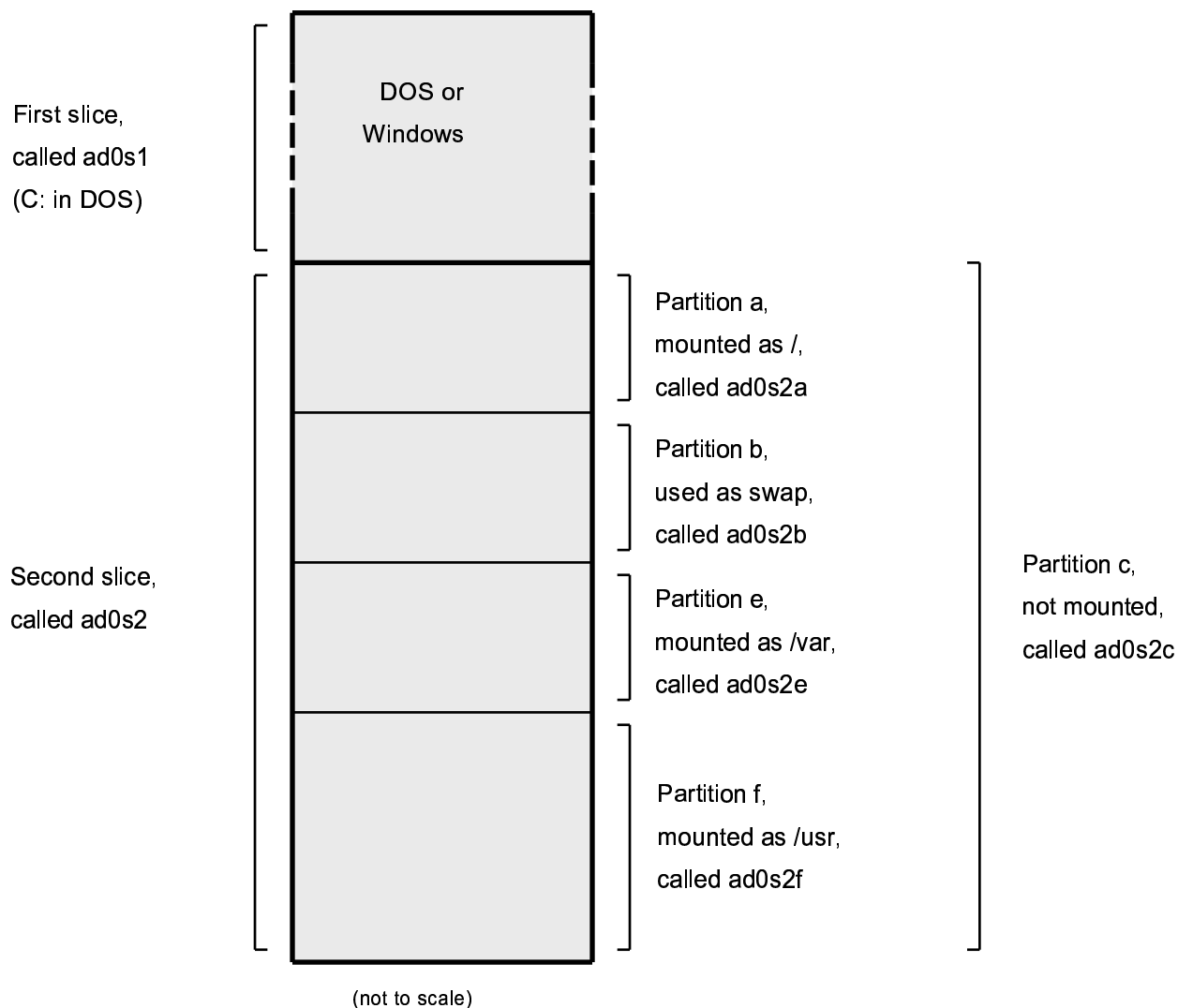
**Пример 3-1. Пример имен диска, слайса, и раздела**

Имя	Значение
ad0s1a	Первый раздел (a) на первом слайсе (s1) первого IDE диска (ad0).
da1s2e	Пятый раздел (e) на втором слайсе (s2) второго SCSI диска (da1).

**Пример 3-2. Концептуальная модель диска**

Эта диаграмма показывает первый подключенный к системе IDE диск с точки зрения FreeBSD. Предположим, что размер диска 4 GB, и он содержит два 2 GB слайса (MS-DOS разделы). Первый слайс содержит MS-DOS диск, с:, а второй слайс содержит установленную FreeBSD. В этом примере в установленной FreeBSD три раздела и раздел подкачки.

В каждом из трех разделов есть файловая система. Раздел a используется для корневой файловой системы, e для иерархии каталогов /var, а f для иерархии каталогов /usr.



### 3.6. Монтирование и размонтирование файловых систем

Файловая система лучше всего представима в виде дерева, с корнем в /. Каталоги, /dev, /usr и прочие – это ветви дерева, которые, в свою очередь, являются корнями для поддеревьев, также имеющих ветви (/usr/local), и т.д.

Хорошей практикой является разнесение некоторых особо важных каталогов на разные файловые системы. Например, /var, содержит log/, spool/, а также всевозможные временные файлы и нередко

может занять все свободное место на диске. Поэтому лучше смонтировать `/var` отдельно, чтобы избежать переполнения `/`.

Часто бывает так, что некоторые разделы файловой системы расположены на разных физических носителях (дисках, CDROM), виртуальных или сетевых (например, сетевая файловая система (Network File System, NFS)). В этом случае узлы файловой иерархии будут расположены на разных файловых системах.

### 3.6.1. Файл `fstab`

Файловые системы, перечисленные в `/etc/fstab`, монтируются автоматически в процессе загрузки (если, конечно, для них не указана опция `noauto`).

Формат файла `/etc/fstab` следующий (файловые системы перечисляются построчно):

*устройство /точка-монтирования тип файловой системы опции частота дампов порядок проверки*

#### устройство

Имя устройства (которое должно присутствовать), как описано в Разд. 16.2.

#### точка монтирования

Каталог (существующий), куда следует смонтировать файловую систему.

#### тип файловой системы

Тип файловой системы, который передается программе `mount(8)`. По умолчанию FreeBSD использует `ufs`.

#### опции

Например, `rw`, для монтирования файловой системы в режиме "чтение-запись", или `ro`, для режима "только чтение", за которыми могут следовать и другие опции. Довольно часто используется опция `noauto`, чтобы не монтировать автоматически файловые системы в процессе загрузки. Об остальных опциях можно прочитать в `mount(8)`.

#### частота дампов

Используется утилитой `dump(8)` для определения файловых систем, с которых необходимо периодически снимать специальные архивные копии. При отсутствии этого параметра принимается равным нулю.

#### порядок проверки

Определяет порядок, в котором следует проверять файловые системы (чаще всего, в случае некорректного размонтирования или внезапной перезагрузки системы). Если файловую систему не нужно проверять, этот параметр должен быть установлен в ноль. Для корневой файловой системы (которая должна быть проверена в первую очередь) установите его в 1. Для всех остальных – 2 или больше. Если две или более файловые системы имеют одинаковое значение `passno`, `fsck(8)` попытается проверять их параллельно (если, конечно, это возможно физически).

Обратитесь к `fstab(5)` за дополнительной информацией о формате файла `/etc/fstab` и различных опциях монтирования.

### 3.6.2. Команда `mount`

Команда `mount(8)` используется, как следует из ее имени, для монтирования файловых систем.

Пример использования (простейший случай):

```
# mount устройство точка-монтирования
```

Перечислим основные опции, которые может принимать команда `mount(8)` (полный список смотрите на странице справочника):

#### Опции монтирования

`-a`

Смонтировать все файловые системы, перечисленные в файле `/etc/fstab`. Исключение составляют помеченные как `"noauto"`, перечисленные после опции `-t` и уже смонтированные.

`-d`

Сделать все, кроме самого системного вызова `mount`. Эта опция полезна вместе с флагом `-v` для определения того, что на самом деле пытается сделать `mount(8)`.

`-f`

Монтировать поврежденный раздел (опасно!), или форсировать отмену всех запросов на запись при изменении режима монтирования с `"чтение-запись"` на `"только чтение"`.

`-r`

Монтировать файловую систему в режиме `"только для чтения"`. То же самое, что и указание аргумента `ro` (`rdonly` для FreeBSD версий старше, чем 5.2) для опции `-o`.

`-t fstype`

Монтировать файловую систему как систему указанного типа, или, в случае опции `-a`, только файловые системы данного типа.

По умолчанию, тип файловой системы – `"ufs"`.

`-u`

Обновить опции монтирования для файловой системы.

`-v`

Выдавать более подробную информацию.

`-w`

Монтировать файловую систему в режиме `"чтение-запись"`.

Опция `-o` принимает разделенные запятыми аргументы, включая нижеперечисленные:

nodev

Не обращать внимание на специальные файлы устройств на файловой системе (то есть не считать их файлами устройств). Весьма полезная опция для повышения безопасности системы.

noexec

Запрет на исполнение бинарных файлов на файловой системе (тоже полезная опция для повышения безопасности системы).

nosuid

Игнорировать `setuid` и `setgid` биты на файловой системе (еще одна полезная опция для повышения безопасности системы).

### 3.6.3. Команда `umount`

Команда `umount(8)` принимает в качестве параметра точку монтирования какой-либо файловой системы, имя устройства, опцию `-a` или `-A`.

Кроме того, вы можете дополнительно указать опцию `-f` для форсированного размонтирования файловой системы, и `-v` для получения более подробной информации. Имейте ввиду, что это в общем случае опасно и потому не рекомендуется, так как тем самым вы можете нарушить работу компьютера или повредить данные на файловой системе.

Опции `-a` и `-A` используются для размонтирования всех файловых систем (разве что вы укажете опцию `-t`). Разница состоит в том, что `-A` не пытается размонтировать корневую файловую систему.

## 3.7. Процессы

FreeBSD является многозадачной операционной системой. Это означает, что одновременно может быть запущена более чем одна программа. Каждая программа, работающая в некоторый момент времени, называется *процессом*. Каждая команда, которую вы запускаете, порождает хотя бы один процесс. Есть несколько системных процессов, запущенных все время и поддерживающих функциональность системы.

У каждого процесса есть уникальный номер, называемый *process ID*, или *PID*, и, как и у файлов, у каждого процесса есть владелец и группа. Информация о владельце и группе процесса используется для определения того, какие файлы и устройства могут быть открыты процессом с учетом прав на файлы, о которых говорилось ранее. Также у большинства процессов есть родительский процесс. Например, при запуске команд из оболочки, оболочка является процессом и любая запущенная команда также является процессом. Для каждого запущенного таким путем процесса оболочка будет являться родительским процессом. Исключением из этого правила является специальный процесс, называемый `init(8)`. `init` всегда первый процесс, его `PID` всегда 1. `init` запускается автоматически ядром во время загрузки FreeBSD.

Две команды очень полезны для просмотра работающих в системе процессов, это `ps(1)` и `top(1)`. Команда `ps` используется для получения списка запущенных процессов и может показать их `PID`, сколько памяти они используют, команду, которой они были запущены и т.д. Команда `top` показывает

запущенные процессы и обновляет экран каждые несколько секунд, что позволяет наблюдать за работой компьютера в реальном времени.

По умолчанию, `ps` показывает только принадлежащие вам процессы. Например:

```
% ps
  PID  TT  STAT      TIME COMMAND
  298  p0  Ss      0:01.10 tcsh
 7078  p0  S        2:40.88 xemacs mdoc.xsl (xemacs-21.1.14)
37393  p0  I        0:03.11 xemacs freebsd.dsl (xemacs-21.1.14)
48630  p0  S        2:50.89 /usr/local/lib/netcape-linux/navigator-linux-4.77.bi
48730  p0  IW       0:00.00 (dns helper) (navigator-linux-)
72210  p0  R+       0:00.00 ps
  390  p1  Is       0:01.14 tcsh
 7059  p2  Is+      1:36.18 /usr/local/bin/mutt -y
 6688  p3  IWs      0:00.00 tcsh
10735  p4  IWs      0:00.00 tcsh
20256  p5  IWs      0:00.00 tcsh
  262  v0  IWs      0:00.00 -tcsh (tcsh)
  270  v0  IW+      0:00.00 /bin/sh /usr/X11R6/bin/startx -- -bpp 16
  280  v0  IW+      0:00.00 xinit /home/nik/.xinitrc -- -bpp 16
  284  v0  IW       0:00.00 /bin/sh /home/nik/.xinitrc
  285  v0  S        0:38.45 /usr/X11R6/bin/sawfish
```

Как вы можете видеть в данном примере, вывод `ps(1)` организован в несколько колонок.

Идентификатор процесса `PID` обсуждался ранее. `PID` назначаются с 1 до 99999 и опять с начала, если последнее число будет превышено. Колонка `TT` показывает терминал (tty), на котором запущена программа (можете пока забыть про это). `STAT` показывает состояние программы и опять же может быть пока проигнорирован. `TIME` это количество времени центрального процессора, использованное программой — это обычно не время, прошедшее с запуска программы, поскольку большинство программы проводят много времени в ожидании некоторого события перед тем, как занять время процессора. Наконец, `COMMAND` это команда, которой программа была запущена.

У `ps(1)` есть множество различных опций, влияющих на выводимую информацию. Один из наиболее полезных наборов опций это `auxww`. `a` позволяет показать информацию о всех запущенных процессах, а не только тех, которыми вы владеете. `u` показывает имя пользователя, владеющего процессом, и информацию об используемой памяти. `x` показывает информацию о процессах-демонах и `ww` указывает `ps(1)` показать всю командную строку, вместо обрезания ее, когда она станет слишком длинной, чтобы уместиться на экран.

Вывод `top(1)` похож на только что описанный. Обычно он выглядит так:

```
% top
last pid: 72257;  load averages:  0.13,  0.09,  0.03    up 0+13:38:33  22:39:10
47 processes:  1 running, 46 sleeping
CPU states: 12.6% user,  0.0% nice,  7.8% system,  0.0% interrupt, 79.7% idle
Mem: 36M Active, 5256K Inact, 13M Wired, 6312K Cache, 15M Buf, 408K Free
Swap: 256M Total, 38M Used, 217M Free, 15% Inuse

  PID USERNAME PRI NICE  SIZE  RES STATE   TIME  WCPU   CPU COMMAND
72257 nik        28  0 1960K 1044K RUN      0:00 14.86% 1.42% top
 7078 nik         2  0 15280K 10960K select  2:54  0.88%  0.88% xemacs-21.1.14
  281 nik         2  0 18636K  7112K select  5:36  0.73%  0.73% XF86_SVGA
```



```

296 nik      2    0  3240K  1644K select  0:12  0.05%  0.05% xterm
48630 nik    2    0 29816K  9148K select  3:18  0.00%  0.00% navigator-linu
175 root     2    0   924K   252K select  1:41  0.00%  0.00% syslogd
7059 nik     2    0  7260K  4644K poll    1:38  0.00%  0.00% mutt
...

```

Вывод разбит на два раздела. Заголовок (первые пять строк) показывает PID последнего запущенного процесса, среднее значение загрузки системы (которое показывает насколько система занята), время работы системы с последней перезагрузки и текущее время. Другие цифры заголовка относятся к количеству запущенных процессов (в данном примере 47), количеству занятой памяти и подкачки и время, занимаемое различными состояниями процессора.

Ниже идут несколько колонок, содержащих похожую на вывод `ps(1)` информацию. Как и раньше, это PID, время процессора, командная строка. `top(1)` показывает также величину занятой процессом памяти. Это значение разбито на две колонки, одна для общего объема, а другая для резидентного — общий объем показывает сколько всего памяти нужно приложению, а резидентный показывает количество памяти, используемой в данный момент. Из этого примера видно, что **Netscape®** требует почти 30 МВ памяти, но в данный момент использует только 9 МВ.

`top(1)` автоматически обновляет экран каждые две секунды; это значение можно изменить опцией `s`.

### 3.8. Демоны, сигналы, уничтожение процессов

Если вы запускаете редактор, им можно легко управлять, открывать в нем файлы и т.д. Вы можете делать это, поскольку редактор предоставляет такие возможности и потому, что редактор присоединен к *терминалу*. Некоторые программы разработаны без поддержки интерфейса пользователя, поэтому они отсоединяются от терминала при первой возможности. Например, веб-сервер целый день отвечает на запросы из сети, и ему как правило не требуется ваше вмешательство. Программы, передающие почту от сервера к серверу — другой пример приложений этого класса.

Мы называем эти программы *демонами*. Демоны это персонажи греческой мифологии; хорошие или плохие, они были спутниками человека и, вообще говоря, выполняли полезную работу для людей. Почти как веб- и почтовые серверы выполняют полезную работу сегодня. Это причина, по которой талисманом BSD долгое время является веселый демон в тапочках и с вилами.

Есть соглашение, по которому имя программы, которая обычно запускается как демон, заканчивается на `'d'`. **BIND** это Berkeley Internet Name Daemon (выполняемая программа называется `named`), программа веб сервера **Apache** называется `httpd`, демон очереди печати это `lpd` и так далее. Это соглашение, а не жесткое правило; например, главный почтовый демон для **Sendmail** называется `sendmail`, а не `maild`, как вы могли бы предположить.

Иногда может потребоваться взаимодействие с процессом демона. Эти можно сделать с помощью *сигналов*, т.е. взаимодействовать с демонами (или с любыми запущенными процессами), посылая им сигнал. Есть множество различных сигналов — некоторые из них имеют специальное значение, другие обрабатываются приложением, реакция которого на эти сигналы должна быть описана в документации. Вы можете посылать сигналы только тем процессам, владельцем которых являетесь. Если вы отправите сигнал какому-то другому процессу с помощью `kill(1)` или `kill(2)`, доступ будет запрещен. Исключением из правил является пользователь `root`, который может отправлять сигналы любому процессу.

В некоторых случаях FreeBSD тоже посылает сигналы приложениям. Если приложение плохо написано и пробует обратиться к области памяти, к которой оно не должно обращаться, FreeBSD посылает процессу сигнал *нарушение сегментации* (SIGSEGV). Если приложение использует системный вызов alarm(3), чтобы получить уведомление по истечении определенного периода времени, будет отправлен сигнал Alarm (SIGALRM) и т.д.

Два сигнала могут быть использованы для завершения процесса, SIGTERM и SIGKILL. SIGTERM это корректный способ завершить процесс; процесс может *поймать* сигнал, определить, что его хотят завершить, закрыть любые файлы, которые он мог открыть, и закончить то, что он делал в момент перед закрытием. В некоторых случаях процесс может даже игнорировать SIGTERM, если выполняет задачу, которая не может быть прервана.

SIGKILL не может быть проигнорирован процессом. Этот сигнал говорит "Меня не волнует что ты делаешь — остановись немедленно". Если вы посылаете процессу SIGKILL, FreeBSD сразу же остановит этот процесс<sup>4</sup>.

Другие сигналы, которые возможно вам понадобятся, SIGHUP, SIGUSR1, и SIGUSR2. Это сигналы общего назначения, различные приложения могут по-разному реагировать на них.

Предположим, что вы изменили файл конфигурации веб сервера — теперь нужно указать ему перечитать конфигурацию. Можно остановить и запустить httpd, но это приведет к кратковременной остановке сервера, которая может быть нежелательна. Большинство демонов написаны так, чтобы при получении сигнала SIGHUP перечитывать файлы конфигурации. Поэтому вместо уничтожения и запуска httpd можно послать сигнал SIGHUP. Поскольку нет стандартного способа реагирования на этот сигнал, различные демоны будут вести себя по разному; прочитайте документацию на демон по этому вопросу.

Сигналы посылаются с помощью команды kill(1), как показано в этом примере.

### Отправка сигнала процессу

Этот пример показывает как послать сигнал inetd(8). Файл конфигурации inetd /etc/inetd.conf, inetd перечитает этот файл, если ему отправить сигнал SIGHUP.

1. Нужно определить PID процесса, которому вы хотите отправить сигнал. Сделайте это с помощью ps(1) и grep(1). Команда grep(1) используется для поиска по заданной строке в выходном потоке. Эта команда запускается под обычным пользователем, а inetd(8) под root, поэтому ps(1) должна быть запущена с параметром ax.

```
% ps -ax | grep inetd
198  ??  IWs      0:00.00 inetd -wW
```

Итак, PID inetd(8) 198. В некоторых случаях в выводе команды может также появиться grep inetd. Это из-за способа, которым ps(1) получает список запущенных процессов.

2. Используйте kill(1) для отправки сигнала. Поскольку inetd(8) запускается из под root, нужно сначала использовать su(1) для получения прав root.

```
% su
Password:
# /bin/kill -s HUP 198
```

Как и большинство команд UNIX, kill(1) ничего не выведет, если отработает нормально. Если вы посылаете сигнал процессу, которым не владеете, на экране появится kill: PID: Operation not permitted. При неправильно набранном PID вы или отправите сигнал другому процессу, что

может привести к неприятностям, или, если повезет, сигнал будет отправлен на PID, который в данный момент не используется — на экране появится `kill: PID: No such process`.

**Зачем использовать `/bin/kill`?** Во многих оболочках команда `kill` встроена; таким образом, оболочка вместо вызова `/bin/kill` сама посылает сигнал. Это может быть очень полезно, но в разных оболочках имя сигнала указывается по-разному. Чем пытаться выучить их все, гораздо проще использовать `/bin/kill` ... непосредственно.

Отправка других сигналов очень похожа, просто замените `TERM` или `KILL` в командной строке на имя другого сигнала.

**Важно:** Уничтожение процессов наугад может быть плохой идеей. В частности, `init(8)`, чей PID 1, имеет особое значение. Выполнение `/bin/kill -s KILL 1` — быстрый способ перегрузить систему. *Всегда* дважды проверяйте параметры запуска `kill(1)` *перед* тем, как нажать **Enter**.

## 3.9. Интерпретатор команд

При работе с FreeBSD, в большинстве случаев для выполнения повседневных задач используется командный интерфейс (так называемая "оболочка", "shell"). Основная задача интерпретатора — принимать вводимые команды и выполнять их. Многие командные интерпретаторы имеют встроенные средства для выполнения повседневной работы, например, операции над файлами и каталогами, редактирование командной строки, командные макросы и переменные окружения. Вместе с FreeBSD поставляется несколько командных интерпретаторов, например, `sh`, или Bourne Shell, и `tcsh`, расширенная версия C-shell. Многие другие интерпретаторы доступны из коллекции портов FreeBSD, например `zsh` и `bash`.

Какой из командных интерпретаторов использовать? Это дело вкуса. Если вы программируете на C, то вам, возможно, понравится `tcsh`. Если вы работали с Linux, или только начинаете работать с интерфейсом командной строки UNIX, попробуйте `bash`. Каждый из названных интерпретаторов имеет свои особенности, которые отличат его от других, и, возможно, повлияют на ваш выбор.

Одна из наиболее часто используемых функций командного интерпретатора — дополнение частичного имени файла до полного. Вы можете набрать только первые несколько символов имени файла, нажать клавишу табуляции (TAB), и командный интерпретатор автоматически завершит имя. Например, у нас есть два файла, названные `foobar` и `foo.bar`. Допустим, мы хотим удалить файл `foo.bar`. Для этого, наберем на клавиатуре `rm fo[Tab].[Tab]`.

Вы увидите следующее: `rm foo[БЕЕР].bar`.

Здесь [БЕЕР] — это так называемый консольный сигнал, оповещающий о том, что интерпретатор не в состоянии закончить имя файла, так как по введенным вами символам невозможно однозначно идентифицировать файл. Например, имена файлов `foobar` и `foo.bar` оба начинаются с `fo`, но после нажатия TAB можно однозначно дополнить только до `foo`. Если же теперь ввести точку (.) и вновь нажать TAB, интерпретатор достроит имя файла целиком.

Дополнительные возможности при работе с интерпретатором дает использование переменных окружения. Переменные окружения это пары переменная=значение, хранящиеся в памяти интерпретатора. Значение переменных окружения может быть прочитано любой программой, запущенной из командного интерпретатора, и часто содержит настройки для многих приложений и утилит. Ниже приведены некоторые наиболее часто встречающиеся переменные окружения и их значения:

Переменная	Описание
USER	Имя текущего пользователя.
PATH	Каталоги, разделенные двоеточием, для поиска исполняемых файлов.
DISPLAY	Сетевое имя виртуального дисплея X11, доступного для подключения.
SHELL	Текущий командный интерпретатор.
TERM	Название (тип) терминала. Используется, чтобы узнать возможности терминала.
TERMCAP	Список escape-последовательностей для управления различными функциями терминала.
OSTYPE	Название (тип) операционной системы. Например, FreeBSD.
MACHTYPE	Архитектура машины (процессора).
EDITOR	Выбранный пользователем текстовый редактор.
PAGER	Выбранная пользователем утилита просмотра файлов.
MANPATH	Каталоги, разделенные двоеточием, для поиска файлов системного справочника.

Установка значений переменных окружения различна для разных оболочек. Например, в интерпретаторах C-стиля, таких как `tcsh` и `csh`, это `setenv`. В интерпретаторах Bourne, таких как `sh` и `bash`, это `export`. Например, чтобы установить или изменить значение переменной `EDITOR` к значению `/usr/local/bin/emacs` в `csh` или `tcsh`, выполните команду:

```
% setenv EDITOR /usr/local/bin/emacs
```

В оболочках Bourne:

```
% export EDITOR="/usr/local/bin/emacs"
```

Чтобы получить значение переменной, например, в командной строке, поместите символ `$` перед именем переменной. Например, команда `echo $TERM` выведет значение переменной `$TERM`.

Командный интерпретатор воспринимает некоторые символы, называемые метасимволами, в качестве управляющих. Один из наиболее часто используемых – символ `*`, который заменяет любое количество символов в имени файла. Метасимволы используются для поиска файлов по маске, например, команда `echo *` делает практически тоже самое, что и команда `ls`, поскольку интерпретатор вызывает команду `echo`, передавая ей имена всех файлов, попадающих под маску `*`.

В некоторых ситуациях требуется, чтобы интерпретатор воспринимал метасимволы как обычные, не несущие специальной смысловой нагрузки. Этого можно достичь, поставив перед символом обратную

косую черту (\). Например, команда `echo $TERM` выведет тип вашего терминала, в то же время команда `echo \ $TERM` выведет именно слово `$TERM`, а не значение переменной `$TERM`.

### 3.9.1. Как изменить командный интерпретатор по умолчанию

Самым простым способом, пожалуй, будет воспользоваться командой `chsh`. Если переменная `EDITOR` определена, то будет загружен соответствующий текстовый редактор, иначе `vi`. Вам нужно будет изменить значение поля "Shell:" и выйти из редактора с сохранением результатов.

Можно также воспользоваться опцией `-s` команды `chsh`. Например, если вы хотите изменить интерпретатор на `bash`, выполните:

```
% chsh -s /usr/local/bin/bash
```

Также можно запустить `chsh` без параметров и изменить интерпретатор по умолчанию в редакторе.

**Замечание:** Интерпретатор, который вы будете использовать, *обязательно* должен быть в файле `/etc/shells`. Обычно, при установке интерпретаторов из коллекции портов, это делается автоматически. Если же это не так, вам нужно будет самостоятельно добавить соответствующую строку в этот файл.

Например, если вы установили `bash` вручную и поместили его в каталог `/usr/local/bin`, нужно набрать:

```
# echo "/usr/local/bin/bash" >> /etc/shells
```

Теперь можно смело использовать команду `chsh`.

## 3.10. Текстовые редакторы

Большинство настроек в FreeBSD производится путем редактирования текстовых файлов. Соответственно, вам нужно будет освоиться с каким-либо текстовым редактором. Вместе с FreeBSD поставляются лишь некоторые из них, гораздо больше редакторов доступно из коллекции портов.

Самым простым в изучении и использовании, по-видимому, можно назвать **ee**, что расшифровывается как "easy editor", т.е. "простой редактор". Чтобы начать редактировать какой-либо файл, наберите в командной строке `ee filename`, где `filename` имя редактируемого файла. Например, для редактирования файла `/etc/rc.conf`, наберите `ee /etc/rc.conf`. В верхней части экрана вы увидите список основных команд редактора. Символ каретки (^) означает клавишу **Ctrl**, таким образом, ^e означает комбинацию клавиш **Ctrl+e**. Чтобы выйти из редактора, нажмите клавишу **Esc**, затем **Enter**. Если остались какие-либо не сохраненные данные, вам потребуется подтвердить выход, сохранив результат работы или оставив файл без изменения.

В FreeBSD присутствует также более мощный текстовый редактор **vi**, а редакторы **emacs** и **vim** можно найти в коллекции портов (`editors/emacs` и `editors/vim`). Эти редакторы более сложны в изучении, но их возможности порой превосходят всякие ожидания! Если в будущем вам потребуется часто редактировать большие объемы текстов, то потраченное на изучение этих редакторов время окупится с лихвой.

## 3.11. Устройства и файлы устройств

Термин "устройство" используется в основном по отношению к аппаратному обеспечению системы, такому как диски, принтеры, графические адаптеры, устройства ввода текста. При загрузке FreeBSD главным образом выводит на экран информацию об обнаруженных устройствах. Вы можете найти эти сообщения в файле `/var/run/dmesg.boot`.

Например, `acd0` это первый диск IDE CDROM, а `kbd0` — клавиатура.

В UNIX доступ к большинству этих устройств можно получить через специальные файлы устройств, расположенные в каталоге `/dev`.

### 3.11.1. Создание файлов устройств

При добавлении в систему нового устройства, или добавление поддержки дополнительных устройств, может понадобиться создать один или несколько файлов устройств для нового оборудования.

#### 3.11.1.1. Скрипт MAKEDEV

В системах без DEVFS (это относится ко всем версиям FreeBSD ниже 5.0), файлы устройств создаются с помощью MAKEDEV(8), как показано ниже:

```
# cd /dev
# sh MAKEDEV ad1
```

В этом примере создается соответствующий файл устройства для вторичного IDE диска.

#### 3.11.1.2. DEVFS (DEVIce File System)

Device filesystem, или DEVFS, предоставляет доступ к пространству устройств ядра через общую файловую систему. Вместо создания и модификации файлов устройств, DEVFS создает специальную файловую систему.

Обращайтесь к `devfs(5)` за дополнительной информацией.

В FreeBSD 5.0 и выше DEVFS используется по умолчанию.

## 3.12. Бинарные форматы

Для понимания того, почему FreeBSD использует формат `elf(5)`, вам потребуется сначала немного узнать о трех "доминирующих" исполняемых форматах для UNIX:

- `a.out(5)`

Старейший и "классический" объектный формат UNIX. Он использует короткий и компактный заголовок с магическим числом в начале, которое часто используется для описания формата (смотрите `a.out(5)` с более подробной информацией). Он содержит три загружаемых сегмента: `.text`, `.data` и `.bss` плюс таблицу символов и таблицу строк.

- COFF

Объектный формат SVR3. Заголовок включает таблицу разделов, так что могут быть сегменты кроме `.text`, `.data` и `.bss`.

- elf(5)

Наследник формата COFF, поддерживающий множественные сегменты и 32-битные или 64-битные значения. Одно важное замечание: ELF был разработан в предположении что есть только по одному ABI на одну архитектуру. Это предположение совершенно неверно, и не только в мире коммерческих SYSV (в котором есть как минимум три ABI: SVR4, Solaris, SCO).

FreeBSD пытается обойти эту проблему, в частности предоставляя утилиту для *оглавления* известного исполняемого файла ELF информацией об ABI с которым он совместим. Обратитесь к странице справочника `brandelf(1)` за более подробной информацией.

FreeBSD имеет произошла из "классического" лагеря и использовала формат `a.out(5)`, технологию опробованную и проверенную на многих поколениях релизов BSD, до начала ветки 3.X. Хотя собирать и запускать родные бинарные файлы ELF (и ядро) в системе FreeBSD можно было несколько раньше, FreeBSD вначале сопротивлялась "проталкиванию" ELF как формата по умолчанию. Почему? Когда лагерь Linux производил болезненный переход к ELF, у него не было большого преимущества перед исполняемым форматом `a.out`, из-за негибкого, основанного на таблице переходов механизма разделяемых библиотек, что делало создание разделяемых библиотек очень трудным для поставщиков и разработчиков. Когда доступные инструменты ELF предоставили решение проблемы разделяемых библиотек, и появилась некоторая перспектива, цена перехода была признана допустимой и он был сделан. Механизм разделяемых библиотек FreeBSD близок по стилю к механизму разделяемых библиотек SunOS™ от Sun, и поэтому очень прост в использовании.

Итак, почему так много разных форматов?

Давно, в темном далеком прошлом, оборудование было простым. Это простое оборудование поддерживало простые, маленькие системы. `a.out` был совершенно адекватен задаче представления бинарных файлов на таких простых системах (PDP-11). Люди, портировавшие UNIX с этих простых систем, оставили `a.out` формат потому, что он был достаточен для ранних портов UNIX на архитектуры, подобные Motorola 68k, VAXen, etc.

Затем какой-то смелый инженер по оборудованию решил, что если он сможет заставить программы исполнять некоторые трюки, то сможет несколько упростить дизайн и заставить ядро CPU работать быстрее. Хотя это было сделано с новым типом оборудования (известного сейчас как RISC), формат `a.out` не подходил для него, и было разработано множество форматов, чтобы получить лучшую производительность на таком оборудовании по сравнению с той, которую мог предоставить простой формат `a.out`. Были изобретены форматы COFF, ECOFF и некоторые другие малоизвестные форматы, и их ограничения были учтены, когда все похоже остановилось на ELF.

Кроме того, размеры программ стали огромны, а диски (и оперативная память) остались относительно малы, поэтому появилась концепция разделяемых библиотек. Система VM также стала более сложной. Хотя все эти усовершенствования были выполнены с форматом `a.out`, его полезность все больше и больше уменьшалась с каждым нововведением. К тому же потребовалась динамическая загрузка во время выполнения, или выгрузка частей программы после выполнения стартового кода для экономии памяти или места на диске. Языки усложнились, и потребовался автоматический вызов кода перед главной программой. Множество изменений было внесено в формат `a.out`, чтобы все это появилось, и

в основном работало некоторое время. Настал момент, когда `a.out` не смог решить все эти проблемы без чрезмерного увеличения размера и сложности. В то время, как ELF решил многие из этих проблем, перевод этого формата с системы на систему болезненен. Поэтому формату ELF пришлось подождать, пока не стало более болезненным оставаться с `a.out`, чем перейти на ELF.

Тем временем, инструменты разработки, от которых произошли инструменты разработки FreeBSD (особенно ассемблер и загрузчик), развивались в двух параллельных направлениях. Направление FreeBSD добавило разделяемые библиотеки и устранило некоторые ошибки. Люди из GNU, написавшие эти программы, переписали их и добавили простую поддержку сборки кросскомпиляторов, подключения различных форматов в будущем и так далее. Многим требовалось собрать кросскомпиляторы для FreeBSD, и это не удалось, поскольку устаревшие исходные тексты FreeBSD для `as` и `ld` не подходили для этой задачи. Новый набор инструментов GNU (**binutils**) поддерживает кросскомпилирование, ELF, разделяемые библиотеки, C++, расширения и т.д. В дополнение, многие поставщики выпустили программы в формате ELF и они хорошо подходят для запуска в FreeBSD.

ELF более выразителен, чем `a.out`, позволяет базовой системе быть более гибкой. ELF лучше поддерживается, и предоставляет поддержку кросскомпиляторов, что важно для многих людей. ELF может быть немного медленнее, чем `a.out`, но замерить это сложно. Есть также множество деталей, отличающихся для этих двух форматов, в том как они отображают страницы, обрабатывают начальный код, и т.д. В этом нет ничего очень важного, но они различаются. В настоящее время поддержка `a.out` убрана из ядра GENERIC, и со временем будет убрана из ядра, как только потребность в запуске старых программ `a.out` останется в прошлом.

## 3.13. Дополнительная информация

### 3.13.1. Системный справочник (man)

Пожалуй, самым полным руководством по FreeBSD является системный справочник (man). Практически каждое приложение или утилита имеют соответствующую страницу (часто не одну), описывающую тот или иной аспект работы программы, всевозможные опции и настройки. Для просмотра этих страниц существует команда `man`:

```
% man command
```

Здесь `command` – это команда, о которой вы хотите получить информацию. Например, чтобы узнать побольше о команде `ls`, наберите:

```
% man ls
```

Содержимое системного справочника для удобства разделено на несколько разделов:

1. Пользовательские команды.
2. Системные вызовы и коды ошибок.
3. Функции стандартных библиотек.
4. Драйверы устройств.
5. Форматы файлов.



6. Развлечения и игры.
7. Дополнительная информация.
8. Команды системного администрирования.
9. Для разработчиков ядра.

В некоторых случаях (не так уж редко), страницы, относящиеся к одной и той же команде, находятся в различных разделах справочника. Например, есть команда `chmod` и системный вызов `chmod( )`. В этом случае, необходимо явно указать раздел `man`, в котором нужно искать соответствующую страницу:

```
% man 1 chmod
```

Эта команда выведет справку об утилите `chmod`. По традиции, конкретный раздел справочника указывается в скобках после команды, например, `chmod(1)` относится к утилите `chmod`, а `chmod(2)` – к соответствующему системному вызову.

Часто бывает так, что вы не знаете название команды, но имеете представление о том, что она должна делать. В этом случае можно попытаться найти нужную команду по ключевым словам, встречающимся в ее описании, используя опцию `-k` программы `man`:

```
% man -k mail
```

Вы получите список команд, имеющих слово "mail" в своих описаниях. Это эквивалентно использованию команды `apropos`.

Или например, вы видите список файлов в каталоге `/usr/bin`, при этом не имея ни малейшего представления о том, какие функции выполняет каждый из них? Просто наберите:

```
% cd /usr/bin
% man -f *
```

или

```
% cd /usr/bin
% whatis *
```

что фактически одно и то же.

### 3.13.2. Файлы GNU Info

FreeBSD поставляется с многочисленными приложениями и утилитами от Фонда Свободного Программного Обеспечения, Free Software Foundation (FSF). В дополнение к страницам справочника, с этими программами поставляется обширная гипертекстовая документация в виде так называемых `info` файлов, которые могут быть просмотрены с помощью команды `info`, или, если установлен **emacs**, в `info` режиме этого редактора.

Чтобы воспользоваться командой `info(1)`, просто наберите в командной строке:

```
% info
```

Вызвать на экран краткое введение можно набрав `h`. Краткий список команд можно получить набрав `?`.

## Примечания

1. Это следует из обозначения `i386`. Обратите внимание, что даже если FreeBSD работает не на процессоре Intel 386, это обозначение будет `i386`. Здесь показан не тип процессора, а его "архитектура".
2. Стартовые скрипты — это программы, запускаемые автоматически во время загрузки FreeBSD. Их основная задача — подготовить систему к запуску других программ и запустить настроенные сервисы, которые будут работать в фоновом режиме, выполняя полезную работу.
3. Подробное техническое описание деталей работы драйверов консолей FreeBSD и клавиатуры можно найти в `syscons(4)`, `atkbd(4)`, `vidcontrol(1)` и `kbdcontrol(1)`. Здесь мы не раскрываем деталей, однако интересующиеся этим вопросом всегда могут обратиться к страницам справочника за более подробным объяснением того, как это работает.
4. Не совсем верно — есть несколько действий, которые не могут быть прерваны. Например, если процесс пытается прочитать файл на другом компьютере в сети, и другой компьютер по какой-то причине не отвечает (был выключен, или в сети произошла ошибка), процесс находится в так называемом "непрерываемом состоянии". В конце концов время ожидания закончится, обычно это происходит через две минуты. Как только время закончится, процесс будет уничтожен.

# Глава 4. Установка приложений: порты и пакеты

*Перевод на русский язык: Андрей Захватов.*

## 4.1. Обзор

Вместе с FreeBSD в составе базового комплекта системы поставляется богатый набор системный утилит. Однако для выполнения какой-то реальной работы очень скоро возникает необходимость в установке дополнительных приложений сторонних разработчиков. FreeBSD даёт две взаимодополняющих технологии для установки программного обеспечения сторонних разработчиков в вашу систему: Коллекция Портов FreeBSD и бинарные пакеты с программным обеспечением. Любая из этих систем может быть использована для установки самых последних версий ваших любимых приложений с локальных носителей или прямо из сети.

После чтения этой главы вы будете знать:

- Как устанавливать бинарные пакеты с программным обеспечением сторонних разработчиков.
- Как строить программное обеспечение сторонних разработчиков при помощи коллекции портов.
- Как удалять ранее установленные пакеты или порты.
- Как переопределить значения, используемые по умолчанию в коллекции портов.
- Как обновить ваши порты.

## 4.2. Обзор установки программного обеспечения

Если вы использовали UNIX-системы ранее, то знаете, что стандартная процедура установки программного обеспечения сторонних разработчиков выглядит примерно так:

1. Загрузка программного обеспечения, которое может распространяться в форме исходных текстов или двоичных файлов.
2. Распаковка программного обеспечения из дистрибутивного формата (обычно tar-архива, сжатого при помощи `compress(1)`, `gzip(1)` или `bzip2(1)`).
3. Поиск документации (возможно, подойдут файлы `INSTALL`, `README` или несколько файлов из подкаталога `doc/`) и её чтение в поиске описания установки программного обеспечения.
4. Если программное обеспечение распространялось в форме исходных текстов, его компиляция. Сюда может быть включено редактирование файла `Makefile`, запуск скрипта `configure` и другие работы.
5. Тестирование и установка программного обеспечения.

И это только всё проходит нормально. Если вы устанавливаете программный пакет, который был специально перенесён на FreeBSD, то вам может даже потребоваться редактировать код для того, чтобы он нормально заработал.

Если вы хотите, то можете продолжать устанавливать программное обеспечение во FreeBSD "традиционным" способом. Однако FreeBSD предоставляет две технологии, которые могут сохранить вам много усилий: пакеты и порты. На момент написания таким образом были доступны более 11,800 сторонних приложений.

Для любого конкретно взятого приложения пакет FreeBSD для такого приложения является одним файлом, который вы должны загрузить. Пакет содержит уже откомпилированные копии всех команд приложения, а также все конфигурационные файлы и документацию. Загруженным файлом пакета можно управлять командами управления пакетами FreeBSD, такими, как `pkg_add(1)` `pkg_delete(1)`, `pkg_info(1)` и так далее. Установка нового приложения может выполняться единственной командой.

Порт FreeBSD для приложения является набором файлов, предназначенных для автоматизации процесса компиляции приложения из исходного кода.

Вспомните, что обычно вы должны выполнить некоторое количество шагов, если компилируете программу самостоятельно (загрузка, распаковка, изменение кода, компиляция, установка). Файлы, составляющие порт, содержат всю информацию, необходимую для того, чтобы система сделала это за вас. Вы задаёте пару простых команд, и исходный код приложения автоматически загружается, распаковывается, модифицируется, компилируется и устанавливается.

Действительно, система портов может также использоваться для генерации пакетов, которые позже могут управляться командой `pkg_add` и другими командами управления пакетами, о которых скоро будет рассказано.

Как пакеты, так и порты принимают во внимание *зависимости*. Предположим, что вы хотите установить приложение, которое зависит от некоторой установленной библиотеки. И приложение, и библиотека доступны во FreeBSD в виде портов и пакетов. Если вы используете команду `pkg_add` или систему портов для добавления приложений, то в обоих случаях будет обнаружено, что библиотека не была установлена, и сначала будет автоматически выполнена установка библиотеки.

Видя, что обе технологии весьма похожи, вы можете удивиться, почему во FreeBSD используются обе. И пакеты, и порты имеют свои преимущества, так что выбор используемой вами системы зависит от ваших собственных предпочтений.

### Преимущества пакетов

- Сжатый tar-архив обычно меньше, чем сжатый tar-архив, содержащий исходный код приложения.
- Пакеты не требуют никакой дополнительной компиляции. Для таких больших приложений, как **Mozilla**, **KDE** или **GNOME**, это может быть важно, в частности, если вы работаете на медленной системе.
- Пакеты не требуют понимания процесса компиляции программного обеспечения во FreeBSD.

### Преимущества портов

- Пакеты обычно компилируются с консервативными параметрами, потому что они должны работать на максимальном количестве систем. При установке из порта вы можете изменять параметры компиляции для того, чтобы (к примеру) генерировался код, специфичный для процессора Pentium IV или Athlon.
- Некоторые приложения имеют опции времени компиляции, связанные с тем, что они могут или не могут делать. К примеру, **Apache** может быть настроен с широким набором различных опций. При построении из порта вы можете не принимать параметры по умолчанию, и задать их самостоятельно.

В некоторых случаях для одного и того же приложения будут иметься несколько пакетов для указания конкретных настроек. Например, **Ghostscript** имеется как пакет `ghostscript` и как пакет `ghostscript-nox11`, в зависимости от того, установили вы сервер X11 или нет. Такой тип грубой настройки возможен при использовании пакетов, но быстро становится недостижим, если приложение имеет более одного или двух параметров времени компиляции.

- Условия лицензирования некоторых дистрибутивов программного обеспечения запрещает распространение в двоичном виде. Они должны распространяться в виде исходного кода.
- Некоторые не доверяют дистрибутивам в двоичном виде. При использовании исходного кода вы по крайней мере (теоретически) можете прочесть его и попытаться найти потенциальные проблемы самостоятельно.
- Если у вас есть собственные патчи, вам нужен исходный код для того, чтобы их применять.
- Некоторым нравится иметь исходный код, чтобы его можно было посмотреть и хакать, заимствовать из него (конечно, при разрешающей лицензионном соглашении) и тому подобное.

Чтобы отслеживать обновления портов, подпишитесь на Список рассылки, посвящённый Портam FreeBSD (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports>) и Список рассылки, посвящённый ошибкам в портах FreeBSD (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports-bugs>).

**Внимание:** Перед установкой любого приложения необходимо зайти на <http://vuxml.freebsd.org/>, где находится информация по вопросам безопасности приложений.

Вы можете также установить `security/portaudit`, который автоматически проверит все установленные приложения на наличие известных уязвимостей, проверка также будет выполняться перед сборкой какого-либо порта. Вы можете использовать `portaudit -F -a` и после установки пакетов.

В оставшейся части главы будет рассказано, как использовать пакеты и порты для установки и управления программным обеспечением сторонних разработчиков во FreeBSD.

### 4.3. Поиск нужного вам приложения

Перед тем, как устанавливать какое-либо приложение, вам нужно знать, что вы хотите и как называется нужное вам приложение.

Список имеющихся для FreeBSD приложений постоянно растёт. К счастью, есть несколько способов найти то, что вам нужно:

- На сайте FreeBSD поддерживается обновляемый список имеющихся приложений для FreeBSD, в котором можно выполнять поиск, по адресу <http://www.FreeBSD.org/ports/../../ports/index.html>. Порты разбиты на категории, и вы можете либо выполнить поиск приложения по имени (если его знаете), либо просмотреть список всех приложений, относящихся к определённой категории.
- Dan Langille поддерживает сайт FreshPorts по адресу <http://www.FreshPorts.org/>. На нём отслеживаются изменения в приложениях из дерева портов, как только они происходят, он позволяет вам "отслеживать" один или несколько портов, и может высылать оповещение по электронной почте при их обновлении.

- Если вы не знаете названия нужного вам приложения, попробуйте воспользоваться сайтом типа FreshMeat (<http://www.freshmeat.net/>) для поиска приложения, а затем возвратитесь на сайт FreeBSD, чтобы проверить, есть ли порт для этого приложения.
- Если вы знаете точное имя порта, и хотите определить, в какой категории он находится, используйте команду `whereis(1)`. Просто наберите в приглашении `"whereis file"`, где *file* - программа, которую вы хотите установить. И если она имеется в системе, об этом будет сообщено, как показано ниже:

```
# whereis lsof
lsof: /usr/ports/sysutils/lsof
```

Это говорит о том, что `lsof` (системная утилита) находится в каталоге `/usr/ports/sysutils/lsof`.

- Ещё одним способом поиска некоторого порта является использование встроенной возможности поиска в коллекции портов. Чтобы ею воспользоваться, вы должны находиться в каталоге `/usr/ports`. Очутившись в этом каталоге, выполните команду `make search name=program-name`, где *program-name* – это название программы, которую вы хотите найти. Например, если вы ищете `lsof`:

```
# cd /usr/ports
# make search name=lsof
Port:    lsof-4.56.4
Path:    /usr/ports/sysutils/lsof
Info:    Lists information about open files (similar to fstat(1))
Maint:   obrien@FreeBSD.org
Index:   sysutils
B-deps:
R-deps:
```

Вам следует обратить особое внимание на строчку `"Path:"`, так как в ней указывается, где найти порт. Остальная сообщаемая информация для установки порта не нужна, поэтому здесь она описываться не будет.

Для выполнения более глубокого поиска вы можете также использовать `make search key=string`, где *string* представляет собой некоторый текст, который ищется. При этом будет выполнен поиск в именах портов, комментариях, описаниях и зависимостях, и его можно использовать для поиска портов, связанных с некоторой темой, если вы не знаете названия программы, которую вы ищете.

В обоих этих случаях строка поиска нечувствительна к регистру. Поиск `"LSOF"` приводит к тому же самому результату, что и поиск `"lsof"`.

## 4.4. Использование системы пакетов

*Текст предоставил Chern Lee.*

### 4.4.1. Установка пакета

Для установки пакетов программного обеспечения для FreeBSD из локальных файлов или с сервера в сети вы можете использовать утилиту `pkg_add(1)`.

**Пример 4-1. Загрузка пакета вручную и его локальная установка**

```
# ftp -a ftp2.FreeBSD.org
Connected to ftp2.FreeBSD.org.
220 ftp2.FreeBSD.org FTP server (Version 6.00LS) ready.
331 Guest login ok, send your email address as password.
230-
230-      This machine is in Vienna, VA, USA, hosted by Verio.
230-      Questions? E-mail freebsd@vienna.verio.net.
230-
230-
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /pub/FreeBSD/ports/packages/sysutils/
250 CWD command successful.
ftp> get lsof-4.56.4.tgz
local: lsof-4.56.4.tgz remote: lsof-4.56.4.tgz
200 PORT command successful.
150 Opening BINARY mode data connection for 'lsof-4.56.4.tgz' (92375 bytes).
100% |*****| 92375          00:00 ETA
226 Transfer complete.
92375 bytes received in 5.60 seconds (16.11 KB/s)
ftp> exit
# pkg_add lsof-4.56.4.tgz
```

Если у вас нет исходных текстов локальных пакетов (например, набор CD-ROM с FreeBSD), то проще всего, наверное, воспользоваться опцией `-r` для `pkg_add(1)`. Это приведёт к тому, что утилита автоматически определит правильный формат объектных файлов и релиз, а затем загрузит и установит пакет с сервера FTP.

```
# pkg_add -r lsof
```

В примере выше нужный пакет будет загружен и установлен без всякого дополнительного взаимодействия с пользователем. Если вместо основного сайта вы хотите указать другое зеркало пакетов FreeBSD, то для переопределения используемых по умолчанию значений вам необходимо задать соответствующим образом значение переменной `PACKAGESITE`. Для загрузки файлов утилита `pkg_add(1)` использует функцию `fetch(3)`, которая принимает во внимание различные переменные окружения, включая `FTP_PASSIVE_MODE`, `FTP_PROXY` и `FTP_PASSWORD`. Если вы находитесь за сетевым экраном или для работы с FTP/HTTP вам необходимо использовать прокси, то определите соответствующие переменные. Обратитесь к справочной странице по `fetch(3)` для получения полного списка переменных. Заметьте, что в примере выше вместо `lsof-4.56.4` используется `lsof`. При использовании функций загрузки с сети номер версии в имени пакета должен быть опущен. Утилита `pkg_add(1)` автоматически загрузит последнюю версию приложения.

Файлы пакетов распространяются в форматах `.tgz` и `.tbz`. Вы можете найти их по адресу `ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/packages` или взять с дистрибутива FreeBSD на CD-ROM. Каждый CD из комплекта FreeBSD на 4 дисках (а также PowerPak и тому подобное) содержит пакеты в каталоге `/packages`. Расположение пакетов похоже на то, как организовано дерево `/usr/ports`. Каждая категория имеет собственный каталог, и каждый пакет помещается в каталог `All`.

Структура каталогов системы пакетов соответствует структуре системы портов; они взаимодействуют друг с другом для формирования единой системы пакетов/портов.

#### 4.4.2. Управление пакетами

`pkg_info(1)` является утилитой для вывода списка и описаний различных установленных пакетов.

```
# pkg_info
cvsup-16.1      A general network file distribution system optimized for CV
docbook-1.2     Meta-port for the different versions of the DocBook DTD
...
```

`pkg_version(1)` является утилитой для вывода отчёта о версиях всех установленных пакетов. Она сравнивает версию имеющегося пакета с текущей версией, находящейся в дереве портов.

```
# pkg_version
cvsup           =
docbook         =
...
```

Символы во второй колонке указывают сравнительную разницу в возрасте установленной версии и версии, находящейся в локальном дереве портов.

Символ	Значение
=	Версия установленного пакета соответствует версии, находящейся в локальном дереве портов.
<	Установленная версия старше, чем та, что имеется в дереве портов.
>	Установленная версия новее чем та, что есть в дереве портов. (Скорее всего, локальное дерево портов устарело.)
?	В индексном файле портов установленный пакет не может быть найден. (Это может случиться, например, если установленный порт был удалён из коллекции портов или переименован.)
*	Имеется несколько версий пакета.

#### 4.4.3. Удаление пакета

Для удаления ранее установленных пакетов с программным обеспечением используйте утилиту `pkg_delete(1)`.

```
# pkg_delete xchat-1.7.1
```



#### 4.4.4. Разное

Вся информация о пакете хранится в каталоге `/var/db/pkg`. Список установленных файлов и описания всех пакетов могут быть найдены среди файлов этого каталога.

## 4.5. Использование Коллекции Портов

В этих разделах описаны основные приемы использования коллекции портов для установки и удаления программ из вашей системы.

### 4.5.1. Получение Коллекции Портов

Перед тем, как вы сможете устанавливать порты, вы сначала должны установить коллекцию портов—она представляет собой в основном набор файлов `Makefile`, патчей, файлов описаний, и она размещается в каталоге `/usr/ports`.

При установке вашей системы FreeBSD утилита **sysinstall** запрашивает, требуется ли вам установка коллекции портов. Если вы ответили отрицательно, то можете следовать следующим указаниям для установки коллекции портов:

#### Метод с использованием Sysinstall

В этом методе повторно используется **sysinstall** (`/stand/sysinstall` в FreeBSD версий ниже, чем 5.2) для ручной установки коллекции портов.

1. Работая как пользователь `root`, запустите `sysinstall` так, как это показано ниже:

```
# sysinstall
```

2. Опуститесь вниз и выберите **Configure**, нажмите **Enter**
3. Опуститесь вниз и выберите **Distributions**, затем нажмите **Enter**
4. Опуститесь вниз к пункту **ports**, нажмите клавишу **Пробел**
5. Поднимитесь вверх к **Exit**, нажмите **Enter**
6. Выберите желаемый носитель для установки, например, **CDROM**, **FTP** и так далее.
7. Перейдите на пункт меню **Exit** и нажмите **Enter**.
8. Нажмите **X** для выхода из **sysinstall**.

Альтернативный метод получения и постоянной актуализации вашей коллекции портов заключается в использовании **CVSup**. Посмотрите на файл **CVSup** из портов, `/usr/share/examples/cvsup/ports-supfile`. Прочтите раздел **Использование CVSup** (Разд. А.5) для получения более полной информации об использовании **CVSup** и этого файла.

#### Метод с использованием CVSup

Это быстрый способ для получения коллекции портов при помощи **CVSup**. Если вы хотите поддерживать ваше дерево портов в актуальном состоянии, или узнать больше о **CVSup**, то прочтите вышеотмеченные разделы.

1. Установите порт `net/cvsup`. Обратитесь к разделу Установка CVSup (Разд. А.5.2) для получения более подробной информации.
2. Работая как пользователь `root`, скопируйте `/usr/share/examples/cvsup/ports-supfile` в новое место, например, в каталог `/root` или в ваш домашний каталог.
3. Отредактируйте `ports-supfile`.
4. Измените `CHANGE_THIS.FreeBSD.org` на близкий к вам сервер **CVSup**. Посмотрите Зеркала CVSup (Разд. А.5.7) для получения полного списка зеркалирующих сайтов.
5. Запустите `cvsup`  

```
# cvsup -g -L 2 /root/ports-supfile
```
6. При повторных запусках этой команды все последние изменения (кроме реального перестроения портов для вашей системы) будут загружаться и переноситься в вашу коллекцию портов.

## 4.5.2. Установка портов

Когда речь заходит о коллекции портов, то первым делом вы должны понять, что именно подразумевается под словом "скелет". По сути скелетом порта является минимальный набор файлов, который указывает вашей системе FreeBSD, как корректно откомпилировать и установить программу. Скелет каждого порта включает:

- `Makefile`. `Makefile` содержит различные директивы, которые определяют, как приложение должно быть откомпилировано и куда в вашей системе оно должно быть установлено.
- Файл `distinfo`. Этот файл содержит информацию о файлах, которые должны существовать на вашей системе для успешной сборки порта, и их контрольные суммы, для проверки при помощи `md5(1)` того, что файлы в процессе загрузки не были повреждены.
- Каталог `files`. Этот каталог содержит патчи, использование которых необходимо для компиляции и установки программы в вашей системе FreeBSD. Патчи - это, как правило, маленькие файлы, в которых содержатся изменения, которые нужно внести в какой-то конкретный файл. Они имеют обычный текстовый формат и в основном содержат указания типа "Удалить строку 10" или "Заменить строку 26 на такую ...". Патчи также называются "diff-файлами" или просто "диффами", потому что они генерируются программой `diff(1)`.

Этот каталог также может содержать другие файлы, используемые для построения порта.

- `pkg-descr`. Это более подробное, зачастую многострочное описание программы.
- `pkg-plist`. Это список всех файлов, которые будут установлены портом. В нем также содержатся указания системе портов на удаление определенных файлов во время удаления порта.

В некоторых портах присутствуют и другие файлы, такие, как `pkg-message`. Система портов использует эти файлы для обработки особых ситуаций. Если вы хотите узнать более подробно об этих файлах и о портах вообще, то обратитесь к Руководству по созданию портов для FreeBSD ([http://www.FreeBSD.org/doc/ru\\_RU.KOI8-R/books/porters-handbook/index.html](http://www.FreeBSD.org/doc/ru_RU.KOI8-R/books/porters-handbook/index.html)).

В порт включаются инструкции относительно того, как выполнять построение из исходного кода, но не сам код. Вы можете получить исходный код с CD-ROM или из Интернет. Исходный код

распространяется в том виде, какой предпочёл выбрать разработчик. Зачастую это tar-файл, обработанный утилитой `gzip`, но он может также быть упакованным каким-то другим инструментом или быть не сжатым. Исходный код программы, в каком бы то виде он ни был, называется "дистрибутивным" (`distfile`). Два метода установки портов FreeBSD описаны ниже.

**Замечание:** Для установки портов вы должны войти в систему как пользователь `root`.

**Внимание:** Перед установкой любого порта необходимо убедиться в наличии свежей коллекции портов и заглянуть на <http://vuxml.freebsd.org/>, где могут освещаться вопросы безопасности, связанные с этим портом.

Проверка на наличие уязвимостей может быть автоматически выполнена **portaudit** перед установкой нового приложения. Эту программу можно найти в коллекции портов (`security/portaudit`). Запустите `portaudit -F` перед установкой нового порта для загрузки свежей базы данных уязвимостей. Проверка безопасности и обновление базы данных должны выполняться при повседневной проверке безопасности системы. За дальнейшей информацией обращайтесь к страницам справочника `portaudit(1)` и `periodic(8)`.

#### 4.5.2.1. Установка портов с CD-ROM

Официальные образы CD-ROM Проекта FreeBSD больше не содержат дистрибутивные файлы. На них есть достаточно места, которое лучше использовать для предкомпилированных пакетов. Продукты на CD-ROM, такие, как FreeBSD PowerPak, включают дистрибутивные файлы, и вы можете заказать их от таких поставщиков, как FreeBSD Mall (<http://www.freebsdmail.com/>). В этом разделе предполагается, что у вас есть такой набор CD-ROM с FreeBSD.

Вставьте ваш CD-ROM с FreeBSD в привод. Смонтируйте его в каталог `/cdrom`. (Если вы используете другую точку монтирования, то установка работать не будет.) Чтобы начать, перейдите в каталог с портом, который вы хотите установить:

```
# cd /usr/ports/sysutils/lsof
```

Оказавшись в каталоге `lsof`, вы увидите структуру порта. Следующим шагом является компиляция, или построение ("build") порта. Это выполняется простой выдачей команды `make` в ответ на приглашение командного процессора. Как только вы это сделаете, то увидите выдачу, выглядящую примерно так:

```
# make
>> lsof_4.57D.freebsd.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
>> Attempting to fetch from file:/cdrom/ports/distfiles/.
==> Extracting for lsof-4.57
...
[выдача при распаковке опущена]
...
>> Checksum OK for lsof_4.57D.freebsd.tar.gz.
==> Patching for lsof-4.57
==> Applying FreeBSD patches for lsof-4.57
==> Configuring for lsof-4.57
...
[выдача при конфигурации опущена]
...
```

```

==> Building for lsof-4.57
...
[выдача при компиляции опущена]
...
#

```

Заметьте, что, как только компиляция закончится, вы снова вернётесь к приглашению вашего командного процессора. Следующим шагом является установка порта. Чтобы это сделать, вам нужно просто добавить одно слово к команде `make`, а именно слово `install`:

```

# make install
==> Installing for lsof-4.57
...
[выдача при установке опущена]
...
==> Generating temporary packing list
==> Compressing manual pages for lsof-4.57
==> Registering installation for lsof-4.57
==> SECURITY NOTE:
      This port has installed the following binaries which execute with
      increased privileges.
#

```

Как только вы вернулись к приглашению вашей оболочки, вы должны суметь запустить приложение, которое только что установили. Так как `lsof` является программой, которая запускается с повышенными правами, то выдаётся предупреждение о безопасности. Во время построения и установки портов вы должны принимать во внимание все выдаваемые предупреждения.

**Замечание:** Вы можете сэкономить лишний шаг, просто выдав команду `make install` вместо `make` и последующей `make install` в двух отдельных шагах.

**Замечание:** Некоторые командные процессоры для ускорения поиска выполнимых файлов и команд кэшируют имена таких программ, которые доступны для вызова из каталогов, перечисленных в переменной окружения `PATH`. Если вы используете один из таких командных процессоров, то перед использованием только что добавленных команд вам может понадобиться вызвать директиву `rehash` после установки порта. Это верно как для оболочек, являющихся частями базового системного набора (например, `tcsh`), так и командных процессоров, доступных в виде портов (например, `shells/zsh`).

**Замечание:** Пожалуйста, будьте готовы к тому, что лицензии некоторых портов не позволяют помещать их на CD-ROM. Это может делаться из-за обязательности заполнения регистрационной формы перед загрузкой или запрета на дальнейшее распространение, либо по какой-то другой причине. Если вы хотите установить порт, которого нет на CD-ROM, вам нужно иметь подключение к Интернет, чтобы это сделать (обратитесь к следующему разделу).

#### 4.5.2.2. Установка портов из Интернет

Как и в предыдущем разделе, здесь предполагается, что у вас имеется работающее подключение к Интернет. Если это не ваш случай, вам нужно выполнять установку с CD-ROM либо поместить копию дистрибутивного файла в каталог `/usr/ports/distfiles` вручную.

Установка порта из Интернет производится точно так же, как если бы делали её с CD-ROM. Единственным отличием между ними является тот факт, что дистрибутивный файл загружается из Интернет, а не считывается с CD-ROM.

Выполняются те же самые шаги:

```
# make install
>> lsof_4.57D.freebsd.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
>> Attempting to fetch from ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/.
Receiving lsof_4.57D.freebsd.tar.gz (439860 bytes): 100%
439860 bytes transferred in 18.0 seconds (23.90 kBps)
==> Extracting for lsof-4.57
...
[выдача при распаковке опущена]
...
>> Checksum OK for lsof_4.57D.freebsd.tar.gz.
==> Patching for lsof-4.57
==> Applying FreeBSD patches for lsof-4.57
==> Configuring for lsof-4.57
...
[выдача процесса конфигурации опущена]
...
==> Building for lsof-4.57
...
[выдача компиляции опущена]
...
==> Installing for lsof-4.57
...
[выдача процедуры установки опущена]
...
==> Generating temporary packing list
==> Compressing manual pages for lsof-4.57
==> Registering installation for lsof-4.57
==> SECURITY NOTE:
      This port has installed the following binaries which execute with
      increased privileges.
#
```

Как вы видите, единственным отличием является строка, в которой указывается, откуда система загружает дистрибутивный файл порта.

Для загрузки файлов система портов использует утилиту `fetch(1)`, которая принимает во внимание различные переменные окружения, включая `FTP_PASSIVE_MODE`, `FTP_PROXY` и `FTP_PASSWORD`. Если вы находитесь за сетевым экраном или для работы с работой с FTP/HTTP вам необходимо использовать прокси, то определите соответствующие переменные. Обратитесь к справочной странице по `fetch(3)` для получения полного списка переменных.

Пользователям, которые не могут быть постоянно подключенными к сети, поможет команда `make fetch`. Просто запустите эту команду в каталоге самого верхнего уровня (`/usr/ports`), и требуемые файлы будут загружены. Эта команда будет работать также и с вложенными категориями, например: `/usr/ports/net`. Заметьте, что если порт имеет зависимости от библиотек или других портов, то он *не будет* также загружать дистрибутивные файлы этих портов. Замените `fetch` на `fetch-recursive`, если вы хотите выполнить загрузку также и всех этих зависимостей порта.

**Замечание:** Вы можете построить все порты в категории за раз, запустив команду `make` в каталоге верхнего уровня, как и в вышеописанном методе с `make fetch`. Однако это опасно, так как некоторые порты не могут сосуществовать. В других случаях некоторые порты могут устанавливать два различных файла с одним и тем же именем.

В некоторых редких случаях пользователям необходимо получить tar-архивы с сайтов, отличающихся от `MASTER_SITES` (это место, откуда файлы обычно загружаются). Вы можете переопределять значение `MASTER_SITES` посредством следующей команды:

```
# cd /usr/ports/directory
# make MASTER_SITE_OVERRIDE= \
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/ fetch
```

В этом примере мы изменили значение переменной `MASTER_SITES` на `ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/`.

**Замечание:** Некоторые порты позволяют (или даже требуют) задавать параметры, которые включают или выключают построение отдельных частей приложения, которые не нужны, некоторые параметры безопасности, а также прочие настройки. К некоторым из них, часто упоминаемым, относятся пакеты `www/mozilla`, `security/gpgme` и `mail/sylpheed-claws`. При наличии подобных параметров будет выдано сообщение.

#### 4.5.2.3. Переименование каталогов с портами по умолчанию

Иногда бывает полезным (или необходимым) использование других каталогов с портами и дистрибутивными файлами. Для переопределения каталогов, используемых по умолчанию, используются переменные `PORTSDIR` и `PREFIX`. К примеру:

```
# make PORTSDIR=/usr/home/example/ports install
```

будет компилировать порт в каталоге `/usr/home/example/ports`, а установит всё в `/usr/local`.

```
# make PREFIX=/usr/home/example/local install
```

будет компилировать его в каталоге `/usr/ports`, а установит в `/usr/home/example/local`.

И, конечно,

```
# make PORTSDIR=../ports PREFIX=../local install
```

использует обе возможности (полная команда слишком длинна для написания на этой странице, однако вы должны уловить общую идею).

Альтернативным способом является задание значений этих переменных в параметрах вашего окружения. Обратитесь к страницам справки по вашему командному процессору для получения инструкций о том, как это сделать.

#### 4.5.2.4. Работа с `imake`

Некоторые порты, использующие `imake` (часть X Window System) нормально не работают с `PREFIX`, и будут пытаться установиться в дерево `/usr/X11R6`. Аналогично некоторые Perl-порты игнорируют значения `PREFIX` и устанавливаются в дерево Perl. Заставить такие порты воспринимать переменную `PREFIX` является сложной или невыполнимой задачей.

### 4.5.3. Удаление установленных портов

Теперь, когда вы знаете, как устанавливать порты, вы, наверное, уже задумывались о том, как же их удалять, просто даже на тот случай, если вы установили один из них, а позже решили, что установили не тот порт. Мы удалим порт из нашего предыдущего примера (а это была программа `lsof`, если вы обратили внимание). Как и при установке портов, первым делом вы должны перейти в каталог с портом, `/usr/ports/sysutils/lsof`. После смены каталога вы готовы к удалению `xchat`. Это делается командой `make deinstall`:

```
# cd /usr/ports/sysutils/lsof
# make deinstall
==> Deinstalling for lsof-4.57
```

Это было достаточно легко. Вы удалили `lsof` из вашей системы. Если вам захочется переустановить эту программу, то это можно будет сделать, выдав команду `make reinstall` из каталога `/usr/ports/sysutils/lsof`.

Последовательность команд `make deinstall` и `make reinstall` не сработает, если вы запустите команду `make clean`. Если вы хотите убрать порт из системы после зачистки каталога, используйте команду `pkg_delete(1)`, как это описано в разделе о пакетах Руководства.

### 4.5.4. Порты и дисковое пространство

Использование коллекции портов со временем требует все больше дискового пространства. По этой причине вы не должны забывать о очистке каталогов `work` при помощи команды `make clean`. При этом будет удаляться каталог `work`, возникающий после построения и установки порта. Вы можете также удалить исходные файлы программы из каталога `distfiles`, а позже удалить установленные порты, если они более не используются.

Некоторые пользователи ограничивают набор доступных категорий портов, помещая в файл `refuse` соответствующую запись. В этом случае, при запуске программы **CVSup** она не будет загружать файлы из указанных категорий. Дополнительную информацию, относящуюся к файлу `refuse`, можно найти в Разд. A.5.3.1.

### 4.5.5. Обновление портов

**Замечание:** После обновления коллекции портов, и перед тем, как обновить приложение из порта, сверьтесь с файлом `/usr/ports/UPDATING`. В нем дана информация по различным вопросам и дополнительным шагам, которые могут быть необходимы для обновления порта.

Поддержание ваших портов в актуальном состоянии может оказаться нудным занятием. К примеру, чтобы обновить порт, вам нужно перейти в каталог с портами, построить порт, удалить старый порт, установить новый, а затем вычистить его после построения. Представьте, сколько действий (пять) нужно сделать! Скучно, да? Это было большой проблемой для системных администраторов, но теперь есть утилиты, выполняющие эти действия за нас. Например, утилита `sysutils/portupgrade` делает всё за вас сама! Просто установите её, как обычный порт, при помощи команды `make install clean`.

Теперь создайте базу данных командой `pkgdb -F`. Она выяснит перечень установленных портов и создаст файл базы данных в каталоге `/var/db/pkg`. Теперь при вызове команды `portupgrade -a`, она будет использовать эту базу данных и файл портов `INDEX`. Наконец, **portupgrade** загрузит, построит, сделает резервную копию, установит и очистит порты, которые были обновлены. **portupgrade** имеет массу параметров для использования в разных ситуациях, из которых наибольшего внимания заслуживает описываемая далее.

Если вы хотите обновить только определённое приложение, но не полностью базу данных, используйте `portupgrade pkgname`, с флагом `-r`, если **portupgrade** должен отработать все пакеты, которые зависят от указанного, и использовать флаг `-R` для отработки всех пакетов, которые требуют указанные пакеты.

Для использования при установке пакетов, а не портов, укажите флаг `-P`. С этим параметром **portupgrade** будет искать пакеты в локальных каталогах, указанных в переменной окружения `PKG_PATH`, или, если не найдет их, загрузит с удаленного сайта. Если пакеты не могут быть найдены локально или загружены удаленно, **portupgrade** использует порты. Чтобы запретить использование портов, укажите `-PP`.

Для простой загрузки дистрибутивных файлов без построения или установки чего бы то ни было, задайте флаг `-F`. Дополнительную информацию можно получить на странице справки по утилите `portupgrade(1)`.

**Замечание:** Важно регулярно обновлять базу данных пакетов при помощи команды `pkgdb -F` для восстановления целостности, особенно когда **portupgrade** запрашивает вас это сделать. Не прерывайте работы **portupgrade** при обновлении базы данных пакетов, это может привести к рассогласованию данных.

Существуют и другие утилиты, которые делают подобные вещи, посмотрите в каталоге `ports/sysutils` и выберите то, что вам подходит.

## 4.6. Действия после установки

После установки нового приложения вам обычно требуется прочесть какую-либо документацию, если она есть, отредактировать нужные конфигурационные файлы, проверить, что приложение



запускается во время загрузки системы (если это демон), и так далее.

Очевидно, что в шаги, в точности требуемые для конфигурации каждого приложения, отличаются. Однако, если вы только что установили новое приложение и вам интересно, “Что же дальше?”, то вам могут помочь следующие советы:

- Воспользуйтесь командой `pkg_info(1)` для определения того, куда и какие файлы были установлены. К примеру, если вы только что установили FooPackage версии 1.0.0, то по команде

```
# pkg_info -l foopackage-1.0.0 | less
```

будут выведен список всех файлов, установленных пакетов. Обратите особое внимание на файлы в каталогах `man/`, которые являются справочными страницами, `etc/`, которые будут являться конфигурационными файлами, и `doc/`, которые будут являться более подробной документацией.

Если вы не уверены, какая версия приложения была только что установлена, то по команде типа

```
# pkg_info | grep -i foopackage
```

будут выведен список всех установленных пакетов, в названии которых присутствует `foopackage`. Замените `foopackage` в командной строке на то, что вам нужно.

- Как только вы определите, куда были установлены справочные страницы приложения, просмотрите их при помощи команды `man(1)`. Подобным же образом просмотрите примеры конфигурационных файлов и всю дополнительную документацию, которая была поставлена.
- Если у приложения имеется веб-сервер, поищите там дополнительную документацию, ответы на часто задаваемые вопросы и так далее. Если вы не уверены, каков адрес веб-сайта, он может быть указан в выводе команды

```
# pkg_info foopackage-1.0.0
```

Строка `www:`, если она есть, должна содержать URL Web-сайта приложения.

- Порты, которые должны запускаться при загрузке системы (такие, как серверы Internet), как правило, устанавливают примерный скрипт в каталог `/usr/local/etc/rc.d`. Вы должны просмотреть скрипт на предмет его корректности и отредактировать или переименовать его, если это нужно. Обратитесь к разделу о Запуске сервисов для получения более полной информации.

## 4.7. Обработка нерабочих портов

Если вы встретили порт, который у вас не работает, есть несколько вещей, которые вы можете сделать, а именно:

1. Исправьте его! В Руководстве по созданию портов ([http://www.FreeBSD.org/doc/ru\\_RU.KOI8-R/books/porters-handbook/index.html](http://www.FreeBSD.org/doc/ru_RU.KOI8-R/books/porters-handbook/index.html)) содержится подробная информация об инфраструктуре “Портов”, так что вы сможете исправить иногда встречающиеся порты с ошибками или даже создать собственные!
2. Надавите—*только по электронной почте*! Сначала пошлите письмо человеку, сопровождающему порт. Наберите команду `make maintainer` или прочтите `makefile`, чтобы найти его адрес. Не

забудьте указать имя и версию порта (скопировав строчку `$FreeBSD:` из файла `Makefile`) и включите в письмо весь вывод, предшествующий возникновению ошибки. Если вы не получили ответа от этого человека, то можете воспользоваться командой `send-pr(1)` для отправки сообщения об ошибке.

3. Загрузите пакет с ближайшего к вам FTP-сервера. "Основная" коллекция пакетов находится на сервере `ftp.FreeBSD.org` в каталоге с пакетами (`ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/packages/`), но *первым делом* проверьте местное зеркало! Это, скорее всего, будет работать, и сделать это, кроме того, будет гораздо быстрее, чем пытаться компилировать порты из исходного кода. Воспользуйтесь программой `pkg_add(1)` для установки пакета в вашей системе.

# Глава 5. X Window System

*Обновил для сервера X.Org's X11 Ken Tom, Marc Fonvieille. Перевод на русский язык: Андрей Захватов.*

## 5.1. Обзор

FreeBSD использует X11 для того, чтобы дать пользователям мощный графический интерфейс. X11 является открытой реализацией X Window System, включая **Xorg** и **XFree86**. В версиях FreeBSD до и включая FreeBSD 4.10-RELEASE и FreeBSD 5.2.1-RELEASE сервером X11 по умолчанию был **XFree86**, выпускаемый The XFree86 Project, Inc. Начиная с FreeBSD 5.3-RELEASE, официальной версией X11 по умолчанию стал **Xorg**, разработанный X.Org Foundation.

Эта глава посвящена установке и настройке X11 в системе FreeBSD, с акцентом на **Xorg**.

За дополнительной информацией по видео оборудованию, поддерживаемому X11, обратитесь к веб сайтам Xorg (<http://www.x.org/>) или XFree86 (<http://www.XFree86.org/>).

После чтения этой главы вы будете знать:

- Как установить и настроить X11.
- Различные компоненты X Window System и их взаимодействие.
- Как установить и использовать различные оконные менеджеры.
- Как использовать шрифты TrueType® в X11.
- Как настроить вашу систему на графический интерфейс входа (**XDM**).

Перед чтением этой главы вы должны:

- Знать, как устанавливать дополнительное программное обеспечение сторонних разработчиков (Гл. 4).

**Замечание:** В этой главе описана установка и настройка серверов X11 и **Xorg** и **XFree86**. По большей части файлы настройки, команды и синтаксис идентичны. Там, где есть различия, приводится синтаксис и **Xorg** и **XFree86**.

## 5.2. Основы X

Первое знакомство с X может оказаться чем-то вроде шока для тех, кто работал с другими графическими системами, такими, как Microsoft Windows или Mac OS.

Хотя нет необходимости вникать во все детали различных компонентов X и их взаимодействия, некоторые базовые знания делают возможным использование сильных сторон X.

### 5.2.1. Почему именно X?

X не является первой оконной системой для UNIX, но она является самой популярной из них. До работы над X команда ее разработчиков трудилась над другой оконной системой. Та система называлась "W" (от "Window"). X была просто следующей буквой в романском алфавите.

X можно называть "X", "X Window System", "X11" и множеством других терминов. Факт использования названия "X Windows" для X11 может задеть интересы некоторых людей; дополнительную информацию по этому поводу можно найти на странице справочной системы X(7).

### 5.2.2. Модель клиент/сервер в X

X изначально разрабатывалась, чтобы быть системой, ориентированной на работу в сети с использованием модели "клиент-сервер".

В модели работы X "X-сервер" работает на компьютере с клавиатурой, монитором и мышью. Ответственность сервера включает управление дисплеем, обработку ввода с клавиатуры и мыши и так далее. Каждое X-приложение (например, **XTerm** или **Netscape**) является "клиентом". Клиент посылает сообщения серверу, такие, как "Пожалуйста, нарисуй окно со следующими координатами", а сервер посылает в ответ сообщения типа "Пользователь только что щёлкнул мышью на кнопке OK".

В случае использования дома или в офисе, сервер и клиенты X как правило будут работать на том же самом компьютере. Однако реально возможно запускать X-сервер на менее мощном настольном компьютере, а приложения X (клиенты) на, скажем, мощной и дорогой машине, обслуживающей целый офис. В этом сценарии X-клиент и сервер общаются через сеть.

Некоторых это вводит в заблуждение, потому что терминология X в точности обратна тому, что они ожидают. Они полагают, что "X-сервер" будет большой мощной машиной, стоящей на полу, а "X-клиентом" является машина, стоящая на их столах.

Важно помнить, что X-сервером является машина с монитором и клавиатурой, а X-клиенты являются программами, выводящими окна.

В протоколе нет ничего, что заставляет машины клиента и сервера работать под управлением одной и той же операционной системы, или даже быть одним и тем же типом компьютера. Определённо возможно запускать X-сервер в Microsoft Windows или Mac OS от Apple, и есть множество свободно распространяемых и коммерческих приложений, которые это реализуют.

Начиная с FreeBSD 5.3-RELEASE, X-сервер, поставляемый с FreeBSD, называется **Xorg**, и он распространяется свободно под лицензией, очень похожей на условия распространения FreeBSD. Имеются и коммерческие X-серверы для FreeBSD.

### 5.2.3. Оконный менеджер

Философия построения X очень похожа на философию построения UNIX, "инструменты, не политика". Это значит, что X не пытаются диктовать то, как должна быть выполнена работа. Вместо этого пользователю предоставляются инструменты, а за пользователем остается принятие решения о том, как использовать эти инструменты.

Этот подход расширен в X тем, что не задается, как окна должны выглядеть на экране, как их двигать мышью, какие комбинации клавиш должны использоваться для переключения между окнами (то есть

**Alt+Tab**, в случае использования Microsoft Windows), как должны выглядеть заголовки окон, должны ли в них быть кнопки для закрытия, и прочее.

Вместо этого X делегирует ответственность за это приложению, которое называется "Window Manager" (Менеджер Окон). Есть десятки оконных менеджеров для X: **AfterStep**, **Blackbox**, **ctwm**, **Enlightenment**, **fvwm**, **Sawfish**, **twm**, **WindowMaker** и другие. Каждый из этих оконных менеджеров предоставляет различные внешние виды и удобства; некоторые из них поддерживают "виртуальные рабочие столы"; некоторые из них позволяют изменять назначения комбинаций клавиш, используемых для управления рабочим столом; в некоторых есть кнопка "Start" или нечто подобное; некоторые поддерживают "темы", позволяя изменять внешний вид, поменяв тему. Эти оконные менеджеры, а также множество других, находятся в категории `x11-wm` коллекции Портов.

Кроме того, оболочки **KDE** и **GNOME** обе имеют собственные оконные менеджеры, которые интегрированы с оболочкой.

Каждый оконный менеджер также имеет собственный механизм настройки; некоторые предполагают наличие вручную созданного конфигурационного файла; некоторые предоставляют графические инструменты для выполнения большинства работ по настройке; по крайней мере один (**Sawfish**) имеет конфигурационный файл, написанный на диалекте языка Lisp.

**Политика фокусирования:** Другой особенностью, за которую отвечает оконный менеджер, является "политика фокусирования" мыши. Каждая оконная система должна иметь некоторый способ выбора окна для активации получения нажатий клавиш, а также визуальную индикацию того, какое окно активно.

Широкоизвестная политика фокусировки называется "click-to-focus". Эта модель используется в Microsoft Windows, когда окно становится активным после получения щелчка мыши.

X не поддерживает никакой конкретной политики фокусирования. Вместо этого менеджер окон управляет тем, какое окно владеет фокусом в каждый конкретный момент времени. Различные оконные менеджеры поддерживают разные методы фокусирования. Все они поддерживают метод щелчка для фокусирования, и большинство из них поддерживают некоторые другие методы.

Самыми популярными политики фокусирования являются:

**focus-follows-mouse** (фокус следует за мышью)

Фокусом владеет то окно, что находится под указателем мыши. Это не обязательно будет окно, которое находится поверх всех остальных. Фокус меняется при указании на другое окно, при этом также нет нужды щёлкать на нём.

**sloppy-focus** (нечеткий фокус)

С политикой **focus-follows-mouse** если мышь помещается поверх корневого окна (или заднего фона), то никакое окно фокус не получает, а нажатия клавиш просто пропадают. При использовании политики нечёткого фокуса он меняется только когда курсор попадает на новое окно, но не когда уходит с текущего окна.

**щелчок для выбора фокуса**

Активное окно выбирается щелчком мыши. Затем окно может быть "поднято" и появиться поверх всех других окон. Все нажатия клавиш теперь будут направляться в это окно, даже если курсор переместится к другому.

Многие оконный менеджер поддерживают и другие политики, а также вариации перечисленных. Обязательно обращайтесь к документации по оконному менеджеру.

### 5.2.4. Виджеты

Подход X, заключающийся в предоставлении инструментов, а не политики, распространяется и на виджеты, которые располагаются на экране в каждом приложении.

”Виджет” (widget) является термином для всего в пользовательском интерфейсе, на чём можно щёлкать или каким-то образом управлять; кнопки, зависимые (radio buttons) и независимые (check boxes) кнопки, иконки, списки и так далее. В Microsoft Windows это называется ”элементами управления” (”controls”).

Microsoft Windows и Mac OS от Apple обе имеют очень жёсткую политику относительно виджетов. Предполагается, что разрабатываемые приложения обязательно должны иметь похожий внешний вид. Что касается X, то было решено, что не нужно требовать обязательного использования какого-то определённого графического стиля или набора виджетов.

В результате не стоит ожидать от X-приложений похожести во внешнем виде. Существует несколько популярных наборов виджетов и их разновидностей, включая оригинальный набор виджетов Athena от MIT, **Motif®** (по образу которого был разработан набор виджетов в Microsoft Windows, все эти скошенные углы и три разновидности серого цвета), **OpenLook** и другие.

В большинстве появляющихся в настоящее время приложений для X будет использоваться современно выглядящий набор виджетов, либо Qt, используемый в **KDE**, либо GTK+, используемый проектом **GNOME**. В этом отношении наблюдается унификация внешнего вида рабочего стола в UNIX, что определённо облегчает жизнь начинающему пользователю.

## 5.3. Установка X11

На FreeBSD могут быть установлены **Xorg** или **XFree86**. Начиная с FreeBSD 5.3-RELEASE, **Xorg** является для FreeBSD версией X11 по умолчанию. **Xorg** это сервер X дистрибутива открытой реализации X Window System, выпущенной X.Org Foundation. **Xorg** основан на коде **XFree86 4.4RC2** и X11R6.6. X.Org Foundation выпустил X11R6.7 в апреле 2004 года, а X11R6.8.1 в сентябре 2004; эта версия доступна из коллекции портов FreeBSD.

Для сборки и установки **Xorg** из коллекции портов, выполните:

```
# cd /usr/ports/x11/xorg
# make install clean
```

**Замечание:** Перед сборкой полной версии **Xorg** удостоверьтесь в наличии хотя бы 4 GB свободного места.

Для сборки и установки **XFree86** из коллекции портов:

```
# cd /usr/ports/x11/XFree86-4
# make install clean
```

Кроме того, X11 может быть установлен непосредственно из пакетов. Бинарные пакеты, устанавливаемые pkg\_add(1), доступны и для X11. Когда pkg\_add(1) используется для удаленной загрузки пакетов, номер версии пакета необходимо удалить. pkg\_add(1) автоматически установит последнюю версию приложения.

Таким образом, для загрузки и установки пакета **Xorg**, просто наберите:

```
# pkg_add -r xorg
```

Пакет **XFree86 4.X** может быть установлен командой:

```
# pkg_add -r XFree86
```

**Замечание:** В примерах выше будет установлен полный дистрибутив X11, включая серверы, клиенты, шрифты и так далее. Также доступны и отдельные пакеты и порты для различных частей X11.

В оставшейся части главы будет рассказано о том, как сконфигурировать X11 и настроить рабочее окружение.

### 5.3.1. Замена XFree86™ на Xorg

Как и с любым портом, вам необходимо проверить файл `/usr/ports/UPDATING` на наличие изменений. Инструкции по переходу с **XFree86** на **Xorg** включены в этот файл.

Используйте **CVSup** для обновления дерева портов перед любой переустановкой. Для замены X11 вам также потребуется установить `sysutils/portupgrade`.

В файл `/etc/make.conf` необходимо добавить переменную `X_WINDOW_SYSTEM=xorg`. Это необходимо, чтобы система знала, какой X11 используется. Старая переменная `XFREE86_VERSION` не используется, она заменена переменной `X_WINDOW_SYSTEM`.

Затем используйте следующие команды:

```
# pkg_delete -f /var/db/pkg/imate-4* /var/db/pkg/XFree86-*
# cd /usr/ports/x11/xorg
# make install clean
# pkgdb -F
```

Команда `pkgdb(1)` является частью программы **portupgrade**, она обновит различные зависимости пакетов.

**Замечание:** Перед сборкой полной версии **Xorg** удостоверьтесь в наличии хотя бы 4 GB свободного места.

## 5.4. Конфигурация X11

*Текст предоставил Christopher Shumway.*

### 5.4.1. Перед тем, как начать

Перед настройкой X11 необходима следующая информация о конфигурируемой системе:

- Характеристики монитора
- Набор микросхем, используемый в видеоадаптере
- Объём видеопамати

Характеристики монитора используются в X11 для определения рабочего разрешения и частоты. Эти характеристики обычно могут быть получены из документации, которая прилагается к монитору или с сайта производителя. Тут нужны два диапазона значений, для частоты горизонтальной развёртки и для частоты вертикальной синхронизации.

Набор микросхем графического адаптера определяет, модуль какого драйвера использует X11 для работы с графическим оборудованием. Для большинства типов микросхем это может быть определено автоматически, но все же его полезно знать на тот случай, когда автоматическое определение не работает правильно.

Объём видеопамати графического адаптера определяет разрешение и глубину цвета, с которым может работать система. Это важно, чтобы пользователь знал ограничения системы.

### 5.4.2. Конфигурирование X11

Процесс настройки X11 является многошаговым. Первый шаг заключается в построении начального конфигурационного файла. Работая с правами суперпользователя, просто запустите:

```
# Xorg -configure
```

Для **XFree86** запустите:

```
# XFree86 -configure
```

При этом в каталоге `/root` будет создан скелет конфигурационного файла X11 под именем `xorg.conf.new` (там, куда после `su(1)` или непосредственного входа будет указывать переменная `$HOME`). Для **XFree86**, этот файл называется `XF86Config.new`. Программа X11 сделает попытку распознать графическое оборудование системы и запишет конфигурационный файл, загружающий правильные драйверы для обнаруженного оборудования в системе.

Следующим шагом является тестирование существующей конфигурации для проверки того, что **Xorg** может работать с графическим оборудованием в настраиваемой системе. Для этого выполните:

```
# Xorg -config xorg.conf.new
```

Пользователям **XFree86** необходимо выполнить:

```
# XFree86 -xf86config XF86Config.new
```

Если появилась чёрно-белая сетка и курсор мыши в виде X, то настройка была выполнена успешно. Для завершения тестирования просто нажмите одновременно **Ctrl+Alt+Backspace**.

**Замечание:** Если мышь не работает, его необходимо настроить. Обратитесь к Разд. 2.9.10 в главе об установке FreeBSD.



Теперь выполните тонкую настройку в файле `xorg.conf.new` по своему вкусу (или `XF86Config.new`, если вы работаете с **XFree86**). Откройте файл в текстовом редакторе, таком, как `emacs(1)` или `ee(1)`. Сначала задайте частоты для монитора. Они обычно обозначаются как частоты горизонтальной и вертикальной синхронизации. Эти значения добавляются в файл `XF86Config.new` в раздел "Monitor":

```
Section "Monitor"
    Identifier      "Monitor0"
    VendorName      "Monitor Vendor"
    ModelName       "Monitor Model"
    HorizSync       30-107
    VertRefresh     48-120
EndSection
```

Ключевых слов `HorizSync` и `VertRefresh` может и не оказаться в файле конфигурации. Если их нет, то они должны быть добавлены, с указанием корректных значений горизонтальной частоты синхронизации после ключевого слова `HorizSync` и вертикальной частоты синхронизации после ключевого слова `VertRefresh`. В примере выше были введены частоты монитора настраиваемой системы.

X позволяет использовать возможности технологии DPMS (Energy Star) с поддерживающими её мониторами. Программа `xset(1)` управляет временными задержками и может явно задавать режимы ожидания, останова и выключения. Если вы хотите включить использование возможностей DPMS вашего монитора, вы должны добавить следующую строку в раздел, описывающий монитор:

```
Option      "DPMS"
```

Пока файл конфигурации `xorg.conf.new` (или `XF86Config.new`) открыт в редакторе, выберите желаемые разрешение и глубину цвета, которые будут использоваться по умолчанию. Они задаются в разделе "Screen":

```
Section "Screen"
    Identifier "Screen0"
    Device     "Card0"
    Monitor    "Monitor0"
    DefaultDepth 24
    SubSection "Display"
        Viewport 0 0
        Depth    24
        Modes     "1024x768"
    EndSubSection
EndSection
```

Ключевое слово `DefaultDepth` описывает глубину цвета, с которой будет работа по умолчанию. Это значение может быть переопределено при помощи параметра командной строки `-depth` для `Xorg(1)` (или `XFree86(1)`). Ключевое слово `Modes` описывает разрешение, с которым нужно работать при данной глубине цвета. Заметьте, что поддерживаются только те стандартные режимы VESA, что определены графическим оборудованием настраиваемой системы. В примере выше глубина цвета по умолчанию равна двадцати четырём битам на пиксел. При такой глубине цвета принимается разрешение в 1024 на 768 точек.

Наконец, запишите конфигурационный файл и протестируйте его при помощи тестового режима, описанного выше.

**Замечание:** При решении проблем могут помочь лог файлы X11, в которых находится информация по каждому устройству, к которому подключен сервер X11. Лог файлам **Xorg** названия даются в формате `/var/log/Xorg.0.log` (лог файлам **XFree86** названия даются в формате `XFree86.0.log`). Имена лог файлов могут даваться от `Xorg.0.log` до `Xorg.8.log` и так далее.

Если все в порядке, то конфигурационный файл нужно установить в общедоступное место, где его сможет найти Xorg(1) (или XFree86(1)). Обычно это `/etc/X11/xorg.conf` или `/usr/X11R6/etc/X11/xorg.conf` (для **XFree86** это `/etc/X11/XF86Config` или `/usr/X11R6/etc/X11/XF86Config`).

```
# cp xorg.conf.new /etc/X11/xorg.conf
```

Для **XFree86**:

```
# cp XF86Config.new /etc/X11/XF86Config
```

Теперь процесс настройки X11 завершен. Для запуска **XFree86 4.X** посредством `startx(1)` установите порт `x11/wrapper`. В **Xorg** уже включен код `wrapper`, и установка его из порта не требуется. X11 можно также запустить через `xdm(1)`.

**Замечание:** Имеется также графический инструмент для настройки, `xorgcfg(1)` (`xf86cfg(1)` для **XFree86**), который включён в дистрибутив X11. Он позволяет выполнить настройку в интерактивном режиме посредством выбора соответствующих драйверов и настроек. Эта программа может быть запущена в консоли командой `xf86cfg -textmode`. Для получения более полной информации обратитесь к странице справочной системы `xorgcfg(1)` или `xf86cfg(1)`.

Кроме того, существует программа настройки `xorgconfig(1)` (`xf86config(1)` для **XFree86**), это консольная утилита, которая менее дружелюбна к пользователю, но может работать в ситуациях, в которых другие утилиты не работают.

### 5.4.3. Тонкие вопросы настройки

#### 5.4.3.1. Конфигурирование при работе с графическими чипсетами Intel® i810

Конфигурирование при работе с интегрированными наборами микросхем Intel i810 требует наличия `agpgart`, программного интерфейса AGP, посредством которого X11 будет управлять адаптером. Драйвер `agp(4)` присутствует в ядре `GENERIC` с момента выпуска 4.8-RELEASE и 5.0-RELEASE. Для предшествующих релизов вам нужно добавлять такую строку:

```
device agp
```

в конфигурационный файл вашего ядра и перестраивать новое ядро. Однако вместо этого вы можете подгружать модуль ядра `agp.ko` автоматически во время загрузки системы при помощи `loader(8)`. Для этого просто добавьте следующую строку в файл `/boot/loader.conf`:

```
agp_load="YES"
```

Затем, в случае использования FreeBSD 4.X или более ранних её версий, для программного интерфейса должен быть создан файл устройств. Для создания файла устройств для AGP запустите MAKEDEV(8) в каталоге /dev:

```
# cd /dev
# sh MAKEDEV agpgart
```

**Замечание:** Во FreeBSD 5.X и более поздних версиях будет использоваться devfs(5) для выделения файлов устройств в прозрачном режиме, поэтому шаг с MAKEDEV(8) не нужен.

Это позволит конфигурировать графическое оборудование точно так же, как и любой другой графический адаптер. Заметьте, что для систем, у которых драйвер agr(4) в ядро не вкомпилирован, попытка погрузить модуль с помощью kldload(8) окончится неудачно. Этот драйвер должен оказаться в ядре во время загрузки, либо вкомпилированным, либо подгруженным посредством /boot/loader.conf.

Если вы используете **XFree86 4.1.0** (или более позднюю версию), и выдаются сообщения о неразрешённых ссылках типа fbPictureInit, попробуйте добавить такую строчку после Driver "i810" в конфигурационном файле X11:

```
Option "NoDDC"
```

## 5.5. Использование шрифтов в X11

*Текст предоставил Murray Stokely.*

### 5.5.1. Шрифты Type1

Шрифты, используемые по умолчанию и распространяемые вместе с X11, вряд ли можно назвать идеально подходящими для применения в обычных издательских приложениях. Большие презентационные шрифты выглядят рвано и непрофессионально, а мелкие шрифты в **Netscape** вообще невозможно разобрать. Однако есть некоторое количество свободно распространяемых высококачественных шрифтов Type1 (PostScript®), которые можно без изменений использовать с X11. К примеру, в наборе шрифтов URW (x11-fonts/urwfonts) имеются высококачественные версии стандартных шрифтов type1 (Times Roman®, Helvetica®, Palatino® и другие). В набор Freefonts (x11-fonts/freefonts) включено ещё больше шрифтов, однако большинство из них предназначено для использования в программном обеспечении для работы с графикой, например, **Gimp**, и они не вполне пригодны для использования в качестве экранных шрифтов. Кроме того, X11 с минимальными усилиями может быть настроена на использование шрифтов TrueType. Более детальная информация находится на странице справочной системы X(7) и в разделе о шрифтах TrueType ниже.

Для установки вышеупомянутых коллекций шрифтов Type1 из коллекции портов выполните следующие команды:

```
# cd /usr/ports/x11-fonts/urwfonts
# make install clean
```

То же самое нужно будет сделать для коллекции **freefont** и других. Чтобы X-сервер обнаруживал этих шрифты, добавьте соответствующую строку в файл настройки X сервера `/etc/X11/(xorg.conf` для **Xorg** и `XF86Config` для **XFree86**), которая должна выглядеть так:

```
FontPath "/usr/X11R6/lib/X11/fonts/URW/"
```

Либо из командной строки при работе с X выполните:

```
% xset fp+ /usr/X11R6/lib/X11/fonts/URW
% xset fp rehash
```

Это сработает, но будет потеряно, когда сеанс работы с X будет закрыт, если эта команда не будет добавлена в начальный файл (`~/.xinitrc` в случае обычного сеанса через `startx` или `~/.xsession` при входе через графический менеджер типа **XDM**). Третий способ заключается в использовании нового файла `/usr/X11R6/etc/fonts/local.conf`: посмотрите раздел об антиалиасинге.

## 5.5.2. Шрифты TrueType®

Как в **XFree86 4.X**, так и в **Xorg** имеется встроенная поддержка шрифтов TrueType. Имеются два модуля, которые могут обеспечить эту функциональность. В нашем примере используется модуль **freetype**, потому что он в большей степени похож на другие механизмы для работы с шрифтами. Для включения модуля **freetype** достаточно в раздел "Module" файла `/etc/X11/xorg.conf` или `/etc/X11/XF86Config` добавить следующую строчку.

```
Load "freetype"
```

В случае **XFree86 3.3.X** требуется отдельный сервер шрифтов TrueType. Для этого обычно используется **Xfstt**. Для установки **Xfstt** просто установите порт `x11-servers/Xfstt`.

Теперь создайте каталог для шрифтов TrueType (к примеру, `/usr/X11R6/lib/X11/fonts/TrueType`) и скопируйте все шрифты TrueType в этот каталог. Имейте в виду, что напрямую использовать шрифты TrueType с Macintosh® нельзя; для использования с X11 они должны быть в формате UNIX/MS-DOS/Windows. После того, как файлы будут скопированы в этот каталог, воспользуйтесь утилитой **ttmkfdir** для создания файла `fonts.dir`, который укажет подсистеме вывода шрифтов X на местоположение этих новых файлов. **ttmkfdir** имеется в Коллекции Портов FreeBSD как `x11-fonts/ttmkfdir`.

```
# cd /usr/X11R6/lib/X11/fonts/TrueType
# ttmkfdir > fonts.dir
```

После этого добавьте каталог со шрифтами TrueType к маршруту поиска шрифтов. Это делается точно также, как описано выше для шрифтов Type1, то есть выполните

```
% xset fp+ /usr/X11R6/lib/X11/fonts/TrueType
% xset fp rehash
```

или добавьте строку `FontPath` в файл `xorg.conf` (или `XF86Config`).

Это всё. Теперь **Netscape**, **Gimp**, **StarOffice™** и все остальные X-приложения должны увидеть установленные шрифты TrueType. Очень маленькие (как текст веб-страницы на дисплее с высоким разрешением) и очень большие (в **StarOffice**) шрифты будут теперь выглядеть гораздо лучше.

### 5.5.3. Антиалиасинг шрифтов

*Обновление выполнил Joe Marcus Clarke.*

Антиалиасинг присутствует в X11 начиная с **XFree86**, версии 4.0.2. Однако настройка шрифтов была довольно громоздка вплоть до появления **XFree86** 4.3.0. Начиная с версии **XFree86** 4.3.0, все шрифты, расположенные в каталогах `/usr/X11R6/lib/X11/fonts/` и `~/.fonts/`, автоматически становятся доступными для применения антиалиасинга в приложениях, использующих Xft. Не все приложения могут использовать Xft, но во многих его поддержка присутствует. Примерами приложений, использующих Xft, является Qt версий 2.3 и более поздних (это инструментальный пакет для оболочки **KDE**), GTK+ версий 2.0 и более поздних (это инструментальный пакет для оболочки **GNOME**), а также **Mozilla** версий 1.2 и более поздних.

Для управления тем, к каким шрифтам применять антиалиасинг, а также для настройки параметров антиалиасинга, создайте (или отредактируйте, если он уже существует) файл `/usr/X11R6/etc/fonts/local.conf`. Некоторые мощные возможности системы шрифтов Xft могут быть настроены при помощи этого файла; в этом разделе описаны лишь некоторые простые возможности. Для выяснения всех деталей, пожалуйста, обратитесь к `fonts-conf(5)`.

Этот файл должен быть сформирован в формате XML. Обратите особое внимание на регистр символов, и удостоверьтесь, что все тэги корректно закрыты. Файл начинается обычным заголовком XML, за которым следуют задание DOCTYPE, а потом тэг `<fontconfig>`:

```
<?xml version="1.0"?>
<!DOCTYPE fontconfig SYSTEM "fonts.dtd">
<fontconfig>
```

Как и говорилось ранее, все шрифты из каталога `/usr/X11R6/lib/X11/fonts/`, а также `~/.fonts/` уже доступны для приложений, использующих Xft. Если вы хотите добавить каталог, отличный от этих двух, добавьте строчку, подобную следующей, в файл `/usr/X11R6/etc/fonts/local.conf`:

```
<dir>/path/to/my/fonts</dir>
```

После добавления новых шрифтов, и особенно новых каталогов со шрифтами, вы должны выполнить следующую команду для перестроения кэшей шрифтов:

```
# fc-cache -f
```

Антиалиасинг делает границы несколько размытыми, что делает очень мелкий текст более читабельным и удаляет "лесенки" из текста большого размера, но может вызвать нечёткость при применении к тексту обычного размера. Для исключения размеров шрифтов, меньших 14, из антиалиасинга, добавьте такие строки:

```
<match target="font">
  <test name="size" compare="less">
    <double>14</double>
  </test>
  <edit name="antialias" mode="assign">
    <bool>>false</bool>
  </edit>
</match>
<match target="font">
```

```

    <test name="pixelsize" compare="less" qual="any">
      <double>14</double>
    </test>
    <edit mode="assign" name="antialias">
      <bool>>false</bool>
    </edit>
  </match>

```

Для некоторых моноширинных шрифтов антиалиасинг может также оказаться неприменимым при определении межсимвольного интервала. В частности, эта проблема возникает с **KDE**. Одним из возможных решений для этого является жесткое задание межсимвольного интервала в 100. Добавьте следующие строки:

```

<match target="pattern" name="family">
  <test qual="any" name="family">
    <string>fixed</string>
  </test>
  <edit name="family" mode="assign">
    <string>mono</string>
  </edit>
</match>
<match target="pattern" name="family">
  <test qual="any" name="family">
    <string>console</string>
  </test>
  <edit name="family" mode="assign">
    <string>mono</string>
  </edit>
</match>

```

(это создаст алиасы для других общеупотребительных имён для шрифтов фиксированного размера как "mono"), а затем добавьте:

```

<match target="pattern" name="family">
  <test qual="any" name="family">
    <string>mono</string>
  </test>
  <edit name="spacing" mode="assign">
    <int>100</int>
  </edit>
</match>

```

С некоторыми шрифтами, такими, как **Helvetica**, могут получиться проблемы при антиалиасинге. Обычно это проявляется в виде шрифта, который наполовину вертикально обрезан. Хуже того, это может привести к сбоям таких приложений, как **Mozilla**. Во избежание этого следует добавить следующее в файл `local.conf`:

```

<match target="pattern" name="family">
  <test qual="any" name="family">
    <string>Helvetica</string>
  </test>
  <edit name="family" mode="assign">
    <string>sans-serif</string>
  </edit>
</match>

```

```
</edit>
</match>
```

После того, как вы закончите редактирование `local.conf`, удостоверьтесь, что вы завершили файл тэгом `</fontconfig>`. Если этого не сделать, то ваши изменения будут проигнорированы.

Набор шрифтов по умолчанию, поставляемый с X11, не очень желателен, если включается антиалиасинг. Гораздо лучший набор шрифтов, используемых по умолчанию, можно найти в порте `x11-fonts/bitstream-vera`. Этот порт установит файл `/usr/X11R6/etc/fonts/local.conf`, если такого ещё не существует. Если файл существует, то порт создаст файл `/usr/X11R6/etc/fonts/local.conf-vera`. Перенесите содержимое этого файла в `/usr/X11R6/etc/fonts/local.conf`, и шрифты Bitstream автоматически заменят используемые по умолчанию в X11 шрифты Serif, Sans Serif и Monospaced.

Наконец, пользователи могут добавлять собственные наборы посредством персональных файлов `.fonts.conf`. Для этого каждый пользователь должен просто создать файл `~/.fonts.conf`. Этот файл также должен быть в формате XML.

И последнее замечание: при использовании дисплея LCD может понадобиться включение разбиения точек. При этом компоненты красного, зелёного и голубого цветов (разделяемые по горизонтали), рассматриваются как отдельные точки для улучшения разрешения экрана по горизонтали; результат может оказаться потрясающим. Для включения этого механизма добавьте такую строчку где-нибудь в файл `local.conf`:

```
<match target="font">
  <test qual="all" name="rgba">
    <const>unknown</const>
  </test>
  <edit name="rgba" mode="assign">
    <const>rgb</const>
  </edit>
</match>
```

**Замечание:** В зависимости от типа дисплея, `rgb` может потребоваться заменить на `bgr`, `vrgb` или `vbgr`: пробуйте и смотрите, что работает лучше.

Антиалиасинг должен быть включен при следующем запуске X-сервера. Однако программы должны знать, как использовать его преимущества. В настоящее время инструментальный пакет Qt умеет ими пользоваться, так что вся оболочка **KDE** может использовать шрифты с антиалиасингом (обратитесь к Разд. 5.7.3.2 о **KDE** для выяснения всех подробностей). GTK+ и **GNOME** также можно заставить использовать антиалиасинг посредством капплета 'Font' (обратитесь к Разд. 5.7.1.3 для выяснения всех подробностей). По умолчанию **Mozilla** версий 1.2 и выше будет автоматически использовать антиалиасинг. Для отмены этого перестройте **Mozilla** с флагом `-DWITHOUT_XFT`.

## 5.6. Менеджеры Экранов (Display Managers) X

Текст предоставил Seth Kingsley.

### 5.6.1. Вступление

Менеджер Экранов X (**XDM**) это необязательный компонент Системы Окон X, который используется для управления login сессиями. Это полезно в ряде ситуаций, например минимальные "X Терминалы", десктопы, большие сетевые сервера Экранов. Так как Система Окон X не зависит от сетей и протоколов, то существует множество различных конфигураций для X клиентов и серверов запущенных на различных компьютерах подключенных к сети. **XDM** предоставляет графический интерфейс для выбора сервера к которому Вы желаете подключиться и введения информации авторизующей пользователя, например комбинацию логина и пароля.

Представляйте себе, что **XDM** предоставляет такие же возможности для пользователей, как и программа `getty(8)` (смотрите Разд. 20.3.2 для подробной информации). И это именно так, **XDM** производит вход в систему для подключенного пользователя и запускает управляющую сессию для пользователя (обычно это менеджер окон X). После этого **XDM** ожидает завершения приложения, означающее завершение пользователем работы и отключает управляющую сессию. После этого **XDM** может снова вывести приглашение к входу в систему и ожидать входа другого пользователя.

### 5.6.2. Использование XDM

Программой демона **XDM** является `/usr/X11R6/bin/xdm`. Эта программа может быть запущена от пользователя `root` в любой момент, и она начнёт управлять дисплеем X на локальной машине. Если **XDM** нужно запускать в фоновом режиме каждый раз при запуске компьютера, то наиболее правильный способ это добавить новую запись в `/etc/ttys`. Для более подробной информации о формате и использовании этого файла смотрите раздел Разд. 20.3.2.1. Это строка, которую необходимо добавить в файл `/etc/ttys` для того чтобы запустить демон **XDM** на виртуальном терминале:

```
ttyv8 "/usr/X11R6/bin/xdm -nodaemon" xterm off secure
```

По умолчанию эта запись отключена; для её включения нужно заменить пятое поле с `off` на `on` и перезапустить `init(8)`, используя метод, описанный в Разд. 20.3.2.2. Первое поле это название терминала, которым будет управлять программа, `ttyv8`. Это означает, что **XDM** будет запущен на 9ом виртуальном терминале.

### 5.6.3. Конфигурирование XDM

Конфигурационные файлы **XDM** находятся в каталоге `/usr/X11R6/lib/X11/xdm`. В нём размещаются несколько файлов, которые используются для изменения поведения и внешнего вида **XDM**. Обычно это следующие файлы:

Файл	Описание
<code>Xaccess</code>	Правила авторизации клиентов.
<code>Xresources</code>	Значения ресурсов X по умолчанию.
<code>Xservers</code>	Список локальных и удаленных экранов.



Файл	Описание
Xsession	Сценарий сессии по умолчанию.
Xsetup_*	Скрипт для запуска приложений до появления приглашения к входу в систему.
xdm-config	Глобальный конфигурационный файл для всех экранов запущенных на локальной машине
xdm-errors	Ошибки сгенерированные серверной программой.
xdm-pid	ID процесса запущенного XDM.

В этом каталоге также находятся несколько командных сценариев и программ используемых для настройки рабочего стола (desktop), когда запускается **XDM**. Назначение каждого из этих файлов будет вкратце описано. Точный синтаксис и информацию по их использованию описаны в xdm(1).

В конфигурации по умолчанию это просто прямоугольное окно приглашения ко входу в систему с именем компьютера большим шрифтом написанным сверху и строками ввода 'Login:' и 'Password:' внизу. Это хорошая отправная точка для изменения внешнего вида экранов **XDM**.

#### 5.6.3.1. Xaccess

Протокол, по которому происходит подключение дисплеев, управляемых **XDM**, называется X Display Manager Connection Protocol (XDMCP). Этот файл представляет собой набор правил для управления XDMCP соединениями с удалёнными машинами. По умолчанию он позволяет подключаться любым клиентам, но это не имеет значения, пока стандартный файл xdm-config не содержит указаний по обслуживанию удалённых соединений.

#### 5.6.3.2. Xresources

Это файл содержащий установки по умолчанию для приложений запущенных в экране выбора серверов и экране приглашения к входу в систему. Именно здесь может быть изменён вид программы входа в систему. Формат этого файла идентичен файлу app-defaults описанному в документации к X11.

#### 5.6.3.3. Xservers

Это список удаленных экранов, которые XDM должен предоставить как варианты для входа в систему.

#### 5.6.3.4. Xsession

Этот файл представляет себя командный сценарий по умолчанию для пользователей вошедших в систему с использованием **XDM**. Обычно каждый пользователь имеет собственный сценарий входа в файле ~/.xsession, который используется вместо этого сценария.

#### 5.6.3.5. Xsetup\_\*

Они запускаются автоматически перед тем, как показывается экран выбора сервера или экран входа в систему. Для каждого экрана (display) есть свой сценарий с именем Xsetup\_, за которым следует

локальный номер экрана (например, `xsetup_0`). Обычно эти сценарии запускают одну или две программы в фоновом режиме, например `xconsole`.

### 5.6.3.6. xdm-config

Здесь содержатся настройки в формате `app-defaults`, которые применимы ко всем экранам данного компьютера.

### 5.6.3.7. xdm-errors

Здесь находится выдача X серверов, которые **XDM** пытается запустить. Если экран, который **XDM** пытается открыть отключается по некоторым причинам, то это хорошее место для поиска сообщений об ошибках. Эти сообщения также записываются в пользовательский файл `~/.xsession-errors` для каждого сеанса.

## 5.6.4. Использование сетевого сервера дисплеев

Для того, чтобы позволить другим клиентам подключаться к серверу дисплеев, отредактируйте правила контроля доступа и включите обслуживание сетевых соединений. По умолчанию они выключены, что является хорошим решением с точки зрения обеспечения безопасности. Для того, чтобы позволить **XDM** принимать сетевые соединения, в первую очередь закомментируйте строку в файле `xdm-config`:

```
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with xdm
DisplayManager.requestPort:      0
```

и потом перезапустите **XDM**. Помните, что комментарии в файлах `app-defaults` начинаются с символа `!”`, а не как обычно, `”#`. Может потребоваться более жёсткий контроль доступа. Взгляните на примеры из `Xaccess` и почитайте Справочник о `xdm(1)`.

## 5.6.5. Замены для XDM

Существует несколько программ, заменяющих **XDM**. Одна из них, **kdm** (поставляемая вместе с **KDE**), описана далее в этой главе. В **kdm** имеется много визуальных и косметических улучшений, а также функциональность, позволяющая пользователям выбирать собственные оконные менеджеры во время входа в систему.

## 5.7. Графические оболочки

*Текст предоставил Valentino Vaschetto.*

В этом разделе описываются различные графические оболочки, доступные в X для FreeBSD. Термин ”графическая оболочка” может использоваться для чего угодно, от простого менеджера окон до полнофункционального набора приложений для рабочего стола типа **KDE** или **GNOME**.

## 5.7.1. GNOME

### 5.7.1.1. О GNOME

**GNOME** является дружественной к пользователю графической оболочкой, позволяющей пользователям легко использовать и настраивать свои компьютеры. В **GNOME** имеется панель (для запуска приложений и отображения их состояния), рабочий стол (где могут быть размещены данные и приложения), набор стандартных инструментов и приложений для рабочего стола, а также набор соглашений, облегчающих совместную работу и согласованность приложений. Пользователи других операционных систем или оболочек при использовании такой мощной графической оболочки, какую обеспечивает **GNOME**, должны чувствовать себя в родной среде. Дополнительную информацию относительно **GNOME** во FreeBSD можно найти на сайте FreeBSD GNOME Project (<http://www.FreeBSD.org/gnome>).

### 5.7.1.2. Установка GNOME

Легче всего установить **GNOME** через меню "Desktop Configuration" в ходе процесса установки FreeBSD, как описано в Разд. 2.9.13 Главы 2. Её также легко установить из пакета или Коллекции Портов:

Для установки пакета **GNOME** из сети, просто наберите:

```
# pkg_add -r gnome2
```

Для построения **GNOME** из исходных текстов используйте дерево портов:

```
# cd /usr/ports/x11/gnome2
# make install clean
```

После установки **GNOME** нужно указать X-серверу на запуск **GNOME** вместо стандартного оконного менеджера. Если файл `.xinitrc` уже откорректирован, то просто замените строку, в которой запускается используемый менеджер окон, на ту, что вызовет `/usr/X11R6/bin/gnome-session`. Если в конфигурационном файле нет ничего особенного, то будет достаточно просто набрать:

```
% echo "/usr/X11R6/bin/gnome-session" > ~/.xinitrc
```

Теперь наберите `startx`, и будет запущена графическая оболочка **GNOME**.

**Замечание:** Если используется менеджер дисплеев типа **XDM**, то это не сработает. Вместо этого создайте выполнимый файл `.xsession` с той же самой командой в нём. Для этого отредактируйте файл, заменив существующую команду запуска оконного менеджера на `/usr/X11R6/bin/gnome-session`:

```
% echo "#!/bin/sh" > ~/.xsession
% echo "/usr/X11R6/bin/gnome-session" >> ~/.xsession
% chmod +x ~/.xsession
```

Ещё одним вариантом является настройка менеджера дисплеев таким образом, чтобы он позволял выбирать оконный менеджер во время входа в систему; в разделе о KDE2 в подробностях описывается, как сделать это для **kdm**, менеджера дисплеев из **KDE**.

### 5.7.1.3. Шрифты с антиалиасингом и GNOME

X11 поддерживает антиалиасинг посредством своего расширения "RENDER". GTK+ 2.0 и более поздние версии (это инструментальный пакет, используемый **GNOME**) могут использовать такую функциональность. настройка антиалиасинга описана в Разд. 5.5.3. Таким образом, при наличии современного **GNOME**. Просто перейдите в Applications—>Desktop Preferences—>Font и выберите либо Best shapes, Best contrast, либо Subpixel smoothing (LCDs). Для приложений GTK+, которые не являются частью оболочки **GNOME**, задайте в качестве значения переменной окружения GDK\_USE\_XFT 1 перед запуском программы.

## 5.7.2. KDE

### 5.7.2.1. О KDE

**KDE** является простой в использовании современной графической оболочкой. Вот лишь некоторое из того, что даёт пользователю **KDE**:

- Прекрасный современный рабочий стол
- Рабочий стол, полностью прозрачный для работы в сети
- Интегрированная система помощи, обеспечивающая удобный и согласованный доступ к системе помощи по использованию рабочего стола **KDE** и его приложений
- Единообразный внешний вид и управление во всех приложениях **KDE**
- Стандартизированные меню и панели инструментов, комбинации клавиш, цветовые схемы и так далее.
- Интернационализация: в **KDE** поддерживается более 40 языков
- Централизованное единообразное конфигурирование рабочего стола в диалоговом режиме
- Большое количество полезных приложений для **KDE**

Для **KDE** существует пакет офисных приложений, который выполнен по технологии "KParts" из **KDE**, состоящий из программы для работы с электронными таблицами, презентационной программы, органайзера, клиента для чтения телеконференций и других программ. С **KDE** также поставляется веб-браузер под названием **Konqueror**, который является серьезным соперником другим браузерам для UNIX-систем. Дополнительную информацию о **KDE** можно найти на веб-сайте KDE (<http://www.kde.org/>). Для получения информации и информационных ресурсов, специфичных для **KDE** во FreeBSD, обратитесь к сайту команды FreeBSD-KDE team (<http://freebsd.kde.org/>).

### 5.7.2.2. Установка KDE

Как и в случае с **GNOME** или любой другой графической оболочкой, легче всего установить **KDE** через меню "Desktop Configuration" во время установки FreeBSD, как это описано в Разд. 2.9.13 Главы 2. Повторимся ещё раз, что программное обеспечение можно легко установить из пакета или из Коллекции Портов:

Для установки пакета **KDE** из сети, просто наберите:

```
# pkg_add -r kde
```

pkg\_add(1) автоматически загрузит самую последнюю версию приложения.

Для построения **KDE** из исходных текстов воспользуйтесь деревом портов:

```
# cd /usr/ports/x11/kde3
# make install clean
```

После установки **KDE** нужно указать X-серверу на запуск этого приложения вместо оконного менеджера, используемого по умолчанию. Это достигается редактированием файла `.xinitrc`:

```
% echo "exec startkde" > ~/.xinitrc
```

Теперь при вызове X Window System по команде `startx` в качестве оболочки будет использоваться **KDE**.

При использовании менеджера дисплеев типа **XDM** настройка несколько отличается. Вместо этого нужно отредактировать файл `.xsession`. Указания для **kdm** описаны далее в этой главе.

### 5.7.3. Более подробно о KDE

Теперь, когда **KDE** установлена в системе, можно узнать много нового из её справочных страниц или просто указанием и щелканием по различным меню. Пользователи Windows или Mac® будут чувствовать себя как дома.

Лучшим справочником по **KDE** является онлайн-документация. **KDE** поставляется с собственным веб-браузером, который называется **Konqueror**, десятками полезных приложений и подробной документацией. В оставшейся части этого раздела обсуждаются технические вопросы, трудные для понимания при случайном исследовании.

#### 5.7.3.1. Менеджер дисплеев KDE

Администратору многопользовательской системы может потребоваться графический экран входа в систему для приглашения пользователей. Вы можете использовать **XDM**, как это описано ранее. Однако в **KDE** имеется альтернативный менеджер **kdm**, который был разработан для того, чтобы выглядеть более привлекательно и иметь большее количество настраиваемых опций для входа в систему. В частности, пользователи могут легко выбирать (посредством меню), какую оболочку (**KDE**, **GNOME** или что-то ещё) запускать после входа в систему.

Чтобы начать, запустите панель управления **KDE**, `kcontrol`, из-под пользователя `root`. Вообще говоря, считается небезопасным работать в X пользователем `root`. Вместо этого запустите менеджер окон как обычный пользователь, откройте окно терминала (такого, как `xterm` или `konsole`) из **KDE**, станьте пользователем `root` по команде `su` (для этого нужно быть членом группы `wheel` из `/etc/group`), а затем наберите `kcontrol`.

Щёлкните на иконке слева с надписью **System**, затем на **Login manager**. Справа имеется много различных параметров настройки, которые более детально описаны в руководстве по **KDE**. Щёлкните на **sessions** справа. Щёлкните на кнопку **New type** для того, чтобы добавить различные оконные менеджеры и графические оболочки. Это просто названия, так что они могут именоваться **KDE** или **GNOME**, а не `startkde` или `gnome-session`.) Включите название `failsafe`.

Поэкспериментируйте также и с другими меню, они в основном носят косметический характер и самоописательны. Когда закончите, щёлкните на **Apply** внизу и завершите работу панели управления.

Чтобы **kdm** понимал, что значат эти названия (**KDE**, **GNOME** и так далее), отредактируйте файлы, которые использует XDM.

**Замечание:** В **KDE 2.2** это изменилось: в **kdm** теперь используются собственные конфигурационные файлы. Пожалуйста, обратитесь к документации по **KDE 2.2** для получения подробной информации.

В окне терминала, работая как пользователь **root**, отредактируйте файл `/usr/X11R6/lib/X11/xdm/Xsession`. В середине есть раздел, выглядящий вот так:

```
case $# in
1)
    case $1 in
failsafe)
        exec xterm -geometry 80x24-0-0
        ;;
    esac
esac
```

Нужно добавить к этому разделу несколько строк. Полагая, что ранее использовались названия **'KDE'** и **'GNOME'**, сделаем следующее:

```
case $# in
1)
    case $1 in
kde)
        exec /usr/local/bin/startkde
        ;;
GNOME)
        exec /usr/X11R6/bin/gnome-session
        ;;
failsafe)
        exec xterm -geometry 80x24-0-0
        ;;
    esac
esac
```

Чтобы выбор **KDE** в качестве оболочки на момент входа в систему был принят, нужно добавить такую строку в `/usr/X11R6/lib/X11/xdm/Xsetup_0`:

```
/usr/local/bin/kdmdesktop
```

Теперь проверьте, что **kdm** перечислена в файле `/etc/ttys` для запуска при следующей загрузке. Для этого просто следуйте инструкциям из предыдущего раздела о XDM, заменив отсылки к программе `/usr/X11R6/bin/xdm` на `/usr/local/bin/kdm`.

### 5.7.3.2. Шрифты с антиалиасингом

X11 поддерживает антиалиасинг через своё расширение 'RENDER', а начиная с версии 2.3, Qt (инструментарий, используемый в **KDE**) поддерживает это расширение. Настройка этого описана в Разд. 5.5.3 по антиалиасингу с шрифтами X11. Таким образом, при работе с современным программным обеспечением в оболочке **KDE** возможно использование антиалиасинга. Просто перейдите в меню KDE, затем к **Preferences** → **Look and Feel** → **Fonts** и поставьте галочку рядом с **Use Anti-Aliasing for Fonts and Icons**. Для работы с приложением Qt, которое не является частью **KDE**, перед его запуском нужно устанавливать переменную окружения QT\_XFT в значение true.

## 5.7.4. XFce

### 5.7.4.1. О XFce

**XFce** является графической оболочкой, построенной на основе инструментального пакета GTK+, используемого в **GNOME**, но гораздо легче и предназначен для тех, кому нужен простой, эффективно работающий рабочий стол, который легко использовать и настраивать. Визуально он выглядит очень похоже на **CDE**, который есть в коммерческих UNIX-системах. Вот некоторые из достоинств **XFce**:

- Простой, лёгкий в обращении рабочий стол
- Полностью настраиваемый при помощи мыши, с интерфейсом drag and drop и так далее
- Главная панель похожа на **CDE**, с меню, апплетами и возможностями по быстрому запуску приложений
- Интегрированный оконный менеджер, менеджер файлов, управление звуком, модуль совместимости с **GNOME** и прочее
- Возможность использования тем (так как использует GTK+)
- Быстрый, легкий и эффективный: идеален для устаревших/слабых машин или для машин с ограниченной памятью

Дополнительную информацию о **XFce** можно найти на сайте XFce (<http://www.xfce.org/>).

### 5.7.4.2. Установка XFce

Для **XFce** имеется (на момент написания этого текста) бинарный пакет. Для его установки просто наберите:

```
# pkg_add -r xfce4
```

Либо в случае построения из исходных текстов используйте Коллекцию Портов:

```
# cd /usr/ports/x11-wm/xfce4
# make install clean
```

Теперь укажите X-серверу на запуск **XFce** при следующем запуске X. Просто наберите вот что:

```
% echo "/usr/X11R6/bin/startxfce4" > ~/.xinitrc
```

При следующем запуске X в качестве рабочего стола будет использоваться **XFce**. Как это сказано выше, если используется менеджер дисплеев, такой, как XDM, создайте файл `.xsession` так, как это описано в разделе о GNOME, но с командой `/usr/X11R6/bin/startxfce4`, либо настройте менеджер дисплеев так, чтобы он разрешил выбор рабочего стола во время входа в систему, как это описано в разделе о kdm.



## II. Общие задачи

Теперь, когда основы были пройдены, в данной части Руководства FreeBSD будут обсуждаться некоторые часто используемые возможности FreeBSD. В этих главах:

- Введение в популярные и полезные графические приложения: браузеры, бизнес приложения, программы просмотра документов и т.д.
- Представлены множество мультимедийных программ, доступных в FreeBSD.
- Описан процесс создания собственного ядра FreeBSD для включения дополнительных функций системы.
- Система печати разобрана в деталях, как для непосредственно подключенных принтеров, так и для принтеров, подключенных через сеть.
- Показано, как запускать приложения Linux в системе FreeBSD.

Перед прочтением некоторых из этих глав необходимо предварительно ознакомиться с предварительной информацией, что указано в кратком обзоре в начале каждой главы.

# Глава 6. Приложения для настольного компьютера

*Предоставил Christophe Juniet. Перевод на русский язык: Денис Пеплин.*

## 6.1. Краткий обзор

FreeBSD может работать с широким кругом приложений для настольного компьютера (десктопа), таких как браузеры и текстовые процессоры. Большинство из них доступны в качестве пакетов или могут быть автоматически собраны из коллекции портов. Многим новым пользователям хотелось бы видеть эти приложения на своем компьютере. В этой главе показано как без усилий установить некоторые популярные приложения для настольного компьютера из пакетов или из коллекции портов.

Обратите внимание, что при установке программ из портов они компилируются из исходных текстов. Это может занять очень много времени, в зависимости от того, что вы собираете, и от скорости процессора вашего компьютера (компьютеров). Большинство программ, имеющих в коллекции портов, могут быть установлены из прекомпилированных пакетов, если сборка из исходных текстов занимает недопустимо много времени.

Поскольку FreeBSD обеспечивает двоичную совместимость с Linux, многие приложения, первоначально разработанные для Linux, доступны и на вашем компьютере. Настоятельно рекомендуется прочитать Гл. 10 перед установкой любого из приложений Linux. Названия многих портов, использующих двоичную совместимость с Linux, начинаются с "linux-". Помните это при поиске отдельного порта, например с помощью `whereis(1)`. Далее в статье подразумевается, что вы включили бинарную совместимость с Linux перед установкой какого-либо приложения Linux.

Вот несколько категорий, о которых пойдет речь в этой главе:

- Браузеры (такие как **Mozilla**, **Netscape**, **Opera**, **Firefox**, **Konqueror**)
- Бизнес приложения (такие как **KOffice**, **AbiWord**, **GIMP**, **OpenOffice.org**)
- Программы просмотра документов (такие как **Acrobat Reader®**, **gv**, **Xpdf**, **GQview**)
- Финансовые программы (такие как **GnuCash**, **Gnumeric**, **Abacus**)

Перед прочтением этой главы вам потребуется:

- Узнать как устанавливать дополнительные программы сторонних производителей (Гл. 4).
- Узнать как устанавливать программы Linux (Гл. 10).

Чтобы получить дополнительную информацию о настройке мультимедиа среды, прочтите Гл. 7. Если вам нужна электронная почта, обратитесь к Гл. 22.

## 6.2. Браузеры

FreeBSD поставляется без предустановленного браузера. Взамен каталог [www](http://www.FreeBSD.org/ru/ports/www.html) (http://www.FreeBSD.org/ru/ports/www.html) коллекции портов содержит множество готовых к установке

браузеров. Если у вас нет времени компилировать все (в некоторых случаях это может занять очень много времени), многие из них доступны в виде пакетов.

В **KDE** и **GNOME** уже есть HTML браузеры. Обратитесь к Разд. 5.7 для подробной информации об установке этих полноценных десктопов.

Если вы ищете облегченный браузер, попробуйте `www/dillo`, `www/links`, или `www/w3m` из коллекции портов.

Этот раздел рассказывает о следующих приложениях:

Название приложения	Потребность в ресурсах	Установка из портов	Основные зависимости
<b>Mozilla</b>	большая	тяжелая	<b>Gtk+</b>
<b>Netscape</b>	большая	легкая	Linux Binary Compatibility
<b>Opera</b>	малая	легкая	Доступны версии для FreeBSD и Linux. Для Linux версии необходимо наличие Linux Binary Compatibility и <b>linux-openmotif</b>
<b>Firefox</b>	средняя	тяжелая	<b>Gtk+</b>
<b>Konqueror</b>	средняя	тяжелая	Библиотеки <b>KDE</b>

### 6.2.1. Mozilla

**Mozilla** это возможно наиболее подходящий браузер для десктопа FreeBSD. Он современный, стабильный и полностью портирован на FreeBSD. Его достоинство в высокой совместимости со стандартами HTML. В нем есть почтовая и новостная программы. В нем даже найдется редактор HTML, если вам потребуется самостоятельно написать несколько веб-страничек. Пользователи **Netscape** найдут общие черты с **Communicator**, поскольку оба браузера имеют одну основу.

На медленных компьютерах с частотой CPU меньше 233MHz или с памятью меньше 64MB, **Mozilla** требует слишком много ресурсов, чтобы быть удобной в использовании. Вместо нее вы можете обратить внимание на браузер **Opera**, описанный ниже в этой главе.

Если вы не можете или не хотите компилировать **Mozilla** по какой-то причине, команда FreeBSD GNOME уже сделала это для вас. Просто установите пакет из сети с помощью:

```
# pkg_add -r mozilla
```

Если пакет недоступен, но у вас достаточно времени и места на диске, вы можете скачать исходные тексты для **Mozilla**, скомпилировать их и установить в вашу систему. Это делается так:

```
# cd /usr/ports/www/mozilla
# make install clean
```

Порт **Mozilla** проверяет правильность установки путем запуска регистрации chrome с привилегиями пользователя root. Если вы хотите загрузить некоторые дополнения, например курсоры мыши, потребуется запустить **Mozilla** под root для их правильной установки.

После завершения установки **Mozilla**, больше не требуется работать под root. Вы можете запустить **Mozilla** в качестве браузера, набрав:

```
% mozilla
```

Вы можете также запустить непосредственно программу чтения почты и новостей, как показано ниже:

```
% mozilla -mail
```

## 6.2.2. Mozilla, Java™, и Macromedia® Flash™

*Предоставил Tom Rhodes.*

Установка **Mozilla** проста, но к сожалению, установка **Mozilla** с поддержкой дополнений, таких как Java™ и Macromedia® Flash™ отнимает и время и место на диске.

Первое, что нужно сделать — загрузку файлов, которые будут использоваться с **Mozilla**. Зайдите с помощью имеющегося веб браузера на <http://www.sun.com/software/java2/download.html> и создайте учетную запись на этом вебсайте. Сохраните имя пользователя и пароль, они могут понадобиться в будущем. Загрузите копию файла `j2sdk-1_3_1-src.tar.gz` и поместите в каталог `/usr/ports/distfiles/`, поскольку порт не загрузит его автоматически. Причина в лицензионном ограничении. Загрузите с этого же сайта "java environment", <http://java.sun.com/webapps/download/Display?BundleId=7905>. Имя файла `j2sdk-1_3_1_08-linux-i586.bin`, он большой (около 25 мегабайт!). Как и раньше, этот файл требуется поместить в `/usr/ports/distfiles/`. Наконец, загрузите копию "java patchkit" с <http://www.eyesbeyond.com/freebsd/java/> и поместите ее в `/usr/ports/distfiles/`.

Установите порт `java/jdk13` стандартной командой `make install clean`, затем установите порт `www/flashpluginwrapper`. Этот порт требует `emulators/linux_base` (большой порт). Есть и другие плагины **Flash** но у меня они не работают.

Установите порт `www/mozilla`, если **Mozilla** еще не установлена.

Теперь скопируйте файлы **Flash** плагина:

```
# cp /usr/local/lib/flash/libflashplayer.so \
/usr/X11R6/lib/browser_plugins/libflashplayer_linux.so

# cp /usr/local/lib/flash/ShockwaveFlash.class \
/usr/X11R6/lib/browser_plugins/
```

Теперь добавьте следующие линии к верхней части (но под `#!/bin/sh`) стартового скрипта **Mozilla**: `/usr/X11R6/bin/mozilla`.

```
LD_PRELOAD=/usr/local/lib/libflashplayer.so.1
export LD_PRELOAD
```

Они включают **Flash** плагин.

Теперь просто запустите **Mozilla**:

```
% mozilla &
```

И войдите в пункт **About Plug-ins** меню **Help**. Должен появиться список со всеми доступными плагинами. В нем должны присутствовать **Java** и **Shockwave® Flash**.

### 6.2.3. Netscape®

Коллекция портов содержит несколько версий браузера Netscape. Поскольку в версии для FreeBSD он содержит серьезную ошибку безопасности, установка этой версии настоятельно не рекомендуется. Вместо этого, используйте более свежую версию для Linux или DIGITAL UNIX.

Последний стабильный релиз браузера Netscape это **Netscape 7**. Он может быть установлен из коллекции портов:

```
# cd /usr/ports/www/linux-netscape7
# make install clean
```

Есть локализованные версии в французском, немецком и японском разделах.

**Предостережение:** Версии **Netscape 4.x** не рекомендуются, поскольку они не совместимы с современными стандартами. Однако, **Netscape 7.x** и более новые версии доступны только для платформы i386.

### 6.2.4. Opera

**Opera** это очень быстрый, полноценный и совместимый со стандартами браузер. Он поставляется в двух вариантах: "родная" для FreeBSD версия и версия, запускаемая в режиме эмуляции Linux. Для каждой операционной системы есть бесплатная версия браузера, показывающая баннеры, и версия без баннеров, которую можно купить на веб сайте Opera (<http://www.opera.com>).

Для работы в сети с помощью FreeBSD версии **Opera** установите пакет:

```
# pkg_add -r opera
```

На некоторых серверах FTP нет всех пакетов, но те же результаты можно получить с помощью коллекции портов, набрав:

```
# cd /usr/ports/www/opera
# make install clean
```

Для установки Linux версии **Opera**, замените `opera` на `linux-opera` в примере выше. Версия для Linux полезна в ситуации, когда требуются плагины, доступные только для Linux, такие как **Adobe Acrobat Reader**. Во всех других отношениях версии для FreeBSD и Linux являются функционально идентичными.

### 6.2.5. Firefox

**Firefox** это браузер следующего поколения, основанный на коде **Mozilla**. **Mozilla** это полный набор приложений, таких как браузер, почтовый клиент, чат клиент и многое другое. **Firefox** это всего лишь браузер, что делает его меньше и быстрее.

Установите пакет, выполнив:

```
# pkg_add -r firefox
```

Вы можете также использовать коллекцию портов, если предпочитаете сборку из исходных текстов:

```
# cd /usr/ports/www/firefox
# make install clean
```

### 6.2.6. Konqueror

**Konqueror** это часть **KDE**, но может быть использован и отдельно от **KDE**, путем установки `x11/kdebase3`. **Konqueror** это гораздо больше чем просто браузер, это также менеджер файлов и программа просмотра мультимедиа.

**Konqueror** поставляется с набором плагинов, доступных из `misc/konq-plugins`.

**Konqueror** поддерживает также **Flash**, документация How To для него доступна по адресу <http://freebsd.kde.org/howto.php>.

## 6.3. Бизнес приложения

В начале работы новые пользователи зачастую стремятся найти хороший офисный пакет или удобный текстовый процессор. Хотя некоторые десктопы, такие как **KDE**, поставляются с готовым офисным пакетом, приложения по умолчанию не существует. В FreeBSD есть все необходимое, кроме графической среды.

Этот раздел описывает следующие приложения:

Название приложения	Потребность в ресурсах	Установка из портов	Основные зависимости
<b>KOffice</b>	малая	тяжелая	<b>KDE</b>
<b>AbiWord</b>	малая	легкая	<b>Gtk+</b> или <b>GNOME</b>
<b>The Gimp</b>	малая	тяжелая	<b>Gtk+</b>
<b>OpenOffice.org</b>	большая	очень тяжелая	<b>GCC 3.1</b> , <b>JDK™ 1.3</b> , <b>Mozilla</b>

### 6.3.1. KOffice

Сообщество KDE предоставляет графическую среду с офисным пакетом, который может быть использован вне **KDE**. Он включает четыре стандартных компонента, встречающиеся и в других офисных пакетах. Текстовый процессор **KWord**, программа электронных таблиц **KSpread**, **KPresenter** для создания презентаций и программа векторной графики **Kontour**.

Перед установкой последней версии **KOffice**, убедитесь в наличии свежей версии **KDE**.

Для установки **KOffice** из пакета, выполните следующую команду:

```
# pkg_add -r koffice
```

Если пакет недоступен, используйте коллекцию портов. Например, для установки **KOffice** для **KDE3**, выполните:

```
# cd /usr/ports/editors/koffice-kde3
# make install clean
```

### 6.3.2. AbiWord

**AbiWord** это свободно распространяемый текстовый процессор, по внешнему виду и поведению очень похожий на **Microsoft Word**. Он подходит для набора документов, писем, отчетов, напоминаний и так далее. Он очень быстр, содержит много новшеств и очень удобен в использовании.

**AbiWord** может импортировать и экспортировать множество файловых форматов, включая патентованный Microsoft .doc.

**AbiWord** доступен в виде пакета. Вы можете установить его так:

```
# pkg_add -r AbiWord2
```

Если пакет недоступен, он может быть собран из коллекции портов, которая должна быть свежей. Это можно сделать командой:

```
# cd /usr/ports/editors/AbiWord2
# make install clean
```

### 6.3.3. GIMP

Для создания и редактирования изображений есть продвинутая программа **GIMP**. Она может быть использована как простая программа рисования и как программа обработки фотографий.

Поддерживается большое количество плагинов и предоставлен интерфейс для скриптов. **GIMP** может читать и записывать файлы многих форматов. Есть интерфейс со сканерами и планшетами.

Вы можете установить пакет, выполнив эту команду:

```
# pkg_add -r gimp
```

Если на вашем сервере FTP нет этого пакета, вы можете использовать коллекцию портов. Каталог graphics (<http://www.FreeBSD.org/ru/ports/graphics.html>) коллекции портов содержит также раздел **Руководство Gimp**. Здесь показано, как его установить:

```
# cd /usr/ports/graphics/gimp
# make install clean
# cd /usr/ports/graphics/gimp-manual-pdf
# make install clean
```

**Замечание:** Каталог graphics (<http://www.FreeBSD.org/ru/ports/graphics.html>) коллекции портов содержит версию **GIMP** для разработчиков в graphics/gimp-devel. HTML версия **Руководства Gimp** находятся в graphics/gimp-manual-html.

### 6.3.4. OpenOffice.org

**OpenOffice.org** включает все обязательные компоненты полноценного офисного пакета: текстовый процессор, программу электронных таблиц, программу управления презентациями и программу векторной графики. Интерфейс пользователя очень похож на другие офисные пакеты, возможен импорт и экспорт различных популярных файловых форматов. Приложение доступно в вариантах для множества разных языков, включая интерфейсы, проверку орфографии и словари.

Текстовый процессор **OpenOffice.org** использует чистый XML формат файлов для увеличения переносимости и гибкости. Программа для работы с текстовыми таблицами предоставляет макроязык и может работать с внешними базами данных. **OpenOffice.org** уже стабильна и существует в версиях для Windows, Solaris™, Linux, FreeBSD, и Mac OS X. Дополнительную информацию об **OpenOffice.org** можно найти на веб сайте OpenOffice (<http://www.openoffice.org/>). Получить специфичную для FreeBSD и загрузить пакеты можно с веб сайта команды портирования OpenOffice на FreeBSD (FreeBSD OpenOffice Porting Team (<http://porting.openoffice.org/freebsd/>)).

Для установки **OpenOffice.org**, выполните:

```
# pkg_add -r openoffice
```

После установки пакета запустите программу настройки и выберите standard workstation installation. Запустите эту команду из под пользователя, который будет использовать **OpenOffice.org**:

```
% openoffice-setup
```

Если пакеты **OpenOffice.org** недоступны, можно выбрать компиляцию порта. Однако, вы должны помнить, что это потребует много места на диске и компиляция будет довольно долгой.

```
# cd /usr/ports/editors/openoffice-1.1
# make install clean
```

Потом запустите настройку под пользователем, который будет использовать **OpenOffice.org** и выберите standard workstation installation:

```
% cd /usr/ports/editors/openoffice
% make install-user
```

Если вы хотите использовать локализованную версию, вот доступные порты:

Язык	Порт
Арабский	arabic/openoffice-1.1
Венгерский	hungarian/openoffice-1.1
Голландский	editors/openoffice-1.1-nl
Греческий	editors/openoffice-1.1-el
Датский	editors/openoffice-1.1-dk
Испанский	editors/openoffice-1.1-es
Итальянский	editors/openoffice-1.1-it
Каталанский	editors/openoffice-1.1-ca
Китайский (традиционный)	chinese/openoffice-1.1-zh_TW



Язык	Порт
Китайский (упрощенный)	chinese/openoffice-1.1-zh_CN
Корейский	korean/openoffice-1.1
Немецкий	german/openoffice-1.1
Польский	polish/openoffice-1.1
Португальский (Бразилия)	portuguese/openoffice-1.1-pt_BR
Португальский	portuguese/openoffice-1.1-pt_PT
Русский	russian/openoffice-1.1
Словацкий	editors/openoffice-1.1-sk
Словенский	editors/openoffice-1.1-sl_SI
Турецкий	editors/openoffice-1.1-tr
Финский	editors/openoffice-1.1-fi
Французский	french/openoffice-1.1
Чешский	editors/openoffice-1.1-cs
Шведский	editors/openoffice-1.1-se
Эстонский	editors/openoffice-1.1-et
Японский	japanese/openoffice-1.1

## 6.4. Программы просмотра документов

Некоторые новые форматы документов приобрели большую популярность. Стандартные программы для их просмотра могут отсутствовать в базовой системе. В этом разделе мы увидим, как их установить.

В разделе говорится о следующих приложениях:

Название приложения	Потребность в ресурсах	Установка из портов	Основные зависимости
<b>Acrobat Reader</b>	малая	легкая	Linux Binary Compatibility
<b>gv</b>	малая	легкая	<b>Xaw3d</b>
<b>Xpdf</b>	малая	легкая	<b>FreeType</b>
<b>GQview</b>	малая	легкая	<b>Gtk+</b> или <b>GNOME</b>

### 6.4.1. Acrobat Reader®

Сейчас многие документы распространяются в формате PDF, аббревиатура для "Portable Document Format". Одна из рекомендованных программ для просмотра этого типа документов, это **Acrobat Reader**, выпущенный Adobe для Linux. Поскольку FreeBSD может запускать исполняемые файлы Linux, он доступен также и для FreeBSD.

Для установки пакета **Acrobat Reader 5** выполните:

```
# pkg_add -r acroread
```

Обычно если пакет недоступен, или вам нужна более поздняя версия, вы можете использовать

коллекцию портов так:

```
# cd /usr/ports/print/acroread
# make install clean
```

### 6.4.2. gv

**gv** это программа просмотра PostScript и PDF. Она разработана на основе **ghostview**, но выглядит лучше благодаря библиотеке **Xaw3d**. Она быстра, а ее интерфейс несложен. У **gv** есть множество функций, таких как выбор ориентации, размера бумаги, масштаба и сглаживание. Почти любая операция может быть выполнена как с клавиатуры, так и мышью.

Для установки **gv** из пакета, выполните:

```
# pkg_add -r gv
```

Если вы не можете получить пакет, используйте коллекцию портов:

```
# cd /usr/ports/print/gv
# make install clean
```

### 6.4.3. Xpdf

Если вам нужна небольшая программа просмотра PDF под FreeBSD, **Xpdf** это легкая и эффективная программа. Она требует очень небольшого количества ресурсов и очень стабильна. Используются стандартные шрифты X, **Motif** или другие пакеты для X не нужны.

Для установки пакета **Xpdf**, выполните эту команду:

```
# pkg_add -r xpdf
```

Если пакет недоступен, или вы предпочитаете коллекцию портов, выполните:

```
# cd /usr/ports/graphics/xpdf
# make install clean
```

После завершения установки вы можете запустить **Xpdf** и использовать правую кнопку мыши для активации меню.

### 6.4.4. GQview

**GQview** это программа для работы с изображениями. Вы можете просмотреть файл одним кликом, запустить внешний редактор, получить миниатюры и многое другое. Еще в нем есть слайдшоу и несколько основных файловых операций. Вы можете управлять коллекциями изображений и легко находить дубликаты. В **GQview** изображения можно просматривать во весь экран, его можно адаптировать к разным языкам.

Если вы хотите установить пакет **GQview**, выполните:

```
# pkg_add -r gqview
```

Если пакет недоступен, или вы предпочитаете использовать коллекцию портов, выполните:

```
# cd /usr/ports/graphics/gqview
# make install clean
```

## 6.5. Финансовые программы

Если по каким-то причинам вам нужно управлять своими финансами на десктопе FreeBSD, есть несколько мощных и простых в использовании приложений. Некоторые из них совместимы с широко распространенными форматами файлов, такими как документы **Quicken®** или **Excel**.

В этом разделе говорится о следующих приложениях:

Название приложения	Потребность в ресурсах	Установка из портов	Основные зависимости
<b>GnuCash</b>	малая	тяжелая	<b>GNOME</b>
<b>Gnumeric</b>	малая	тяжелая	<b>GNOME</b>
<b>Abacus</b>	малая	легкая	<b>Tcl/Tk</b>

### 6.5.1. GnuCash

**GnuCash** это часть проекта **GNOME**, который стремится предоставить дружелюбные к пользователю приложения с широким набором функций. С **GnuCash** вы можете отслеживать доходы и расходы, банковские счета или акции. Интуитивный интерфейс программы не мешает ей оставаться очень профессиональной.

**GnuCash** предоставляет интеллектуальный журнал записей, иерархическую систему учетных записей, множество клавиатурных сокращений и метод автозавершения. Он может разбивать одну транзакцию на несколько частей, детализируя ее. **GnuCash** может импортировать и присоединять файлы **Quicken** QIF. Он также работает с основными международными форматами дат и валютами.

Для установки **GnuCash** в вашу систему, выполните:

```
# pkg_add -r gnucash
```

Если пакет недоступен, вы можете использовать коллекцию портов:

```
# cd /usr/ports/finance/gnucash
# make install clean
```

### 6.5.2. Gnumeric

**Gnumeric** это электронная таблица, часть графической среды **GNOME**. Она использует удобное автоматическое "угадывание" ввода пользователя в зависимости от формата ячейки и систему автозаполнения для множества последовательностей. Она может импортировать файлы нескольких популярных форматов, таких как **Excel**, **Lotus 1-2-3**, или **Quattro Pro**. **Gnumeric** работает с диаграммами через **math/guppi**. В ней множество встроенных функций, можно использовать обычные форматы ячеек: число, валюта, дата, время и многие другие.

Для установки **Gnumeric** из пакета, введите:

```
# pkg_add -r gnumeric
```

Если пакет недоступен, вы можете использовать коллекцию портов:

```
# cd /usr/ports/math/gnumeric
# make install clean
```

### 6.5.3. Abacus

**Abacus** это небольшая и простая в использовании программа электронных таблиц. В ней много встроенных функций из нескольких областей, таких как статистика, финансы и математика. Она может импортировать и экспортировать файлы **Excel**. **Abacus** также может печатать PostScript.

Для установки **Abacus** из пакета, выполните:

```
# pkg_add -r abacus
```

Если пакет недоступен, вы можете использовать коллекцию портов, выполнив:

```
# cd /usr/ports/deskutils/abacus
# make install clean
```

## 6.6. Итоги

Хотя FreeBSD популярна в основном среди провайдеров из-за стабильности и высокой производительности, на сегодняшний день она вполне готова к использованию в качестве десктопа. С несколькими тысячами приложений, доступных в виде пакетов (<http://www.FreeBSD.org/ru/where.html>) или портов (<http://www.FreeBSD.org/ru/ports/index.html>), вы можете создать прекрасный десктоп, отвечающий всем вашим потребностям.

После первой установки десктопа, вы можете попробовать сделать шаг вперед с `misc/instant-workstation`. Этот "мета-порт" позволяет вам собрать типичный набор портов для рабочей станции. Вы можете настроить его, редактируя `/usr/ports/misc/instant-workstation/Makefile`. Следуйте синтаксису существующего файла при добавлении и удалении портов, соберите порт как обычно. В конечном итоге, вы можете создать большой пакет, соответствующий вашему собственному десктопу, и установить его на другие рабочие станции!

Вот небольшой обзор всех графических приложений, о которых говорилось в этой главе:

Имя приложения	Имя пакета	Имя порта
<b>Mozilla</b>	mozilla	www/mozilla
<b>Netscape</b>	linux-netscape7	www/linux-netscape7
<b>Opera</b>	linux-opera	www/linux-opera
<b>Firefox</b>	firefox	www/firefox

Имя приложения	Имя пакета	Имя порта
KOffice	koffice-kde3	editors/koffice-kde3
AbiWord	AbiWord2	editors/AbiWord
The GIMP	gimp	graphics/gimp1
OpenOffice.org	openoffice	editors/openoffice
Acrobat Reader	acroread	print/acroread5
gv	gv	print/gv
Xpdf	xpdf	graphics/xpdf
GQview	gqview	graphics/gqview
GnuCash	gnucash	finance/gnucash
Gnumeric	gnumeric	math/gnumeric
Abacus	abacus	deskutils/abacus

# Глава 7. Мультимедиа

*Предоставил Moses Moore. Перевод на русский язык: Александр Коваленко, Денис Пеплин.*

## 7.1. Краткий обзор

FreeBSD поддерживает большое количество различных звуковых карт, что позволяет вам насладиться высококачественным звуком. Также это дает вам возможность записывать и воспроизводить звуковые файлы в формате MPEG Audio Layer 3 (MP3), WAV, Ogg Vorbis, а также во множестве других форматов. Коллекция Портов FreeBSD также содержит ряд приложений, позволяющих редактировать записанные звуковые файлы, добавлять звуковые эффекты, управлять подключенными MIDI устройствами.

Также FreeBSD может поддерживать воспроизведение видеофайлов и DVD. Количество приложений, позволяющих кодировать, преобразовывать и воспроизводить различные форматы видео, существенно меньше количества приложений для работы со звуком. Например, на время написания этого документа в Коллекции Портов FreeBSD не существовало хорошего приложения для преобразования видео, которое могло бы быть использовано для преобразований между разными форматами, как, например, `audio/sox`. Впрочем, ситуация в этой области меняется быстро.

Эта глава описывает необходимые шаги для настройки вашей звуковой карты. Настройка и установка X11 (Гл. 5) уже охватывает вопросы, связанные с аппаратными установками вашей видеокарты, хотя могут быть возможности дополнительной настройки для улучшения воспроизведения.

После прочтения этой главы вы будете знать:

- Как настроить систему так, чтобы звуковая карта была опознана.
- Методы проверки работы звуковой карты при помощи тестовых приложений.
- Как исправить проблемы, возникающие при работе со звуковыми картами.
- Как прослушать и создать MP3 и другие форматы.
- Как X сервер поддерживает видео.
- Некоторые проигрыватели и кодировщики видео, которые показывают хорошие результаты.
- Как воспроизвести DVD, `.mpg` и `.avi` файлы.
- Как скопировать информацию с CD и DVD в файлы.
- Как настроить ТВ тюнер.
- Как настроить сканер.

Прежде, чем начать читать эту главу вы должны:

- Знать как конфигурировать и устанавливать новое ядро (Гл. 8).

**Внимание:** Попытка смонтировать аудио CD при помощи команды `mount(8)` как минимум, сообщит об ошибке и, как максимум, может привести к *панике ядра*. Эти носители имеют специальные форматы, которые отличны от обычной файловой системы ISO.

## 7.2. Настройка звуковой карты

*Предоставил Moses Moore. Расширил для FreeBSD 5.X Marc Fonvieille.*

### 7.2.1. Настройка системы

Перед тем как начать, определите модель вашей карты, процессор, который она использует, и интерфейс карты: PCI или ISA. FreeBSD поддерживает множество разных PCI и ISA карт. Сверьтесь со списком поддерживаемых аудио устройств в Информации об оборудовании

(<http://www.FreeBSD.org/releases/5.3R/hardware.html>), чтобы проверить, поддерживается ли ваша карта.

Этот документ также содержит информацию о том, какой драйвер поддерживает вашу карту.

Для того, чтобы использовать звуковую карту, вы должны загрузить соответствующий драйвер устройства. Этого можно достигнуть двумя путями. Простейший способ – это просто загрузить соответствующий вашей карте модуль ядра используя `kldload(8)`, что можно сделать или из командной строки:

```
# kldload snd_emu10k1
```

или добавлением соответствующей строки к файлу `/boot/loader.conf`:

```
snd_emu10k1_load="YES"
```

Эти примеры приведены для звуковой карты Creative SoundBlaster® Live!. Другие имеющиеся модули драйверов звуковых карты приведены в `/boot/defaults/loader.conf`. Если вы не уверены, какой драйвер использовать, попробуйте загрузить `snd_driver`:

```
# kldload snd_driver
```

Это мета-драйвер, загружающий сразу все наиболее распространенные драйверы сразу. Это повышает скорость поиска правильного драйвера. Возможна также загрузка всех звуковых драйверов через `/boot/loader.conf`.

**Замечание:** В FreeBSD 4.X для загрузки всех звуковых драйверов вам будет необходимо загрузить модуль `snd` вместо `snd_driver`.

Другой способ заключается в добавлении статического драйвера в ядро. В разделе ниже дана более подробная информация о том, что вам нужно сделать для добавления поддержки оборудования. Более подробно о конфигурации ядра описана в Гл. 8.

#### 7.2.1.1. Настройка собственного ядра с поддержкой звука

Первое, что необходимо сделать, это добавить в ядро общий звуковой драйвер `sound(4)`. Добавьте в файл конфигурации ядра следующую строку:

```
device sound
```

В FreeBSD 4.X, вам необходимо использовать следующую строку:

```
device pcm
```

Затем необходимо добавить поддержку имеющейся звуковой карты. Следовательно, нужно знать какой драйвер поддерживает карту. Для этого сверьтесь со списком поддерживаемых устройств из Информации об оборудовании (<http://www.FreeBSD.org/releases/5.3R/hardware.html>). Например, звуковая карта Creative SoundBlaster Live! поддерживается драйвером `snd_emu10k1(4)`. Для добавления поддержки этой карты, используйте:

```
device "snd_emu10k1"
```

Прочтите страницу справочника драйвера, чтобы узнать, какой синтаксис использовать. Информация, относящаяся к синтаксису включения звуковых драйверов в файл конфигурации ядра, может быть также найдена в файле `(/usr/src/sys/i386/conf/LINT` для FreeBSD 4.X).

Не-PnP ISA карты могут потребовать включения в ядро информации о настройках звуковой карты (IRQ, I/O port, и т.д.). Эта информация добавляется редактированием файла `/boot/device.hints`. Во время загрузки системы `loader(8)` прочтет этот файл и настройки ядру. Например, старая ISA не-PnP карта Creative SoundBlaster 16 использует драйвер `snd_sbc(4)` со следующей строкой, добавленной к файлу настройки ядра:

```
device snd_sbc
```

и со следующей информацией в `/boot/device.hints`:

```
hint.sbc.0.at="isa"
hint.sbc.0.port="0x220"
hint.sbc.0.irq="5"
hint.sbc.0.drq="1"
hint.sbc.0.flags="0x15"
```

В данном случае, карта использует порт ввода-вывода `0x220` и `IRQ 5`.

Синтаксис, используемый в файле `/boot/device.hints`, описан в справочной странице драйвера. В FreeBSD 4.X эти настройки записываются непосредственно в файл конфигурации ядра. В случае ISA карты используется такая строка:

```
device sbc0 at isa? port 0x220 irq 5 drq 1 flags 0x15
```

Установки, приведенные выше, используются по умолчанию. В некоторых случаях вам может потребоваться изменить `IRQ` или другие настройки в соответствии с настройками карты. За более подробной информацией обратитесь к странице справочника `snd_sbc(4)`.

**Замечание:** В FreeBSD 4.X некоторые системы со встроенной в материнскую плату звуковой картой могут потребовать следующей строки в файле настройки ядра:

```
options PNPBIOS
```

## 7.2.2. Тестирование звуковой карты

После перезагрузки модифицированного ядра, или после загрузки необходимого модуля, звуковая карта должна появиться в буфере системных сообщений (`dmesg(8)`) примерно так:



```
pcm0: <Intel ICH3 (82801CA)> port 0xdc80-0xdcbf,0xd800-0xd8ff irq 5 at device 31.5 on pci0
pcm0: [GIANT-LOCKED]
pcm0: <Cirrus Logic CS4205 AC97 Codec>
```

Статус звуковой карты может быть проверен через файл `/dev/sndstat`:

```
# cat /dev/sndstat
FreeBSD Audio Driver (newpcm)
Installed devices:
pcm0: <Intel ICH3 (82801CA)> at io 0xd800, 0xdc80 irq 5 bufsz 16384
kld snd_ich (1p/2r/0v channels duplex default)
```

Вывод этой команды для вашей системы может отличаться. Если устройства `pcm` не появились, вернитесь назад и проверьте выполненные действия. Проверьте файл настройки ядра еще раз и убедитесь, что выбрано подходящее устройство. Часто встречающиеся проблемы приведены в Разд. 7.2.2.1.

Если всё пройдет удачно, звуковая карта заработает. Если CD-ROM или DVD-ROM привод правильно подключён к звуковой карте, вы можете вставить CD в привод и воспроизвести его при помощи `cdcontrol(1)`.

```
% cdcontrol -f /dev/acd0 play 1
```

Различные приложения, например `audio/workman` может предоставить более дружелюбный пользователю интерфейс. Вы можете также установить приложения для прослушивания звуковых файлов MP3, как например `audio/mpg123`. Быстрым способом тестирования звуковой карты является отправка данных в файл `/dev/dsp`, как здесь:

```
% cat filename > /dev/dsp
```

где *filename* может быть любым файлом. Результатом выполнения этой команды станет шум, который означает, что звуковая карта на самом деле работает.

**Замечание:** Пользователям FreeBSD 4.X потребуется создать устройства звуковой карты перед тем, как ее использовать. Если в буфере системных сообщений карта появилась как `pcm0`, вам необходимо выполнить от `root` следующее:

```
# cd /dev
# sh MAKEDEV snd0
```

Если карта обнаружена как `pcm1`, выполните то же самое, заменив `snd0` на `snd1`.

`MAKEDEV` создаст группу файлов устройств, которые будут использоваться различными звуковыми приложениями.

Уровни громкости звука могут быть изменены командой `mixer(8)`. Более подробная информация находится на странице справочной системы `mixer(8)`.

### 7.2.2.1. Часто встречающиеся проблемы

Ошибка	Решение
--------	---------

Ошибка	Решение
unsupported subdevice XX	Одно или более устройств не были правильно созданы. Повторите приведенные выше шаги.
sb_dspwr(XX) timed out	Порт ввода-вывода указан неправильно.
bad irq XX	IRQ установлен неправильно. Убедитесь, что настройки в системе и на карте одинаковы.
xxx: gus pcm not attached, out of memory	Для использования устройства недостаточно памяти.
xxx: can't open /dev/dsp!	Проверьте с помощью <code>fstat   grep dsp</code> , не занято ли устройство другим приложением. Создать проблемы могут <b>esound</b> и поддержка звука в <b>KDE</b> .

### 7.2.3. Использование нескольких источников звука

*Предоставил Munish Chopra.*

Достаточно часто встречается необходимость иметь несколько источников звука, которые должны воспроизводить одновременно, например когда **esound** или **artsd** не поддерживают совместное использование звукового устройства с некоторым приложением.

FreeBSD позволяет делать это при помощи *виртуальных звуковых каналов*, которые могут быть настроены с помощью `sysctl(8)`. Виртуальные каналы позволяют вам мультиплексировать каналы воспроизведения звуковой карты, смешивая звук в ядре.

Для установки количества виртуальных каналов вы можете использовать две переменные `sysctl`, которые, если вы пользователь `root`, могут быть установлены таким образом:

```
# sysctl hw.snd.pcm0.vchans=4
# sysctl hw.snd.maxautovchans=4
```

В этом примере выделяются четыре виртуальных канала, чего вполне достаточно для повседневного использования. `hw.snd.pcm0.vchans` это количество виртуальных каналов устройства `pcm0`, оно может быть установлено сразу же, как только устройство было подключено. `hw.snd.maxautovchans` это количество виртуальных каналов, которые выделяются новому аудио устройству, когда оно подключается при помощи `kldload(8)`. Так как модуль `pcm` может быть загружен независимо от аппаратных драйверов, `hw.snd.maxautovchans` может указывать количество виртуальных каналов для любых устройств, которые будут подключены позже.

Если вы не используете `devfs(5)`, необходимо будет указать приложениям `/dev/dsp0.x`, где `x` это число от 0 до 3, если `hw.snd.pcm0.vchans` установлено в значение 4. Для системы, использующей `devfs(5)`, вышеуказанные настройки будут сделаны автоматически прозрачно для пользователя.

### 7.2.4. Установка значений по умолчанию для каналов **mixer**

*Предоставил Josef El-Rayes.*

Значения по умолчанию для различных каналов **mixer** жестко прописаны в исходных текстах драйвера

pcm(4). Существует множество различных приложений и демонов, которые позволяют устанавливать значения для mixer, которые они запоминают и выставляют каждый раз при запуске, но это не совсем правильное решение, нам нужны значения по умолчанию на уровне драйвера. Они могут быть установлены путем указания в /boot/device.hints. Например:

```
hint.pcm.0.vol="100"
```

Установит значение для канала volume в значение по умолчанию 100, как только будет загружен модуль pcm(4).

**Замечание:** Только FreeBSD 5.3 и более свежие релизы поддерживают эту возможность.

## 7.3. Звук MP3

*Предоставил Chern Lee.*

MP3 (MPEG Layer 3 Audio) достигает качества звука, близкого к CD, и нет причин не воспользоваться им на вашей рабочей станции.

### 7.3.1. Проигрыватели MP3

На данный момент наиболее популярным MP3-проигрывателем для **XFree86** является **XMMS** (X Multimedia System). Скины приложения **WinAMP** могут быть использованы для **XMMS** так как графический интерфейс пользователя практически идентичен интерфейсу программы **WinAMP** от Nullsoft. **XMMS** поддерживает также собственные расширения.

**XMMS** может быть установлен из порта или пакета multimedia/xmms.

Интерфейс **XMMS** интуитивно понятен и включает в себя список песен, графический эквалайзер и многое другое. Те, кто знаком с **WinAMP** найдут **XMMS** очень простым в использовании.

Порт audio/mpg123 является альтернативой, это MP3-проигрыватель для командной строки.

**mpg123** может быть запущен с указанием звукового устройства и файла MP3 в командной строке как показано ниже:

```
# mpg123 -a /dev/dsp1.0 Foobar-GreatestHits.mp3
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layer 1, 2 and 3.
Version 0.59r (1999/Jun/15). Written and copyrights by Michael Hipp.
Uses code from various people. See 'README' for more!
THIS SOFTWARE COMES WITH ABSOLUTELY NO WARRANTY! USE AT YOUR OWN RISK!
```

```
Playing MPEG stream from Foobar-GreatestHits.mp3 ...
MPEG 1.0 layer III, 128 kbit/s, 44100 Hz joint-stereo
```

/dev/dsp1.0 должно быть заменено соответствующим устройством dsp для вашей системы.

### 7.3.2. Копирование аудио дорожек с CD

Перед тем как преобразовывать CD или дорожку CD в MP3, аудио данные на CD должны быть скопированы на жёсткий диск. Это можно сделать путём копирования данных CDDA (CD Digital Audio) в файл WAV.

Утилита `cdda2wav`, которая является частью пакета `sysutils/cdrtools`, может быть использована для копирования аудио информации с CD, а также различной связанной информации.

Когда музыкальный CD находится в приводе, следующая команда может быть выполнена под `root` для того, чтобы скопировать весь CD в отдельные (один на каждую дорожку) WAV файлы:

```
# cdda2wav -D 0,1,0 -B
```

`cdda2wav` поддерживает ATAPI (IDE) приводы CDRom. Для копирования с IDE привода, укажите имя устройства вместо номеров SCSI. Например, для того, чтобы скопировать 7-ую аудио дорожку с IDE-привода:

```
# cdda2wav -D /dev/acd0a -t 7
```

Параметр `-D 0,1,0` указывает устройство SCSI 0,1,0, соответственно результату работы `cdrecord -scanbus`.

Для того, чтобы копировать отдельные дорожки, используйте параметр `-t` как показано ниже:

```
# cdda2wav -D 0,1,0 -t 7
```

Этот пример показывает как скопировать 7-ю дорожку музыкального CD. Для того чтобы скопировать набор дорожек, например, с первой по седьмую, укажите диапазон:

```
# cdda2wav -D 0,1,0 -t 1+7
```

Утилита `dd(1)` также может быть использована для копирования аудио дорожек на приводах ATAPI, для того, чтобы узнать больше об этом, прочитайте Разд. 16.6.5.

### 7.3.3. Создание (encoding) файлов MP3

На сегодняшний день наилучшим выбором программы для создания mp3 является **lame**. **Lame** находится в дереве портов в подкаталоге `audio/lame`.

Используя скопированные файлы WAV следующая команда преобразует `audio01.wav` в `audio01.mp3`:

```
# lame -h -b 128 \
--tt "Foo Song Title" \
--ta "FooBar Artist" \
--tl "FooBar Album" \
--ty "2001" \
--tc "Ripped and encoded by Foo" \
--tg "Genre" \
audio01.wav audio01.mp3
```

Частота 128 килобит является стандартом "де факто" для MP3. Многие, однако, используют более высокие частоты для получения лучшего качества, 160 или 192 килобита. Чем выше частота, тем больше дискового пространства будет занимать получаемый MP3, но качество будет выше. Параметр

-h включает режим "лучшее качество, но меньше скорость". Параметры, начинающиеся с --t указывают теги ID3, которые обычно содержат информацию о песне, включаемую в файл MP3. О дополнительных настройках преобразования можно узнать, прочитав страницу руководства lame.

### 7.3.4. Декодирование MP3

Для того, чтобы записать музыкальный CD из файлов MP3, они должны быть преобразованы в несжатый формат WAV. Как **XMMS**, так и **mpg123** поддерживают вывод MP3 в распакованный формат файлов.

Запись на диск в **XMMS**:

1. Запустите **XMMS**:
2. Нажмите правой кнопкой мыши в главном окне **XMMS** для того, чтобы показать меню.
3. Выберите Preferences (либо **Свойства**, если у вас локализованная версия **XMMS**) в Options.
4. Измените расширение вывода на "Disk Writer Plugin" (или "Расширение записи на диск", если у вас локализованная версия **XMMS**).
5. Нажмите Configure (или "Настройка", если у вас локализованная версия **XMMS**).
6. Введите (или выберите при помощи обзора) каталог, в который следует сохранять распакованные файлы.
7. Загрузите файл MP3 в **XMMS** как вы это делаете обычно. Установите громкость на 100% и отключите эквалайзер.
8. Нажмите **Воспроизвести** — **XMMS** будет выглядеть так же как и при обычном воспроизведении MP3, но самой музыки слышно не будет. На самом деле MP3 воспроизводится в файл.
9. Убедитесь, что вы установили расширение вывода таким, как оно было до этого, для того, чтобы снова слушать MP3.

Запись в stdout в **mpg123**:

1. Запустите `mpg123 -s audio01.mp3 > audio01.pcm`

**XMMS** записывает файл в формате WAV, в то время как **mpg123** преобразовывает MP3 в простые аудио данные PCM. Оба формата могут быть использованы **cdrecord** для создания музыкальных CD. Для использования **burncd(8)** вам потребуются простые аудио данные PCM. Если же вы будете использовать файлы в формате WAV, то заметите небольшой щелчок в начале каждой аудио дорожки, этот щелчок - заголовок файла в формате WAV. Вы очень просто можете избавиться от него путём удаления заголовка WAV при помощи утилиты **SoX** (она может быть установлена из порта audio/sox или соответствующего пакета:

```
% sox -t wav -r 44100 -s -w -c 2 track.wav track.raw
```

Прочтите Разд. 16.6 для того, чтобы узнать больше о записи CD в FreeBSD.

## 7.4. Воспроизведение видео

*Предоставил Ross Lippert.*

Воспроизведение видео является очень новой и быстро развивающейся областью применения. Будьте терпеливы. Не всё будет работать так бесппроблемно, как это было со звуком.

Прежде, чем вы начнёте, определите модель видеокарты и чипсет, который она использует. Хотя **Xorg** и **XFree86** поддерживают множество различных видеокарт, только их малая часть показывает хорошую скорость воспроизведения видео. Для того, чтобы получить список расширений, поддерживаемых X-сервером, который используется вашей видеокартой, используйте команду `xdpyinfo(1)` во время работы X11.

Неплохо также иметь небольшой файл MPEG, который бы использовался как тестовый файл для проверки различных проигрывателей и настроек. Так как некоторые проигрыватели DVD будут искать носитель DVD как `/dev/dvd` по умолчанию или быть жёстко настроены на него, возможно будет полезно сделать символические ссылки на правильные устройства:

```
# ln -sf /dev/acd0c /dev/dvd
# ln -sf /dev/racd0c /dev/rdvd
```

Для FreeBSD 5.X, которая использует `devfs(5)` рекомендуется немного другой набор ссылок:

```
# ln -sf /dev/acd0 /dev/dvd
# ln -sf /dev/acd0 /dev/rdvd
```

Обратите внимание, природа `devfs(5)` такова, что такие созданные вручную ссылки не сохраняются при перезагрузке системы. Для автоматического создания символических ссылок при каждой загрузке системы добавьте в `/etc/devfs.conf` следующие строки:

```
link acd0 dvd
link acd0 rdvd
```

Кроме того, декодирование DVD требует доступа к специальным функциям DVD-ROM, поэтому должен быть доступ на запись для устройств DVD.

Некоторые из обсуждаемых портов полагаются на наличие некоторых параметров ядра для правильной сборки. Перед тем, как собирать порты, добавьте эти параметры в файл конфигурации ядра, пересоберите ядро и перезагрузите систему:

```
option CPU_ENABLE_SSE
option USER_LDT
```

**Замечание:** `option USER_LDT` в FreeBSD 5.X не существует.

Для того, чтобы улучшить работу разделяемой памяти X11, рекомендуется увеличить значения некоторых переменных `sysctl(8)`:

```
kern.ipc.shmmax=67108864
kern.ipc.shmall=32768
```

### 7.4.1. Определение возможностей видео

Существует несколько возможных путей отображения видео под X11. Что именно будет действительно работать, во многом зависит от аппаратного обеспечения. Каждый из описанных методов будет работать с различным качеством на разном аппаратном обеспечении. Во-вторых, воспроизведение видео в X11, это тема, которой уделяется достаточно много внимания последнее время, и с каждой новой версией **Xorg** или **XFree86** могут наблюдаться значительные улучшения.

Список наиболее часто используемых видеоинтерфейсов:

1. X11: обычный вывод X11 с использованием разделяемой памяти.
2. XVideo: расширение интерфейса X11, которое поддерживает видео в любом объекте X11.
3. SDL: the Simple Directmedia Layer.
4. DGA: the Direct Graphics Access — Прямой Доступ для Графики.
5. SVGAlib: низкоуровневый доступ к графике на консоли.

#### 7.4.1.1. XVideo

**Xorg** и **XFree86 4.X** включают в себя расширение, называющееся *XVideo* (также известное как Xvideo, Xv и xv), которое позволяет отображать видео прямо на объектах при помощи специального ускорения. Это расширение предоставляет очень хорошее качество воспроизведения даже на low-end машинах.

Для того чтобы проверить, работает ли это расширение, используйте команду `xvinfo`:

```
% xvinfo
```

XVideo поддерживается вашей видеокартой, если результат выглядит приблизительно так:

```
X-Video Extension version 2.2
screen #0
  Adaptor #0: "Savage Streams Engine"
    number of ports: 1
    port base: 43
    operations supported: PutImage
    supported visuals:
      depth 16, visualID 0x22
      depth 16, visualID 0x23
    number of attributes: 5
      "XV_COLORKEY" (range 0 to 16777215)
        client settable attribute
        client gettable attribute (current value is 2110)
      "XV_BRIGHTNESS" (range -128 to 127)
        client settable attribute
        client gettable attribute (current value is 0)
      "XV_CONTRAST" (range 0 to 255)
        client settable attribute
        client gettable attribute (current value is 128)
      "XV_SATURATION" (range 0 to 255)
        client settable attribute
        client gettable attribute (current value is 128)
      "XV_HUE" (range -180 to 180)
```

```

        client settable attribute
        client gettable attribute (current value is 0)
maximum XvImage size: 1024 x 1024
Number of image formats: 7
  id: 0x32595559 (YUY2)
    guid: 59555932-0000-0010-8000-00aa00389b71
    bits per pixel: 16
    number of planes: 1
    type: YUV (packed)
  id: 0x32315659 (YV12)
    guid: 59563132-0000-0010-8000-00aa00389b71
    bits per pixel: 12
    number of planes: 3
    type: YUV (planar)
  id: 0x30323449 (I420)
    guid: 49343230-0000-0010-8000-00aa00389b71
    bits per pixel: 12
    number of planes: 3
    type: YUV (planar)
  id: 0x36315652 (RV16)
    guid: 52563135-0000-0000-0000-000000000000
    bits per pixel: 16
    number of planes: 1
    type: RGB (packed)
    depth: 0
    red, green, blue masks: 0x1f, 0x3e0, 0x7c00
  id: 0x35315652 (RV15)
    guid: 52563136-0000-0000-0000-000000000000
    bits per pixel: 16
    number of planes: 1
    type: RGB (packed)
    depth: 0
    red, green, blue masks: 0x1f, 0x7e0, 0xf800
  id: 0x31313259 (Y211)
    guid: 59323131-0000-0010-8000-00aa00389b71
    bits per pixel: 6
    number of planes: 3
    type: YUV (packed)
  id: 0x0
    guid: 00000000-0000-0000-0000-000000000000
    bits per pixel: 0
    number of planes: 0
    type: RGB (packed)
    depth: 1
    red, green, blue masks: 0x0, 0x0, 0x0

```

Следует заметить, что перечисленные форматы (YUV2, YUV12 и т.п.) не присутствуют в каждой реализации XVideo и их отсутствие может быть помехой для некоторых проигрывателей.

Если результат выглядит так:

```

X-Video Extension version 2.2
screen #0
no adaptors present

```



то, возможно, XVideo не поддерживается для вашей видеокарты.

Если XVideo не поддерживается вашей видеокартой, то это всего лишь означает, что будет сложнее получить приемлемые для воспроизведения видео вычислительные мощности. В зависимости от вашей видеокарты и процессора, возможно, вы сможете получить удовлетворительный результат. Возможно, вы должны будете прочитать о путях улучшения производительности здесь: Разд. 7.4.3.

#### 7.4.1.2. Simple Directmedia Layer

SDL был задуман как уровень абстракции для разработки кроссплатформенных приложений под Microsoft Windows, BeOS и UNIX, позволяя им эффективно использовать звук и графику. SDL предоставляет низкоуровневые абстракции для аппаратного обеспечения, что, зачастую, может быть более эффективным чем интерфейс X11.

SDL есть в Коллекции портов FreeBSD: `devel/sdl12`.

#### 7.4.1.3. Прямой доступ для графики (DGA)

DGA это расширение **XFree86** которое позволяет программам напрямую изменять кадровый буфер (framebuffer) без участия X-сервера. Поскольку DGA основывается на низкоуровневом доступе к памяти, программы, которые используют его должны исполняться от пользователя `root`.

Расширение DGA может быть протестировано при помощи `dga(1)`. Когда `dga` запущена, она изменяет цвета на экране при каждом нажатии клавиш. Для того, чтобы выйти из неё, используйте `q`.

### 7.4.2. Порты и пакеты для работы с видео

Этот раздел обсуждает программное обеспечение для работы с видео из Коллекции Портов FreeBSD. Воспроизведение видео является очень активной сферой разработок программного обеспечения и возможности различных приложений могут несколько отличаться от описанных здесь.

Во-первых, важно помнить, что многие приложения для работы с видео, которые работают на FreeBSD, были разработаны как приложения Linux. Многие из этих приложений все еще бета-качества. Вот некоторые проблемы, которые могут встретиться в работе видео пакетов на FreeBSD:

1. Приложение не может воспроизвести файл, который создало другое приложение.
2. Приложение не может воспроизвести файл, который создало само.
3. Одно и то же приложение на разных машинах, скомпилированное на каждой машине специально для неё, воспроизводит один и тот же файл различно.
4. Кажущийся тривиальным фильтр, например фильтр изменения размеров изображения, приводит к очень плохим "артефактам" из-за неправильной функции изменения размера.
5. Приложение часто не работает (оставляет core-файл).
6. Документация не устанавливается вместе с портом и может быть найдена лишь на сайте или в каталоге порта `work`.

Многие из этих приложений могут также проявлять "линуксизмы". Так, это могут быть некоторые проблемы, связанные со способом реализации некоторых стандартных библиотек в дистрибутивах

Linux, или некоторыми дополнительными возможностями ядра Linux, которые авторы приложений посчитали существующими везде. Эти проблемы не всегда могут быть обнаружены людьми, поддерживающими порт (порты), вследствие чего могут возникнуть проблемы, сходные с нижеперечисленными:

1. Использование `/proc/cpuinfo` для того, чтобы определить характеристики процессора.
2. Неправильное использование нитей (threads), которое может привести к зависанию программы при завершении вместо нормального выхода.
3. Программного обеспечения, которое обычно используется совместно с данным приложением, ещё нет в Коллекции Портов FreeBSD.

Таким образом, разработчики этих приложений должны сотрудничать с людьми, поддерживающими порты, для того, чтобы минимизировать количество обходных путей, необходимых для портирования.

#### 7.4.2.1. MPlayer

**MPlayer** это недавно разработанный и быстро развивающийся проигрыватель видео. Задачами команды разработчиков **MPlayer** являются скорость и гибкость при работе на Linux и других Unix-системах. Проект был начал, когда его основатель стал сыт по горло плохой производительностью и качеством проигрывателей того времени. Некоторые могут сказать, что графический интерфейс был принесён в жертву рационализированному дизайну. Однако, как только вы привыкнете к опциям командной строки MPlayer и его управлению с клавиатуры, всё будет хорошо.

##### 7.4.2.1.1. Компиляция MPlayer

**MPlayer** находится в `multimedia/mplayer`. MPlayer производит различные тесты аппаратного обеспечения во время процесса компиляции, в результате чего полученные исполняемые модули не могут быть перенесены с одной системы на другую. Поэтому важно собирать его из портов, а не использовать бинарный пакет. Также, при сборке вы можете указать различные установки при помощи параметров командной строки `make`, как описывается в `Makefile` в начале сборки:

```
# cd /usr/ports/multimedia/mplayer
# make
N - O - T - E
```

```
Take a careful look into the Makefile in order
to learn how to tune mplayer towards you personal preferences!
For example,
make WITH_GTK1
builds MPlayer with GTK1-GUI support.
If you want to use the GUI, you can either install
/usr/ports/multimedia/mplayer-skins
or download official skin collections from
http://www.mplayerhq.hu/homepage/dload.html
```

Параметры порта по умолчанию должны подходить большинству пользователей. Однако, если вам необходим кодек XviD, необходимо указать в командной строке параметр `WITH_XVID`. Устройство DVD по умолчанию также может быть указано в командной строке параметром `WITH_DVD_DEVICE`, по умолчанию используется `/dev/acd0`.

На время написания данного документа порт **MPlayer'a** собирает и устанавливает свою документацию в формате HTML и два исполняемых файла, `mplayer` и `mencoder`, который является утилитой для перекодировки видео.

Документация к **MPlayer** очень информативна. Если читатель найдет информацию этой главы о аппаратном обеспечении для поддержки видео и интерфейсам недостаточной, то документация **MPlayer** будет очень хорошим дополнением. Обязательно уделите время чтению документации **MPlayer**, если вам нужна информация о поддержке видео под UNIX.

#### 7.4.2.1.2. Использование MPlayer

Каждый пользователь **MPlayer** должен создать подкаталог `.mplayer` в своем домашнем каталоге. Для того, чтобы его создать, выполните следующие действия:

```
% cd /usr/ports/multimedia/mplayer
% make install-user
```

Параметры для `mplayer` перечислены в страничке руководства `mplayer`. За более подробной информацией вы можете обратиться к документации в формате HTML. В этом разделе мы опишем несколько самых распространённых случаев использования `mplayer`.

Для того, чтобы воспроизвести файл, например `testfile.avi` через один из многих видеointерфейсов, используйте параметр `-vo`:

```
% mplayer -vo xv testfile.avi
% mplayer -vo sdl testfile.avi
% mplayer -vo x11 testfile.avi
# mplayer -vo dga testfile.avi
# mplayer -vo 'sdl:dga' testfile.avi
```

Стоит попробовать все варианты интерфейсов, так как их производительность зависит от множества факторов и будет заметно меняться в зависимости от аппаратного обеспечения.

Для того, чтобы воспроизвести DVD, замените `testfile.avi` на `dvd://N -dvd-device DEVICE`, где `N` является номером дорожки, с которой следует начать воспроизведение и `DEVICE` файл устройства привода DVD. Например, для того, чтобы воспроизвести дорожку 3 с `/dev/dvd`:

```
# mplayer -vo xv dvd://3 -dvd-device /dev/dvd
```

**Замечание:** Устройство DVD по умолчанию может быть определено во время сборки порта **MPlayer** параметром `WITH_DVD_DEVICE`. По умолчанию, это устройство `/dev/acd0`. Дополнительную информацию можно найти в `Makefile` порта.

Для того, чтобы остановить, приостановить или продолжить воспроизведение, воспользуйтесь привязкой клавиш, информация о которой может быть получена посредством запуска `mplayer -h`, либо на страничке документации.

Дополнительные, достаточно важные параметры воспроизведения: `-fs -zoom`, которые включают полноэкранный режим и `-framedrop`, который улучшает производительность на медленных системах.

Для того, чтобы командная строка запуска `mplayer` не становилась слишком большой, пользователь может создать файл `.mplayer/config` и установить параметры по умолчанию там:

```
vo=xv
fs=yes
zoom=yes
```

Также `mplayer` может быть использован для копирования дорожек DVD в `.vob` файлы. Для того, чтобы скопировать вторую дорожку DVD необходимо выполнить следующую команду:

```
# mplayer -dumpstream -dumpfile out.vob dvd://2 -dvd-device /dev/dvd
```

Полученный файл, `out.vob`, будет представлять собой MPEG, с которым можно производить различные действия при помощи программ, которые будут описаны далее в этом разделе.

#### 7.4.2.1.3. mencoder

Перед использованием `mencoder`, было бы неплохо ближе ознакомиться с его параметрами, используя документацию в формате HTML. Также существует страничка справочника `mplayer`, но она не очень полезна без HTML документации. Существует бесчисленное множество способов улучшения качества, снижения битрейта и изменения формата; и некоторые из этих приёмов могут влиять на производительность. Ниже приведено несколько примеров использования `mencoder`. Во-первых, простое копирование:

```
% mencoder input.avi -oac copy -ovc copy -o output.avi
```

Неправильная комбинация параметров командной строки привести к появлению файлов, которые невозможно будет воспроизвести даже `mplayer`. Поэтому, если вы хотите скопировать в файл, лучше использовать только параметр `mplayer -dumpfile`.

Для того, чтобы преобразовать `input.avi` в MPEG4 со звуком в формате MPEG Audio Layer 3 (MP3) (требуется `audio/lame`):

```
% mencoder input.avi -oac mp3lame -lameopts br=192 \
    -ovc lavc -lavcopts vcodec=mpeg4:vhq -o output.avi
```

Эта команда создаст файл, воспроизводимый `mplayer` и `xine`.

`input.avi` может быть заменён на `dvd://1 -dvd-device /dev/dvd` и `mplayer`, запущенный от пользователя `root`, будет преобразовывать дорожку DVD напрямую. Так как первый раз, скорее всего, вы будете недовольны полученными результатами, всё же рекомендуется копировать дорожку в файл и работать затем с файлом.

#### 7.4.2.2. Проигрыватель xine

**xine** - это большой проект, в задачи которого входит не только создание решения для видео все-в-одном, но и создание базовой библиотеки с возможностью расширения возможностей путем

использования плагинов (расширений). Поставляется он как в виде порта, так и в виде пакета, multimedia/xine.

**xine** все еще несовершенен, но, все-таки это хорошее начало. На практике **xine** требует либо быстрого процессора с быстрой видеокартой или поддержку расширения XVideo. Графический интерфейс можно использовать, но он все еще немного неуклюж.

На время написания этого документа в поставке **xine** не существовало модуля ввода, который бы мог воспроизводить DVD, закодированные по алгоритму CSS. Существуют сборки, в которых есть такой модуль, но ни одна из них не входит в Коллекцию Портов FreeBSD.

По сравнению с **MPlayer**, **xine** является более дружелюбным к пользователю, но, в то же время, скрывает более тонкие настройки и управление от пользователя. Также **xine** лучше работает на XVideo интерфейсах.

По умолчанию, **xine** запускается с графическим интерфейсом. Для открытия файлов используются меню.

```
% xine
```

В качестве альтернативы можно использовать его для запуска файла непосредственно, без GUI, следующей командой:

```
% xine -g -p mymovie.avi
```

### 7.4.2.3. Утилиты transcode

Приложение **transcode** не является проигрывателем. Это набор инструментов для преобразования .avi и .mpeg файлов. При помощи **transcode** можно объединять видеофайлы, исправлять поврежденные файлы, использовать инструменты командной строки для работы с потоками ввода/вывода stdin/stdout.

Так же, как и **MPlayer**, **transcode** является очень экспериментальным программным продуктом, который необходимо собирать из порта multimedia/transcode. Этот порт имеет огромное количество параметров для команды make при сборке. Рекомендуется:

```
# make WITH_LIBMPEG2=yes
```

Если вы планируете установить multimedia/avifile, то добавьте параметр WITH\_AVIFILE в строку запуска make как показано ниже:

```
# make WITH_AVIFILE=yes WITH_LIBMPEG2=yes
```

Ниже приводятся два примера использования transcode для преобразования видео, в результате которого изменяется размер изображения. Первый пример преобразовывает данные в openDIVX AVI файл, в то время как второй использует более переносимый формат MPEG.

```
% transcode -i input.vob -x vob -V -Z 320x240 \
-y opendivx -N 0x55 -o output.avi
```

```
% transcode -i input.vob -x vob -V -Z 320x240 \
-y mpeg -N 0x55 -o output.tmp
```

```
% tcmplex -o output.mpg -i output.tmp.mlv -p output.tmp.mpa -m 1
```

Существует страничка справочника для transcode, но для различных утилит tc\* (таких как tcsmplex), которые тоже устанавливаются в систему, документации очень немного. Тем не менее, для получения краткой справки по использованию команд, может быть задан параметр -h.

По сравнению с mencoder, transcode работает заметно медленнее, но имеет больше шансов создать более переносимый файл, который можно будет воспроизвести на большем количестве систем. Например MPEG, созданные при помощи transcode, воспроизводятся в **Windows Media® Player** и **Apple Quicktime®**.

### 7.4.3. Дальнейшее чтение

Различные пакеты видео программ для FreeBSD интенсивно разрабатываются. Очень возможно, что в ближайшем будущем многие обсуждаемые здесь проблемы разрешатся. Это займет время, и те, кто желает получить максимум от аудио/видео возможностей FreeBSD, должны будут собирать необходимые знания из нескольких списков часто задаваемых вопросов и обучающих статей, а также использовать различные приложения. Этот раздел существует для того, чтобы читатель мог получить указания на несколько источников дополнительной информации.

Документация **MPlayer** (<http://www.mplayerhq.hu/DOCS/>) очень содержательна в техническом плане. Возможно, эти документы должны использоваться любым человеком, желающим получить высокий уровень знаний о видео на UNIX системах. Список рассылки **MPlayer** враждебен для любого, кто не потрудился прочитать документацию, так что, если у вас есть желание сообщать о найденных ошибках, прочитайте вначале документацию.

**xine** HOWTO ([http://dvd.sourceforge.net/xine-howto/en\\_GB/html/howto.html](http://dvd.sourceforge.net/xine-howto/en_GB/html/howto.html)) содержит главу об улучшении производительности, которая применима к любому проигрывателю.

Наконец, существует несколько многообещающих приложений, которые читатель может испробовать:

- Avifile (<http://avifile.sourceforge.net/>), для которого также существует порт `multimedia/avifile`.
- Ogle (<http://www.dtek.chalmers.se/groups/dvd/>), для которого также существует порт `multimedia/ogle`.
- Xtheater (<http://xtheater.sourceforge.net/>)
- `multimedia/dvdauthor`, пакет с открытыми текстами для распространения DVD контента.

## 7.5. Настройка ТВ тюнеров

*Первоначально предоставил Josef El-Rayes. Улучшил и адаптировал Marc Fonvieille.*

### 7.5.1. Введение

ТВ тюнеры предназначены для просмотра широкоэмитательного или кабельного телевидения на компьютере. Большинство тюнеров поддерживают композитный видео вход RCA или S-video, а некоторые из них поставляются с FM радио тюнером.

FreeBSD поддерживает PCI ТВ тюнеры, использующие Brooktree Bt848/849/878/879 или Conexant CN-878/Fusion 878a Video Capture Chip через драйвер bktr(4). Вы должны также убедиться, что тюнер поддерживается; обратитесь к странице справочника bktr(4) за списком поддерживаемых тюнеров.

### 7.5.2. Добавление драйвера

Для использования карты потребуется загрузить драйвер bktr(4), что можно сделать, добавив в `/boot/loader.conf` следующую строку:

```
bktr_load="YES"
```

В качестве альтернативы, вы можете статически скомпилировать ядро с поддержкой ТВ тюнера; добавьте следующие строки в файл конфигурации ядра:

```
device bktr
device iicbus
device iicbb
device smbus
```

Эти дополнительные драйвера устройств необходимы, поскольку компоненты карты соединены через шину I2C. Затем соберите и установите новое ядро.

Как только поддержка тюнера будет добавлена в систему, перезагрузите компьютер. Во время загрузки TV карта должна отобразить примерно такие строки:

```
bktr0: <BrookTree 848A> mem 0xd7000000-0xd7000fff irq 10 at device 10.0 on pci0
iicbb0: <I2C bit-banging driver> on bti2c0
iicbus0: <Philips I2C bus> on iicbb0 master-only
iicbus1: <Philips I2C bus> on iicbb0 master-only
smbus0: <System Management Bus> on bti2c0
bktr0: Pinnacle/Miro TV, Philips SECAM tuner.
```

Конечно, эти сообщения будут различаться на разном оборудовании. Тем не менее, проверьте, что тюнер определяется правильно; возможна перезапись параметров, определенных ядром, с помощью `sysctl(8)` MIB и параметров в файле настройки ядра. Например, если вы хотите указать, что это Philips SECAM тюнер, добавьте следующую строку к файлу настройки ядра:

```
options OVERRIDE_TUNER=6
```

или прямо задайте переменную `sysctl(8)`:

```
# sysctl hw.bt848.tuner=6
```

Обратитесь к странице bktr(4) и файлу `/usr/src/sys/conf/NOTES` за более детальной информацией о доступных параметрах (если вы работаете в FreeBSD 4.X, вместо `/usr/src/sys/conf/NOTES` используйте `/usr/src/sys/i386/conf/LINT`).

### 7.5.3. Полезные приложения

Для использования ТВ тюнера вам потребуется установить одно из следующих приложений:

- multimedia/fxrtv предоставляет возможности ТВ-в-окне и захвата изображений/аудио/видео.
- multimedia/xawtv это также приложение для ТВ тюнера, с теми же, что и у **fxrtv** возможностями.
- misc/alevt раскодирует и отображает видеотекст/телетекст.
- audio/xmradio, приложение для использования с FM радио тюнером, поставляемым с некоторыми ТВ тюнерами.
- audio/wmtune, это удобное приложение для радио тюнеров.

В коллекции портов FreeBSD можно найти и другие приложения.

### 7.5.4. Решение проблем

Если вы столкнулись с какой-либо проблемой, связанной с ТВ тюнером, проверьте в первую очередь поддержку микросхемы захвата видео и тюнера драйвером bktr(4), а также правильность установки параметров. За дальнейшей поддержкой и с вопросами о ТВ тюнере вы можете обращаться в [freebsd-multimedia](http://lists.FreeBSD.org/mailman/listinfo/freebsd-multimedia) (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-multimedia>) и использовать его архивы.

## 7.6. Сканеры

*Написана Marc Fonvieille. Перевод на русский язык: Дмитрий Морозовский.*

### 7.6.1. Введение

FreeBSD, как и прочие современные операционные системы, поддерживает сканеры. Стандартный доступ к сканерам обеспечивается программой **SANE** (Scanner Access Now Easy), обеспечивающей универсальный интерфейс (API) и доступной в коллекции портов FreeBSD. Для общения со сканерами **SANE** использует некоторые драйвера устройств FreeBSD.

FreeBSD поддерживает сканеры с интерфейсом как SCSI, так и USB. Убедитесь, что ваш сканер поддерживается **SANE** перед тем, как приступить к конфигурации. Список устройств, поддерживаемых **SANE**, можно найти здесь (<http://sane-project.org/sane-supported-devices.html>). Кроме того, страница справочника `usanner(4)` также перечисляет поддерживаемые устройства.

### 7.6.2. Конфигурация ядра

Как уже отмечалось, поддерживаются как SCSI, так и USB сканеры. В зависимости от интерфейса вашего сканера требуется поддержка разных драйверов устройств.

#### 7.6.2.1. USB

Стандартное ядро GENERIC включает в себя драйвера, необходимые для поддержки USB сканеров. Если вы компилируете собственное ядро, убедитесь, что в его конфигурации присутствуют строки

```
device usb
device uhci
device ohci
```



```
device usscanner
```

В зависимости от чипсета USB, встроенного в вашу материнскую плату, потребуется лишь один из драйверов `device uhci` или `device ohci`, однако, наличие обеих строк в конфигурации ядра никому не повредит.

Если вы не хотите перестраивать ядро, и при этом ваше ядро не является стандартным (GENERIC), вы можете загрузить модуль драйвера поддержки сканеров `uscanner(4)` при помощи команды `kldload(8)`:

```
# kldload usscanner
```

Для автоматической загрузки модуля при старте системы добавьте в файл `/boot/loader.conf` строку

```
uscanner_load="YES"
```

После перезагрузки с новым ядром или загрузки модуля подключите ваш USB сканер. В буфере системных сообщений (`dmesg(8)`) должно появиться сообщение о найденном устройстве, например:

```
uscanner0: EPSON EPSON Scanner, rev 1.10/3.02, addr 2
```

В данном случае сканер будет использовать устройство `/dev/uscanner0`.

**Замечание:** В FreeBSD 4.X для корректного распознавания некоторых USB устройств должен быть запущен демон USB (`usbd(8)`). Для этого добавьте строку `usbd_enable="YES"` в файл конфигурации системы `/etc/rc.conf`.

### 7.6.2.2. SCSI

Если ваш сканер имеет интерфейс SCSI, важно знать, к какому контроллеру он подключен. В зависимости от контроллера потребуются различные драйвера в файле конфигурации ядра. Стандартное ядро GENERIC поддерживает большинство распространенных SCSI-контроллеров. Внимательно прочитайте файл NOTES (в FreeBSD 4.X файл LINT) и добавьте необходимые строки в файл конфигурации вашего ядра. Помимо строки для драйвера адаптера, вам потребуются следующие строки:

```
device scbus
device pass
```

После сборки и загрузки нового ядра, в буфере системных сообщений должны появиться строки о вашем сканере, например:

```
pass2 at aic0 bus 0 target 2 lun 0
pass2: <AGFA SNAPSCAN 600 1.10> Fixed Scanner SCSI-2 device
pass2: 3.300MB/s transfers
```

Если сканер не был включен в момент загрузки, его можно принудительно опознать, выполнив сканирование SCSI шины при помощи команды `camcontrol(8)`:

```
# camcontrol rescan all
Re-scan of bus 0 was successful
```

```
Re-scan of bus 1 was successful
Re-scan of bus 2 was successful
Re-scan of bus 3 was successful
```

После этого сканер должен появиться в списке устройств:

```
# camcontrol devlist
<IBM DDRS-34560 S97B>          at scbus0 target 5 lun 0 (pass0,da0)
<IBM DDRS-34560 S97B>          at scbus0 target 6 lun 0 (pass1,da1)
<AGFA SNAPSCAN 600 1.10>      at scbus1 target 2 lun 0 (pass3)
<PHILIPS CDD3610 CD-R/RW 1.00> at scbus2 target 0 lun 0 (pass2,cd0)
```

More details about SCSI devices, are available in the `scsi(4)` and `camcontrol(8)` manual pages.

### 7.6.3. Конфигурация SANE

Система **SANE** состоит из двух частей: аппаратной поддержки (`backend`, `graphics/sane-backends`) и программной поддержки (`frontend`, `graphics/sane-frontends`). Первая часть обеспечивает собственно доступ к сканеру. Список поддерживаемых устройств (<http://sane-project.org/sane-supported-devices.html>) **SANE** содержит информацию о необходимом вам аппаратном модуле. Вторая часть обеспечивает графический интерфейс для сканирования (**xscanimage**).

В первую очередь следует установить порт или пакет `graphics/sane-backends`, после чего при помощи команды `sane-find-scanner` проверить поддержку сканера системой **SANE**:

```
# sane-find-scanner -q
found SCSI scanner "AGFA SNAPSCAN 600 1.10" at /dev/pass3
```

В выводе должны присутствовать интерфейс сканера и имя используемого устройства. Производитель и модель сканера могут отсутствовать: это нормально.

**Замечание:** Некоторым USB сканерам может потребоваться загрузка прошивки. Подробности смотрите в страницах справочника драйвера сканера, `sane-find-scanner(1)` и `sane(7)`.

Теперь необходимо убедиться, что сканер опознан программой графического интерфейса. В состав системы **SANE** входит утилита `scanimage(1)`, позволяющая работать со сканером из командной строки. Опция `-L` используется для показа информации о сканере:

```
# scanimage -L
device `snapscan:/dev/pass3' is a AGFA SNAPSCAN 600 flatbed scanner
```

Отсутствие сообщений или сообщение об отсутствии устройств означает, что утилита `scanimage(1)` не смогла идентифицировать сканер. В этом случае вам потребуется отредактировать файл конфигурации аппаратного модуля и указать устройство, используемое сканером. Все файлы настройки находятся в каталоге `/usr/local/etc/sane.d/`. Такие проблемы присущи некоторым моделям USB сканеров.

Например, в случае USB сканера, описанного в разделе Разд. 7.6.2.1, утилита `sane-find-scanner` выдаст следующую информацию:

```
# sane-find-scanner -q
found USB scanner (UNKNOWN vendor and product) at device /dev/usb/lp0
```

Сканер обнаружен корректно, он использует интерфейс USB и доступен через устройство /dev/usb/lp0. Теперь попробуем идентифицировать его:

```
# scanimage -L
```

```
No scanners were identified. If you were expecting something different,
check that the scanner is plugged in, turned on and detected by the
sane-find-scanner tool (if appropriate). Please read the documentation
which came with this software (README, FAQ, manpages).
```

Поскольку сканер не идентифицирован, нам потребуется изменить файл конфигурации /usr/local/etc/sane.d/epson.conf. В нашем примере использован сканер EPSON Perfection® 1650, так что мы знаем, что будет использоваться драйвер epson. Не забудьте прочитать комментарии в файле конфигурации. Требуемые изменения весьма просты: закомментируйте все строки, описывающие интерфейсы, не соответствующий интерфейсу вашего сканера (в нашем случае, все строки, начинающиеся со scsi: наш сканер использует интерфейс USB), и добавьте в конец файла строку, содержащую интерфейс и имя использованного устройства. Мы добавим строку

```
usb /dev/usb/lp0
```

Пожалуйста, прочтите комментарии в файле конфигурации, а также страницы справочника для более полной информации. Теперь мы можем проверить, что наш сканер опознан:

```
# scanimage -L
device 'epson:/dev/usb/lp0' is a Epson GT-8200 flatbed scanner
```

Наш USB сканер опознан. Не столь важно, что имя и номер модели не совпадают, главное, что используются правильные имя устройства и драйвер: 'epson:/dev/usb/lp0'.

После того как команда scanimage -L опознала сканер, конфигурация завершена. Все готово к сканированию.

Хотя утилита scanimage(1) позволяет производить сканирование из командной строки, как правило, для сканирования предпочтительнее использовать графический интерфейс. Для этого в состав **SANE** входит простая, но эффективная утилита **xscanimage** (graphics/sane-frontends).

Другой популярной программой графического интерфейса к сканеру является **Xsane** (graphics/xsane). Эта программа поддерживает такие расширенные возможности, как разные режимы сканирования (фотокопия, факс и т.п.), цветокоррекцию, потоковое сканирование и другие. Оба приложения пригодны для использования в качестве плагинов сканирования для **GIMP**.

#### 7.6.4. Доступ к сканеру для других пользователей

Все описанные операции выполнялись нами с привилегиями суперпользователя (root). Вам может потребоваться дать доступ к сканеру другим пользователям. Для этого необходимо разрешить доступ на чтение и запись к файлу устройства, обслуживающему сканер. В нашем примере USB сканер использует устройство /dev/usb/lp0, принадлежащее группе operator. Добавление пользователя joe в группу operator разрешит ему использовать сканер:

```
# pw groupmod operator -m joe
```

За подробностями обращайтесь к странице справочника `pw(8)`. Вам также потребуется установить нужные права доступа (0660 или 0664) к устройству `/dev/usb/lcd0`, поскольку по умолчанию группа `operator` может лишь читать из него. Это достигается добавлением следующей строки в файл `/etc/devfs.rules`:

```
[system=5]  
add path usb/lcd0 mode 660
```

Затем добавьте в файл конфигурации системы `/etc/rc.conf` такую строку (после чего перезагрузите систему):

```
devfs_system_ruleset="system"
```

Подробную информацию о правах на файлы устройств вы найдете на странице справочника `devfs(8)`. В FreeBSD 4.X группа `operator` по умолчанию имеет права как на чтение, так и на запись, к устройству `/dev/usb/lcd0`.

**Замечание:** Разумеется, по соображениям безопасности, вы должны как следует подумать, прежде чем добавлять пользователя в другие группы, в особенности в группу `operator`.

# Глава 8. Настройка ядра FreeBSD

*Обновлено и реструктуризовано Jim Mock. Предоставлено Jake Hamby. Переведено Александром Коваленко.*

## 8.1. Краткий обзор

Ядро - это основная часть операционной системы FreeBSD. Оно ответственно за управление памятью, параметры безопасности, работу с сетью, доступ к дискам и многое другое. Несмотря на то, что FreeBSD становится всё более динамически конфигурируемой, иногда приходится собирать собственное ядро.

После прочтения этой главы вы узнаете:

- Почему вам может понадобиться сборка собственного ядра.
- Как написать файл конфигурации ядра или изменить существующий.
- Как использовать файл конфигурации ядра для того, чтобы создать и собрать новое ядро.
- Как установить новое ядро.
- Как создать записи в `/dev`, которые могут оказаться необходимы.
- Что делать, если что-то не работает или работает не так, как должно.

Все команды, приводимые в этой главе в качестве примера, должны выполняться от пользователя `root`.

## 8.2. Зачем собирать собственное ядро?

Традиционно в FreeBSD использовалось так называемое "монолитное" ядро. Это означает, что ядро - это одна большая программа, которая поддерживает фиксированный набор устройств и в случае, если необходимо изменить его поведение, требуется сборка нового ядра и перезагрузка компьютера уже с новым ядром.

На сегодняшний день FreeBSD быстро продвигается к модели, в которой большая часть функциональности содержится в модулях, которые могут быть при необходимости динамически загружены и выгружены из ядра. Это позволяет ядру использовать устройства, которые "внезапно" появились в системе (например, устройства PCMCIA в ноутбуке) или добавлять новую функциональность в ядро, которая не была необходима в момент первоначальной сборки ядра. Такой подход известен как модульность ядра.

Несмотря на это, всё ещё иногда бывает необходимо, чтобы некоторая функциональность была вкомпилирована в ядро статически. В некоторых случаях это продиктовано тем, что эта функциональность настолько сильно привязана к ядру, что не может быть динамически загружаемой. В других случаях это может быть просто потому, что никто не уделит время написанию динамически загружаемого модуля для этой функциональности.

Сборка собственного ядра - один из наиболее важных ритуалов, через который должен пройти практически любой пользователь BSD. Несмотря на длительность этого процесса, ваша FreeBSD

останется только в выигрыше. В отличие от ядра GENERIC, которое должно поддерживать широкий спектр аппаратного обеспечения, собственное ядро содержит поддержку аппаратного обеспечения только *вашего* компьютера. Это может давать следующие плюсы:

- Меньшее время загрузки. Поскольку ядро будет пытаться определить только то аппаратное обеспечение, которое установлено в вашем компьютере, время, которое потребуется системе для загрузки, может значительно уменьшиться.
- Уменьшение использования памяти. Собственное ядро часто использует меньше памяти, чем ядро GENERIC, что очень важно, поскольку ядро всегда находится в физической памяти. Именно по этой причине собственное ядро особенно полезно при использовании систем с малым объемом оперативной памяти.
- Поддержка дополнительного аппаратного обеспечения. Собственное ядро позволяет вам добавить поддержку таких устройств, поддержка которых отсутствует в ядре GENERIC.

### 8.3. Сборка и установка собственного ядра

Во-первых, давайте сделаем краткий обзор каталога, в котором будет происходить сборка ядра. Все каталоги, которые будут упоминаться, будут относительными по отношению к основному каталогу `/usr/src/sys`, который также доступен как каталог `/sys`. Этот каталог содержит множество подкаталогов, представляющих собой различные части ядра, но наиболее важным для нас будет каталог `arch/conf`, в котором вы будете редактировать конфигурационный файл ядра и в котором находится каталог `compile`, где будет собираться ваше ядро. `arch` может быть `i386`, `alpha`, `amd64`, `ia64`, `powerpc`, `sparc64` или `pc98` (альтернативная ветвь аппаратного обеспечения, популярная в Японии). Все, что находится внутри каталога определенной архитектуры, относится только к этой архитектуре; остальной код является машинно независимым и общим для всех платформ, на которые FreeBSD может быть потенциально портирована. Обратите внимание на логическую структуру каталогов, в которой каждое поддерживаемое устройство, каждая файловая система и каждая опция размещается в своём собственном каталоге. Версии FreeBSD до 5.X поддерживали только архитектуры `i386`, `alpha` и `pc98`.

В примерах этой главы подразумевается, что вы используете архитектуру `i386`. Если это не так, измените каталоги, указанные в примерах, в соответствии с архитектурой вашей системы.

**Замечание:** Если в вашей системе *отсутствует* каталог `/usr/src/sys`, это означает, что исходные тексты ядра не были установлены. Наиболее простой способ установить их - запустить `/stand/sysinstall` как `root`, выбрать `Configure`, потом `Distributions`, потом `src`, потом `sys`. Если вы испытываете отвращение к `sysinstall` и у вас есть доступ к "официальному" FreeBSD CDRом, вы также можете установить исходные тексты при помощи командной строки:

```
# mount /cdrom
# mkdir -p /usr/src/sys
# ln -s /usr/src/sys /sys
# cat /cdrom/src/ssys.[a-d]* | tar -xzvf -
```

Затем, перейдите в каталог `arch/conf` и скопируйте файл конфигурации GENERIC в файл с выбранным вами именем. Например:

```
# cd /usr/src/sys/i386/conf
# cp GENERIC MYKERNEL
```

По традиции имя состоит из букв в верхнем регистре, и если вы поддерживаете несколько компьютеров FreeBSD на различном оборудовании, хорошая идея добавлять это имя к имени хоста. Мы назвали ядро MYKERNEL в этом примере.

**Подсказка:** Помещение файла конфигурации ядра в `/usr/src` может быть плохой идеей. Если вы испытываете проблемы, их можно решить удалив `/usr/src` и начав все с начала. После этого обычно требуется несколько секунд, чтобы понять, что вы удалили собственный файл настройки ядра. Не редактируйте непосредственно `GENERIC`, он может быть также перезаписан и при следующем обновлении дерева исходных текстов, и изменения ядра будут потеряны.

Вы можете сохранить файл конфигурации ядра в другом месте, а затем создать символическую ссылку на этот файл в каталоге `i386`.

Например:

```
# cd /usr/src/sys/i386/conf
# mkdir /root/kernels
# cp GENERIC /root/kernels/MYKERNEL
# ln -s /root/kernels/MYKERNEL
```

Теперь отредактируйте файл MYKERNEL в своём любимом текстовом редакторе. Если вы только начинаете, единственным доступным редактором скорее всего будет **vi**, который слишком сложен для того, чтобы объяснить, как им пользоваться, здесь, но в библиографии перечислено множество книг, в которых его использование хорошо освещено. Однако FreeBSD предоставляет более простой редактор **ee**, который, если вы - новичок, подойдёт вам больше всего. Не стесняйтесь изменять строки комментариев в начале файла, с тем, чтобы отобразить вашу конфигурацию или изменения, которые вы сделали по сравнению с `GENERIC`.

Если вам приходилось собирать ядро для SunOS или какой-либо другой операционной системы типа BSD, многое из того, что содержится в этом файле будет очень знакомо вам. Если же вы, напротив, использовали другую операционную систему, такую как DOS, файл конфигурации `GENERIC` может показаться вам крайне сложным, поэтому следуйте инструкциям в разделе Конфигурационный файл медленно и внимательно.

**Замечание:** Если вы синхронизируете дерево исходных текстов с деревом проекта FreeBSD, не забудьте свериться с файлом `/usr/src/UPDATING` перед обновлением. В этом файле описаны все важные вопросы и области исходного кода, требующие особого внимания. `/usr/src/UPDATING` всегда соответствует версии ваших исходных текстов FreeBSD, поэтому является более актуальным источником информации, чем это руководство.

Теперь вы должны скомпилировать ядро. Существует два способа, которые позволяют это сделать. Какой из них выбрать, зависит от того, почему вам понадобилось пересобрать ядро и версии FreeBSD, которую вы используете.

- Если вы установили *только* исходные тексты ядра, используйте способ 1.

- Если вы используете FreeBSD версии более ранней, чем 4.0 и вы *не* обновляете систему до FreeBSD 4.0 или выше, используя `make buildworld`, используйте способ 1.
- Если вы собираете новое ядро, не обновляя исходные тексты (возможно, для того, чтобы добавить новую опцию, например `IPFIREWALL`), вы можете использовать любой способ.
- Если вы пересобираете ядро как часть процесса `make buildworld`, используйте способ 2.

Если вы *не* обновляли дерево исходных текстов с момента последнего успешного завершения цикла `buildworld-installworld`, (то есть не запускали **CVSup**, **CTM**, или не использовали **anoncvs**), возможно использование последовательности `config, make depend, make, make install`.

### Способ 1. Сборка ядра "традиционным" способом

1. Запустите `confi g(8)` для того, чтобы сгенерировать исходные тексты ядра.  

```
# /usr/sbin/config MYKERNEL
```
2. Перейдите в каталог, в котором будет собираться ядро. Запущенный как показано выше `confi g(8)` выведет имя этого каталога.  

```
# cd ../compile/MYKERNEL
```

Для FreeBSD версий более ранних, чем 5.0, вместо этого используйте:

```
# cd ../../compile/MYKERNEL
```
3. Соберите ядро.  

```
# make depend
# make
```
4. Установите новое ядро.  

```
# make install
```

### Способ 2. Сборка ядра "новым" способом.

1. Перейдите в каталог `/usr/src`.  

```
# cd /usr/src
```
2. Соберите ядро.  

```
# make buildkernel KERNCONF=MYKERNEL
```
3. Установите новое ядро.  

```
# make installkernel KERNCONF=MYKERNEL
```

**Замечание:** В FreeBSD 4.2 и в более ранних, вам будет необходимо заменить `KERNCONF=` на `KERNEL=`. 4.2-STABLE, исходные тексты которой были обновлены до 2 февраля 2001 года, не понимает параметра `KERNCONF=`.

Новое ядро будет скопировано в каталог `/boot/kernel` как `/boot/kernel/kernel`, а старое ядро будет перемещено в `/boot/kernel.old/kernel`. Теперь перезагрузите систему для того, чтобы использовать новое ядро. Если что-то пойдёт не так, вы можете обратиться к разделу Если что-то пошло не так в



конец этой главы, который может оказаться полезен. Не забудьте прочитать раздел, который объясняет как исправить то, что ядро не загружается.

**Замечание:** В FreeBSD 4.X и более ранних, ядро устанавливается в `/kernel`, модули в `/modules`, а старое ядро сохраняется в `/kernel.old`. Другие файлы, относящиеся к процессу загрузки, такие как загрузчик (`loader(8)`) и его конфигурационные файлы, размещаются в `/boot`. Модули сторонних производителей могут быть помещены в `/modules`, хотя пользователи должны знать, что очень важно, чтобы модули были синхронизированы с собранным ядром. Модули, не рассчитанные на работу с собранным ядром, могут вызвать нестабильность и некорректность работы.

**Замечание:** Если вы добавили новые устройства (такие, как звуковая карта) и используете FreeBSD 4.X или более ранние версии, то, возможно, вам понадобится создать некоторые файлы устройств в каталоге `/dev` перед тем, как вы сможете их использовать. За дополнительной информацией вы можете обратиться к разделу Создание файлов устройств далее в этой главе.

## 8.4. Конфигурационный файл

*Обновил для FreeBSD 5.X Joel Dahl.*

Формат конфигурационного файла достаточно прост. Каждая строка представляет собой ключевое слово и один или более аргументов. Для простоты большинство строк содержат только один аргумент. Всё, что следует за символом `#` является комментарием и игнорируется. Следующие разделы описывают каждый параметр, в основном в порядке, в котором они появляются в `GENERIC`, хотя некоторые родственные параметры сгруппированы вместе в одном разделе (например Работа с сетью), даже если на самом деле они разбросаны по всему файлу `GENERIC`. За полным списком архитектурно-зависимых параметров и устройств обратитесь к файлу `NOTES` в том же каталоге, что и `GENERIC`. Архитектурно независимые параметры находятся в `/usr/src/sys/conf/NOTES`.

**Замечание:** `NOTES` не существует в FreeBSD 4.X. Детальное описание параметров и устройств из `GENERIC` находится в файле `LINT`. В 4.X `LINT` служит двум целям: предоставить справочное руководство для выбора параметров ядра при сборке ядра по умолчанию и предоставить конфигурацию ядра с параметрами, установленными в значения, отличные от значений по умолчанию (настолько много параметров, насколько это возможно). Это было сделано по той причине, что такая конфигурация хорошо помогала (и все еще помогает) при тестировании нового кода и изменений в существующем коде, поскольку этот код и изменения могут конфликтовать с другими частями ядра. Однако, инфраструктура конфигурации ядра в 5.X прошла через множество серьезных изменений; Одно из таких изменений — перемещение параметров конфигурации драйверов в `hints`, и теперь они могут быть изменены и загружены динамически во время загрузки системы, а в `LINT` их больше нет. По этой и другим причинам `LINT` был переименован в `NOTES` и в основном по-прежнему соответствует своему назначению: информирование пользователей о доступных параметрах.

В FreeBSD 5.X и более поздних версиях вы все еще можете создать собираемый `LINT`, выполнив:

```
# cd /usr/src/sys/i386/conf && make LINT
```

Это пример конфигурационного файла ядра GENERIC с различными дополнительными комментариями, которые могут понадобиться для ясности. Этот пример должен совпадать с вашей копией в `/usr/src/sys/i386/conf/GENERIC` практически полностью.

```
#
# GENERIC -- Generic kernel configuration file for FreeBSD/i386
#
# For more information on this file, please read the handbook section on
# Kernel Configuration Files:
#
#   http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/handbook/kernelconfig-config.html
#
# The handbook is also available locally in /usr/share/doc/handbook
# if you've installed the doc distribution, otherwise always see the
# FreeBSD World Wide Web server (http://www.FreeBSD.org/) for the
# latest information.
#
# An exhaustive list of options and more detailed explanations of the
# device lines is also present in the ../../conf/NOTES and NOTES files.
# If you are in doubt as to the purpose or necessity of a line, check first
# in NOTES.
#
# $FreeBSD: /repoman/r/ncvs/src/sys/i386/conf/GENERIC,v 1.413 2004/08/11 01:34:18 rwatson Exp $
```

Следующие параметры обязательны для *любого* ядра, которое вы будете собирать:

```
machine      i386
```

Это архитектура машины. Она должна быть одной из следующих: `alpha`, `amd64`, `i386`, `ia64`, `pc98`, `powerpc`, или `sparc64`.

```
cpu          I486_CPU
cpu          I586_CPU
cpu          I686_CPU
```

Эта опция указывает тип процессора, который используется в вашей системе. В конфигурационном файле может быть несколько вхождений этой опции (например, если вы не уверены, какой из типов процессора необходимо использовать — `I586_CPU` или `I686_CPU`), но для собственного ядра лучше указывать только тот тип процессора, который установлен в вашей системе. Если вы не уверены, какой тип необходимо использовать вам, вы можете воспользоваться файлом `/var/run/dmesg.boot`, чтобы увидеть протокол загрузки системы.

Поддержка типа `I386_CPU` все ещё существует в исходных текстах FreeBSD, но по умолчанию отключена в ветках `-STABLE` и `-CURRENT`. Это означает, что для того, чтобы установить FreeBSD на систему с процессором `i386`, вы можете использовать следующие способы:

- Установить более раннюю версию FreeBSD и пересобрать систему из исходных текстов в соответствии с Разд. 8.3.
- Собрать мир и ядро на более новой машине и установить на 386 используя уже собранные файлы в `/usr/obj` (за дополнительной информацией можете обратиться к Разд. 19.5).

- Сделать собственный релиз FreeBSD, который включает поддержку I386\_CPU в ядре установочного CD-ROM.

Первый из этих способов, возможно, наиболее простой из всех предложенных, но вам понадобится много дискового пространства, что может быть затруднительно для систем класса 386.

```
ident                GENERIC
```

Этот параметр определяет "метку" ядра. Необходимо, чтобы она соответствовала названию файла конфигурации ядра, например MYKERNEL, если вы следовали инструкциям в предыдущих примерах. Значение, которое вы присвоите параметру `ident` будет выводиться в процессе загрузки, поэтому полезно давать новым ядрам другие имена для того, чтобы отличать их от обычного ядра (например, если вы хотите собрать экспериментальное ядро).

```
maxusers             n
```

Параметр `maxusers` определяет размеры многих важных системных таблиц. Это число должно быть приблизительно равно количеству пользователей, одновременно использующих вашу систему.

Начиная с FreeBSD 4.5, система автоматически настроит этот параметр если в конфигурационном файле вы установили его в значение 0<sup>1</sup>. В FreeBSD 5.X и выше, `maxusers` по умолчанию равен 0, если не установлено другое значение. Если вы используете FreeBSD версии более ранней, чем 4.5 или хотите самостоятельно управлять этим значением, рекомендуется устанавливать значение `maxusers` равным минимум 4, особенно если вы используете X Window System или компиляторы. Причина этого кроется в том, что одной из основных таблиц, размер которой определяет `maxusers`, является максимальное количество процессов, которое устанавливается равным  $20 + 16 * \text{maxusers}$ , таким образом, если вы установите `maxusers` равным 1, то получите возможность выполнять максимум 36 одновременных процессов, включая 18 или около того процессов, которые система запускает при загрузке, и 15 или около того вы создадите, когда запустите X Window System. Даже такая простая задача, как чтение страницы справочной системы (`man(1)`), потребует запуска девяти процессов, для того, чтобы использовать фильтры, распаковать и просмотреть её. Установка `maxusers` в значение 64 позволит запускать до 1044 одновременных процессов, что должно быть достаточно для большинства применений. Однако, если вы видите ошибку `proc table full`, когда пытаетесь запустить программу или поддерживаете сервер с большим количеством пользователей (таким как `ftp.FreeBSD.org`), вы всегда можете увеличить это число и пересобрать ядро.

**Замечание:** `maxusers` не ограничивает количество пользователей, которые могут зайти на вашу машину. Он просто устанавливает размеры различных таблиц в разумные значения, учитывая максимальное количество пользователей, которое будет у вас в системе и как много процессов будет запускать каждый из них. Параметр ядра, который действительно ограничивает количество одновременных удалённых сессий и окон X терминала, это `pseudo-device` `pty` 16. С FreeBSD 5.X, вам не нужно беспокоиться об этом номере, поскольку драйвер `pty(4)` "автоклонируемый"; используйте в файле конфигурации строку `device` `pty`.

```
# Floating point support - do not disable.
device                npx
```

`npx` - это интерфейс к модулю операций с плавающей точкой в FreeBSD, который может использовать как аппаратный сопроцессор, так и программную эмуляцию. Этот параметр является обязательным.

```
# Pseudo devices
```

```
device    loop                # Network loopback
```

Стандартное устройство обратной связи для TCP/IP. Если вы запускаете telnet или FTP на localhost (он же 127.0.0.1), то соединение пройдет через это устройство. Этот параметр *обязателен*. В FreeBSD 4.X вам потребуется использовать pseudo-device loop.

Все остальные параметры более или менее опциональны. Обращайте внимание на информацию о каждой опции, которая размещается рядом с ней.

```
#To statically compile in device wiring instead of /boot/device.hints
#hints                "GENERIC.hints"          # Default places to look for devices.
```

В FreeBSD 5.X и более поздних device.hints(5) используются для настройки параметров драйверов устройств. Путь по умолчанию, который loader(8) будет проверять при загрузке - /boot/device.hints. Используя опцию hints вы можете вкомпилировать эти параметры статически в ваше ядро. В этом случае не требуется создавать файл device.hints в каталоге /boot.

```
#makeoptions          DEBUG=-g                # Build kernel with gdb(1) debug symbols
```

Обычный процесс сборки FreeBSD не включает генерацию отладочной информации при сборке ядра и удаляет большую часть вспомогательной информации (symbols) после того как ядро слинковано для того, чтобы сэкономить немного места. Если вы собираетесь проводить тестирование ядер в ветви -CURRENT или разрабатывать собственные правки для ядра FreeBSD, возможно вам будет нужно раскомментировать эту строку. Тем самым вы включите использования опции -g, которая включает генерацию отладочной информации gcc(1). Того же самого эффекта можно добиться используя опцию conf(8)-g, в случае, если вы используете "традиционный" способ сборки ядра (обратитесь к Разд. 8.3 за дополнительной информацией).

```
options              SCHED_4BSD              # 4BSD scheduler
```

Традиционный планировщик FreeBSD. В зависимости от загруженности системы, вы можете повысить производительность системы, используя новый планировщик FreeBSD ULE, который был специально разработан для SMP, но отлично работает также и на однопроцессорных системах. Если вы хотите попробовать этот планировщик, замените SCHED\_4BSD на SCHED\_ULE в файле конфигурации ядра.

```
options              INET                    # InterNETworking
```

Поддержка сетевых возможностей. Оставьте эту опцию включенной, даже если вы не планируете подключаться к сети. Большинство программ требуют, чтобы работал хотя бы интерфейс обратной связи (loopback) (т.е. создание сетевых соединений внутри вашего ПК), так что эта опция в принципе является обязательной.

```
options              INET6                  # IPv6 communications protocols
```

Включает поддержку коммуникационных протоколов IPv6.

```
options              FFS                    # Berkeley Fast Filesystem
```

Включает поддержку основной файловой системы. Не удаляйте эту опцию, если вы планируете загружаться с жесткого диска.

```
options              UFS_ACL                # Support for access control lists
```

Этот параметр, присутствующий только в FreeBSD 5.X, включает в ядре поддержку списков управления доступом (ACL). Основывается на использовании расширенных атрибутов и UFS2, детальное описание вы сможете найти в Разд. 14.13. ACL включены по умолчанию и не должны выключаться в случае если они ранее использовались на файловой системе, так как это удалит списки управления доступом и изменит то, как защищены файлы, непредсказуемым образом.

```
options          UFS_DIRHASH          # Improve performance on big directories
```

Эта опция включает функциональность, которая повышает скорость дисковых операций на больших каталогах в обмен на использование дополнительной памяти. Для большого сервера или рабочей станции рекомендуется оставить ее включенной, и выключить для системы, для которой более приоритетна память, чем скорость доступа к дискам, например сетевой экран (firewall).

```
options          SOFTUPDATES          # Enable FFS Soft Updates support
```

Этот параметр включает в ядре технологию Soft Updates, которая повышает скорость записи на диски. Несмотря на то, что эта технология включена в ядре, она должна быть включена для отдельных дисков. Просмотрите вывод команды mount(8) чтобы определить, включены ли Soft Updates для дисков вашей системы. Если вы не увидите параметр soft-updates, вам будет необходимо активировать его при помощи команды tuneefs(8) (для существующих файловых систем) или команды newfs(8) (для новых файловых систем).

```
options          MD_ROOT              # MD is a potential root device
```

Этот параметр включает поддержку использования дисков в памяти для корневой файловой системы.

```
options          NFSCLIENT           # Network Filesystem Client
options          NFSSERVER            # Network Filesystem Server
options          NFS_ROOT             # NFS usable as /, requires NFSCLIENT
```

Сетевая файловая система. Если вы не планируете монтировать разделы с файлового сервера UNIX через TCP/IP, вы можете исключить этот параметр из конфигурационного файла ядра.

```
options          MSDOSFS              # MSDOS Filesystem
```

Файловая система MS-DOS. Если вы не собираетесь монтировать форматированный в DOS раздел жесткого диска в момент загрузки, вы можете безопасно закомментировать этот параметр. Необходимый модуль будет автоматически загружен когда вы в первый раз смонтируете раздел DOS, так, как это описано ниже. Также, замечательный пакет emulators/mtools позволяет получить доступ к DOS дискетам без необходимости монтировать и размонтировать их (и не требует наличия MSDOSFS).

```
options          CD9660               # ISO 9660 Filesystem
```

Файловая система ISO 9660 для компакт-дисков. Если у вас нет привода CDRW или вы будете лишь изредка монтировать компакт-диски с данными, закомментируйте эту строку, так как необходимый модуль будет загружен автоматически при первом монтировании компакт-диска с данными. Для использования звуковых компакт-дисков эта файловая система не потребуется.

```
options          PROCFS               # Process filesystem
```

Файловая система процессов. Это "виртуальная" файловая система монтируемая в `/proc`, которая позволяет таким приложениям, как `ps(1)` выдавать вам больше информации о запущенных процессах. В FreeBSD 5.X и выше, использование PROCFS не требуется, так как большинство мониторинговых и отладочных инструментов было адаптировано для работы без PROCFS: в отличие от FreeBSD 4.X, новые системы FreeBSD 5.X по умолчанию не монтируют файловую систему процессов. Более того, ядра 6.X-CURRENT, которые используют PROCFS, должны также включать поддержку PSEUDofs:

```
options          PSEUDofs          # Pseudo-filesystem framework
```

PSEUDofs недоступна в FreeBSD 4.X.

```
options          GEOM_GPT          # GUID Partition Tables.
```

Этот параметр делает возможным наличие большого количества разделов на одном диске.

```
options          COMPAT_43         # Compatible with BSD 4.3 [KEEP THIS!]
```

Совместимость с 4.3BSD. Не выключайте эту опцию; некоторые приложения будут вести себя странно, если этой опции не будет в ядре.

```
options          COMPAT_FREEBSD4   # Compatible with FreeBSD4
```

Эта опция требуется в FreeBSD 5.X для платформ i386 и Alpha для поддержки приложений, собранных на более старых версиях FreeBSD, которые используют старые интерфейсы вызовов. Рекомендуется использовать данную опцию на всех системах на платформах i386 и Alpha, на которых могут запускаться старые приложения; платформы, поддержка которых появилась только в FreeBSD 5.X, например ia64 и Sparc64®, не требуют этой опции.

```
options          SCSI_DELAY=15000  # Delay (in ms) before probing SCSI
```

Этот параметр заставляет ядро приостановиться на 15 секунд перед тем, как идентифицировать каждое устройство SCSI в вашей системе. Если у вас установлены только жесткие диски IDE, вы можете игнорировать эту опцию, в противном случае, возможно, вы захотите уменьшить это число, например до 5 секунд, для того, чтобы ускорить загрузку. Естественно, что если вы сделаете это, а у FreeBSD появятся проблемы с распознаванием ваших устройств SCSI, необходимо будет увеличить этот параметр.

```
options          KTRACE            # ktrace(1) support
```

Включает поддержку трассировки процессов, что удобно при отладке.

```
options          SYSVSHM           # SYSV-style shared memory
```

Этот параметр предоставляет поддержку разделяемой памяти System V. Наиболее распространенное применение этого - расширение XSHM в X, которое многие приложения, интенсивно работающие с графикой, будут автоматически использовать для повышения скорости работы. Если вы используете X, эта опция будет необходима.

```
options          SYSVSEM           # SYSV-style semaphores
```

Поддержка семафоров System V. Не настолько часто используемая возможность, но в ядро добавляет всего несколько сотен байт.

```
options          SYSVMSG          # SYSV-style message queues
```

Поддержка сообщений System V. Опять-таки, этот параметр добавляет в ядро всего лишь несколько сотен байт.

**Замечание:** Команда `ipcs(1)` с параметром `-p` покажет все процессы, которые используют любую из этих возможностей System V.

```
options          _KPOSIX_PRIORITY_SCHEDULING # POSIX P1003.1B real-time extensions
```

Расширения реального времени, добавленные 1993 POSIX®. Определенные приложения из коллекции используют их, например **StarOffice**.

```
options          KBD_INSTALL_CDEV  # install a CDEV entry in /dev
```

Этот параметр относится к клавиатуре. Он добавляет CDEV в `/dev`.

```
options          AHC_REG_PRETTY_PRINT      # Print register bitfields in debug
                                                    # output. Adds ~128k to driver.
options          AHD_REG_PRETTY_PRINT      # Print register bitfields in debug
                                                    # output. Adds ~215k to driver.
```

Распечатка регистров для облегчения отладки.

```
options          ADAPTIVE_GIANT          # Giant mutex is adaptive.
```

**Giant** — имя механизма защиты (“спящего” мьютекса) для крупных наборов ресурсов ядра. На нынешний момент **Giant** представляется фактически непригодным для использования в связи с серьезными потерями в производительности, и активно заменяется на механизмы, защищающие отдельные ресурсы ядра. Параметр `ADAPTIVE_GIANT` включает **Giant** в число адаптивных мьютексов: в случае, когда нить ядра нуждается в **Giant**, а он уже захвачен нитью, выполняющейся на другом процессоре, первая нить будет продолжать выполнение и ждать освобождения **Giant**. В норме нить должна была бы уснуть, пока не настанет очередной момент ее выполнения. Если вы не уверены, оставьте этот параметр в покое.

```
device           apic                # I/O APIC
```

Устройство `apic` разрешает использование набора I/O APIC для распределения прерываний. Оно может быть использовано как с однопроцессорными, так и с многопроцессорными ядрами (для последних наличие `apic` является обязательным). Для поддержки многопроцессорности добавьте строку `options SMP`.

```
device           isa
```

Все компьютеры, поддерживаемые FreeBSD, имеют хотя бы одно устройство ISA. Не удаляйте эту строку, даже если в вашем компьютере нет слотов ISA. Для архитектуры IBM PS/2 (Микроканальная Архитектура (MCA)), FreeBSD предоставляет ограниченную поддержку данной шины. За дальнейшей информацией о поддержке MCA обратитесь к файлу `/usr/src/sys/i386/conf/NOTES`.

```
device           eisa
```

Включите эту опцию если у вас материнская плата EISA. Это включает автоопределение и конфигурирование поддержки всех устройств на шине EISA.

```
device          pci
```

Включите этот параметр, если у вас материнская плата с поддержкой PCI. Это включит автоопределение карт PCI и проксирование из шины PCI в шину ISA.

```
device          agp
```

Включите эту опцию, если у вас есть AGP карта в системе. Это включит поддержку AGP и AGP GART для тех карт, которые поддерживают эту возможность.

```
# Floppy drives
device          fd
```

Контроллер флоппи-диска.

```
# ATA and ATAPI devices
device          ata
```

Этот драйвер поддерживает все устройства ATA и ATAPI. Вам необходима только одна строка `device ata` в ядре для того, чтобы обнаружить все PCI устройства ATA/ATAPI в современных машинах.

```
device          atadisk          # ATA disk drives
```

Эта строка необходима вместе с `device ata` для поддержки дисков ATA.

```
device          ataraid          # ATA RAID drives
```

Эта строка необходима вместе с `device ata` для поддержки дисков ATA RAID.

```
device          atapicd          # ATAPI CDROM drives
```

Поддержка приводов ATAPI CDROM. Используется вместе с `device ata`.

```
device          atapifd          # ATAPI floppy drives
```

Поддержка флоппи-приводов ATAPI. Используется вместе с `device ata`.

```
device          atapist          # ATAPI tape drives
```

Поддержка ленточных приводов ATAPI (стримеров). Используется вместе с `device ata`.

```
options          ATA_STATIC_ID          # Static device numbering
```

Заставляет драйвер нумеровать устройства статически; в противном случае происходит динамическая нумерация.

```
# SCSI Controllers
device          ahb          # EISA AHA1742 family
device          ahc          # AHA2940 and onboard AIC7xxx devices
device          ahd          # AHA39320/29320 and onboard AIC79xx devices
device          amd          # AMD 53C974 (Teckram DC-390(T))
```



```

device      isp      # Qlogic family
device      mpt      # LSI-Logic MPT-Fusion
#device     ncr      # NCR/Symbios Logic
device      sym      # NCR/Symbios Logic (newer chipsets)
device      trm      # Tekram DC395U/UW/F DC315U adapters

device      adv      # Advansys SCSI adapters
device      adw      # Advansys wide SCSI adapters
device      aha      # Adaptec 154x SCSI adapters
device      aic      # Adaptec 15[012]x SCSI adapters, AIC-6[23]60.
device      bt       # Buslogic/Mylex MultiMaster SCSI adapters

device      ncv      # NCR 53C500
device      nsp      # Workbit Ninja SCSI-3
device      stg      # TMC 18C30/18C50

```

Контроллеры SCSI. Закомментируйте те, которых у вас в системе нет. Если у вас в системе исключительно IDE устройства, вы можете удалить все эти строки.

```

# SCSI peripherals
device      scbus    # SCSI bus (required for SCSI)
device      ch       # SCSI media changers
device      da       # Direct Access (disks)
device      sa       # Sequential Access (tape etc)
device      cd       # CD
device      pass     # Passthrough device (direct SCSI access)
device      ses      # SCSI Environmental Services (and SAF-TE)

```

Периферийные устройства SCSI. Опять-таки, закомментируйте те, которых у вас в системе нет или, если у вас в наличии исключительно IDE, можете удалить все.

**Замечание:** USB umass(4) драйвер (и некоторые другие драйверы) используют подсистему SCSI, хотя и не являются настоящими SCSI устройствами. Следовательно, вам необходимо сохранить поддержку SCSI, если какой-либо из этих драйверов включен в конфигурацию ядра.

```

# RAID controllers interfaced to the SCSI subsystem
device      amr      # AMI MegaRAID
device      asr      # DPT SmartRAID V, VI and Adaptec SCSI RAID
device      ciss     # Compaq Smart RAID 5*
device      dpt      # DPT Smartcache III, IV - See NOTES for options
device      hptmv    # Highpoint RocketRAID 182x
device      iir      # Intel Integrated RAID
device      ips      # IBM (Adaptec) ServerRAID
device      mly      # Mylex AcceleRAID/eXtremeRAID
device      twa      # 3ware 9000 series PATA/SATA RAID

# RAID controllers
device      aac      # Adaptec FSA RAID
device      aacp     # SCSI passthrough for aac (requires CAM)
device      ida      # Compaq Smart RAID
device      mlx      # Mylex DAC960 family

```

```
device      pst      # Promise Supertrak SX6000
device      twe      # 3ware ATA RAID
```

Поддерживаемые RAID-контроллеры. Если у вас нет таковых, можете их закомментировать или удалить эти строки.

```
# atkbdc0 controls both the keyboard and the PS/2 mouse
device      atkbdc    # AT keyboard controller
```

Контроллер клавиатуры (atkbdc) предоставляет средства ввода/вывода для клавиатуры AT и PS/2 устройств. Этот контроллер необходим драйверу клавиатуры (atkbd) и PS/2 устройств (psm).

```
device      atkbd     # AT keyboard
```

Драйвер atkbd вместе с контроллером atkbdc предоставляет доступ к клавиатуре AT 84 или улучшенной клавиатуре AT, которая подключена к контроллеру AT клавиатуры.

```
device      psm       # PS/2 mouse
```

Используйте это устройство, если ваша мышь включается в порт PS/2.

```
device      vga       # VGA video card driver
```

Драйвер видеокарты.

```
# splash screen/screen saver
device      splash     # Splash screen and screen saver support
```

Заставка при загрузке. Хранители экрана также требуют этого устройства. В FreeBSD 4.X используйте строку pseudo-device splash.

```
# syscons is the default console driver, resembling an SCO console
device      sc
```

sc - это драйвер консоли по умолчанию, который имитирует консоль SCO. Так как большая часть консольных полноэкранных приложений обращаются к консоли через терминальную библиотеку termcap, вас не должно волновать, будете ли вы использовать этот драйвер, либо драйвер vt, который является VT220-совместимым драйвером консоли. Если у вас возникнут какие-либо проблемы с приложениями, работающими с этим драйвером консоли, установите переменную окружения TERM в значение scoansi.

```
# Enable this for the pcvt (VT220 compatible) console driver
#device      vt
#options     XSERVER          # support for X server on a vt console
#options     FAT_CURSOR       # start with block cursor
```

VT220-совместимый драйвер консоли, обратно совместимый с VT100/102. Он работает лучше на некоторых ноутбуках, у которых возникают проблемы несовместимости с sc. Также, установите переменную окружения TERM в значение vt100 или vt220. Этот драйвер также может быть полезен в случаях подключения к большому количеству различных машин через сеть, на которых параметры для устройства sc для termcap или terminfo могут отсутствовать — vt100 присутствует практически на любой платформе.

```
# Power management support (see NOTES for more options)
#device          apm
```

Поддержка Advanced Power Management. Чаще всего используется в ноутбуках, хотя в ядре GENERIC FreeBSD 5.X и выше отключена по умолчанию.

```
# Add suspend/resume support for the i8254.
device          pmtimer
```

Устройство таймера для управления энергопотреблением, APM и ACPI.

```
# PCCARD (PCMCIA) support
# PCMCIA and cardbus bridge support
device          cbb          # cardbus (yenta) bridge
device          pccard       # PC Card (16-bit) bus
device          cardbus      # CardBus (32-bit) bus
```

Поддержка PCMCIA. Включите ее, если вы используете ноутбук.

```
# Serial (COM) ports
device          sio          # 8250, 16[45]50 based serial ports
```

Четыре последовательных порта, которые известны как COM порты в мире MS-DOS/Windows

**Замечание:** Если у вас есть внутренний модем на COM4 и последовательный порт COM2, вам понадобится поменять IRQ модема на 2 (по непонятным техническим причинам  $IRQ2 = IRQ9$ ) для того, чтобы получить к нему доступ из FreeBSD. Если у вас есть многопортовая карта с последовательными портами, ознакомьтесь с `sio(4)` чтобы узнать корректные значения для добавления в `/boot/device.hints`. Некоторые видеокарты (в частности те, что используют чипы S3) используют адреса ввода/вывода в форме `0x*2e8` и, так как многие дешевые последовательные карты не полностью раскодируют шестнадцатитбитное пространство адресов ввода/вывода, они конфликтуют с этими картами, в итоге COM4 оказывается практически недоступным.

Каждый последовательный порт требует уникального IRQ (кроме тех случаев, когда вы используете мультипортовую карту, которая поддерживает совместное использование прерываний), поэтому значения IRQ по умолчанию для COM3 и COM4 не могут быть использованы.

```
# Parallel port
device          ppc
```

Интерфейс параллельного порта на шине ISA.

```
device          ppbus        # Parallel port bus (required)
```

Поддержка шины параллельного порта.

```
device          lpt          # Printer
```

Поддержка принтеров на параллельном порту.

**Замечание:** Все три последних устройства необходимы для поддержки принтеров на параллельном порту.

```
device      plip      # TCP/IP over parallel
```

Драйвер TCP/IP через параллельный порт.

```
device      ppi      # Parallel port interface device
```

Поддержка ввода/вывода общего назначения ("geek port") + IEEE1284 ввода/вывода.

```
#device     vpo      # Requires scbus and da
```

Драйвер привода Iomega Zip. Требуется наличия scbus и da. Наилучшая производительность достигается с портами в режиме EPP 1.9.

```
#device     puc
```

Раскомментируйте это устройство, если у вас есть "простая" последовательная или параллельная PCI карта, поддерживаемая драйвером puc(4).

```
# PCI Ethernet NICs.
```

```
device      de      # DEC/Intel DC21x4x ("Tulip")
device      em      # Intel PRO/1000 adapter Gigabit Ethernet Card
device      ixgb    # Intel PRO/10GbE Ethernet Card
device      txp     # 3Com 3cR990 ("Typhoon")
device      vx      # 3Com 3c590, 3c595 ("Vortex")
```

Драйвера сетевых карт PCI. Закомментируйте или удалите драйвера тех карт, которые отсутствуют в вашей системе.

```
# PCI Ethernet NICs that use the common MII bus controller code.
# NOTE: Be sure to keep the 'device miibus' line in order to use these NICs!
device      miibus  # MII bus support
```

Поддержка шины MII требуется для некоторых PCI 10/100 Ethernet карт, которые используют MII-совместимые передатчики или реализовывают интерфейс управления передатчиком, который имитирует MII. Добавление device miibus в конфигурационный файл ядра включает поддержку стандартный API miibus и все драйверы РНУ, включая стандартный для тех РНУ, которые не обрабатываются специфическим образом конкретным драйвером.

```
device      dc      # DEC/Intel 21143 and various workalikes
device      miibus  # MII bus support
device      bfe     # Broadcom BCM440x 10/100 Ethernet
device      bge     # Broadcom BCM570xx Gigabit Ethernet
device      dc      # DEC/Intel 21143 and various workalikes
device      fxp     # Intel EtherExpress PRO/100B (82557, 82558)
device      pcn     # AMD Am79C97x PCI 10/100 (precedence over 'lnc')
device      re      # RealTek 8139C+/8169/8169S/8110S
device      rl      # RealTek 8129/8139
device      sf      # Adaptec AIC-6915 ("Starfire")
device      sis     # Silicon Integrated Systems SiS 900/SiS 7016
device      sk      # SysKonnect SK-984x & SK-982x gigabit Ethernet
device      ste     # Sundance ST201 (D-Link DFE-550TX)
device      ti      # Alteon Networks Tigon I/II gigabit Ethernet
device      tl      # Texas Instruments ThunderLAN
```

```

device      tx          # SMC EtherPower II (83c170 "EPIC")
device      vr          # VIA Rhine, Rhine II
device      wb          # Winbond W89C840F
device      xl          # 3Com 3c90x ("Boomerang", "Cyclone")

```

Драйвера, которые используют контроллер шины MII.

```

# ISA Ethernet NICs. pccard NICs included.
device      cs          # Crystal Semiconductor CS89x0 NIC
# 'device ed' requires 'device miibus'
device      ed          # NE[12]000, SMC Ultra, 3c503, DS8390 cards
device      ex          # Intel EtherExpress Pro/10 and Pro/10+
device      ep          # Etherlink III based cards
device      fe          # Fujitsu MB8696x based cards
device      ie          # EtherExpress 8/16, 3C507, StarLAN 10 etc.
device      lnc         # NE2100, NE32-VL Lance Ethernet cards
device      sn          # SMC's 9000 series of Ethernet chips
device      xe          # Xircom pccard Ethernet

```

```

# ISA devices that use the old ISA shims
#device      le

```

Драйвера сетевых карт ISA. Ознакомьтесь с файлом `/usr/src/sys/i386/conf/NOTES`, чтобы узнать, какие сетевые карты поддерживаются каким драйвером.

```

# Wireless NIC cards
device      wlan        # 802.11 support
device      an          # Aironet 4500/4800 802.11 wireless NICs.
device      awi         # BayStack 660 and others
device      wi          # WaveLAN/Intersil/Symbol 802.11 wireless NICs.
#device      wl         # Older non 802.11 Wavelan wireless NIC.

```

Поддержка различных беспроводных карт.

```

device      mem          # Memory and kernel memory devices

```

Устройства системной памяти.

```

device      io          # I/O device

```

Этот параметр позволяет процессу получение привилегий ввода/вывода. Он полезен для написания пользовательских программ, работающих с оборудованием непосредственно. Необходим для работы X Window system.

```

device      random      # Entropy device

```

Генератор случайных чисел для криптографической защиты.

```

device      ether        # Ethernet support

```

ether необходим лишь в случае, если у вас есть сетевая карта. Он включает поддержку стандартного кода протокола Ethernet. В FreeBSD 4.X используйте строку `pseudo-device ether`.

```

device      sl          # Kernel SLIP

```

sl - это поддержка SLIP. SLIP был практически вытеснен PPP, который легче настраивается, лучше подходит для соединений модем-модем и имеет больше возможностей. В FreeBSD 4.X используйте строку `pseudo-device sl`.

```
device    ppp                # Kernel PPP
```

Поддержка PPP в ядре для соединений dial-up. Также существует версия PPP, реализованного как приложение использующее tun и предлагает большую гибкость и большее количество возможностей, как, например, соединение при необходимости (наличии обращения к сети). В FreeBSD 4.X используйте строку `pseudo-device ppp`.

```
device    tun                # Packet tunnel.
```

Используется пользовательским программным обеспечением PPP. Ознакомьтесь с разделом PPP этой книги за дальнейшей информацией. В FreeBSD 4.X используйте строку `pseudo-device tun`.

```
device    pty                # Pseudo-ttys (telnet etc)
```

”псевдо-терминал” или имитированный порт для входа. Используется входящими telnet и rlogin-сессиями, приложением **xterm** и некоторыми другими приложениями, такими как **Emacs**.

**Замечание:** В FreeBSD 4.X используйте строку `pseudo-device pty number`. Число после pty определяет количество pty, которое необходимо создать. Если вам необходимо более, чем стандартное (16) количество окон **xterm** и/или удаленных входов, увеличьте это число в соответствии с вашими потребностями, но не более чем до 256.

```
device    md                 # Memory "disks"
```

Псевдоустройства дисков в памяти. В FreeBSD 4.X используйте строку `pseudo-device md`

```
device    gif                # IPv6 and IPv4 tunneling
```

Поддержка туннелирования IPv6 через IPv4, IPv4 через IPv6, IPv4 через IPv4 и IPv6 через IPv6. Начиная с FreeBSD 4.4 устройство gif является ”автоклонируемым” и вы должны использовать строку `pseudo-device gif`. Более ранним версиям FreeBSD 4.X требуется номер, например `pseudo-device gif 4`.

```
device    faith              # IPv6-to-IPv4 relaying (translation)
```

Это псевдоустройство захватывает пакеты, которые были посланы ему и перенаправляет их демону трансляции IPv4/IPv6. В FreeBSD 4.X используйте строку `pseudo-device faith 1`

```
# The 'bpf' device enables the Berkeley Packet Filter.
# Be aware of the administrative consequences of enabling this!
device    bpf                # Berkeley packet filter
```

Фильтр пакетов Berkeley. Это псевдоустройство позволяет переводить сетевые интерфейсы в ”неразборчивый” (promiscuous) режим, в котором перехватывается любой пакет в широковещательной сети (например ethernet). Эти пакеты могут быть сохранены на диск и/или исследованы при помощи tcpdump(1). В FreeBSD 4.X используйте строку `pseudo-device bpf`.

**Замечание:** Устройство bpf(4) также используется программой dhclient(8) для того, чтобы получить адрес шлюза по умолчанию и т.п. Если вы используете DHCP, не удаляйте эту опцию.

```
# USB support
device      uhci          # UHCI PCI->USB interface
device      ohci          # OHCI PCI->USB interface
device      usb           # USB Bus (required)
#device      udbp          # USB Double Bulk Pipe devices
device      ugen          # Generic
device      uhid          # "Human Interface Devices"
device      ukbd          # Keyboard
device      ulpt          # Printer
device      umass         # Disks/Mass storage - Requires scbus and da
device      ums           # Mouse
device      urio          # Diamond Rio 500 MP3 player
device      uscanner      # Scanners
# USB Ethernet, requires mii
device      aue           # ADMtek USB Ethernet
device      axe           # ASIX Electronics USB Ethernet
device      cue           # CATC USB Ethernet
device      kue           # Kawasaki LSI USB Ethernet
device      rue           # RealTek RTL8150 USB Ethernet
```

Поддержка различных USB устройств.

```
# FireWire support
device      firewire      # FireWire bus code
device      sbp           # SCSI over FireWire (Requires scbus and da)
device      fwe           # Ethernet over FireWire (non-standard!)
```

Поддержка различных устройств Firewire.

За дальнейшей информацией о дополнительных устройствах, поддерживаемых FreeBSD, обратитесь к файлу `/usr/src/sys/i386/conf/NOTES`.

### 8.4.1. Конфигурации с большим количеством оперативной памяти (PAE)

Машины с большим количеством оперативной памяти, в которых требуется более 4 гигабайт в пользовательском адресном пространстве и адресном пространстве ядра (User+Kernel Virtual Address, KVA) в обычном случае не смогут использовать более 4 гигабайт. Для решения этой проблемы Intel добавили поддержку 36-битной адресации в Pentium Pro и более поздних моделях процессоров.

Расширение физического адресного пространства (PAE) в процессорах Intel Pentium Pro и более поздних позволяет использовать до 64 гигабайт оперативной памяти. FreeBSD имеет поддержку для этой возможности посредством опции ядра PAE, доступной в FreeBSD 4.9-RELEASE и FreeBSD 5.1-RELEASE и более поздних. В связи с ограничениями архитектуры Intel, не делается никакого различия между памятью ниже или выше 4 гигабайт. Память, размещенная выше 4 гигабайт, просто добавляется к доступной памяти.

Для того, чтобы включить PAE в ядре, просто добавьте приведенную строку в конфигурационный файл ядра:

**Замечание:** Поддержка PAE в FreeBSD существует только для процессоров Intel IA-32. Также следует заметить, что PAE в FreeBSD не было полностью протестировано и должно считаться находящимся в состоянии бета-тестирования по сравнению с другими, стабильными возможностями FreeBSD.

Поддержка PAE в FreeBSD имеет следующие ограничения:

- Процесс не может получить доступ к более, чем 4 гигабайтам пространства VM.
- KLD модули не могут быть загружены в ядро с включенной поддержкой PAE из-за разницы в окружении для сборки модулей и самого ядра.
- Драйверы устройств, которые не используют интерфейс `bus_dma(9)` приведут к повреждению информации в ядре с включенным PAE. Не рекомендуется использовать такие драйвера. По этой причине в FreeBSD 5.X включен конфигурационный файл ядра PAE, из которого удалены все драйверы, о которых известно, что они не работают при включенной поддержке PAE.
- Некоторые системные переменные определяют использование ресурсов памяти по количеству доступной физической памяти. Такие переменные могут привести к ненужному чрезмерному выделению памяти из-за особенностей работы системы PAE. Один из таких примеров - переменная `kern.maxvnodes`, которая управляет максимальным количеством vnode разрешенных в ядре. Рекомендуется установить такую и подобные ей переменные вручную в адекватные значения.
- Возможно, понадобится увеличить пространство виртуальных адресов ядра (KVA) или уменьшить какую-либо переменную (см. выше), значение которой было неоправданно велико и могло привести к исчерпанию KVA. Для этого может быть использована опция ядра `KVA_PAGES`.

В случае сомнений относительно производительности и стабильности рекомендуется обратиться к странице руководства `tuning(7)`. Страница руководства `paе(4)` содержит свежую информацию о поддержке PAE в FreeBSD.

## 8.5. Создание файлов устройств

*Если вы работаете с FreeBSD 5.0 или старше, вы можете пропустить этот раздел. Эти версии используют `devfs(5)` для выделения файлов устройств прозрачно для пользователя.*

Практически все устройства в ядре имеют соответствующий файл, называемый "node" в каталоге `/dev`. Они выглядят как обычные файлы, но на самом деле это специальные "точки входа" в ядро, которые необходимы приложениям для доступа к устройствам. Скрипт для `/bin/sh /dev/MAKEDEV`, который выполняется при первой установке операционной системы, создает практически все поддерживаемые файлы устройств. Однако он не создаёт их *все*, таким образом, когда вы добавляете поддержку нового устройства, следует убедиться, что соответствующие файлы есть в `/dev`, и если их там нет, создать их. Вот простой пример:

Предположим, вы добавили поддержку привода IDE CD-ROM в ядро. Строка, которую вы добавили:

```
device acd0
```



Это означает, что вам необходимо проверить наличие файлов, название которых начинаются строкой `acd0` в каталоге `/dev`, за которой, возможно, следует буква, например `c`, или предшествует `r`, что означает, что это "сырое" (raw) устройство, которое используется для прямого доступа к аппаратному устройству. Если оказывается, что таких файлов в каталоге нет, вы должны перейти в каталог `/dev` и набрать:

```
# sh MAKEDEV acd0
```

Когда скрипт завершит работу, вы увидите, что файлы `acd0c` и `racd0c` созданы, соответственно скрипт отработал корректно.

Для звуковых карт нижеследующая команда создает необходимые файлы:

```
# sh MAKEDEV snd0
```

**Замечание:** Если вы создаете файлы устройств для таких устройств, как звуковые карты и если другие люди имеют доступ к вашей машине, было бы не лишним защитить их от использования извне при помощи добавления их в файл `/etc/fstab`. За дальнейшей информацией обратитесь к странице руководства `fstab(5)`.

Следуйте этой простой процедуре для любых других устройств, которые отсутствуют в ядре `GENERIC` и которые не имеют соответствующих файлов в `/dev`.

**Замечание:** Все контроллеры SCSI используют одинаковый набор файлов в `/dev`, поэтому вам не нужно создавать их. Также, сетевые карты и псевдоустройства `SLIP` и `PPP` не имеют вхождений в `/dev`, поэтому и о них вы не должны беспокоиться.

## 8.6. Если что-то пошло не так

Существует пять категорий проблем, которые могут возникнуть при сборке собственного ядра. Вот они:

Не удаётся отработать команде `config`:

Если команда `confi g(8)` не может отработать, то, скорее всего, вы допустили где-нибудь маленькую ошибку. К счастью `confi g(8)` выведет номер проблемной строки, поэтому вы можете быстро найти строку, содержащую ошибку. Например, если вы видите:

```
config: line 17: syntax error
```

Убедитесь, что опция введена верно путём сравнения с файлом `GENERIC` или другим источником.

Не удаётся отработать команде `make`:

Если не удаётся отработать команде `make`, обычно это означает ошибку в описании конфигурации ядра, которая не достаточно тривиальна для того, чтобы `confi g(8)` мог обнаружить её. Опять-таки, просмотрите файл конфигурации и, если вы все еще не можете решить проблему, напишите

письмо в Список рассылки, посвящённый вопросам и ответам пользователей FreeBSD (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions>) включив в письмо файл конфигурации ядра. Скорее всего проблема будет решена быстро.

Не удаётся установить новое ядро:

Если ядро собралось без проблем, но не удалось его установить (команда `make install` или `make installkernel` сообщила об ошибке), первое, что необходимо проверить, не запущена ли ваша система с уровнем безопасности 1 или выше (обратитесь к `init(8)`). Процедура установки пробует снять флаг неизменяемости с текущего ядра и установить этот флаг для нового ядра. Так как на уровне безопасности 1 и выше запрещено снятие флага неизменяемости для всех файлов в системе, установка должна производиться на уровне безопасности 0 и ниже.

Линия выше применима только к FreeBSD 4.X и более ранним версиям. FreeBSD 5.X, как и более поздние версии, не устанавливает этот флаг на ядро и ошибка при установке ядра возможно указывает на более фундаментальную проблему.

Ядро не загружается:

Если ваше новое ядро не загружается или ему не удаётся обнаружить ваши устройства - не паникуйте! К счастью, в FreeBSD существует отличный механизм для восстановления после установки несовместимого ядра. Просто выберите ядро, которое хотите загрузить в загрузчике FreeBSD. К нему вы можете получить доступ, когда система делает обратный отсчёт от 10 в меню загрузки. Нажмите любую клавишу, кроме **Enter**, введите `unload` и наберите `boot /boot/kernel.old/kernel`, или используйте любое другое ядро, которое загрузится без проблем. Во время переконфигурирования ядра всегда полезно оставлять копию ядра, о котором известно, что оно рабочее.

После загрузки с рабочим ядром вы можете проверить ваш файл конфигурации и попробовать собрать ядро опять. Очень полезным в данном случае окажется файл `/var/log/messages`, в котором, среди других записей, имеются сообщения ядра от каждой успешной загрузки. Также, команда `dmesg(8)` выведет сообщения ядра от текущей загрузки.

**Замечание:** Если у вас возникли проблемы со сборкой ядра, убедитесь, что вы сохранили ядро `GENERIC` или другое рабочее ядро под другим именем, чтобы оно не было удалено при следующей сборке. Вы не можете использовать `kernel.old`, потому что при установке нового ядра `kernel.old` перезаписывается последним установленным ядром, которое может оказаться нерабочим. Также, как можно скорее переместите рабочее ядро в `/boot/kernel`, так как некоторые команды, такие как `ps(1)` будут работать некорректно. Для этого просто переместите каталог, содержащий работоспособное ядро:

```
# mv /boot/kernel /boot/kernel.bad
# mv /boot/kernel.good /boot/kernel
```

В версиях FreeBSD до 5.X, для восстановления работоспособного ядра используйте команду:

```
# chflags noschg /kernel
```

Если вы обнаружите, что не можете этого сделать, скорее всего вы работаете на уровне безопасности (`securelevel(8)`) большем нуля. Отредактируйте переменную `kern_securelevel` в файле `/etc/rc.conf`, установите её в `-1` и перезагрузитесь. Вы можете установить её в предыдущее состояние когда будете довольны своим новым ядром.

И, если вы захотите "закрыть" ваше новое ядро или любой другой файл, таким образом, что он не сможет быть перемещен или изменен, используйте следующую команду:

```
# chflags schg /kernel
```

Ядро работает, но ps(1) больше не работает:

Если вы установили версию ядра отличную от той, с которой были собраны ваши системные утилиты, например, ядро от 5.X на системе 4.X, большая часть системных команд, таких как ps(1) и vmstat(8) не будут больше работать. Вам потребуется перекомпилировать и установить систему той же версии исходных текстов, что и ядро. Это одна из причин, по которой не следует использовать версию ядра, отличную от версии всей остальной системы.

## Примечания

1. Алгоритм автоматической настройки устанавливает maxusers равным количеству памяти в системе, минимум 32 и максимум 384.

# Глава 9. Печать

*Написал Sean Kelly. Реструктурировал и обновил Jim Mock. Перевод на русский язык: Валерий Кравчук.*

## 9.1. Краткий обзор

FreeBSD можно использовать для печати на широком спектре принтеров, от старых матричных до новейших лазерных, без исключений, что позволяет создавать высококачественные распечатки из используемых приложений.

FreeBSD можно также сконфигурировать для работы в качестве сервера печати в сети; в этом качестве FreeBSD может получать задания печати от множества других компьютеров, включая другие компьютеры под управлением ОС FreeBSD, хосты Windows и Mac OS. FreeBSD будет гарантировать печать заданий по одному и может сохранять информацию о том, какие пользователи и машины выполняют основную часть печати, выдавать страницы-«баннеры», показывающие, кому принадлежит распечатка, и многое другое.

При прочтении этой главы вы узнаете:

- Как конфигурировать спулер печати FreeBSD.
- Как устанавливать фильтры печати для специфической обработки определенных заданий печати, включая преобразование поступающих на печать документов в форматы, которые понимает принтер.
- Как включить при печати колонтитулы или выдачу страниц-баннеров.
- Как печатать на принтеры, подключенные к другим компьютерам.
- Как печатать на принтеры, подключенные непосредственно к сети.
- Как задавать ограничения для принтера, включая ограничение размера заданий печати и запрет печати для отдельных пользователей.
- Как сохранять статистическую информацию о печати и учитывать использование принтера.
- Как решать проблемы печати.

Прежде чем читать эту главу, вы должны:

- Знать, как сконфигурировать и установить новое ядро (Гл. 8).

## 9.2. Введение

Для использования принтеров в ОС FreeBSD надо настроить их для работы с системой спулинга печати Беркли (Berkeley line printer spooling system), также известной как система спулинга **LPD**. Это - стандартная система управления принтером во FreeBSD. В этой главе представлена система спулинга **LPD**, которую часто называют просто **LPD**, и описано ее конфигурирование.

Если вы уже знакомы с **LPD** или другой системой спулинга печати, вы можете сразу перейти к разделу Настройка системы спулинга.

**LPD** управляет всеми аспектами работы принтеров хоста. Она отвечает за несколько вещей:

- Она управляет доступом к непосредственно подключенным принтерам и принтерам, подключенным к другим хостам в сети.
- Она позволяет пользователям посылать файлы на печать; эти послыки называют *заданиями*.
- Она предотвращает одновременный доступ к принтеру нескольких пользователей путем поддержки *очереди* для каждого принтера.
- Она позволяет печатать *страницы заголовка* (их также называют *баннерными* или *начальными* страницами), чтобы пользователи могли легко находить распечатанные задания в пачке распечаток.
- Она обеспечивает установку параметров взаимодействия для принтеров подключенных, к последовательным портам.
- Она может отправлять задания по сети спулеру **LPD** на другом хосте.
- Она может применять специальные фильтры для форматирования заданий для печати на разных языках описания страниц или задействования специфических возможностей принтера.
- Она учитывает использование принтера.

С помощью файла конфигурации (`/etc/printcap`) и за счет предоставления специальных программ фильтрования, можно потребовать от системы **LPD** выполнять все или некоторые из перечисленных выше функций на широком спектре принтерного оборудования.

### 9.2.1. Зачем использовать спулер

Если вы - единственный пользователь системы, вы можете спросить, зачем возиться со спулером, если управление доступом, страницы заголовка или учет использования принтера вам не нужно. Хотя можно обеспечить непосредственный доступ к принтеру, в любом случае следует использовать спулер, поскольку:

- **LPD** печатает задания в фоновом режиме; вам не придется ждать, пока данные будут скопированы на принтер.
- **LPD** позволяет легко пропустить задание печати через фильтры для добавления заголовков с датой/временем или преобразования специального формата файлов (такого как файла `TeX DVI`) в формат, который понимает принтер. Вам не придется выполнять эти шаги вручную.
- Многие свободно распространяемые и коммерческие программы, обеспечивающие возможность печати, обычно предполагают взаимодействие со спулером системы. Путем настройки системы спулинга вы упростите поддержку другого программного обеспечения, которое может быть добавлено в дальнейшем или уже установлено.

## 9.3. Основная настройка

Для использования принтеров с системой спулинга **LPD**, необходимо настроить как сам принтер, так и программное обеспечение **LPD**. Этот документ описывает два уровня настройки:

- См. раздел Простая настройка принтера, чтобы узнать, как подключить принтер, объяснить **LPD**, как с ним взаимодействовать, и отправлять на принтер простые текстовые файлы.
- См. раздел Расширенная настройка принтера, чтобы узнать, как печатать файлы множества специальных форматов, как печатать страницы заголовка, печатать по сети, управлять доступом к принтерам и учитывать использование принтера.

### 9.3.1. Простая настройка принтера

В этом разделе описано, как сконфигурировать принтер и программное обеспечение **LPD** для использования принтера. Здесь рассматриваются следующие вопросы:

- В разделе Настройка оборудования представлены советы по подключению принтера к порту компьютера.
- В разделе Настройка программного обеспечения показано, как настроить файл конфигурации спулера **LPD** (`/etc/printcap`).

Если вы настраиваете принтер, использующий для принятия заданий печати вместо последовательного или параллельного интерфейса сетевой протокол, см. раздел Принтеры с сетевыми интерфейсами для приема данных.

Хотя этот раздел и назван "Простая настройка принтера", это, на самом деле, достаточно сложно. Заставить принтер работать с компьютером и спулером **LPD** - самая сложная часть. Расширенные опции, вроде выдачи страниц заголовков и учета использования, установить несложно, как только принтер заработает.

#### 9.3.1.1. Настройка оборудования

В этом разделе описаны различные способы подключения принтера к ПК. Рассматриваются различные порты и кабели, а также параметры конфигурации ядра, которые может потребоваться установить, чтобы ОС FreeBSD могла взаимодействовать с принтером.

Если вы уже подключили ваш принтер и успешно печатали на него в другой операционной системе, можете перейти к разделу Настройка программного обеспечения.

##### 9.3.1.1.1. Порты и кабели

Принтеры, которые продаются сегодня для использования на ПК, обычно поддерживают один или несколько из следующих интерфейсов:

- *Последовательные* интерфейсы, также известные как RS232C или RS232D, или COM-порты, используют для отправки данных на принтер последовательный порт компьютера. Последовательные интерфейсы широко распространены в компьютерной индустрии, кабели для них легко найти и просто сделать. Для последовательных интерфейсов иногда нужны специальные кабели, и для их использования может потребоваться настраивать достаточно сложные опции взаимодействия. Большинство последовательных портов ПК имеют максимальную скорость передачи 115200 бит/сек, поэтому печатать через них большие графические задания неудобно.
- *Параллельные* интерфейсы используют параллельный порт компьютера для отправки данных на принтер. Параллельные интерфейсы широко распространены на рынке ПК и работают быстрее, чем последовательные RS232. Кабели легко найти, но сделать самостоятельно сложнее. При

использовании параллельных интерфейсов опции взаимодействия обычно задавать не надо, что делает их конфигурирование существенно проще.

Параллельные интерфейсы иногда называют интерфейсами "Centronics", по названию типа разъема на принтере.

- Интерфейсы USB (сокращение от Universal Serial Bus - универсальная последовательная шина), могут работать на еще больших скоростях, чем параллельные или последовательные интерфейсы RS232. Кабели для них - простые и дешевые. USB превосходит последовательный RS232 и параллельный интерфейсы для печати, но не слишком хорошо поддерживается в UNIX-системах. Обойти эту проблему можно, купив принтер с двумя интерфейсами, USB и параллельным, как у многих принтеров.

В общем случае, параллельные интерфейсы обычно обеспечивают только одностороннюю передачу (с компьютера на принтер), тогда как последовательные и USB поддерживают двустороннюю. Более новые параллельные порты (EPP и ECP) и принтеры могут взаимодействовать в обоих направлениях под FreeBSD, если используется кабель, соответствующий стандарту IEEE1284.

Двустороннее взаимодействие с принтером через параллельный порт обычно выполняется одним из двух способов. Первый метод опирается на использование специально созданного драйвера принтера для FreeBSD, который поддерживает специфический язык данного принтера. Этот метод типичен для струйных принтеров и может использоваться для получения информации об уровне чернил и другой информации о состоянии. Второй метод используется, когда принтер поддерживает PostScript.

Фактически, задания PostScript являются программами, посылаемыми для выполнения принтеру; они вообще могут не выдавать результат на бумагу и возвращать их непосредственно компьютеру. PostScript также использует двустороннее взаимодействие для сообщения компьютеру о проблемах, таких как ошибки в PostScript-программе или замятие бумаги. Такая информация может пригодиться пользователям. Более того, лучший способ эффективного учета использования PostScript-принтера требует двустороннего взаимодействия: вы запрашиваете у принтера значение счетчика страниц (сколько страниц напечатал принтер за все время существования), затем посылаете задание пользователя, затем снова запрашиваете значение его счетчика страниц. Вычитаем одно значение из другого, и узнаем, сколько бумаги потратил пользователь.

#### 9.3.1.1.2. Параллельные порты

Для подключения принтера через параллельный интерфейс, соедините принтер и компьютер кабелем Centronics. Инструкции для принтера, для компьютера или обе должны полностью описывать эту процедуру.

Помните, какой параллельный порт компьютера вы использовали. Первый параллельный порт в ОС FreeBSD - /dev/ppc0; второй - /dev/ppc1, и так далее. Имена устройств для принтеров используют ту же схему: /dev/lpt0 для принтера на первом параллельном порту и т.д.

#### 9.3.1.1.3. Последовательные порты

Для подключения принтера через последовательный интерфейс, соедините принтер с компьютером подходящим последовательным кабелем. Инструкции для принтера, для компьютера или обе должны полностью описывать эту процедуру.

Если вы не знаете, что такое "подходящий последовательный кабель", можете попробовать использовать один из следующих:

- *Модемный* кабель соединяет каждый штырёк на одном конце кабеля напрямую с соответствующим штырьком на другом конце. Кабель такого типа также называют кабелем "DTE-to-DCE".
- *Нуль-модемный* кабель соединяет часть штырьков напрямую, другие - меняет (пересылку данных на приём данных, например), а некоторые - закорачивает на каждом разъеме. Кабель такого типа также называют кабелем "DTE-to-DTE" cable.
- Кабель *последовательного принтера*, необходимый для некоторых редко используемых принтеров, похож на нуль-модемный кабель, но посылает часть сигналов на соответствующие штырьки, а не закорачивает их.

Вам надо также настроить эти параметры взаимодействия с принтером, обычно - через элементы управления на лицевой панели или переключатели (DIP switches) на принтере. Выберите максимальную скорость передачи bps (бит в секунду, иногда - *baud rate*), которую могут поддерживать как компьютер, так и принтер. Выберите 7 или 8 битов данных; четность none, even или odd; и 1 или 2 стоп-бита. Также надо выбрать протокол управления передачей: none или XON/XOFF (также известный как "внутриполосный" или "программный"). Запомните выбранные установки для последующего конфигурирования программного обеспечения.

### 9.3.1.2. Настройка программного обеспечения

В этом разделе описана настройка программного обеспечения, необходимая для печати с помощью системы спулинга **LPD** в ОС FreeBSD.

Вот план действий, которые необходимо выполнить:

1. При необходимости, сконфигурировать в ядре поддержку порта, к которому подключен принтер; в разделе Конфигурирование ядра описано, что надо сделать.
2. Установить режим взаимодействия для параллельного порта, если используется параллельный порт; детали представлены в разделе Настройка режима взаимодействия для параллельного порта.
3. Проверить, может ли операционная система посылать данные на принтер. В разделе Проверка взаимодействия с принтером даны советы, как это сделать.
4. Настроить **LPD** для принтера, изменяя файл /etc/printcap. Как это сделать описано далее в этой главе.

#### 9.3.1.2.1. Конфигурирование ядра

Ядро операционной системы компилируется для работы с конкретным набором устройств. Последовательный или параллельный интерфейс для принтера входит в этот набор. Поэтому может понадобиться добавить поддержку для дополнительного последовательного или параллельного порта, если он еще не сконфигурирован в ядре.

Чтобы узнать, поддерживает ли используемое в настоящий момент ядро последовательный интерфейс, наберите:



```
# grep sioN /var/run/dmesg.boot
```

Где  $N$  - номер последовательного порта, начиная с нуля. Если вы получаете результат, подобный следующему:

```
sio2 at port 0x3e8-0x3ef irq 5 on isa
sio2: type 16550A
```

значит, ядро поддерживает порт.

Чтобы узнать, поддерживает ли ядро параллельный интерфейс, наберите:

```
# grep ppcN /var/run/dmesg.boot
```

Где  $N$  номер параллельного порта, начиная с нуля. Если вы получаете результат, подобный следующему:

```
ppc0: <Parallel port> at port 0x378-0x37f irq 7 on isa0
ppc0: SMC-like chipset (ECP/EPP/PS2/NIBBLE) in COMPATIBLE mode
ppc0: FIFO with 16/16/8 bytes threshold
```

значит, ядро поддерживает порт.

Может потребоваться переконфигурировать ядро, чтобы операционная система распознала и использовала параллельный или последовательный порт, используемый для подключения принтера.

Чтобы добавить поддержку последовательного порта, обратитесь к разделу, посвященному конфигурированию ядра. Чтобы добавить поддержку параллельного порта, почитайте этот же раздел и следующий раздел.

### 9.3.1.3. Добавление файлов `/dev` для портов

**Замечание:** FreeBSD 5.0 включает файловую систему `devfs`, которая автоматически создает специальные файлы устройств по мере необходимости. Если вы используете версию FreeBSD с включенной поддержкой `devfs`, можно пропустить этот раздел.

Хотя ядро может уже поддерживать взаимодействие по последовательному или параллельному порту, вам необходим программный интерфейс, через которых работающие в системе программы смогут посылать и принимать данные. Именно для этого и существуют специальные файлы устройств в каталоге `/dev`.

*Для добавления файла в каталог `/dev` для порта:*

1. Станьте пользователем `root` с помощью команды `su(1)`. Введите пароль пользователя `root` в ответ на приглашение.
2. Перейдите в каталог `/dev`:  
# `cd /dev`
3. Введите команду:

```
# ./MAKEDEV port
```

Где *port* - имя специального файла устройства для порта, который вы хотите создать.

Используйте *lpt0* для принтера на первом параллельном порту, *lpt1* для принтера на втором порту и т.д.; используйте *ttud0* для первого последовательного порта, *ttud1* - для второго и так далее.

#### 4. Введите команду:

```
# ls -l port
```

чтобы убедиться, что требуемый специальный файл устройства создан.

#### 9.3.1.3.1. Настройка режима взаимодействия для параллельного порта

При использовании параллельного интерфейса можно выбрать, должна ли ОС FreeBSD взаимодействовать с принтером на основе прерываний или путем опроса. Универсальный драйвер принтера (*lpt(4)*) во FreeBSD 4.X и 5.X использует систему *ppbus(4)*, которая управляет чипсетом порта с помощью драйвера *ppc(4)*.

- Метод взаимодействия *на основе прерываний* является стандартным для ядра GENERIC. По этому методу, операционная система использует линию запроса прерывания (IRQ line) для определения готовности принтера к приему данных.
- Метод взаимодействия *путем опроса* требует от операционной системы постоянно запрашивать принтер, готов ли он к приему данных. Когда он отвечает, что готов, ядро посылает дополнительные данные.

Метод взаимодействия на основе прерываний обычно работает несколько быстрее, но использует ценную линию запроса прерывания. Про некоторые новые принтеры HP утверждают, что они работают некорректно в режиме взаимодействия на основе прерываний, вероятно, из-за некоторой (еще не вполне понятной) проблемы синхронизации. Для этих принтеров необходимо устанавливать режим опроса. Используйте тот режим, который работает. Некоторые принтеры будут работать в обоих режимах, но оказываются крайне медленными в режиме на основе прерываний.

Режим взаимодействия можно установить двумя способами: конфигурируя ядро или с помощью программы *lptcontrol(8)*.

*Для установки режима взаимодействия путем конфигурирования ядра:*

1. Отредактируйте файл конфигурации ядра. Найдите запись *ppc0*. Если вы настраиваете второй параллельный порт, ищите запись *ppc1*. Используйте запись *ppc2* для третьего порта, и так далее.

- Если необходимо установить режим на основе прерываний, для FreeBSD 4.X добавьте спецификацию *irq*:

```
device ppc0 at isa? irq N
```

Где *N* - номер IRQ для параллельного порта компьютера.

Для FreeBSD 5.X, отредактируйте следующую строку:

```
hint.ppc.0.irq="N"
```

в файле */boot/device.hints*, заменив *N* соответствующим номером IRQ. Файл конфигурации ядра также должен содержать драйвер *ppc(4)*:

```
device ppc
```

- Если необходимо установить режим опроса, не добавляйте спецификацию `irq`:

Для FreeBSD 4.X используйте следующую строку в файле конфигурации ядра:

```
device ppc0 at isa?
```

Для FreeBSD 5.X просто удалите из файла `/boot/device.hints` следующую строку:

```
hint.ppc.0.irq="N"
```

В некоторых случаях, этого недостаточно для перевода порта в режим опроса под FreeBSD 5.X. Чаще всего, проблема связана с драйвером `aspi(4)`, который может опрашивать и подключать устройства и, тем самым, управлять режимом доступа к порту принтера. Чтобы решить эту проблему, проверьте конфигурацию `aspi(4)`.

2. Сохраните файл. Затем сконфигурируйте, соберите и установите ядро и перезагрузите систему. Подробнее см. в разделе конфигурирование ядра.

*Для настройки режима взаимодействия с помощью утилиты `lptcontrol(8)`:*

1. Введите команду:

```
# lptcontrol -i -d /dev/lptN
```

для установки режима взаимодействия на основе прерываний для `lptN`.

2. Введите команду:

```
# lptcontrol -p -d /dev/lptN
```

для установки режима взаимодействия по опросу для `lptN`.

Вы можете поместить эти команды в файл `/etc/rc.local` для установки требуемого режима при каждой загрузке системы. Дополнительную информацию об этом ищите на странице справочного руководства `lptcontrol(8)`.

### 9.3.1.3.2. Проверка взаимодействия с принтером

Прежде чем переходить к конфигурированию системы спулинга, надо убедиться, что операционная система может успешно посылать данные на принтер. Намного проще отлаживать взаимодействие с принтером и систему спулинга отдельно.

Для тестирования принтера мы пошлем на него текст. Для принтеров, которые могут непосредственно печатать посланные на них символы, программа `lptest(1)` идеально подходит: она генерирует все 96 печатных символов ASCII в 96 строках.

Для PostScript- (или основанного на другом языке) принтера, необходим более сложный тест. Подойдет небольшая PostScript-программа, вроде следующей:

```
%!PS
100 100 moveto 300 300 lineto stroke
310 310 moveto /Helvetica findfont 12 scalefont setfont
```

```
(Is this thing working?) show
showpage
```

Представленный выше PostScript-код можно поместить в файл и использовать, как показано в примерах в следующих разделах.

**Замечание:** Когда в этом документе речь идет о языке принтера, подразумевается язык типа PostScript, а не PCL компании Hewlett Packard. Хотя PCL имеет прекрасные функциональные возможности, в нем можно смешивать обычный текст с его управляющими последовательностями. PostScript не позволяет непосредственно печатать обычный текст, и это язык принтера именно того рода, для которого надо выполнять специальные настройки.

#### 9.3.1.3.2.1. Проверка параллельного принтера

В этом разделе описано, как проверить, может ли ОС FreeBSD взаимодействовать с принтером, подключенным к параллельному порту.

*Для тестирования принтера на параллельном порту:*

1. Станьте пользователем root с помощью команды su(1).
2. Пошлите данные на принтер.

- Если принтер может печатать обычный текст, используйте утилиту lptest(1). Введите команду:

```
# lptest > /dev/lptN
```

Где *N* - номер параллельного порта, начиная с нуля.

- Если принтер понимает PostScript или другой язык принтера, пошлите на принтер небольшую программу. Введите команду:

```
# cat > /dev/lptN
```

Затем, построчно, *внимательно* введите программу, поскольку вы не сможете отредактировать строку после нажатия клавиши RETURN или ENTER. По окончании ввода программы, нажмите CONTROL+D или другую комбинацию клавиш, используемую для ввода символа конца файла.

Можно также поместить программу в файл и выполнить команду:

```
# cat file > /dev/lptN
```

Где *file* - имя файла, содержащего программу, которую вы хотите послать принтеру.

Вы должны увидеть распечатку. Не переживайте, если текст выглядит не так, как предполагалось; этими проблемами мы займемся позже.

### 9.3.1.3.2.2. Проверка последовательного принтера

В этом разделе описано, как проверить, может ли ОС FreeBSD взаимодействовать с принтером, подключенным к последовательному порту.

*Для тестирования принтера на последовательном порту:*

1. Станьте пользователем root с помощью команды `su(1)`.
2. Отредактируйте файл `/etc/remote`. Добавьте следующую запись:

```
printer:dv=/dev/port:br#bps-rate:pa=parity
```

Где *port* - специальный файл устройства для последовательного порта (`tttyd0`, `tttyd1` и т.д.), *bps-rate* - скорость обработки данных принтером, в битах в секунду, а *parity* - требуемая принтером четность (значение `even`, `odd`, `none` или `zero`).

Вот пример записи для принтера, подключенного к третьему последовательному порту на скорости 19200 bps без четности:

```
printer:dv=/dev/ttyd2:br#19200:pa=none
```

3. Подключитесь к принтеру с помощью `tip(1)`. Введите команду:

```
# tip printer
```

Если этот шаг не срабатывает, снова отредактируйте файл `/etc/remote` и попробуйте использовать устройство `/dev/cuaaN` вместо `/dev/ttydN`.

4. Пошлите данные на принтер.

- Если принтер может печатать обычный текст, используйте утилиту `lptest(1)`. Введите команду:

```
% $lptest
```

- Если принтер понимает PostScript или другой язык принтера, пошлите на принтер небольшую программу. Вводите программу, построчно, *очень внимательно*, поскольку нажатие клавиши Backspace или других клавиш редактирования может иметь значение для принтера. Может также понадобится нажать специальную комбинацию клавиш, обозначающую конец файла, чтобы принтер понял, что получена вся программа. Для PostScript-принтеров нажмите CONTROL+D.

Можно также поместить программу в файл и ввести команду:

```
% >file
```

Где *file* - имя файла, содержащего программу. После того, как утилита `tip(1)` пошлет файл, нажмите требуемую для ввода признака конца файла комбинацию клавиш.

Вы должны увидеть распечатку. Не переживайте, если текст выглядит не так, как предполагалось; этими проблемами мы займемся позже.

#### 9.3.1.4. Включение спулера: файл `/etc/printcap`

Сейчас ваш принтер уже должен быть подключен, ядро (при необходимости) - сконфигурировано для взаимодействия с ним, и вы смогли послать на принтер простые данные. Теперь мы готовы к конфигурированию системы **LPD** для управления доступом к принтеру.

Система **LPD** конфигурируется путем редактирования файла `/etc/printcap`. Система спулинга **LPD** читает этот файл при каждом использовании спулера, так что, изменения в файле сразу же учитываются.

Формат файла `printcap(5)` прост. Используйте свой любимый текстовый редактор для изменения файла `/etc/printcap`. Формат файла идентичен формату других файлов, описывающих характеристики, например, `/usr/share/misc/termcap` и `/etc/remote`. Полная информация о формате представлена на странице справочного руководства `cagetent(3)`.

Простое конфигурирование спулера включает следующие шаги:

1. Выберите имя (и несколько удобных псевдонимов) для принтера и поместите их в файл `/etc/printcap`; подробнее об именовании см. в разделе Именованное принтера.
2. Отключите выдачу начальных страниц (которые по умолчанию выдаются), вставив характеристику `sh`; подробнее об этом см. в разделе Подавление выдачи начальных страниц.
3. Создайте каталог для спулинга и укажите его местонахождение с помощью характеристики `sd`; подробнее об этом см. в разделе Создание каталога спулинга.
4. Выберите специальный файл устройства `/dev` для использования с принтером и укажите его в файле `/etc/printcap` с помощью характеристики `lp`; подробнее об этом см. в разделе Выбор устройства для принтера. Кроме того, если принтер подключен к последовательному порту, настройте параметры взаимодействия с помощью характеристики `ms#`, которая обсуждается в разделе Конфигурирование параметров взаимодействия для спулера.
5. Установите фильтр для обычного текста; подробнее об этом см. в разделе Установка текстового фильтра.
6. Проверьте настройку, напечатав что-нибудь с помощью команды `lpr(1)`. Подробнее об этом см. в разделах Проверка и Выявление проблем.

**Замечание:** Принтеры, использующие специальные языки, например, PostScript-принтеры, не могут непосредственно печатать обычный текст. Простая настройка, представленная выше и описанная в следующих разделах, предполагает, что, если вы устанавливаете такой принтер, то будете печатать только файлы, которые он может обработать.

Пользователи часто предполагают, что они могут печатать обычный текст на любом из установленных в системе принтеров. Программы, взаимодействующие для обеспечения печати с системой **LPD**, обычно исходят из этого же предположения. Если вы устанавливаете такой принтер и хотите иметь возможность посылать на печать задания на языке принтера *и* в виде обычного текста, настоятельно рекомендуется добавить дополнительный шаг к представленной выше простой последовательности настройки: установите программу автоматического преобразования обычного текста в PostScript (или другой язык принтера). В разделе Прием заданий с обычным текстом на PostScript-принтеры рассказано, как это сделать.

#### 9.3.1.4.1. Именованние принтера

Первый (простой) шаг - выбрать имя для принтера. На самом деле, не важно, выберете ли вы функциональное имя или причудливое, поскольку для принтера можно также задать несколько псевдонимов.

По крайней мере, один из принтеров, указанных в файле `/etc/printcap`, должен иметь псевдоним `lp`. Это - стандартное имя принтера. Если пользователи не установят переменную среды `PRINTER` и не укажут имя принтера в командной строке при вводе любой команды системы **LPD**, по умолчанию для ее выполнения будет использован принтер `lp`.

Также широко распространена практика в качестве последнего псевдонима для принтера задавать полное его описание, включая производителя и модель.

После выбора имени и нескольких популярных псевдонимов поместите их в файл `/etc/printcap`. Имя принтера должно начинаться с крайнего левого столбца. Каждый псевдоним отделяйте вертикальной чертой, а после последнего псевдонима поместите двоеточие.

В следующем примере мы начнем со скелетного файла `/etc/printcap`, определяющего два принтера (построчный принтер Diablo 630 и лазерный PostScript-принтер Panasonic KX-P4455):

```
#
# /etc/printcap для хоста rose
#
rattan|line|diablo|lp|Diablo 630 Line Printer:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:
```

В этом примере первый принтер назван `rattan` и ему заданы псевдонимы `line`, `diablo`, `lp` и `Diablo 630 Line Printer`. Поскольку у него есть псевдоним `lp`, он является стандартным принтером. Второму принтеру дано имя `bamboo` и ему заданы псевдонимы `ps`, `PS`, `S`, `panasonic` и `Panasonic KX-P4455 PostScript v51.4`.

#### 9.3.1.4.2. Подавление выдачи начальных страниц

Система спулинга **LPD** будет по умолчанию печатать *заголовочную страницу* для каждого задания. Заголовочная страница содержит имя пользователя, отправившего задание, хост, с которого поступило задание, и имя задания, красивыми большими буквами. К сожалению, все эти дополнительные тексты мешают отладке простой настройки принтера, поэтому мы будем отключать выдачу начальных страниц.

Для подавления выдачи начальных страниц добавьте характеристику `sh` к записи принтера в файле `/etc/printcap`. Вот пример файла `/etc/printcap` с добавлением `sh`:

```
#
# /etc/printcap для хоста rose - никаких начальных страниц
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
    :sh:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
    :sh:
```

Обратите внимание, как мы использовали правильный формат: первая строка начинается с самого левого столбца, а последующие строки смещены на один символ табуляции. Каждая строка в записи, кроме последней, завершается символом обратной косой.

#### 9.3.1.4.3. Создание каталога для спулинга

Следующий шаг в простой настройке спулера - создать *каталог для спулинга*, каталог, в котором находятся задания печати, пока не будут напечатаны, и находятся еще несколько других файлов для поддержки спулера.

Из-за присущих каталогам спулинга постоянных изменений, принято помещать эти каталоги в каталог `/var/spool`. Кроме того, не нужно создавать резервные копии содержимого каталогов спулинга. Пересоздать их можно с помощью простой команды `mkdir(1)`.

Принято также задавать для каталога имя, совпадающее с именем принтера, как показано ниже:

```
# mkdir /var/spool/имя-принтера
```

Однако при наличии большого количества принтеров в сети может иметь смысл поместить все каталоги спулинга в один каталог, который просто резервируется для печати с помощью **LPD**. Мы сделаем это для наших двух принтеров, `rattan` и `bamboo`:

```
# mkdir /var/spool/lpd
# mkdir /var/spool/lpd/rattan
# mkdir /var/spool/lpd/bamboo
```

**Замечание:** Если вас интересует конфиденциальность заданий, отправляемых пользователями на печать, можно защитить каталог спулинга, чтобы он не был общедоступным. Каталоги спулинга должны принадлежать и быть доступны на чтение, запись и просмотр содержимого пользователю `daemon` и группе `daemon`, и никому больше. Мы установим это для каталогов спулинга принтеров из нашего примера:

```
# chown daemon:daemon /var/spool/lpd/rattan
# chown daemon:daemon /var/spool/lpd/bamboo
# chmod 770 /var/spool/lpd/rattan
# chmod 770 /var/spool/lpd/bamboo
```

Наконец, надо сообщить системе **LPD** об этих каталогах с помощью файла `/etc/printcap`. Полное имя каталога спулинга задается с помощью характеристики `sd`:

```
#
# /etc/printcap для хоста rose - добавлены каталоги спулинга
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
    :sh:sd=/var/spool/lpd/rattan:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
    :sh:sd=/var/spool/lpd/bamboo:
```



Обратите внимание, что имя принтера начинается с самого первого столбца, а все последующие строки смещены на один символ табуляции и каждая строка в записи, кроме последней, завершается символом обратной косой.

Если вы не зададите каталог спулинга с помощью характеристики `sd`, система спулинга будет использовать по умолчанию каталог `/var/spool/lpd`.

#### 9.3.1.4.4. Выбор устройства для принтера

В разделе Добавление файлов `/dev` для портов мы выяснили, какой специальный файл устройства в каталоге `/dev FreeBSD` будет использовать для взаимодействия с принтером. Теперь мы сообщаем эту информацию системе **LPD**. Когда у системы спулинга есть задание для печати, она будет открывать указанное устройство от имени программы-фильтра (которая отвечает за передачу данных на принтер).

Задайте полное имя устройства `/dev` в файле `/etc/printcap` с помощью характеристики `lp`.

В нашем текущем примере давайте предположим, что принтер `rattan` подключен к первому параллельному порту, а принтер `bamboo` - к шестому последовательному порту; вот что нужно добавить в файл `/etc/printcap`:

```
#
# /etc/printcap для хоста rose - указано, какие устройства использовать
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
    :sh:sd=/var/spool/lpd/rattan:\
    :lp=/dev/lpt0:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
    :sh:sd=/var/spool/lpd/bamboo:\
    :lp=/dev/ttyd5:
```

Если вы не укажете характеристику `lp` для принтера в файле `/etc/printcap`, система **LPD** использует по умолчанию устройство `/dev/lp`. Устройство `/dev/lp` сейчас в ОС FreeBSD не существует.

Если устанавливаемый принтер подключен к параллельному порту, перейдите к разделу Установка текстового фильтра. Иначе выполните сначала инструкции, представленные в следующем разделе.

#### 9.3.1.4.5. Конфигурирование параметров взаимодействия спулера

Для принтеров на последовательных портах система **LPD** может устанавливать скорость передачи, четность и другие параметры взаимодействия через последовательных порт от имени программы-фильтра, которая посылает данные на принтер. Это полезно потому, что:

- Позволяет опробовать различные параметры взаимодействия, просто редактируя файл `/etc/printcap`; программу-фильтр перекомпилировать не нужно.
- Позволяет системе спулинга использовать одну и ту же программу-фильтр для нескольких принтеров, которые могут иметь различные установки для взаимодействия через последовательный порт.

Следующие характеристики в файле `/etc/printcap` задают параметры взаимодействия через последовательный порт для устройства, указанного в качестве значения характеристики `lp`:

`br#bps-rate`

Устанавливает скорость взаимодействия для устройства в `bps-rate`, где `bps-rate` может иметь значение 50, 75, 110, 134, 150, 200, 300, 600, 1200, 1800, 2400, 4800, 9600, 19200, 38400, 57600 или 115200 бит в секунду.

`ms#stty-mode`

Устанавливает опции для терминального устройства после открытия устройства.  
Поддерживаемые опции описаны на странице справочного руководства `stty(1)`.

Когда система **LPD** открывает устройство, заданное характеристикой `lp`, она устанавливает опции устройства в соответствии со значением характеристики `ms#`. Наибольший интерес представляют режимы `parenb`, `parodd`, `cs5`, `cs6`, `cs7`, `cs8`, `cstopb`, `crtcts` и `ixon`, которые описаны на странице справочного руководства `stty(1)`.

Давайте зададим опции для нашего принтера на шестом последовательном порту. Мы установим скорость передачи 38400. В качестве режима установим без четности с помощью `-parenb`, 8-битовые символы с помощью `cs8`, отсутствие модемного управления с помощью `clocal` и аппаратное управление потоком с помощью опции `crtcts`:

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
      :sh:sd=/var/spool/lpd/bamboo:\
      :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtcts:
```

#### 9.3.1.4.6. Установка текстового фильтра

Теперь мы готовы задать системе **LPD**, какой текстовый фильтр использовать для отправки заданий на принтер. *Текстовый фильтр*, известный также как *входной фильтр*, - это программа, которую система **LPD** запускает при получении задания на печать. Когда система **LPD** запускает текстовый фильтр для принтера, она направляет на стандартный входной поток фильтра задание печати, а его стандартный выходной поток - на устройство принтера, заданное характеристикой `lp`. Предполагается, что фильтр прочитает задание из стандартного входного потока, выполнит все необходимые для принтера преобразования и выдаст результат в стандартный выходной поток, который и будет напечатан. Подробнее о текстовом фильтре см. в разделе *Фильтры*.

Для простой настройки принтера в качестве текстового фильтра можно задать небольшой скрипт командного интерпретатора, который просто выполняет `/bin/cat` для отправки задания на принтер. В составе FreeBSD поставляется другой фильтр, `lpf`, обрабатывающий забор и подчеркивание для принтеров, которые не слишком хорошо справляются с потоком данных, содержащих такие символы. И, конечно же, вы можете использовать любую другую необходимую программу-фильтр. Фильтр `lpf` детально описан в разделе `lpf: текстовый фильтр`.

Сначала давайте создадим скрипт командного интерпретатора `/usr/local/libexec/if-simple` для простого тестового фильтра. Поместите в этот файл следующий текст с помощью любимого текстового редактора:

```
#!/bin/sh
#
```

```
# if-simple - Простой фильтр входного текста для lpd
# Установлен в /usr/local/libexec/if-simple
#
# Просто копирует stdin в stdout. Игнорирует все аргументы фильтра.

/bin/cat && exit 0
exit 2
```

Сделайте этот файл выполняемым:

```
# chmod 555 /usr/local/libexec/if-simple
```

А теперь потребуйте от системы **LPD** его использовать, указав его в качестве значения характеристики `if` в файле `/etc/printcap`. Мы добавим его для двух принтеров, имеющих пока в примере файла `/etc/printcap`:

```
#
# /etc/printcap для хоста rose - добавлен текстовый фильтр
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
    :sh:sd=/var/spool/lpd/rattan:\ :lp=/dev/lpt0:\
    :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
    :sh:sd=/var/spool/lpd/bamboo:\
    :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:\
    :if=/usr/local/libexec/if-simple:
```

**Замечание:** Копию скрипта `if-simple` можно найти в каталоге `/usr/share/examples/printing`.

#### 9.3.1.4.7. Запуск системы **LPD**

Демон `lpd(8)` запускается из `/etc/rc`, а необходимость запуска задается переменной `lpd_enable`. Эта переменная по умолчанию имеет значение `NO`. Если вы еще этого не сделали, добавьте строку:

```
lpd_enable="YES"
```

в файл `/etc/rc.conf`, а затем либо перезапустите машину, либо просто выполните команду `lpd(8)`.

```
# lpd
```

#### 9.3.1.4.8. Проверка

Вы добрались до конца простой настройки системы **LPD**. К сожалению, поздравлять вас еще рано, поскольку надо еще проверить настройку и устранить все выявленные проблемы. Для проверки настройки, попытайтесь что-то распечатать. Для печати с помощью системы **LPD** используется команда `lpr(1)`, которая посылает задание на печать.

Можно скомбинировать `lpr(1)` с программой `lptest(1)`, представленной в разделе Проверка взаимодействия с принтером, генерирующей тестовый текст.

Для тестирования простой настройки **LPD**:

Введите команду:

```
# lptest 20 5 | lpr -Pprinter-name
```

Где *printer-name* - имя (или псевдоним) принтера, заданное в файле `/etc/printcap`. Для проверки стандартного принтера, введите команду `lpr(1)` без аргумента `-P`. Как уже отмечалось, если тестируется принтер, предполагающий использование PostScript, пошлите ему PostScript-программу вместо использования утилиты `lptest(1)`. Это можно сделать, поместив программу в файл и выполнив команду `lpr file`.

Для PostScript-принтера вы должны получить результаты выполнения программы. Если вы используете `lptest(1)`, ваши результаты должны иметь такой вид:

```
!"#$%&'()*+,-./01234
"#$%&'()*+,-./012345
#$%&'()*+,-./0123456
$%&'()*+,-./01234567
%&'()*+,-./012345678
```

Для дальнейшего тестирования принтера, попытайтесь загрузить программы побольше (для принтеров, поддерживающих определенный язык) или выполните команду `lptest(1)` с другими аргументами. Например, команда `lptest 80 60` выдаст 60 строк по 80 символов в каждой.

Если принтер не работает, см. раздел Выявление проблем.

## 9.4. Расширенная настройка принтера

В этом разделе описаны фильтры для печати специально сформатированных файлов, начальных страниц, печати по сети, ограничения и учета использования принтера.

### 9.4.1. Фильтры

Хотя система **LPD** поддерживает сетевые протоколы, очереди, контроль доступа и другие аспекты печати, большая часть *реальной* работы происходит в *фильтрах*. Фильтры - это программы, взаимодействующие с принтером и обеспечивающие учет особенностей устройства и специальных требований. При простой настройке принтера мы установили фильтр для обычного текста—крайне простой, который должен работать с большинством принтеров (см. раздел Установка текстового фильтра).

Однако, чтобы обеспечить преобразования формата, учет использования принтера и индивидуальных особенностей отдельных принтеров и т.п., надо разобраться, как работают фильтры. В конечном итоге, всеми этими аспектами печати должен заниматься фильтр. А плохая новость состоит в том, что, в большинстве случаев, вы *сами* должны предоставить соответствующие фильтры. Хорошая новость состоит в том, что многие фильтры общедоступны; а если подходящих нет, их обычно легко написать.

Кроме того, в составе ОС FreeBSD поставляется один фильтр, `/usr/libexec/lpr/lpf`, работающий со многими принтерами, которые могут печатать обычный текст. (Он обрабатывает символы забора и табуляции в файле, выполняет учет использования, но и не более того.) Есть также ряд фильтров и компонентов фильтров в наборе портов FreeBSD.

Вот что вы найдете в этом разделе:

- В разделе Как работают фильтры сделана попытка дать обзор роли фильтра в процессе печати. Прочтите этот раздел, чтобы понять, что происходит "за кадром", когда система **LPD** использует фильтры. Это понимание поможет предвидеть и решать проблемы, с которыми вы можете столкнуться при добавлении дополнительных фильтров для каждого из принтеров.
- Система **LPD** предполагает, что каждый принтер, по умолчанию, может печатать обычный текст. Это проблематично для PostScript-принтеров (или принтеров на базе другого языка), поскольку они не могут печатать обычный текст непосредственно. В разделе Прием заданий с обычным текстом на PostScript-принтеры описано, что нужно сделать, чтобы решить эту проблему. Прочтите этот раздел, если используете PostScript-принтер.
- PostScript - популярный формат выдачи для многих программ. Некоторые люди даже пишут PostScript-код непосредственно. К сожалению, PostScript-принтеры дороги. В разделе Имитация PostScript на не-PostScript принтерах описано, как можно дополнительно изменить текстовый фильтр принтера для приема и печати данных PostScript не *не-PostScript* принтере. Прочтите этот раздел, если ваш принтер не поддерживает PostScript.
- В разделе Фильтры преобразования описан способ автоматизации преобразования определенных форматов файлов, например, графики или данных для печатного станка, в форматы, которые может обработать ваш принтер. После чтения этого раздела вы сможете настроить свои принтеры так, что пользователи смогут выполнять команду `lpr -t` для печати данных troff, или `lpr -d` для печати данных TeX DVI, или `lpr -v` - для печати растровых изображений, и так далее. Я рекомендую прочитать этот раздел.
- В разделе Выходные фильтры описана не часто используемая возможность задавать выходные фильтры в системе **LPD**. Если только вы не печатаете начальные страницы (см. Начальные страницы), можно, пожалуй, вообще пропустить этот раздел.
- В разделе lpf: текстовый фильтр описана команда `lpf`, - достаточно полный, хотя и простой текстовый фильтр для строчных принтеров (и лазерных принтеров, работающих как строчные), поставляемый в составе ОС FreeBSD. Если надо быстро настроить учет использования принтера для обычного текста или если используется принтер, из которого при получении символов забора идет дым, несомненно, стоит подумать об использовании `lpf`.

**Замечание:** Различные скрипты, описанные далее, можно найти в каталоге `/usr/share/examples/printing`.

#### 9.4.1.1. Как работают фильтры

Как уже упоминалось, фильтр - это выполняемая программа, запускаемая системой **LPD** для поддержки специфических особенностей устройства при взаимодействии с принтером.

Когда системе **LPD** надо напечатать входящий в задание файл, она запускает программу-фильтр. Стандартный входной поток фильтра связывается с файлом, который надо распечатать, его стандартный выходной поток - с принтером, а стандартный поток ошибок перенаправляется в файл

регистрации ошибок (задается характеристикой `lf` в файле `/etc/printcap`, или используется стандартное устройство `/dev/console`).

Запускаемый системой **LPD** фильтр и его аргументы зависят от того, что указано в файле `/etc/printcap`, и какие аргументы указал пользователь для задания в команде `lpr(1)`. Например, если пользователь ввел команду `lpr -t`, система **LPD** должна запустить фильтр `troff`, заданный характеристикой `tf` для соответствующего принтера. Если пользователь хочет печатать обычный текст, система должна запустить фильтр `if` (это верно в большинстве случаев: подробнее см. в разделе Выходные фильтры).

В файле `/etc/printcap` можно задавать три вида фильтров:

- *Текстовый фильтр*, который в документации **LPD** двусмысленно называют *входным фильтром*, обеспечивает печать обычного текста. Рассматривайте его как стандартный фильтр. Система **LPD** предполагает, что любой принтер может по умолчанию печатать обычный текст, а на текстовый фильтр возлагается задача обеспечить, чтобы символы забора, табуляции или другие специальные символы не сбивали принтер с толку. Если вы работаете в среде, где надо учитывать использование принтера, текстовый фильтр должен также учитывать количество напечатанных страниц, обычно, подсчитывая количество напечатанных строк и сравнивая их с количеством строк на страницу, поддерживаемых принтером. Текстовый фильтр запускается со следующим списком аргументов:

**имя-фильтра [-c] -wширина -ldлина -isdвиг -n имя-пользователя -h хост учетный-файл**

где

**-c**

указывается, если задание послано командой `lpr -l`

**ширина**

значение из характеристики `pw` (page width - ширина страницы), указанной в файле `/etc/printcap`, по умолчанию - 132

**длина**

значение из характеристики `pl` (page length - длина страницы), по умолчанию - 66

**сдвиг**

сдвиг, заданный командой `lpr -i`, по умолчанию - 0

**имя-пользователя**

регистрационное имя пользователя, печатающего файл

**хост**

имя хоста, с которого было послано задание

**учетный-файл**

имя учетного файла, задаваемое характеристикой `af`.

- *Фильтр преобразования* преобразует специфичный формат файла в то, что принтер может воспроизвести на бумаге. Например, данные системы набора dtroff нельзя печатать непосредственно, но можно установить фильтр преобразования для файлов dtroff, чтобы преобразовывать данные dtroff в тот вид, который принтер может воспринять и напечатать. В разделе Фильтры преобразования описано всё об этих фильтрах. Фильтры преобразования также необходимы для учета, если предполагается учет использования принтера. Фильтры преобразования запускаются со следующими аргументами:

`имя-фильтра -хширина-пиксела -увысота-пиксела -п имя-пользователя -h хост учетный-файл`

где *ширина-пиксела* - значение характеристики *rx* (по умолчанию - 0), а *высота-пиксела* - значение характеристики *ry* (по умолчанию - 0).

- *Выходной фильтр* используется только если нет текстового фильтра или если включена выдача начальных страниц. Судя по моему опыту, выходные фильтры используются редко. Они описаны в разделе Выходные фильтры. У выходного фильтра есть всего два аргумента:

`имя-фильтра -wширина -lдлина`

которые идентичны аргументам `-w` и `-l` текстового фильтра.

Фильтры также должны *завершать работу* со следующим статусом выхода:

`exit 0`

Если фильтр успешно напечатал файл.

`exit 1`

Если фильтр не смог напечатать файл, но хочет, чтобы система **LPD** попыталась распечатать файл ещё раз. Система **LPD** перезапустит фильтр, если его работа завершена с этим статусом.

`exit 2`

Если фильтр не смог напечатать файл и не хочет, чтобы система **LPD** пыталась его печатать еще раз. Система **LPD** удалит файл.

Поставляемый в составе FreeBSD текстовый фильтр `/usr/libexec/lpr/lpr` использует аргументы, задающие ширину и длину страницы для определения того, когда посылать символ прогона страницы (form feed) и как учитывать использование принтера. Он использует переданные в качестве аргументов имя пользователя, хост и учетный файл для внесения учетных записей.

При поиске фильтров убедитесь, что они совместимы с системой **LPD**. Если да, они должны поддерживать описанные выше списки аргументов. Если вы планируете создавать фильтры для общего использования, позаботьтесь о поддержке этих списков аргументов и кодов выхода.

#### 9.4.1.2. Прием заданий с обычным текстом на PostScript®-принтеры

Если вы - единственный пользователь компьютера и PostScript-принтера (или принтера на основе другого языка), и вы обещаете никогда не посылать на принтер обычный текст и никогда не использовать возможностей различных программ, требующих посылки на принтер обычного текста, вам можно не заботиться о том, что описано в этом разделе.

Но, если вы хотите посылать на принтер как задания PostScript, так и обычный текст, рекомендуется дополнить настройку принтера. Для этого надо, чтобы текстовый фильтр определял, является ли поступающее задание обычным текстом или программой на языке PostScript. Все PostScript-задания должны начинаться с `%!` (для других языков принтеров обратитесь к соответствующей документации). Если первые два символа в задании - именно эти, речь идет о PostScript, и мы можем остальную часть задания передавать непосредственно. Если же первые два символа в файле - другие, фильтр будет преобразовывать текст в PostScript и печатать результат.

Как нам это сделать?

Если вы используете последовательный принтер, хороший способ достичь поставленной цели состоит в установке `lprps`. `lprps` - это фильтр для PostScript-принтера, выполняющий двустороннее взаимодействие с принтером. Он обновляет файл состояния принтера, помещая в него подробную информацию, выданную принтером, так что пользователи и администраторы могут узнать, в каком именно состоянии (например, `toner low` или `paper jam`) находится принтер. Но еще важнее, что он включает программу `psif`, которая определяет, является ли входящее задание обычным текстом, и вызывает `textps` (еще одну программу, поставляемую вместе с `lprps`) для преобразования его в PostScript. Затем `lprps` посылает преобразованное задание на принтер.

`lprps` входит в набор портов FreeBSD (см. Набор портов). Вы, конечно, можете загрузить, собрать и установить его самостоятельно. После установки `lprps` просто укажите путь к программе `psif`, входящей в состав пакета `lprps`. Если вы установили `lprps` из набора портов, используйте следующий текст в записи для последовательного PostScript-принтера в файле `/etc/printcap`:

```
:if=/usr/local/libexec/psif:
```

Надо также задать характеристику `rw`; она требует от системы **LPD** открывать принтер в режиме чтения и записи.

При использовании параллельного PostScript-принтера (что не позволяет обеспечить двустороннее взаимодействие с принтером, необходимое для системы `lprps`), можно использовать в качестве текстового фильтра следующий скрипт командного интерпретатора:

```
#!/bin/sh
#
#  psif - Печать PostScript или обычного текста на PostScript-принтере
#  Скрипт, а НЕ версия, входящая в состав lprps
#  Установлен в /usr/local/libexec/psif
#

IFS="" read -r first_line
first_two_chars=$(expr "$first_line" : '\(..\)')

if [ "$first_two_chars" = "%!" ]; then
    #
    #  Задание PostScript, печатать его.
    #
    echo "$first_line" && cat && printf "\004" && exit 0
    exit 2
else
    #
    #  Обычный текст, преобразовать его, а затем напечатать.
    #
```



```
( echo "$first_line"; cat ) | /usr/local/bin/textps && printf "\004" && exit 0
exit 2
fi
```

В представленном выше скрипте, `textps` - отдельно установленная программа для преобразования обычного текста в PostScript. Можно использовать любую программу преобразования текста в PostScript. Набор портов FreeBSD (см. Набор портов) включает полнофункциональную программу преобразования текста в PostScript под названием `a2ps`, которую тоже можно попробовать использовать.

#### 9.4.1.3. Имитация PostScript на не-PostScript принтерах

PostScript является *фактическим* стандартом для высококачественного набора и печати. PostScript, однако, - *дорогой* стандарт. К счастью, благодаря компании Aladdin Enterprises есть свободный аналог PostScript под названием **Ghostscript**, который работает с FreeBSD. Ghostscript может читать большинство PostScript-файлов и выдавать соответствующие страницы на множество устройств, включая много моделей не-PostScript принтеров. Установив Ghostscript и используя специальный текстовый фильтр для принтера, можно заставить ваш не-PostScript принтер работать фактически как PostScript-принтер.

Ghostscript входит в набор портов FreeBSD, если вы хотите устанавливать его оттуда. Вы можете также легко загрузить, собрать и установить его самостоятельно.

Для имитации PostScript надо, чтобы текстовый фильтр определял, печатается ли PostScript-файл. Если нет, фильтр будет передавать файл на принтер непосредственно; в противном случае, он будет использовать Ghostscript, чтобы сначала преобразовать файл в формат, который поймет принтер.

Рассмотрим пример: следующий сценарий представляет собой текстовый фильтр для принтеров Hewlett Packard DeskJet 500. Для других принтеров замените аргумент `-sDEVICE` в команде `gs` (Ghostscript). (Введите команду `gs -h` для получения списка устройств, поддерживаемых установленной версией Ghostscript.)

```
#!/bin/sh
#
# ifhp - Печать Ghostscript-эмулированного PostScript на DeskJet 500
# Установлен в /usr/local/libexec/ifhp
#
# Обращать LF как CR+LF (чтобы избежать "эффекта ступенек"
# на принтерах HP/PCL:
#
printf "\033&k2G" || exit 2

#
# Прочитать первые два символа файла
#
IFS="" read -r first_line
first_two_chars=`expr "$first_line" : '\(..\)'`

if [ "$first_two_chars" = "%!" ]; then
#
# Это PostScript; используем Ghostscript для чтения, преобразования и печати.
```

```

#
/usr/local/bin/gs -dSAFER -dNOPAUSE -q -sDEVICE=djet500 \
-sOutputFile=- - && exit 0
else
#
# Обычный текст или HP/PCL, поэтому просто печатаем его напрямую; печатаем в
# конце символ прогона страницы, чтобы была выдана последняя страница.
#
echo "$first_line" && cat && printf "\033&l0H" &&
exit 0
fi

exit 2

```

Наконец, надо указать системе **LPD**, какой фильтр использовать, задав характеристику `if`:

```
:if=/usr/local/libexec/ifhp:
```

Вот и все. Теперь можно выполнять `lpr plain.text` и `lpr whatever.ps`, и обе команды должны успешно печатать.

#### 9.4.1.4. Фильтры преобразования

После завершения простой настройки, описанной в разделе Простая настройка принтера, прежде всего, вам может потребоваться установить фильтры преобразования для любимых форматов файлов (кроме обычных текстов ASCII).

##### 9.4.1.4.1. Зачем устанавливать фильтры преобразования?

Фильтры преобразования упрощают печать различного рода файлов. В качестве примера, предположим, что активно используется издательская система  $\text{\TeX}$  и имеется PostScript-принтер. При каждой генерации DVI-файла из  $\text{\TeX}$ , мы не можем печатать его непосредственно, пока не преобразуем в PostScript. Для этого используется такая последовательность команд:

```
% dvips seaweed-analysis.dvi
% lpr seaweed-analysis.ps
```

Установив фильтр преобразования для файлов DVI, мы можем не конвертировать файл каждый раз вручную, возложив эту задачу на систему **LPD**. Теперь при каждом получении DVI-файла нас от его распечатки отделяет только один шаг:

```
% lpr -d seaweed-analysis.dvi
```

Мы заставили систему **LPD** автоматически преобразовывать DVI-файл, указав опцию `-d`. Все опции преобразования представлены в разделе Опции форматирования и преобразования.

Для каждой из опций преобразования, которая должна поддерживаться принтером, установите *фильтр преобразования* и укажите его полное имя в файле `/etc/printcap`. Фильтр преобразования аналогичен текстовому фильтру для простой настройки принтера (см. раздел Установка текстового фильтра), но вместо печати обычного текста он преобразует файл в формат, который может понять принтер.

#### 9.4.1.4.2. Какие фильтры преобразования следует устанавливать?

Устанавливать надо те фильтры преобразования, которые предполагается использовать. Если вы часто печатаете файлы DVI, значит, фильтр преобразования DVI необходим. Если вам часто приходится печатать результаты работы troff, может потребоваться фильтр troff.

В следующей таблице представлены фильтры, с которыми работает система **LPD**, их соответствующие характеристики для файла `/etc/printcap`, а также способ их вызова в команде `lpr`:

Тип файла	Характеристика <code>/etc/printcap</code>	Опция <code>lpr</code>
cifplot	cf	-c
DVI	df	-d
plot	gf	-g
ditroff	nf	-n
Текст на языке FORTRAN	rf	-f
troff	tf	-f
растровое изображение	vf	-v
обычный текст	if	никакой, -p или -l

В нашем примере использование `lpr -d` означает, что для принтера должна быть задана характеристика `df` в записи в файле `/etc/printcap`.

Вопреки мнению многих, форматы вроде текста на языке FORTRAN и `plot`, вероятно, устарели. У себя на машине вы можете дать новые значения этим или любым другим опциям форматирования, установив соответствующие специализированные фильтры. Например, пусть необходимо напрямую печатать файлы Printerleaf (файлы настольной издательской системы Interleaf), но вообще вы не собираетесь печатать файлы типа `plot`. Можно установить фильтр преобразования Printerleaf в качестве значения характеристики `gf` и научить своих пользователей, что команда `lpr -g` означает "печатать файлы Printerleaf".

#### 9.4.1.4.3. Установка фильтров преобразования

Поскольку фильтры преобразования представляют собой программы, не входящие в базовую поставку FreeBSD, их, видимо, надо помещать в каталоге `/usr/local`. Популярное местонахождение - каталог `/usr/local/libexec`, поскольку эти фильтры являются специализированными программами для выполнения системой **LPD**; обычным пользователям никогда не понадобится их выполнять.

Для включения фильтра преобразования, укажите его полное имя в качестве значения соответствующей характеристики для принтера в файле `/etc/printcap`.

В качестве примера, давайте добавим фильтр преобразования DVI в запись для принтера `bamboo`. Вот опять пример файла `/etc/printcap`, с новой характеристикой `df` для принтера `bamboo`.

```
#
# /etc/printcap для хоста rose - добавлен фильтр df для bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
    :sh:sd=/var/spool/lpd/rattan:\
    :lp=/dev/lpt0:\
    :if=/usr/local/libexec/if-simple:
```

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
:sh:sd=/var/spool/lpd/bamboo:\
:lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
:if=/usr/local/libexec/psif:\
:df=/usr/local/libexec/psdf:
```

Фильтр DVI - скрипт командного интерпретатора по имени /usr/local/libexec/psdf. Вот его текст:

```
#!/bin/sh
#
#  psdf - фильтр принтера, преобразующий DVI в PostScript
#  Установлен в /usr/local/libexec/psdf
#
#  Вызывается системой lpd при выполнении пользователем команды lpr -d
#
exec /usr/local/bin/dvips -f | /usr/local/libexec/lprps "$@"
```

Это скрипт выполняет команду dvips в режиме фильтрации (аргумент -f) входного потока, представляющего собой задание для печати. Затем запускается фильтр PostScript-принтера lprps (см. раздел Прием заданий с обычным текстом на PostScript-принтеры) с аргументами, переданными системой **LPD** этому скрипту. Команда lprps будет использовать эти аргументы для учета распечатанных страниц.

#### 9.4.1.4.4. Дополнительные примеры фильтров преобразования

Поскольку нет фиксированного набора шагов для установки фильтров преобразования, я просто представлю дополнительные примеры. Используйте их в качестве руководства при создании собственных фильтров. Используйте их непосредственно, если нужно.

Следующий пример фильтра преобразует растровый файл (точнее, GIF-файл) для печати на принтере Hewlett Packard LaserJet III-Si:

```
#!/bin/sh
#
#  hpvf - Преобразовать GIF-файлы в HP/PCL и напечатать
#  Установлен в /usr/local/libexec/hpvf

PATH=/usr/X11R6/bin:$PATH; export PATH
giftopnm | pmtopgm | pgmtopbm | pbmtolj -resolution 300 \
    && exit 0 \
    || exit 2
```

Он работает путем преобразования GIF-файла в переносимый формат аунтар, его - в переносимый формат граунтар, затем - в переносимый bitmap, а уже его - в данные, подходящие для LaserJet/PCL.

Вот файл /etc/printcap с записью для принтера, в которой используется представленный выше фильтр:

```
#
#  /etc/printcap для хоста orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
```

```
:lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak:mx#0:\
:if=/usr/local/libexec/hpif:\
:vf=/usr/local/libexec/hpvf:
```

Следующий скрипт является фильтром преобразования для печати данных troff, получаемых из системы набора groff, на PostScript-принтере bamboo:

```
#!/bin/sh
#
# pstf - Преобразует выдаваемые groff данные troff в PS и печатает.
# Установлен в /usr/local/libexec/pstf
#
exec grops | /usr/local/libexec/lprps "$@"
```

Представленный выше скрипт снова использует команду lprps для взаимодействия с принтером. Если принтер подключен к параллельному порту, придется использовать следующий скрипт:

```
#!/bin/sh
#
# pstf - Преобразует выдаваемые groff данные troff в PS и печатает.
# Установлен в /usr/local/libexec/pstf
#
exec grops
```

Вот и все. Вот какую запись надо добавить в файл /etc/printcap, чтобы включить этот фильтр:

```
:tf=/usr/local/libexec/pstf:
```

Вот пример, который пригодится старым специалистам по языку FORTRAN. Это фильтр для печати текста программы на языке FORTRAN на любом принтере, который может непосредственно печатать обычный текст. Мы установим его для принтера teak:

```
#!/bin/sh
#
# hprf - Фильтр текста на языке FORTRAN для LaserJet 3si:
# Установлен в /usr/local/libexec/hprf
#
printf "\033&k2G" && fpr && printf "\033&l0H" &&
exit 0
exit 2
```

Нужно добавить следующую строку к записи в файле /etc/printcap для принтера teak, чтобы включить этот фильтр:

```
:rf=/usr/local/libexec/hprf:
```

Перейдем к последнему, более сложному примеру. Мы добавим фильтр DVI для уже использовавшегося принтера LaserJet по имени teak. Сначала простая часть: изменить файл /etc/printcap, указав местонахождение фильтра DVI:

```
:df=/usr/local/libexec/hpdf:
```

А теперь - часть посложнее: создать фильтр. Для этого нам понадобится программа преобразования DVI в LaserJet/PCL. Набор портов FreeBSD (см. Набор портов) содержит одну: соответствующий пакет называется `dvi2xx`. Установка этого пакета дает нам необходимую программу, `dvilj2p`, которая преобразует DVI в коды, подходящие для LaserJet IIp, LaserJet III и LaserJet 2000.

Команда `dvilj2p` требует создания достаточно сложного фильтра `hpdf`, поскольку она не может читать стандартный входной поток. Она хочет работать с именем файла. Что еще хуже, имя файла должно завершаться расширением `.dvi`, так что использование стандартного входного потока `/dev/fd/0` тоже проблематично. Мы можем обойти эту проблему, создав (символическую) связь (с именем, завершающимся суффиксом `.dvi`) с устройством `/dev/fd/0`, тем самым, заставив команду `dvilj2p` читать из стандартного входного потока.

Единственная оставшаяся проблема состоит в том, что мы не можем создавать временную связь в каталоге `/tmp`. Символьные связи принадлежат пользователю и группе `bin`. Фильтр же работает от имени пользователя `daemon`. А у каталога `/tmp` установлен `sticky bit`. Фильтр сможет создать связь, но не сможет почистить за собой и удалить ее, поскольку связь будет принадлежать другому пользователю.

Вместо этого, фильтр будет создавать символическую связь в текущем рабочем каталоге, которым является каталог спулинга (задаваемый характеристикой `sd` в файле `/etc/printcap`). Это отличное место для выполнения фильтрами своих действий, особенно потому, что (иногда) в каталоге спулинга места больше, чем в `/tmp`.

Вот, наконец, и сам фильтр:

```
#!/bin/sh
#
#  hpdf - Печать данных DVI на принтере HP/PCL
#  Установлен в /usr/local/libexec/hpdf

PATH=/usr/local/bin:$PATH; export PATH

#
#  Определяем функцию для удаления временных файлов. Они существуют
#  в текущем каталоге - в каталоге спулинга для принтера.
#
cleanup() {
    rm -f hpdf$$dvi
}

#
#  Определяем функцию для обработки критических ошибок: напечатать заданное
#  сообщение и выйти с кодом 2. Код выхода 2 сообщает системе LPD, что не
#  надо повторно пытаться печатать задание.
#
fatal() {
    echo "$@" 1>&2
    cleanup
    exit 2
}

#
#  Если пользователь удаляет задание, система LPD будет посылать сигнал SIGINT,
```

```

# поэтому перехватываем SIGINT (и пару других сигналов), чтобы убрать за собой.
#
trap cleanup 1 2 15

#
# Гарантируем, что не конфликтуем с существующими файлами.
#
cleanup

#
# Связываем входной файл DVI со стандартным входным потоком (файлом для печати).
#
ln -s /dev/fd/0 hpdf$$dvi || fatal "Cannot symlink /dev/fd/0"

#
# Заменяем LF = CR+LF
#
printf "\033&k2G" || fatal "Cannot initialize printer"

#
# Преобразуем и печатаем. Значение, возвращаемое программой dviIj2p, не надежно,
# так что мы его игнорируем.
#
dviIj2p -Ml -q -e- dfhp$$dvi

#
# Убираем за собой и завершаем работу
#
cleanup
exit 0

```

#### 9.4.1.4.5. Автоматизированное преобразование: альтернатива фильтрам преобразования

Все эти фильтры преобразования многое дают для среды печати, но требуют от пользователя указывать (в командной строке `lpr(1)`), какой именно фильтр использовать. Если пользователи не особенно разбираются в компьютерах, необходимость указывать опцию фильтра будет их раздражать. Что еще хуже, однако, при неправильном указании опции фильтрования может быть применен фильтр, не соответствующий типу файла, и принтер испортит несколько сотен страниц бумаги.

Вместо установки фильтров преобразования, можно попытаться заставить текстовый фильтр (поскольку он применяется по умолчанию) определять тип файла, который его попросили напечатать, и затем автоматически вызывать соответствующий фильтр преобразования. В этом могут помочь утилиты вроде `file`. Конечно, будет сложно различать *некоторые* типы файлов—и, конечно же, можно задавать фильтры преобразования только для них.

В наборе портов FreeBSD есть текстовый фильтр, выполняющий автоматическое преобразование; это `apsfilter`. Он может выявлять обычный текст, PostScript и файлы DVI, выполнять соответствующие преобразования и печатать результат.

#### 9.4.1.5. Выходные фильтры

Система спулинга **LPD** поддерживает еще один тип фильтров, который мы еще не рассматривали: выходные фильтры. Выходной фильтр предназначен только для печати обычного текста, как текстовый фильтр, но с множеством упрощений. Если вы используете выходной фильтр, а текстовый фильтр не задан, то:

- Система **LPD** запускает выходной фильтр один раз для всего задания, а не для каждого файла задания.
- Система **LPD** не пытается определить начало или конец файлов в задании для выходного фильтра.
- Система **LPD** не передает выходному фильтру имя пользователя или хоста, так что этот фильтр не предназначен для учета использования принтера. Фактически, он получает всего два аргумента:

*имя-фильтра -wширина -lдлина*

Где *ширина* берется из характеристики *rw*, а *длина* - из характеристики *rl* для соответствующего принтера.

Не соблазняйтесь простотой выходного фильтра. Если вы хотите, чтобы каждый файл в задании начинал печататься с новой страницы, выходной фильтр *не поможет*. Используйте текстовый фильтр (также известный как входной); см. раздел Установка текстового фильтра. Более того, выходной фильтр, фактически, - *более сложный*, поскольку он должен проверять посылаемый ему поток байтов в поисках специальных символов-флагов и посылать себе сигналы от имени системы **LPD**.

Однако выходной фильтр *необходим*, если надо выдавать начальные страницы и требуется посылать управляющие последовательности или другие строки инициализации, чтобы можно было напечатать начальную страницу. (Но он *не поможет*, если необходимо учитывать начальные страницы для пользователя, поскольку система **LPD** не передает выходному фильтру никакой информации о пользователе или хосте.)

На одном принтере система **LPD** позволяет совместно с выходным использовать текстовый или другие фильтры. В таких случаях, система **LPD** будет запускать выходной фильтр только для печати начальной страницы (см. раздел Начальные страницы). Система **LPD** затем предполагает, что выходной фильтр *остановится*, посылая ему два байта: ASCII 031 и ASCII 001. Когда выходной фильтр видит эти два байта (031, 001), он должен остановиться, посылая себе сигнал SIGSTOP. Когда система **LPD** закончит выполнение остальных фильтров, она перезапускает выходной фильтр, посылая ему сигнал SIGCONT.

Если есть выходной фильтр, но *нет* текстового, и система **LPD** обрабатывает задания с обычным текстом, **LPD** использует для выполнения задания выходной фильтр. Как уже было сказано, выходной фильтр будет печатать все файлы задания последовательно, без прогонов страниц или других настроек бумаги, а это *вряд ли* вас устроит. Почти во всех случаях необходим текстовый фильтр.

Программа *lpf*, которую мы представили ранее как текстовый фильтр, может также работать как выходной фильтр. Если срочно необходим простой выходной фильтр, но вы не хотите писать код для выявления байтов и посылки сигнала, попробуйте использовать *lpf*. Можно также поместить *lpf* в скрипт командного интерпретатора для обработки любых кодов инициализации, которые может потребовать принтер.



#### 9.4.1.6. lprf: текстовый фильтр

Программа `/usr/libexec/lpr/lpf`, поставляемая в составе двоичного дистрибутива FreeBSD, представляет собой текстовый (входной) фильтр, который может печатать с отступом (если задание послано командой `lpr -i`), пропускать все символы буквально (если задание послано командой `lpr -l`), настраивать позицию печати при получении в задании символов забора и табуляции, а также учитывать количество напечатанных страниц. Она может также использоваться как выходной фильтр.

Программа `lpf` подходит для многих сред печати. И хотя она не позволяет посылать на принтер инициализационные последовательности, легко написать скрипт командного интерпретатора, который будет выполнять необходимую инициализацию, а затем вызывать `lpf`.

Чтобы программа `lpf` корректно выполняла учет страниц, ей необходимо указать корректные значения характеристик `pw` и `pl` в файле `/etc/printcap`. Она использует эти значения для определения того, сколько текста может поместиться на странице и сколько страниц было в задании пользователя. Подробнее об учете использования принтера см. в разделе

Учет использования принтера.

### 9.4.2. Начальные страницы

При наличии *множества* пользователей, использующих различные принтеры, вероятно, можно считать *начальные страницы* неизбежным злом.

Начальные страницы, которые также называют *баннерными* или *разделительными страницами*, идентифицируют, кому принадлежат задания после их печати. Обычно информация на них выдается большими, жирными буквами, возможно, с декоративными рамочками, чтобы в пачке распечаток они отличались от реальных документов, образующих задания пользователей. Они позволяют пользователям быстро находить свои задания. Очевидный недостаток выдачи начальных страниц состоит в том, что для каждого задания надо печатать на одну страницу больше, причем, страница эта хоть сколько-нибудь нужна несколько минут, а затем она оказывается в мусорной корзине или сдается в макулатуру. (Учтите, что начальная страница выдается в начале задания, а не перед каждым файлом, так что бумаги может теряться не так уж и много.)

Система **LPD** может выдавать заголовочные страницы для ваших распечаток автоматически, *если* ваш принтер может непосредственно печатать обычный текст. Если используется PostScript-принтер, потребуется внешняя программа для генерации начальной страницы; см.

Начальные страницы на PostScript-принтерах.

#### 9.4.2.1. Включение выдачи начальных страниц

В разделе Простая настройка принтера мы отключили выдачу начальных страниц, задав характеристику `sh` (что означает "suppress header") в файле `/etc/printcap`. Для включения. Для обеспечения выдачи начальных страниц на принтер, просто удалите характеристику `sh`.

Кажется слишком просто, правда?

Вы правы. *Может* потребоваться задать выходной фильтр для посылки строк инициализации на принтер. Вот пример выходного фильтра для Hewlett Packard PCL-совместимых принтеров:

```
#!/bin/sh
```

```
#
# hprof - Выходной фильтр для Hewlett Packard PCL-совместимых принтеров
# Установлен в /usr/local/libexec/hprof

printf "\033&k2G" || exit 2
exec /usr/libexec/lpr/lpf
```

Задайте полное имя выходного фильтра в качестве значения характеристики of. Подробнее об этом см. в разделе Выходные фильтры.

Вот пример файла /etc/printcap для принтера teak, который мы представили ранее; мы включили выдачу начальных страниц и добавили показанный выше выходной фильтр:

```
#
# /etc/printcap для хоста orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
    :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:\
    :if=/usr/local/libexec/hpif:\
    :vf=/usr/local/libexec/hpvf:\
    :of=/usr/local/libexec/hprof:
```

Теперь, когда пользователи выдают задания на принтер teak, они получают начальную страницу с каждым заданием. Если пользователи хотят тратить время на поиск своих распечаток, они могут подавить вывод начальных страниц, посылая задание с опцией `lpr -h`; другие опции `lpr(1)` см. в разделе Опции начальных страниц.

**Замечание:** Система **LPD** выдает символ прогона страницы (form feed) после начальной страницы. Если ваш принтер использует другой символ или последовательность символов для выброса напечатанной страницы, укажите их в качестве значения характеристики `ff` в файле /etc/printcap.

#### 9.4.2.2. Управление начальными страницами

Включая выдачу начальных страниц, система **LPD** будет выдавать длинный *длинный заголовок*, целую страницу с большими буквами, идентифицирующими пользователя, хост и задание. Ниже представлен пример (kelly напечатала задание по имени outline с хоста rose):

```

k          11      11
k          1       1
k          1       1
k  k      eeee    1       1      y   y
k  k      e   e   1       1      y   y
k k      eeeee   1       1      y   y
kk k      e       1       1      y   y
k  k      e   e   1       1      y  yy
k  k      eeee   111     111     yyy y
                                y
                                y
                                y  y
                                yyyy
```

```

          ll
          t   l   i
          t   l
oooo    u   u   ttttt   l   ii   n nnn   eeee
o   o   u   u   t       l   i   nn   n   e   e
o   o   u   u   t       l   i   n   n   eeeee
o   o   u   u   t       l   i   n   n   e
o   o   u   uu   t t     l   i   n   n   e   e
oooo    uuu u   tt      ll   iii   n   n   eeee

```

```

r rrr    oooo    ssss    eeee
rr   r   o   o   s   s   e   e
r      o   o   ss      eeeee
r      o   o   ss      e
r      o   o   s   s   e   e
r      oooo    ssss    eeee

```

```
Job:  outline
```

```
Date: Sun Sep 17 11:04:58 1995
```

Система **LPD** добавляет прогон страницы после этого текста, чтобы задание начиналось с новой страницы (если только вы не указали характеристику `sf` (suppress form feeds) в записи соответствующего принтера в файле `/etc/printcap`).

Если вы предпочитаете, чтобы система **LPD** создавала *короткий заголовок*, укажите характеристику `sb` (short banner) в файле `/etc/printcap`. Начальная страница будет иметь следующий вид:

```
rose:kelly Job: outline Date: Sun Sep 17 11:07:51 1995
```

Также по умолчанию система **LPD** печатает начальную страницу перед заданием. Для изменения порядка на обратный, укажите характеристику `hl` (header last) в файле `/etc/printcap`.

### 9.4.2.3. Учет начальных страниц

Использование встроенных начальных страниц системы **LPD** порождает определенную парадигму учета использования принтера: начальные страницы пользователи *не должны оплачивать*.

Почему?

Поскольку выходной фильтр - единственная внешняя программа, управляющая выдачей начальных страниц, которая может выполнять учет, а ей не передают информацию о *пользователе или хосте* и учётный файл, так что, она не имеет никакого представления о том, на чей счет отнести использование принтера. Также недостаточно просто "добавлять одну страницу" в текстовом фильтре или в любом из фильтров преобразований (которые имеют информацию о пользователе и хосте), поскольку пользователи могут подавлять выдачу начальных страниц с помощью опции `lpr -h`. И их заставят оплачивать начальные страницы, которые они не печатали. Понятно, что опцию `lpr -h` будут использовать в большинстве случаев те, кто озабочен проблемами окружающей среды, но вы никак не можете стимулировать ее использование.

Также *недостаточно*, чтобы каждый из фильтров генерировал собственные начальные страницы (и, тем самым, мог их учитывать). Если пользователи захотят отказаться от выдачи начальных страниц и укажут опцию `lpr -h`, они все равно их получают, и будут вынуждены оплатить, поскольку система **LPD** не передает информации о наличии опции `-h` ни одному из этих фильтров.

Итак, что же вы можете сделать?

Вы можете:

- Принять парадигму системы **LPD** и сделать начальные страницы бесплатными.
- Установить альтернативную систему вместо **LPD**, такую как **LPRng**. В разделе Альтернативы стандартному спулеру представлена дополнительная информация о других системах спулинга, которые можно использовать вместо **LPD**.
- Написать *умный* выходной фильтр. Обычно выходной фильтр не предназначен для выполнения чего-то кроме инициализации принтера и простых преобразований символов. Он подходит для начальных страниц и заданий с обычным текстом (когда нет текстового (входного) фильтра). Но, если есть текстовый фильтр для заданий с обычным текстом, то система **LPD** будет запускать выходной фильтр только для начальных страниц. И выходной фильтр может анализировать текст начальной страницы, которую генерирует система **LPD**, чтобы определить, на счет какого пользователя и хоста отнести начальную страницу. Единственная проблема этого метода в том, что выходной фильтр все равно не знает, какой учётный файл использовать (ему не передают имя файла, заданное в качестве значения характеристики `af`), но при наличии хорошо известного учётного файла, его имя можно явно указать в выходном фильтре. Для упрощения этапа анализа задайте характеристику `sh` (short header) в файле `/etc/printcap`. Повторимся, что это может оказаться слишком сложным, и пользователи, несомненно, больше оценят великодушного системного администратора, который сделает начальные страницы бесплатными.

#### 9.4.2.4. Начальные страницы на PostScript-принтерах

Как было описано выше, система **LPD** может генерировать начальную страницу в виде обычного текста, что подходит для многих принтеров. Конечно, PostScript-принтеры не могут непосредственно печатать обычный текст, так что, для них возможность выдачи начальных страниц системы **LPD** бесполезна—или почти бесполезна.

Один очевидный способ получить начальные страницы — заставить каждый фильтр преобразования и текстовый фильтр генерировать начальную страницу. Эти фильтры должны использовать аргументы имя пользователя и хост для генерации соответствующей начальной страницы. Недостаток этого метода состоит в том, что пользователи будут всегда получать начальные страницы, даже если будут посылать задания с помощью команды `lpr -h`.

Давайте рассмотрим этот метод детально. Следующий сценарий принимает три аргумента (регистрационное имя пользователя, имя хоста и имя задания) и создает простую начальную страницу на языке PostScript:

```
#!/bin/sh
#
# make-ps-header - выдать начальную страницу на языке PostScript в stdout
# Установлен в /usr/local/libexec/make-ps-header
#
#
# Это единицы измерения PostScript (72 на дюйм). Измените значения для A4 или
# другого используемого формата бумаги:
#
page_width=612
page_height=792
border=72
#
# Проверяем аргументы
#
if [ $# -ne 3 ]; then
    echo "Usage: `basename $0` <user> <host> <job>" 1>&2
    exit 1
fi

#
# Сохраняем значения в переменных, в основном, для упрощения понимания
# последующего PostScript-кода.
#
user=$1
host=$2
job=$3
date=`date`

#
# Пошляем PostScript-код в stdout.
#
exec cat <<EOF
%!PS

%
% Гарантируем, что не будем влять на следующее далее задание пользователя
%
save

%
% Делаем тонкую некрасивую рамку по краям бумаги.
%
$border $border moveto
$page_width $border 2 mul sub 0 rlineto
0 $page_height $border 2 mul sub rlineto
currentscreen 3 -1 roll pop 100 3 1 roll setscreen
```

```

$border 2 mul $page_width sub 0 rlineto closepath
0.8 setgray 10 setlinewidth stroke 0 setgray

%
% Выдаем регистрационное имя пользователя, красивыми, большими и рельефными буквами
%
/Helvetica-Bold findfont 64 scalefont setfont
$page_width ($user) stringwidth pop sub 2 div $page_height 200 sub moveto
($user) show

%
% Теперь выдаем всякие детали
%
/Helvetica findfont 14 scalefont setfont
/y 200 def
[ (Job:) (Host:) (Date:) ] {
200 y moveto show /y y 18 sub def }
forall

/Helvetica-Bold findfont 14 scalefont setfont
/y 200 def
[ ($job) ($host) ($date) ] {
270 y moveto show /y y 18 sub def
} forall

%
% Вот и все
%
restore
showpage
EOF

```

Теперь, каждый из фильтров преобразования и текстовый фильтр может вызвать этот сценарий, чтобы сначала сгенерировать начальную страницу, а затем напечатать задание пользователя. Вот фильтр преобразования DVI, представленные ранее в этом документе, измененный для выдачи начальной страницы:

```

#!/bin/sh
#
# psdf - фильтр преобразования DVI в PostScript
# Установлен в /usr/local/libexec/psdf
#
# Вызывается системой lpd при выполнении пользователем команды lpr -d
#

orig_args="$@"

fail() {
    echo "$@" 1>&2
    exit 2
}

while getopts "x:y:n:h:" option; do

```

```

case $option in
    x|y) ;; # Ignore
    n)   login=$OPTARG ;;
    h)   host=$OPTARG ;;
    *)   echo "LPD started `basename $0` wrong." 1>&2
        exit 2
        ;;
esac
done

[ "$login" ] || fail "No login name"
[ "$host" ] || fail "No host name"

( /usr/local/libexec/make-ps-header $login $host "DVI File"
  /usr/local/bin/dvips -f ) | eval /usr/local/libexec/lprps $orig_args

```

Обратите внимание, как фильтр должен анализировать список аргументов, чтобы определить имя пользователя и имя хоста. Анализ аргументов в других фильтрах аргументов выполняется точно так же. Текстовый фильтр принимает, однако, немного другой набор аргументов (см. раздел Как работают фильтры).

Как уже упоминалось, представленная выше схема хотя и достаточно проста, но не позволяет учесть опцию "подавить вывод начальной страницы" (опция `-h`) команды `lpr`. Если пользователи хотят сберечь деревья (или несколько копеек, если вы берете деньги и за начальные страницы), они не смогут этого сделать, поскольку каждый фильтр будет выдавать начальную страницу для каждого задания.

Чтобы позволить пользователям отключать выдачу начальной страницы для отдельного задания, надо будет использовать прием, представленный в разделе Учет начальных страниц: написать выходной фильтр, который анализирует сгенерированную системой LPD начальную страницу и выдает ее PostScript-версию. Если пользователь посылает задание командой `lpr -h`, система **LPD** не будет генерировать начальную страницу, как и ваш выходной фильтр. В противном случае, ваш выходной фильтр будет читать текст, полученный от системы **LPD**, и посылать на принтер соответствующий PostScript-код для начальной страницы.

Если вы используете PostScript-принтер с последовательным интерфейсом, можно использовать систему `lprps`, которая включает выходной фильтр, `psuf`, делающий то, что описано выше. Помните, что программа `psuf` не учитывает напечатанные пользователями начальные страницы.

### 9.4.3. Печать по сети

FreeBSD поддерживает печать по сети: посылку заданий на удаленные принтеры. Печатью по сети обычно называют две разные ситуации:

- Работа с принтером, подключенным к удаленному хосту. Вы устанавливаете принтер с обычным последовательным или параллельным интерфейсом на одном хосте. Затем, вы настраиваете систему **LPD** для обеспечения доступа к принтеру с других хостов в сети. В разделе Принтеры, установленные на удаленных хостах описано, как это сделать.

- Работа с принтером, подключенным непосредственно к сети. Принтер имеет сетевой интерфейс, кроме (или вместо) более традиционного последовательного или параллельного. Такой принтер может работать следующим образом:
  - Он может понимать протокол **LPD** и даже поддерживать очереди заданий с удаленных хостов. В этом случае, он работает просто как обычный хост с системой **LPD**. Для настройки такого принтера следуйте той же процедуре, которая описана в разделе Принтеры, установленные на удаленных хостах.
  - Он может поддерживать получение потока данных по сети. В этом случае, вы "подключаете" принтер к одному из хостов в сети, делая этот хост ответственным за поддержку очередей заданий и их посылку на принтер. В разделе Принтеры с интерфейсами для получения потока данных по сети представлен ряд советов по установке таких принтеров.

#### 9.4.3.1. Принтеры, установленные на удаленных хостах

Система спуллинга **LPD** имеет встроенную поддержку для посылки заданий на другие хосты, на которых тоже работает система **LPD** (или совместимая с **LPD**). Это позволяет установить принтер на одном хосте и сделать его доступным с других хостов. Она также работает с принтерами, имеющими сетевые интерфейсы и понимающими протокол **LPD**.

Для обеспечения такого рода удаленной печати, сначала установите принтер на одном хосте, *хосте принтера*, с помощью процедуры, описанной в разделе Простая настройка принтера. Выполните любые необходимые дополнительные настройки, как описано в разделе

Расширенная настройка принтера. Не забудьте протестировать принтер и убедиться, обеспечивает ли он заданные возможности системы **LPD**. Также проверьте, что *локальный хост* имеет право использовать службу **LPD** на *удаленном хосте* (см. раздел Ограничение приема заданий с удаленных хостов).

Если вы используете принтер с сетевым интерфейсом, совместимый с системой **LPD**, упомянутым в обсуждении выше *хостом принтера* будет сам принтер, а в качестве *имена принтера* будет выступать имя, которое вы сконфигурировали для принтера. См. документацию, поставляемую с принтером и/или сетевым интерфейсом принтера.

**Подсказка:** Если вы используете Hewlett Packard Laserjet, то при задании принтеру имени `text` будет автоматически выполняться преобразование символа `LF` в последовательность `CRLF`, так что, сценарий `hpif` не понадобится.

Затем, на других хостах, для которых вы хотите обеспечить доступ к принтеру, создайте запись в их файлах `/etc/printcap` со следующими компонентами:

1. Дайте записи любое подходящее имя. Для простоты, однако, имеет смысл задавать такое же имя и псевдонимы, как и на хосте принтера.
2. Характеристику `lp` оставьте пустой, указав это явно (`:lp=:`).
3. Создайте каталог спуллинга и укажите его местонахождение в характеристике `sd`. Система **LPD** будет сохранять задания в нем, прежде чем они будут посланы на хост принтера.



4. Укажите имя хоста принтера в качестве значения характеристики `rm`.

5. Укажите имя принтера на *хосте принтера* в качестве значения характеристики `rp`.

Вот и все. Не нужно перечислять фильтры преобразования, размеры страницы и вообще ничего больше в файле `/etc/printcap`.

Рассмотрим пример. На хосте `rose` есть два принтера, `bamboo` и `rattan`. Мы позволим пользователям хоста `orchid` печатать на эти принтеры. Вот файл `/etc/printcap` для хоста `orchid` (из раздела Включение выдачи начальных страниц). В нем уже есть запись для принтера `teak`; мы добавили две записи для принтеров на хосте `rose`:

```
#
# /etc/printcap для хоста orchid - добавлены (удаленные) принтеры на rose
#

#
# teak - локальный принтер; он подключен непосредственно к orchid:
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
    :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:\
    :if=/usr/local/libexec/ifhp:\
    :vf=/usr/local/libexec/vfhp:\
    :of=/usr/local/libexec/ofhp:

#
# rattan подключен к rose; посылать задания для rattan на хост rose:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
    :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan:

#
# bamboo тоже подключен к rose:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
    :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo:
```

Затем достаточно только создать каталоги спулинга на `orchid`:

```
# mkdir -p /var/spool/lpd/rattan /var/spool/lpd/bamboo
# chmod 770 /var/spool/lpd/rattan /var/spool/lpd/bamboo
# chown daemon:daemon /var/spool/lpd/rattan /var/spool/lpd/bamboo
```

Теперь пользователи хоста `orchid` могут печатать на принтеры `rattan` и `bamboo`. Если, например, пользователь на `orchid` выполнит команду

```
% lpr -P bamboo -d sushi-review.dvi
```

система **LPD** на `orchid` будет копировать задание в каталог спулинга `/var/spool/lpd/bamboo` и учтет, что печатается задание **DVI**. Как только на хосте `rose` появится место в каталоге спулинга принтера `bamboo`, две системы **LPD** передадут файл на хост `rose`. Файл будет ждать в очереди на `rose` пока, наконец, не будет напечатан. Он будет преобразован из формата **DVI** в **PostScript** (поскольку `bamboo` является **PostScript**-принтером) на хосте `rose`.

### 9.4.3.2. Принтеры с интерфейсами для получения потока данных по сети

Часто при покупке сетевой карты для принтера можно приобрести две версии: эмулирующую спулер (более дорогая версия) или просто позволяющую принимать на принтер данные так, как если бы использовался последовательный или параллельный порт (более дешевая версия). В этом разделе описано, как использовать более дешёвую версию. Использование более дорогой версии описано в предыдущем разделе Принтеры, установленные на удаленных хостах.

Формат файла `/etc/printcap` позволяет указывать, какой последовательный или параллельный интерфейс использовать, и (при использовании последовательного интерфейса), какую установить скорость, использовать ли управление потоком, размер отступов для табуляций, преобразование символов новой строки и другие параметры. Но нет способа указать подключение к принтеру, прослушивающему TCP/IP или другой сетевой порт.

Для отправки данных на подключенный к сети принтер, надо разработать программу взаимодействия, которую могут вызывать текстовый фильтр и фильтры преобразований. Вот один из примеров: скрипт `netprint` принимает все данные со стандартного входного потока и посылает их на принтер, подключенный к сети. Мы указываем имя хоста принтера в качестве первого аргумента, а номер порта, к которому надо подключаться - в качестве второго аргумента команды `netprint`. Учтите, что поддерживается только одностороннее взаимодействие (с ОС FreeBSD на принтер); многие сетевые принтеры поддерживают двустороннее взаимодействие, и вы можете захотеть его использовать (для получения состояния принтера, учета и т.п.).

```
#!/usr/bin/perl
#
# netprint - Текстовый фильтр для принтера, подключенного к сети
# Установлен в /usr/local/libexec/netprint
#
$#ARGV eq 1 || die "Usage: $0 <printer-hostname> <port-number>";

$printer_host = $ARGV[0];
$printer_port = $ARGV[1];

require 'sys/socket.ph';

($ignore, $ignore, $protocol) = getprotobyname('tcp');
($ignore, $ignore, $ignore, $ignore, $address)
    = gethostbyname($printer_host);

$sockaddr = pack('S n a4 x8', &AF_INET, $printer_port, $address);

socket(PRINTER, &PF_INET, &SOCK_STREAM, $protocol)
    || die "Can't create TCP/IP stream socket: $!";
connect(PRINTER, $sockaddr) || die "Can't contact $printer_host: $!";
while (<STDIN>) { print PRINTER; }
exit 0;
```

Затем можно использовать этот сценарий в различных фильтрах. Пусть у нас есть строчный принтер Diablo 750-N, подключенный к сети. Принтер принимает данные на печать через порт 5100. Имя хоста для принтера - `scrivener`. Вот текстовый фильтр для этого принтера:

```
#!/bin/sh
#
```

```
# diablo-if-net - Текстовый фильтр для принтера Diablo 'scrivener',
# прослушивающего порт 5100. Установлен в /usr/local/libexec/diablo-if-net
#
exec /usr/libexec/lpr/lpf "$@" | /usr/local/libexec/netprint scrivener 5100
```

## 9.4.4. Ограничение использования принтера

В этом разделе представлена информация об ограничении доступа к принтеру. Система **LPD** позволяет управлять тем, кто может обращаться к принтеру, как локально, так и удаленно, смогут ли они печатать несколько копий, насколько большими могут быть их задания и насколько могут разрастаться очереди печати.

### 9.4.4.1. Ограничение количества копий

Система **LPD** позволяет пользователям легко печатать несколько копий файла. Пользователи могут печатать задания с помощью команды `lpr -#5` (например) и получать пять копий каждого файла в задании. Хорошо это или нет - решать вам.

Если вы считаете, что многочисленные копии только изнашивают ваши принтеры, можете отключить опцию `-#` команды `lpr(1)`, добавив характеристику `sc` в файл `/etc/printcap`. Когда пользователи пошлют задания с опцией `-#`, они увидят:

```
lpr: multiple copies are not allowed
```

Учтите, что если вы настроили удаленный доступ к принтеру (см. раздел [Принтеры, установленные на удаленных хостах](#)), необходимо задать характеристику `sc` также и в файлах `/etc/printcap` удаленных хостов, иначе пользователи все равно смогут посылать задания с несколькими копиями с других хостов.

Рассмотрим пример. Вот файл `/etc/printcap` для хоста `rose`. Принтер `rattan` вполне надежен, поэтому мы разрешим печатать на него несколько копий, но лазерный принтер `bamboo` несколько более изношен, поэтому мы отключим для него печать нескольких копий, добавив характеристику `sc`:

```
#
# /etc/printcap для хоста rose - запрещает печать нескольких копий на bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
    :sh:sd=/var/spool/lpd/rattan:\
    :lp=/dev/lpt0:\
    :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
    :sh:sd=/var/spool/lpd/bamboo:sc:\
    :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
    :if=/usr/local/libexec/psif:\
    :df=/usr/local/libexec/psdf:
```

Теперь нам также нужно добавить характеристику `sc` в файле `/etc/printcap` на хосте `orchid` (и раз уж мы его меняем, давайте отключим печать нескольких копий для принтера `teak`):

```
#
# /etc/printcap для хоста orchid - отключена печать нескольких копий на
# локальном принтере teak и на удаленном принтере bamboo
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
    :lp=/dev/lpt0:sd=/var/spool/lpd/teak:mx#0:sc:\
    :if=/usr/local/libexec/ifhp:\
    :vf=/usr/local/libexec/vfhp:\
    :of=/usr/local/libexec/ofhp:

rattan|line|diablo|lp|Diablo 630 Line Printer:\
    :lp=:rm=rose:rp=rattan:sd=/var/spool/lpd/rattan:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
    :lp=:rm=rose:rp=bamboo:sd=/var/spool/lpd/bamboo:sc:
```

С помощью характеристики `sc` мы предотвращаем использование команды `lpr -#`, но это не мешает пользователям просто выполнить команду `lpr(1)` несколько раз или просто послать один и тот же файл несколько раз в одном задании следующим образом:

```
% lpr forsale.sign forsale.sign forsale.sign forsale.sign forsale.sign
```

Есть много способов предотвратить такое некорректное использование (включая его игнорирование), которые вы можете разработать самостоятельно.

#### 9.4.4.2. Ограничение доступа к принтерам

Вы можете управлять тем, кто и на какие принтеры может печатать, с помощью механизма групп UNIX и характеристики `rg` в файле `/etc/printcap`. Просто поместите пользователей, которым необходимо предоставить доступ к принтеру, в определенную группу, а затем укажите эту группу в качестве значения характеристики `rg`.

Пользователи, не входящие в эту группу (включая `root`) будут получать уведомление `lpr: Not a member of the restricted group` при попытке печатать на контролируемый принтер.

Как и в случае с характеристикой `sc` (подавить выдачу нескольких копий), при необходимости, надо указывать характеристику `rg` и на удаленных хостах, имеющих доступ к вашим принтерам (см. раздел Принтеры, установленные на удаленных хостах).

Например, давайте разрешим всем обращаться к принтеру `rattan`, но только пользователи группы `artists` смогут использовать принтер `bamboo`. Вот знакомый уже файл `/etc/printcap` для хоста `rose`:

```
#
# /etc/printcap для хоста rose - ограничение группы для bamboo
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
    :sh:sd=/var/spool/lpd/rattan:\
    :lp=/dev/lpt0:\
    :if=/usr/local/libexec/if-simple:

bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
    :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:\
    :lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
```

```
:if=/usr/local/libexec/psif:\
:df=/usr/local/libexec/psdf:
```

Давайте не будем менять другой рассматриваемый файл `/etc/printcap` (для хоста `orchid`). Конечно, в результате, любой пользователь `orchid` может печатать на `bamboo`. Возможно, на хосте `orchid` учетных записей и так немного, и вы хотите, чтобы все они имели доступ к принтеру. Или нет.

**Замечание:** Для принтера может быть только одна ограниченная группа.

#### 9.4.4.3. Контроль размеров посылаемых заданий

Если к принтеру обращается несколько пользователей, вам, возможно, понадобится установить ограничение на максимальный размер файлов, которые пользователи могут посылать и печатать. В конечном итоге, размер файловой системы, в которой находятся каталоги спулинга, ограничен, и надо гарантировать, что в нем останется место для заданий других пользователей.

Система **LPD** ограничить максимально допустимый размер файла в задании с помощью характеристики `mx`. Размер задается в блоках, размер которых, `BUFSIZ`, составляет 1024 байта. Если задать этой характеристике значение ноль, размер файла ограничиваться не будет; однако, если характеристика `mx` вообще не задана, то будет использоваться стандартное ограничение - 1000 блоков.

**Замечание:** Ограничение применяется к *файлам в задании*, а не к общему размеру задания.

Система **LPD** не откажется печатать файл больше максимально допустимого для принтера размера. Вместо этого, она поставит в очередь часть файла до заданного предела, и она будет напечатана. Остальное не будет напечатано. Правильность такого поведения не бесспорна.

Давайте установим ограничения для принтеров из наших примеров, `rattan` и `bamboo`. Поскольку PostScript-файлы этих художников обычно бывают весьма большими, мы ограничим их размер пятью мегабайтами. Мы не будем ограничивать использование обычного текстового строчного принтера:

```
#
# /etc/printcap для хоста rose
#

#
# Без ограничения на размер задания:
#
rattan|line|diablo|lp|Diablo 630 Line Printer:\
    :sh:mx#0:sd=/var/spool/lpd/rattan:\
    :lp=/dev/lpt0:\
    :if=/usr/local/libexec/if-simple:

#
# Размер файла - не более пяти мегабайт:
#
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
    :sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:mx#5000:\
```

```
:lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:\
:if=/usr/local/libexec/psif:\
:df=/usr/local/libexec/psdf:
```

Опять таки, ограничения применяются только для локальных пользователей. Если вы настроили удаленный доступ к принтерам, для удаленных пользователей эти ограничения не действуют. Надо задать характеристику `mx` и в файлах `/etc/printcap` удаленных хостов. Более детальную информацию по удаленной печати см. в разделе Принтеры, установленные на удаленных хостах.

Есть еще один специализированный способ ограничить размер заданий печати с удаленных принтеров; см. раздел Ограничение печати заданий с удаленных хостов.

#### 9.4.4.4. Ограничение печати заданий с удаленных хостов

Система спулинга **LPD** обеспечивает несколько способов ограничить посылку заданий с удаленных хостов:

##### Ограничения хостов

Вы можете управлять тем, с каких удаленных хостов локальная система **LPD** принимает запросы, с помощью файлов `/etc/hosts.equiv` и `/etc/hosts.lpd`. Система **LPD** проверяет, поступает ли входящий запрос с хоста, указанного в одном из этих файлов. Если нет, система **LPD** отвергает запрос.

Формат этих файлов простой: по одному имени хоста в строке. Учтите, что файл `/etc/hosts.equiv` также используется протоколом `ruserok(3)` и влияет на программы `rsh(1)` и `rsp(1)`, так что, будьте внимательны.

Например, вот файл `/etc/hosts.lpd` для хоста `rose`:

```
orchid
violet
madrigal.fishbaum.de
```

Это означает, что хост `rose` будет принимать запросы с хостов `orchid`, `violet` и `madrigal.fishbaum.de`. Если любой другой хост попытается обратиться к системе **LPD** хоста `rose`, его задание будет отвергнуто.

##### Ограничения размера

Вы можете управлять тем, сколько свободного места должно оставаться в файловой системе, в которой находится каталог спулинга. Создайте файл с именем `minfree` в каталоге спулинга для локального принтера. Вставьте в этот файл число, задающее, сколько блоков диска (по 512 байтов) должно быть свободными, чтобы удаленное задание было принято.

Это позволяет гарантировать, что удаленные пользователи не заполнят вашу файловую систему. Можно также использовать этот механизм для предоставления определенного преимущества локальным пользователям: они смогут ставить задания в очередь еще долго после того, как свободного места на диске станет меньше, чем указано в файле `minfree`.

Например, давайте добавим файл `minfree` для принтера `bamboo`. Найдем в файле `/etc/printcap` каталог спулинга для этого принтера; вот запись для принтера `bamboo`:

```
bamboo|ps|PS|S|panasonic|Panasonic KX-P4455 PostScript v51.4:\
:sh:sd=/var/spool/lpd/bamboo:sc:rg=artists:mx#5000:\
:lp=/dev/ttyd5:ms#-parenb cs8 clocal crtscts:rw:mx#5000:\
:if=/usr/local/libexec/psif:\
:df=/usr/local/libexec/psdf:
```

Каталог спулинга задается характеристикой `sd`. Укажем, что в файловой системе должно быть три мегабайта (что составляет 6144 блоков диска) свободного места, чтобы система **LPD** принимала удаленные задания:

```
# echo 6144 > /var/spool/lpd/bamboo/minfree
```

## Ограничения пользователей

Вы можете управлять тем, какие удаленные пользователи смогут печатать на локальные принтеры, задавая характеристику `rs` в файле `/etc/printcap`. Когда характеристика `rs` указана в записи для локально подключенного принтера, система **LPD** будет принимать задания с удаленных хостов, *если* пользователь, посылающий задание, также имеет учетную запись с тем же именем на локальном хосте. В противном случае, система **LPD** отвергает задание.

Эта возможность особенно полезна в среде, где есть, например, несколько отделов, совместно использующих сеть, и некоторые пользователи могут переходить из отдела в отдел. Если дать им учетные записи в системах, они смогут использовать принтеры из систем в своих отделах. Если вы хотели бы позволить им использовать *только* принтеры, но не остальные ресурсы вашего компьютера, можно дать им "формальные" учетные записи, без начального каталога и с бесполезным начальным командным интерпретатором вроде `/usr/bin/false`.

## 9.4.5. Учет использования принтера

Итак, вам надо брать деньги за распечатки. А почему нет? Бумага и чернила стоят денег. А есть еще и затраты на поддержку в работоспособном состоянии—принтеры имеют множество движущихся частей и склонны к поломкам. Вы проанализировали состояние принтеров, объемы использования и затраты на их эксплуатацию, и получили определенную стоимость страницы (или фута, метра или чего угодно). Теперь, как же начать реально учитывать распечатки?

Итак, плохая новость состоит в том, что система спулинга **LPD** в этом не сильно поможет. Учет сильно зависит от типа используемого принтера, форматов распечаток и *ваших* требований к оплате использования принтеров.

Для реализации учета надо изменить текстовый фильтр принтера (чтобы учитывать обычные текстовые задания) и фильтры преобразования (чтобы учитывать другие форматы файлов), для подсчета страниц или запроса количества напечатанных страниц у принтера. Не получится обойтись использованием простого выходного фильтра, поскольку он не может выполнять учет. См. раздел **Фильтры**.

Обычно есть два способа выполнения учета:

- *Периодический учет* - более распространенный способ, возможно, потому, что он проще. Когда кто-то печатает задание, фильтр регистрирует пользователя, хост и количество страниц в учетном файле. Каждый месяц, семестр, год или раз в любой желаемый период времени, вы собираете учетные файлы для различных принтеров, суммируете напечатанные каждым пользователем страницы и выставляете суммы за использование. Затем вы очищаете все регистрационные файлы, начиная с чистого листа новый отчетный период.
- *Постоянный учет* используется реже, вероятно, потому, что сложнее в реализации. Этот метод требует от фильтров выставлять пользователям суммы за распечатки сразу после использования принтеров. Как и проверка дисковых квот, этот учет выполняется немедленно. Вы можете не давать пользователям печатать, если баланс на их счету стал отрицательным, а также предоставить им способ проверить и изменить свои "квоты печати". Но этот метод требует поддержки базы данных для отслеживания пользователей и квот.

Система спулинга **LPD** легко поддерживает оба метода: поскольку вы (в большинстве случаев) должны предоставить фильтры, вам придется предоставить и код для учета. Но есть и положительный момент: методы учета могут быть сколько угодно гибкими. Например, можно выбрать периодический или постоянный учет. Можно выбрать, какую именно информацию регистрировать: имена пользователей, имена хостов, типы заданий, количество напечатанных страниц, квадратные метры использованной бумаги, продолжительность печати заданий, и т.д. Это делается путем изменения фильтров так, чтобы они сохраняли соответствующую информацию.

#### 9.4.5.1. Простая система учета использования принтера

В составе FreeBSD поставляется две программы, которые можно сразу использовать для организации простой системы периодического учета. Речь идет о текстовом фильтре `lpf`, описанном в разделе `lpf`: текстовый фильтр, и о программе `rac(8)`, обеспечивающей сбор и суммирование записей из учетных файлов принтеров.

Как уже упоминалось в разделе, посвященном фильтрам (Фильтры), система **LPD** при запуске текстового фильтра и фильтров преобразований передает им имя учетного файла в командной строке. Фильтры могут использовать соответствующий аргумент, чтобы определить, куда записывать учетную информацию. Имя этого файла берется из значения характеристики `af` в файле `/etc/printcap` и, если не заданно как абсолютное, интерпретируется относительно каталога спулинга.

Система **LPD** запускает `lpf` с аргументами ширины и длины страницы (которые берутся из характеристик `rw` и `pl`). Программа `lpf` использует эти аргументы для определения количества бумаги, которая будет использована. После отправки файла на принтер она вносит запись в учетный файл. Эти записи имеют следующий вид:

```
2.00 rose:andy
3.00 rose:kelly
3.00 orchid:mary
5.00 orchid:mary
2.00 orchid:zhang
```

Следует использовать отдельный учетный файл для каждого принтера, поскольку программа `lpf` не реализует механизм блокирования файлов, и два экземпляра `lpf` могут повредить записи друг друга, если записывают одновременно в один и тот же файл. Простой способ выделить отдельный учетный файл для каждого принтера - использовать характеристику `af=acct` в файле `/etc/printcap`. Тогда каждый учетный файл окажется в каталоге спулинга соответствующего принтера и будет назван `acct`.



Когда вы будете готовы выставить пользователям счет за распечатки, запустите программу `рас(8)`. Просто перейдите в каталог спулинга принтера, учетную информацию которого вы хотите обработать, и введите команду `рас`. Вы получите итоговые суммы в долларах, как показано ниже:

Login	pages/feet	runs	price
orchid:kelly		5.00	1 \$ 0.10
orchid:mary		31.00	3 \$ 0.62
orchid:zhang		9.00	1 \$ 0.18
rose:andy		2.00	1 \$ 0.04
rose:kelly	177.00	104	\$ 3.54
rose:mary	87.00	32	\$ 1.74
rose:root	26.00	12	\$ 0.52
total	337.00	154	\$ 6.74

Команда `рас(8)` принимает следующие аргументы:

**-Р *принтер***

По какому *принтеру* подсчитывать итоговые суммы. Эта опция работает, только если в качестве значения характеристики `af` в файле `/etc/printcap` указано абсолютное имя.

**-с**

Сортировать отчет по сумме, а не по имени пользователя в алфавитном порядке.

**-т**

Игнорировать имя хоста в учетных файлах. При указании этой опции, пользователь `smith` на хосте `alpha` считается тем же, что и пользователь `smith` на хосте `gamma`. Обычно эти пользователи считаются разными.

**-р *стоимость***

Вычислять суммы из расчета *стоимость* долларов за страницу или за фут, вместо использования значения характеристики `pc` в файле `/etc/printcap`, или двух центов (как принято по умолчанию). Можно задавать *стоимость* как число с плавающей запятой.

**-r**

Изменить порядок сортировки.

**-s**

Создать итоговый учетный файл и очистить учетный файл.

***имя* . . .**

Выдать учетную информацию только для пользователей с заданными *именами*.

В стандартном отчете, который создает команда `рас(8)`, выдается количество страниц, напечатанное каждым из пользователей с различных хостов. Если для вас хосты не имеют значения (поскольку пользователи могут работать на любом хосте), выполните команду `рас -т` для получения следующих итогов:

Login	pages/feet	runs	price
-------	------------	------	-------

andy	2.00	1	\$	0.04
kelly	182.00	105	\$	3.64
mary	118.00	35	\$	2.36
root	26.00	12	\$	0.52
zhang	9.00	1	\$	0.18
total	337.00	154	\$	6.74

Для получения сумм в долларах программа `рас(8)` использует значение характеристики `рс` в файле `/etc/printcap` (по умолчанию - 200, или 2 цента за страницу). Укажите в качестве значения этой характеристики, в сотых долях цента, стоимость страницы или фута, исходя из которой вы хотите брать деньги за распечатки. Это значение можно переопределить при вызове команды `рас(8)` с помощью опции `-р`. Но при использовании опции `-р` стоимость надо указывать в долларах, а не в сотых долях цента. Например, команда

```
# рас -р1.50
```

приводит к тому, что страница будет стоить один доллар пятьдесят центов. Используя эту опцию, можно фактически начинать грести деньги лопатой.

Наконец, при выполнении команды `рас -s` итоговая информация будет сохранена в итоговом учетном файле, имя которого строится как имя учетного файла принтера с суффиксом `_sum`. Затем учетный файл принтера очищается. Когда команда `рас(8)` выполняется повторно, она пересчитывает итоговый файл для получения начальных сумм, а затем добавляет информацию из обычного учетного файла.

#### 9.4.5.2. Как можно подсчитать количество напечатанных страниц?

Для выполнения хоть отдаленно точного учета надо уметь определять, сколько бумаги использовано для печати задания. Это - основная проблема учета использования принтеров.

Для обычных текстовых заданий решить эту проблему не сложно: надо считать, сколько строк входит в задание, и сравнивать с количеством строк на страницу, которые поддерживает принтер. Не забывайте учитывать символы забоя в файлах, которые приводят к перепечатке строк поверх, а также длинные логические строки, которые переносятся на несколько физических.

Текстовый фильтр `lpf` (представленный в разделе `lpf`: текстовый фильтр) учитывает эти вещи при выполнении учета. Если вы пишете текстовый фильтр, который должен осуществлять учет, может иметь смысл просмотреть исходный код программы `lpf`.

Но как обрабатывать файлы других форматов?

Ну, для преобразования из `DVI` в формат `LaserJet` или из `DVI` в `PostScript`, можно в фильтре анализировать диагностические результаты команды `dvilj` или `dvips`, чтобы определить, сколько страниц было преобразовано. Может оказаться возможным применить этот прием и для других форматов файлов и программ преобразования.

Но эти методы несовершенны из-за того, что принтер мог фактически и не напечатать все эти страницы. Например, он мог замять бумагу, в нем мог закончиться тонер или он мог вообще взорваться—и пользователю все равно пришлось бы платить.

Так что же делать?

Есть только один *надежный* способ *точного* учета. Купите принтер, который может сообщать, сколько бумаги он использовал, и подключите его через последовательный порт или по сети.

Практически все PostScript-принтеры поддерживают такую возможность. Другие модели - тоже (сетевые лазерные принтеры Imagen, например). Измените фильтры для этих принтеров так, чтобы получать количество использованных страниц после печати каждого задания, и пусть они записывают учетную информацию *только* на основе этого значения. Не надо ни считать строки, ни выполнять чреватую ошибками обработку файла.

Конечно, всегда можно поступить великодушно и не брать денег за распечатки.

## 9.5. Использование принтеров

В этом разделе описано, как использовать настроенные принтеры в ОС FreeBSD. Вот сводка команд пользовательского уровня:

`lpr(1)`

Печать заданий

`lpq(1)`

Проверка очередей принтеров

`lprm(1)`

Удаление заданий из очередей принтеров

Есть также административная команда, `lpc(8)`, описанную в разделе Администрирование спулера **LPD**, используемая для управления принтерами и их очередями.

Все три команды, `lpr(1)`, `lprm(1)` и `lpq(1)`, поддерживают опцию `-P имя-принтера`, позволяющую указать, с каким принтером/очередью из указанных в файле `/etc/printcap` работать. Это позволяет посылать, удалять и проверять задания на разных принтерах. Если вы не используете опцию `-P`, эти команды используют принтер, указанный в качестве значения переменной среды `PRINTER`. Наконец, если переменная среды `PRINTER` не задана, эти команды по умолчанию направляются на принтер по имени `lp`.

Далее термин *стандартный принтер* означает принтер, указанный переменной среды `PRINTER` или принтер по имени `lp`, если переменная среды `PRINTER` не задана.

### 9.5.1. Задания печати

Для печати файлов, выполните команду:

```
% lpr имя-файла ...
```

Эта команда печатает каждый из перечисленных файлов на стандартный принтер. Если файлы не указаны, команда `lpr(1)` читает данные для печати со стандартного входного потока. Например, следующая команда печатает некоторые важные системные файлы:

```
% lpr /etc/host.conf /etc/hosts.equiv
```

Для выбора конкретного принтера, введите:

```
% lpr -P имя-принтера имя-файла ...
```

Следующая команда печатает подробный листинг текущего каталога на принтере rattan:

```
% ls -l | lpr -P rattan
```

Поскольку для команды `lpr(1)` файлы не указаны, команда `lpr` читает данные для печати из стандартного входного потока, который содержит результат выполнения команды `ls -l`.

Команда `lpr(1)` может также принимать множество опций для управления форматированием, применения преобразований, печати нескольких копий и т.д. Дополнительную информацию см. в разделе Опции печати.

## 9.5.2. Проверка заданий

При печати с помощью команды `lpr(1)`, данные, которые надо напечатать, помещаются вместе в пакет, который называют "заданием печати", и посылаются системе спулинга **LPD**. Каждый принтер имеет очередь заданий, и ваше задание ждет в этой очереди вместе с другими вашими заданиями и заданиями других пользователей. Принтер печатает эти задания по принципу первым пришло, первым выполнено.

Для получения очереди стандартного принтера, введите команду `lpq(1)`. Чтобы указать конкретный принтер, используйте опцию `-P`. Например, команда

```
% lpq -P bamboo
```

показывает очередь для принтера по имени bamboo. Вот пример результатов выполнения команды `lpq`:

```
bamboo is ready and printing
Rank  Owner   Job  Files                                Total Size
active kelly    9    /etc/host.conf, /etc/hosts.equiv    88 bytes
2nd    kelly    10    (standard input)                   1635 bytes
3rd    mary     11    ...                                78519 bytes
```

Показано, что в очереди bamboo есть три задания. Первое задание, посланное пользователем kelly, получило "номер задания" 9. Каждое задание для принтера получает уникальный номер задания. В большинстве случаев номер задания можно игнорировать, но он потребуется, если надо будет отменить задание; подробнее об этом см. в разделе Удаление заданий.

Задание номер 9 состоит из двух файлов; несколько файлов, указанных в командной строке `lpr(1)`, считаются частью одного задания. Это задание является текущим активным (обратите внимание на слово `active` в столбце "Rank"), т.е. принтер должен сейчас печатать это задание. Второе задание состоит из данных, передаваемых в качестве стандартного входного потока команде `lpr(1)`. Третье задание послано пользователем mary; оно намного больше по объему. Полное имя файла, который печатается, слишком длинное и не помещается, поэтому команда `lpq(1)` просто выдает три точки.

Самая первая строка результатов команды `lpq(1)` тоже полезна: она говорит о том, что сейчас делает принтер (или, по крайней мере, что он делает по мнению системы **LPD**).

Команда `lpq(1)` также поддерживает опцию `-l` для генерации подробного длинного листинга. Вот пример результатов выполнения команды `lpq -l`:

```
waiting for bamboo to become ready (offline ?)
```

```

kelly: 1st                               [job 009rose]
      /etc/host.conf                      73 bytes
      /etc/hosts.equiv                   15 bytes

kelly: 2nd                               [job 010rose]
      (standard input)                   1635 bytes

mary: 3rd                                [job 011rose]
      /home/orchid/mary/research/venus/alpha-regio/mapping 78519 bytes

```

### 9.5.3. Удаление заданий

Если вы передумали печатать задание, можно удалить его из очереди заданий с помощью команды `lprm(1)`. Часто можно использовать `lprm(1)` для удаления активного задания, но часть задания или даже все задание все равно может быть напечатано.

Для удаления задания со стандартного принтера сначала используйте команду `lpq(1)` для поиска номера задания. Затем введите команду:

```
% lprm номер-задания
```

Для удаления задания с указанного принтера, задайте опцию `-P option`. Следующая команда удаляет задание номер 10 из очереди заданий принтера `bamboo`:

```
% lprm -P bamboo 10
```

Для команды `lprm(1)` есть ряд сокращений:

**lprm -**

Удаляет все задания (со стандартного принтера), принадлежащие пользователю, который выполнил команду.

**lprm *пользователь***

Удаляет все задания (для стандартного принтера), принадлежащие указанному *пользователю*. Суперпользователь может удалять задания других пользователей; обычный пользователь может удалять только собственные задания.

**lprm**

Если в командной строке не указаны номер задания, имя пользователя, или указана опция `-`, команда `lprm(1)` удаляет текущее активное задание на стандартном принтере, если оно принадлежит вам. Суперпользователь может удалять любое активное задание.

Добавьте опцию `-P` для любого из перечисленных выше сокращений, чтобы работать с любым необходимым принтером вместо стандартного. Например, следующая команда удаляет все задания текущего пользователя из очереди принтера по имени `rattan`:

```
% lprm -P rattan -
```

**Замечание:** Если вы работаете в сетевой среде, команда `lprm(1)` позволит вам удалять задания только с хоста, с которого они были посланы, даже если тот же принтер доступен и с других хостов. Следующая последовательность команд демонстрирует это:

```
% lpr -P rattan myfile
% rlogin orchid
% lpq -P rattan
Rank  Owner      Job  Files      Total Size
active seeyan      12   ...      49123 bytes
2nd   kelly       13  myfile     12 bytes
% lprm -P rattan 13
rose: Permission denied
% logout
% lprm -P rattan 13
dfa013rose dequeued
cfa013rose dequeued
```

### 9.5.4. Не только обычный текст: опции печати

Команда `lpr(1)` поддерживает несколько опций, управляющих форматированием текста, преобразованием графики и других форматов файлов, выдачей нескольких копий, обработкой задания и др. В этом разделе описаны эти опции.

#### 9.5.4.1. Опции форматирования и преобразования

Следующие опции команды `lpr(1)` управляют форматированием файлов в задании. Используйте эти опции, если задание содержит не простой текст или если вы хотите сформатировать простой текст с помощью утилиты `pr(1)`.

Например, следующая команда печатает файл DVI (из системы верстки  $\text{\TeX}$ ) по имени `fish-report.dvi` на принтере `bamboo`:

```
% lpr -P bamboo -d fish-report.dvi
```

Эти опции применяются для каждого файла в задании, так что нельзя смешивать (например) файлы DVI и `ditroff` в одном задании. Вместо этого посылайте однотипные файлы отдельными заданиями, используя для каждого задания соответствующие опции преобразования.

**Замечание:** Все эти опции, кроме `-r` и `-T`, требуют наличия установленных для целевого принтера фильтров преобразования. Например, опция `-d` требует фильтра преобразования DVI. Подробнее см. в разделе **Фильтры преобразования**.

-с

Печать файлов `cifplot`.

-d

Печать файлов DVI.

-f

Печать текстовых файлов на языке FORTRAN.

-g

Печать графиков.

-i *число*

Сдвинуть результат вправо на *число* столбцов; если *число* не указано, сдвиг выполняется на 8 столбцов. Эта опция работает только с определенными фильтрами преобразования.

**Замечание:** Не помещайте пробелы между -i и числом.

-l

Печать текстовых данных буквально, включая управляющие символы.

-n

Печать данных ditroff (device independent troff).

-p

Форматировать обычный текст перед печатью утилитой pr(1). Подробнее см. pr(1).

-T *заголовок*

Использовать указанный *заголовок* в колонтитуле pr(1) вместо имени файла. Эта опция учитывается только при использовании вместе с опцией -p.

-t

Печать данных troff.

-v

Печать растровых данных.

Вот пример: следующая команда печатает красиво сформатированную версию справочного руководства по команде ls(1) на стандартный принтер:

```
% zcat /usr/share/man/man1/ls.1.gz | troff -t -man | lpr -t
```

Команда zcat(1) распаковывает исходный код страницы справочного руководства ls(1) и передает его команде troff(1), которая форматирует его и выдает результат в формате GNU troff, передаваемый команде lpr(1), посылающей задание спулеру **LPD**. Поскольку мы использовали опцию -t команды lpr(1), спулер при печати задания будет преобразовывать результат GNU troff в формат, понятный стандартному принтеру.

### 9.5.4.2. Опции обработки заданий

Следующие опции команды `lpr(1)` требуют от системы **LPD** специальной обработки задания:

#### **-# *копий***

Выдавать указанное количество *копий* каждого файла в задании вместо одной. Администратор может отключить эту опцию для уменьшения износа принтера и поощрения использования ксерокса. См. раздел Ограничение количества копий.

В следующем примере на стандартный принтер печатается три копии файла `parser.c`, а затем - три копии `parser.h`:

```
% lpr -#3 parser.c parser.h
```

#### **-m**

Посылать почту после завершения задания печати. При указании этой опции, система **LPD** будет посылать почту на ваше имя после завершения обработки вашего задания. В сообщении будет сказано, выполнено ли задание успешно или по ходу была ошибка, и (часто) - в чем она состояла.

#### **-s**

Не копировать файлы в каталог спулинга, а сделать там на них символические связи.

Эту опцию имеет смысл использовать при печати больших заданий. Она экономит место в каталоге спулинга (ваше задание может занять все свободное место в файловой системе, в которой находится каталог спулинга). Она также экономит время, поскольку системе **LPD** не придется копировать каждый байт задания в каталог спулинга.

Есть, однако, и недостаток: поскольку система **LPD** будет ссылаться на исходные файлы непосредственно, вы не сможете изменять или удалять их, пока они не будут распечатаны.

**Замечание:** Если вы печатаете на удаленный принтер, система **LPD** будет вынуждена, так или иначе, скопировать файлы с локального хоста на удаленный, поэтому опция `-s` экономит место только в локальном каталоге спулинга, но не в удаленном. Но, она все равно полезна.

#### **-Г**

Удалять файлы в задании после копирования в каталог спулинга или после печати, если указана опция `-s`. Будьте внимательны при использовании этой опции!

### 9.5.4.3. Опции начальных страниц

Эти опции команды `lpr(1)` изменяют текст, который обычно выдается на начальной странице задания.

Если выдача начальных страниц для целевого принтера отключена, эти опции не действуют.

Информацию по настройке начальных страниц см. в разделе Начальные страницы.



**-С текст**

Заменить имя хоста на начальной странице *текстом*. Обычно на ней выдается имя хоста, с которого было послано задание.

**-J текст**

Заменить имя задания на начальной странице *текстом*. Имя задания обычно совпадает с именем первого файла в задании или имеет значение stdin, если печатается стандартный входной поток.

**-h**

Не выдавать начальной страницы.

**Замечание:** В некоторых организациях эта опция может не действовать, что определяется способом генерации начальных страниц. Подробнее см. в разделе Начальные страницы.

### 9.5.5. Администрирование принтеров

Как администратор принтеров, вы должны их установить, настроить и протестировать. С помощью команды lpc(8) вы можете взаимодействовать с принтерами и другими способами. С помощью lpc(8) вы можете:

- Запускать и останавливать принтеры
- Включать и отключать их очереди
- Изменять порядок заданий в каждой очереди.

Начнем с замечания по терминологии: если принтер *остановлен*, он не будет печатать ничего из своей очереди. Пользователи могут продолжать посылать задания, которые будут ждать в очереди, пока принтер не будет *запущен* или пока очередь не будет очищена.

Если очередь *отключена*, ни один пользователь (кроме root) не может посылать задания на принтер. Во *включенную* очередь можно посылать задания. Принтер для отключенной очереди может быть *запущен*; при этом он будет продолжать печатать находящиеся в очереди задания, пока очередь не станет пустой.

В общем случае, для использования команды lpc(8) необходимо иметь привилегии root. Обычные пользователи могут использовать команду lpc(8) только для получения состояния принтера и перезапуска зависшего принтера.

Далее представлена сводка команд lpc(8). Большинство команд принимает аргумент *имя-принтера*, задающий, с каким принтером работать. Можно использовать значение all вместо *имени-принтера*, означающее все принтеры, перечисленные в файле /etc/printcap.

**abort имя-принтера**

Снять текущее задание и остановить принтер. Пользователи могут продолжать посылать задания, если очередь включена.

**clean *имя-принтера***

Удалить старые файлы из каталога спулинга принтера. Иногда файлы, составляющие задание, не удаляются как положено системой **LPD**, особенно если в ходе печати были ошибки и выполнялось много административных действий. Эта команда находит файлы, не принадлежащие каталогу спулинга, и удаляет их.

**disable *имя-принтера***

Отключить постановку новых заданий в очередь. Если принтер работает, он продолжит печатать задания, остающиеся в очереди. Суперпользователь (**root**) всегда может посылать задания, даже в отключенную очередь.

Эта команда полезна при тестировании вновь установленного принтера или фильтра: отключаем очередь и посылаем задания как **root**. Другие пользователи не смогут посылать задания, пока вы не закончите тестирование и не включите очередь повторно командой **enable**.

**down *имя-принтера* *сообщение***

Отключить принтер. Аналогична последовательности команд **disable** и **stop**. Указанное *сообщение* выдается как состояние принтера при проверке пользователем очереди принтера с помощью **lpq(1)** или запросе его состояния командой **lpc status**.

**enable *имя-принтера***

Включить очередь для принтера. Пользователи могут посылать задания, но принтер не будет их печатать, пока не будет запущен.

**help *имя-команды***

Выдать справочную информацию по команде *имя-команды*. Если *имя-команды* не указано, выдает сводку по имеющимся командам.

**restart *имя-принтера***

Перезапустить принтер. Обычные пользователи могут использовать эту команду, если в результате неких чрезвычайных обстоятельств система **LPD** зависла, но они не могут запустить принтер, остановленный командами **stop** или **down**. Команда **restart** эквивалентна последовательности команд **abort** и **start**.

**start *имя-принтера***

Запустить принтер. Принтер будет печатать задания, находящиеся в его очереди.

**stop *имя-принтера***

Остановить принтер. Принтер закончит печать текущего задания и больше ничего из очереди печатать не будет. Хотя принтер и остановлен, пользователи могут посылать задания во включенную очередь.

**topq *имя-принтера* *задание-или-имя-пользователя***

Переупорядочить очередь для указанного принтера, помещая указанные по номеру *задания* или задания указанного по имени *пользователя* в начало очереди. Для этой команды нельзя

использовать `all` в качестве *имени-принтера*.

ур *имя-принтера*

Включить принтер; команда по действию противоположна команде `down`. Эквивалентна последовательности команд `start` и `enable`.

Утилита `lpc(8)` принимает перечисленные выше команды в командной строке. Если команда не указана, утилита `lpc(8)` входит в интерактивный режим, в котором можно вводить команды, пока не будет введена команда `exit`, `quit` или символ конца файла.

## 9.6. Альтернативы стандартному спулеру

Если вы прочитали все это руководство, к этому моменту вы знаете практически все, что надо знать о системе спулинга **LPD**, входящей в состав ОС FreeBSD. Вы, возможно, уже осознали многие из ее недостатков, что, естественно, приводит к вопросу: "Какие еще системы спулинга существуют (и работают с ОС FreeBSD)?"

### LPRng

Система **LPRng**, имя которой означает "LPR: the Next Generation" (LPR: следующее поколение) - это полностью переписанная система PLP. Патрик Пауэл (Patrick Powell) и Джастин Мейсон (Justin Mason) (основной специалист, занимающийся поддержкой PLP) объединили усилия для создания системы **LPRng**. Основной сайт по системе **LPRng** - <http://www.lprng.org/>.

### CUPS

Система **CUPS** (сокращение от Common UNIX Printing System) предоставляет переносимый механизм печати для операционных систем, основанных на UNIX. Она была разработана компанией Easy Software Products в качестве стандартного механизма печати для всех производителей и пользователей UNIX.

Система **CUPS** использует протокол Internet Printing Protocol (IPP) для управления заданиями и очередями. Протоколы Line Printer Daemon (LPD), Server Message Block (SMB) и AppSocket (известный также как JetDirect) также поддерживаются, но с меньшими возможностями. Система CUPS добавляет поиск сетевых принтеров и опции печати на основе PostScript Printer Description (PPD), для поддержки практической печати в UNIX.

Основной сайт по системе **CUPS** - <http://www.cups.org/>.

## 9.7. Выявление проблем

После выполнения простого тестирования с помощью команды `lpctest(1)` вы можете получить один из следующих результатов вместо корректной распечатки:

Все работает, после определенной задержки; или не выдается распечатанная страница.

Принтер напечатал все, что нужно, но он на некоторое время задумывался и ничего не делал. Фактически, могло потребоваться нажать кнопку PRINT REMAINING или FORM FEED на принтере, чтобы результаты были выданы.

Если это произошло, вероятно, принтер ждал, нет ли в задании еще данных, прежде чем что бы то ни было печатать. Для решения этой проблемы можно посылать в текстовом фильтре на принтер символ FORM FEED (или любую необходимую последовательность символов). Этого обычно достаточно, чтобы принтер немедленно распечатал любой остающийся в его внутреннем буфере текст. Также полезно убедиться, что каждое задание печати заканчивается полной страницей, чтобы следующее задание не начиналось где-то с середины последней страницы предыдущего задания.

Следующий измененный скрипт командного интерпретатора `/usr/local/libexec/if-simple` выдает символ прогона страницы после отправки задания на принтер:

```
#!/bin/sh
#
# if-simple - Простой текстовый входной фильтр для lpd
# Установлен в /usr/local/libexec/if-simple
#
# Просто копирует stdin в stdout. Игнорирует все аргументы фильтра.
# Выдает символ прогона страницы (\f) после печати задания.

/bin/cat && printf "\f" && exit 0
exit 2
```

Принтер печатает "лесенкой".

Вы получаете на бумаге следующее:

```
! "$%&' ( ) * + , - . / 0 1 2 3 4
      "$%&' ( ) * + , - . / 0 1 2 3 4 5
                "$%&' ( ) * + , - . / 0 1 2 3 4 5 6
```

Вы стали очередной жертвой *эффекта лесенки*, вызванного различными интерпретациями того, какие символы должны обозначать новую строку. Операционные системы UNIX-стиля используют один символ: ASCII-код 10, перевод строки (line feed - LF). MS-DOS, OS/2® и другие используют пару символов, ASCII-код 10 и ASCII-код 13 (возврат каретки, carriage return или CR). Многие принтеры используют соглашение MS-DOS для представления новых строк.

При печати из FreeBSD в тексте используется только символ перевода строки. Принтер, встретив символ перевода строки, переходит на следующую строку, но оставляет ту же горизонтальную позицию на строке для следующего печатаемого символа. Вот зачем нужен символ возврата каретки: чтобы перенести следующий печатаемый символ на левый край бумаги.

Вот что ОС FreeBSD хочет от принтера:

Принтер получает CR

Принтер получает LF

Принтер печатает CR

Принтер печатает CR + LF

Вот несколько способов этого добиться:

- Использовать переключатели конфигурации принтера или панель управления, чтобы изменить его интерпретацию этих символов. Поищите как это сделать в руководстве по своему принтеру.

**Замечание:** Если вы загружаете другие операционные системы, кроме FreeBSD, может иметь смысл *переконфигурировать* принтер для использования такой интерпретации символов CR и LF, которая принята в этих операционных системах. Затем можно использовать одно из представленных далее решений.

- Заставить драйвер последовательного порта FreeBSD автоматически преобразовывать LF в CR+LF. Конечно, это подойдет *только* для принтеров, подключенных к последовательным портам. Для включения этой возможности используйте характеристику `ms#` и установите режим `onlcr` для принтера в файле `/etc/printcap`.
- Послать *управляющий код* на принтер, заставляющий его временно обрабатывать символы LF по-другому. Управляющие коды, которые может поддерживать ваш принтер, поищите в руководстве своего принтера. Когда найдете соответствующий управляющий код, измените текстовый фильтр для отправки сначала этого кода, а затем - задания печати.

Вот пример текстового фильтра для принтеров, понимающих управляющие последовательности языка Hewlett-Packard PCL. Этот фильтр заставляет принтер обрабатывать символы LF как LF и CR; затем он посылает задание; наконец, он посылает символ прогона страницы для выдачи последней страницы задания. Он должен работать практически со всеми принтерами Hewlett Packard.

```
#!/bin/sh
#
# hpfif - Простой текстовый входной фильтр для lpd для принтеров на базе HP-PCL
# Установлен в /usr/local/libexec/hpfif
#
# Просто копирует stdin в stdout. Игнорирует все аргументы фильтра.
# Требуется от принтера обрабатывать LF как CR+LF. Выдает страницу по окончании.

printf "\033&k2G" && cat && printf "\033&l0H" && exit 0
exit 2
```

Вот пример файла `/etc/printcap` с хоста `orchid`. К нему через первый параллельный порт подключен один принтер, Hewlett Packard LaserJet 3Si, по имени `teak`. Для него в качестве текстового фильтра используется представленный выше скрипт:

```
#
# /etc/printcap для хоста orchid
#
teak|hp|laserjet|Hewlett Packard LaserJet 3Si:\
    :lp=/dev/lpt0:sh:sd=/var/spool/lpd/teak:mx#0:\
    :if=/usr/local/libexec/hpfif:
```

Строки напечатаны одна поверх другой.

Принтер так и не перешел на следующую строку. Все строки текста были напечатаны одна поверх другой, на одной строке.

Эта проблема "обратна" эффекту лесенки, описанному выше, и встречается намного реже.

Каким-то образом, символы LF, которые ОС FreeBSD использует для завершения строк, обрабатывались как символы CR и вызывали перевод позиции печати на левый край бумаги, но не переход на следующую строку.

Используйте переключатели конфигурации принтера или панель управления для обеспечения следующей интерпретации символов LF и CR:

Принтер получает	Принтер печатает
CR	CR
LF	CR + LF

Принтер теряет символы.

По ходу печати принтер не печатает несколько символов в каждой строке. Проблема со временем может становиться все хуже, так что теряется все больше символов.

Проблема состоит в том, что принтер не справляется с той скоростью, с которой компьютер посылает данные по последовательной линии (эта проблема не должна возникать на принтерах, подключенных к параллельным портам). Есть два способа решить проблему:

- Если принтер поддерживает управление потоком XON/XOFF, заставить FreeBSD использовать его, указав режим `ixon` в характеристике `ms#`.
- Если принтер поддерживает управление несущим потоком (carrier flow control), укажите режим `crtsets` в характеристике `ms#`. Убедитесь, что кабель, соединяющий принтер с компьютером, правильно распаян для управления несущим потоком.

Напечатан мусор.

Принтер напечатал нечто похожее на случайный мусор, а не требуемый текст.

Это, обычно, - еще один симптом неправильных параметров взаимодействия с последовательным принтером. Перепроверьте скорость взаимодействия в характеристике `br` и установку четности в характеристике `ms#`; проверьте, что принтер использует те же установки, которые заданы в файле `/etc/printcap`.

Ничего не произошло.

Если ничего не произошло, проблема, вероятно, связана с FreeBSD, а не с оборудованием. Добавьте характеристику журнального файла (`lf`) в файл `/etc/printcap` для принтера, работу с которым отлаживаете. Например, вот запись для принтера `rattan` с характеристикой `lf`:

```
rattan|line|diablo|lp|Diablo 630 Line Printer:\
:sh:sd=/var/spool/lpd/rattan:\
```

```
:lp=/dev/lpt0:\  
:if=/usr/local/libexec/if-simple:\  
:lf=/var/log/rattan.log
```

Затем попытайтесь напечатать снова. Поищите в журнальном файле (в нашем примере - /var/log/rattan.log) возможные сообщения об ошибках. На основе полученных сообщений попытайтесь решить проблему.

Если вы не зададите характеристику lf, система **LPD** использует по умолчанию /dev/console.

# Глава 10. Двоичная совместимость с Linux

*Реструктурировал и частично обновил Jim Mock. Первоначально предоставил Brian N. Handy, Rich Murphey. Перевод на русский язык: Алексей Докучаев. Перевод на русский язык и редактирование: Валерий Кравчук.*

## 10.1. Краткий обзор

FreeBSD предоставляет двоичную совместимость с несколькими другими UNIX-подобными операционными системами, включая Linux. Возможно, вы спрашиваете себя, зачем FreeBSD нужно уметь работать с приложениями, написанными для Linux? Ответ на этот вопрос достаточно прост: многие компании и разработчики производят программное обеспечение только для Linux, так как эта операционная система очень быстро завоевала огромную популярность в компьютерном мире. Пользователям же FreeBSD приходится обращаться к этим компаниям и разработчикам с просьбами выпустить версии своих программ специально для FreeBSD. Проблема в том, что большинство производителей программного обеспечения не осознают, насколько бы увеличился их рынок сбыта, выпускай они также FreeBSD-версии, и продолжают разрабатывать только под Linux. Что же делать пользователям FreeBSD? В этой ситуации на помощь приходит двоичная совместимость FreeBSD с Linux.

Вкратце, эта совместимость позволяет пользователям FreeBSD работать с 90% приложений для Linux без каких-либо модификаций последних. Среди этих приложений: **StarOffice**, Linux-версия **Netscape**, **Adobe® Acrobat®**, **RealPlayer®**, **VMware™**, **Oracle**, **WordPerfect®**, **Doom**, **Quake**, и многие другие. Есть сведения, что в некоторых ситуациях эти Linux-программы показывали более высокую производительность при работе под FreeBSD, чем под Linux.

Конечно, существуют некоторые особенности Linux, которые не поддерживаются в полной мере под FreeBSD. Например, не будут работать приложения Linux, использующие файловую систему `/proc` (реализация которой отличается от файловой системы `/proc` в ОС FreeBSD) или специфичные вызовы `i386`, такие как переключение в виртуальный режим `8086`.

При чтении этой главы вы узнаете:

- Как включить двоичную совместимость с Linux в вашей системе.
- Как установить дополнительные совместно используемые (shared) библиотеки Linux.
- Как установить приложения Linux в систему FreeBSD.
- Детали реализации совместимости с Linux в ОС FreeBSD.

Перед прочтением этой главы вам потребуется:

- Узнать как устанавливать дополнительное программное обеспечение сторонних разработчиков (Гл. 4).



## 10.2. Установка

Двоичная совместимость с Linux не включена по умолчанию. Простейший способ включения этой функциональности заключается в загрузке KLD-объекта `linux` ("Kernel Loadable object"). Вы можете загрузить этот модуль, просто набрав `linux` в командной строке.

Если вы хотите, чтобы совместимость с Linux была включена постоянно, необходимо добавить в `/etc/rc.conf` следующую строку:

```
linux_enable="YES"
```

Для проверки того, загружен ли модуль, может быть использована команда `kldstat(8)`:

```
% kldstat
Id Refs Address      Size      Name
  1    2 0xc0100000 16bdb8    kernel
  7    1 0xc24db000 d000      linux.ko
```

Если по какой-либо причине вы не хотите или не можете загрузить KLD, вы можете статически включить поддержку Linux в ядро, добавив опцию `options COMPAT_LINUX` в файл конфигурации ядра. Затем соберите и установите новое ядро, следуя описанию в Гл. 8.

### 10.2.1. Установка необходимых библиотек Linux

Установить все требуемые библиотеки можно двумя путями: либо используя порт `linux_base`, либо установив их вручную.

#### 10.2.1.1. Установка с помощью порта `linux_base`

Этот метод является самым простым, и мы рекомендуем воспользоваться именно им. Процесс аналогичен установке любого другого порта из коллекции портов (`/usr/ports/`). Просто выполните следующие команды:

```
# cd /usr/ports/emulators/linux_base
# make install distclean
```

Теперь вы можете работать с приложениями для Linux. Некоторые программы, возможно, будут сообщать о несоответствии подверсий некоторых системных библиотек. Однако обычно это не вызывает каких-либо неудобств.

**Замечание:** Возможно наличие нескольких версий порта `emulators/linux_base`, соответствующих различным версиям разных дистрибутивов Linux. Вы должны установить порт, наиболее близко соответствующий требованиям приложений Linux, которые будут установлены.

#### 10.2.1.2. Установка библиотек вручную

Если у вас не установлена коллекция портов, можно установить требуемые библиотеки вручную. Вам понадобятся совместно используемые библиотеки для Linux, которые нужны программам, и `runtime`-компоновщик. Вам также потребуется создать "теневого корневого каталог", `/compat/linux`,

где будут расположены Linux-библиотеки. Если Linux-программе нужно загрузить какую-либо совместно используемую библиотеку, FreeBSD сперва будет пытаться найти ее в этом дереве. Так, если программа загружает, например, `/lib/libc.so`, FreeBSD попытается открыть `/compat/linux/lib/libc.so`, и если такого файла не существует, будет пытаться открыть `/lib/libc.so`. Разделяемые библиотеки должны находиться в теновом дереве, а не в каталогах, выдаваемых загрузчиком Linux `ld.so`.

Обычно вам придется добавлять совместно используемые библиотеки, от которых зависят Linux-программы, только при нескольких первых установках приложений Linux на вашу систему FreeBSD. По мере работы, у вас в системе накопится достаточный набор совместно используемых библиотек Linux для запуска новых Linux-программ без дополнительных действий.

### 10.2.1.3. Как установить дополнительные совместно используемые библиотеки

Что, если при установленном `linux_base` порте ваше приложение все равно сообщает об отсутствии необходимой библиотеки? Как узнать, какая именно нужна библиотека и где ее взять? В принципе, есть два способа. Вам необходимо иметь привилегии пользователя `root` для их осуществления.

Если у вас есть доступ к машине, на которой установлен Linux, узнайте, какие библиотеки использует Linux-приложение, и просто скопируйте из на свою машину. Рассмотрим следующий пример:

Допустим, вы скачали по FTP Linux-версию **Doom** и установили ее на Linux-машине. Вы можете узнать, какие совместно используемые библиотеки нужны **Doom**, с помощью команды `ldd linuxdoom`:

```
% ldd linuxdoom
libXt.so.3 (DLL Jump 3.1) => /usr/X11/lib/libXt.so.3.1.0
libX11.so.3 (DLL Jump 3.1) => /usr/X11/lib/libX11.so.3.1.0
libc.so.4 (DLL Jump 4.5p126) => /lib/libc.so.4.6.29
```

Вам потребуются все файлы, перечисленные в последнем столбце. Скопируйте их в дерево `/compat/linux` на вашей системе, а также создайте символические ссылки на эти файлы с именами из первого столбца, соответственно. В итоге, у вас в системе FreeBSD должны быть следующие файлы:

```
/compat/linux/usr/X11/lib/libXt.so.3.1.0
/compat/linux/usr/X11/lib/libXt.so.3 -> libXt.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3 -> libX11.so.3.1.0
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

**Замечание:** Учтите, что если у вас уже есть совместно используемая библиотека Linux с соответствующим первому столбцу результатов `ldd` основным номером версии, вам не обязательно копировать файл, указанный в последнем столбце, в вашу систему. Уже существующий файл должен подойти. Рекомендуется, однако, все равно скопировать совместно используемую библиотеку, если ее версия новее. Предыдущую версию библиотеки можно удалить, если создали символическую ссылку на новую. Итак, если у вас в системе есть следующие библиотеки:

```
/compat/linux/lib/libc.so.4.6.27
/compat/linux/lib/libc.so.4 -> libc.so.4.6.27
```

и какое-либо приложение требует библиотеку более поздней версии, судя по результатам команды `ldd`:

```
libc.so.4 (DLL Jump 4.5p126) -> libc.so.4.6.29
```

Если версии немного отличаются в последней цифре, копировать `/lib/libc.so.4.6.29` необязательно, так как программа, скорее всего, будет нормально работать и с немного устаревшей версией. Тем не менее, вы можете заменить `libc.so`:

```
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

**Замечание:** Символические ссылки важны *только* для Linux-программ. Runtime-компоновщик FreeBSD самостоятельно подберет правильные номера версий библиотек, и вам не нужно об этом беспокоиться.

### 10.2.2. Установка двоичных файлов Linux ELF

Для ELF-файлов иногда требуется сделать так называемый "branding" (маркировать его). Если попытаться запустить немаркированный ELF-файл, вы получите следующее сообщение об ошибке:

```
% ./моя-linux-elf-программа
ELF binary type not known
Abort
```

Чтобы помочь ядру FreeBSD отличить ELF-файл FreeBSD от двоичного файла Linux, используется утилита `brandelf(1)`:

```
% brandelf -t Linux моя-linux-elf-программа
```

В настоящее время набор инструментальных средств GNU (GNU toolchain) помещает необходимую маркировочную информацию в двоичные ELF-файлы автоматически, поэтому необходимость в этом действии возникает всё реже.

### 10.2.3. Конфигурирование распознавания имен хостов

Если DNS не работает или вы получаете это сообщение:

```
resolv+: "bind" is an invalid keyword resolv+:
"hosts" is an invalid keyword
```

то вам нужно создать (изменить) файл `/compat/linux/etc/host.conf`, содержащий:

```
order hosts, bind
multi on
```

Таким образом, вы указываете, то сначала производится поиск в файле `/etc/hosts`, а только затем запрашивается DNS. Когда файл `/compat/linux/etc/host.conf` отсутствует, Linux-приложения находят файл `/etc/host.conf` для FreeBSD и сообщают о несовместимом синтаксисе. Если вы не

настраивали сервер имен с помощью файла `/etc/resolv.conf`, уберите `bind` из файла `/compat/linux/etc/host.conf`.

## 10.3. Установка Mathematica®

*Обновил для Mathematica 4.X Murray Stokely. Объединено с работой Bojan Bistrovic.*

Ниже описано, как установить Linux-версию пакета **Mathematica 4.X** на систему FreeBSD.

Linux-версия **Mathematica** отлично работает под FreeBSD, однако выполняемые файлы, поставляемые Wolfram, должны быть маркированы утилитой `brandelf(1)`, чтобы FreeBSD знала о необходимости использования Linux ABI для их выполнения.

Linux-версию пакета **Mathematica** или **Mathematica for Students** можно заказать непосредственно в компании Wolfram по адресу <http://www.wolfram.com/>.

### 10.3.1. Маркировка двоичных файлов Linux

Двоичные файлы Linux находятся в каталоге Unix на компакт-диске с дистрибутивом **Mathematica** от Wolfram. Скопируйте этот каталог (вместе со всеми подкаталогами) на ваш жесткий диск, чтобы промаркировать (утилитой `brandelf(1)`) двоичные файлы Linux перед инсталляцией:

```
# mount /cdrom
# cp -rp /cdrom/Unix/ /localdir/
# brandelf -t Linux /localdir/Files/SystemFiles/Kernel/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/FrontEnd/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/Installation/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/Graphics/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/Converters/Binaries/Linux/*
# brandelf -t Linux /localdir/Files/SystemFiles/LicenseManager/Binaries/Linux/mathlm
# cd /localdir/Installers/Linux/
# ./MathInstaller
```

Вы также можете установить марку ELF Linux по умолчанию для всех немаркированных двоичных файлов с помощью команды:

```
# sysctl kern.fallback_elf_brand=3
```

FreeBSD будет считать, что все немаркированные двоичные ELF-файлы используют Linux ABI, и вы сможете запустить `MathInstaller` прямо с CDROM.

### 10.3.2. Получение пароля к пакету Mathematica

Прежде чем запускать **Mathematica**, вам необходимо получить от компании Wolfram пароль, соответствующий "идентификатору машины".

После того, как вы установили необходимые библиотеки и распаковали файлы **Mathematica**, получите "идентификатор машины", запустив программу `mathinfo` в инсталляционном каталоге.

Идентификатор машины зависит от MAC-адреса вашей первой Ethernet-карты.

```
# cd /localdir/Files/SystemFiles/Installation/Binaries/Linux
# mathinfo
disco.example.com 7115-70839-20412
```

При регистрации по электронной почте, по телефону или по факсу вы сообщаете "идентификатор машины", а в ответ получаете пароль, состоящий из нескольких групп чисел. Введите этот пароль при первом запуске **Mathematica**, как и для любой другой платформы.

### 10.3.3. Использование интерфейса Mathematica по сети

**Mathematica** использует специальные шрифты для отображения некоторых символов, которые отсутствуют в стандартных шрифтах (символы интегралов, сумм, греческий алфавит и другие). Протокол X требует, чтобы эти шрифты были установлены *локально*. Это означает, что вы должны скопировать эти шрифты с компакт-диска или хоста, на котором установлена **Mathematica**, на вашу машину. Обычно эти шрифты находятся в каталоге /cdrom/Unix/Files/SystemFiles/Fonts компакт-диска или в каталоге /usr/local/mathematica/SystemFiles/Fonts на диске. Собственно файлы со шрифтами находятся в подкаталогах Type1 и X. О том, как их использовать, читайте ниже.

Можно просто скопировать их в один из существующих каталогов шрифтов в каталоге /usr/X11R6/lib/X11/fonts. В этом случае придётся отредактировать файл fonts.dir, добавив в него названия шрифтов и изменив число шрифтов в первой строке. Можно также запустить программу mkfontdir(1), находясь в том каталоге, куда вы скопировали шрифты.

Есть альтернативный способ: скопировать каталоги в /usr/X11R6/lib/X11/fonts:

```
# cd /usr/X11R6/lib/X11/fonts
# mkdir X
# mkdir MathType1
# cd /cdrom/Unix/Files/SystemFiles/Fonts
# cp X/* /usr/X11R6/lib/X11/fonts/X
# cp Type1/* /usr/X11R6/lib/X11/fonts/MathType1
# cd /usr/X11R6/lib/X11/fonts/X
# mkfontdir
# cd ../MathType1
# mkfontdir
```

Теперь добавьте каталоги с новыми шрифтами в путь к шрифтам:

```
# xset fp+ /usr/X11R6/lib/X11/fonts/X
# xset fp+ /usr/X11R6/lib/X11/fonts/MathType1
# xset fp rehash
```

Если вы используете сервер **XFree86**, то можно просто прописать эти каталоги в файле XF86Config.

Если на вашем компьютере *нет* каталога /usr/X11R6/lib/X11/fonts/Type1, замените MathType1 на Type1 в предыдущем примере.

## 10.4. Установка Maple™

*Предоставил Aaron Kaplan. Благодарности: Robert Getschmann.*

**Maple™** - коммерческая математическая программа, аналогичная **Mathematica**. Это программное обеспечение надо купить у <http://www.maplesoft.com/>, а потом зарегистрироваться там для получения файла лицензии. Для установки это программного обеспечения в ОС FreeBSD используется следующая последовательность простых шагов.

1. Выполните скрипт командного интерпретатора `INSTALL` из дистрибутива. Выберите опцию **RedHat**, когда будет предложено программой установки. Обычно установка выполняется в каталог `/usr/local/maple`.
2. Если вы этого ещё не сделали, купите лицензию на **Maple** в компании Maple Waterloo Software (<http://register.maplesoft.com/>) и скопируйте ее в файл `/usr/local/maple/license/license.dat`.
3. Установите диспетчер лицензий **FLEXlm**, выполнив скрипт установки `INSTALL_LIC`, входящий в состав **Maple**. Укажите основное имя хоста вашей машины для сервера лицензий.
4. Исправьте файл `/usr/local/maple/bin/maple.system.type` с помощью следующего патча:

```
----- snip -----
*** maple.system.type.orig      Sun Jul  8 16:35:33 2001
--- maple.system.type      Sun Jul  8 16:35:51 2001
*****
*** 72,77 ***
--- 72,78 ----
        # the IBM RS/6000 AIX case
        MAPLE_BIN="bin.IBM_RISC_UNIX"
        ; ;
+   "FreeBSD" | \
    "Linux")
        # the Linux/x86 case
        # We have two Linux implementations, one for Red Hat and
----- snip end of patch -----
```

Учтите, что после `"FreeBSD" | \` не должно быть никаких пробелов.

Этот патч заставляет **Maple** распознавать **FreeBSD** как тип Linux-системы. Скрипт командного интерпретатора `bin/maple` вызывает скрипт `bin/maple.system.type`, который, в свою очередь, вызывает `uname -a` для получения имени операционной системы. В зависимости от имени ОС он определяет, какие двоичные модули использовать.

5. Запустите сервер лицензий.

Следующий скрипт, установленный в файл `/usr/local/etc/rc.d/lmgrd.sh`, обеспечивает удобный способ запуска `lmgrd`:

```
----- snip -----

#! /bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin
PATH=${PATH}:/usr/local/maple/bin:/usr/local/maple/FLEXlm/UNIX/LINUX
export PATH

LICENSE_FILE=/usr/local/maple/license/license.dat
```

```
LOG=/var/log/lmgrd.log

case "$1" in
start)
lmgrd -c ${LICENSE_FILE} 2>> ${LOG} 1>&2
echo -n " lmgrd"
;;
stop)
lmgrd -c ${LICENSE_FILE} -x lmdown 2>> ${LOG} 1>&2
;;
*)
echo "Usage: `basename $0` {start|stop}" 1>&2
exit 64
;;
esac

exit 0

----- snip -----
```

#### 6. Выполните тестовый запуск **Maple**:

```
% cd /usr/local/maple/bin
% ./xmaple
```

Программа должна запуститься и работать. Не забудьте написать в Maplesoft и сообщить, что хотели бы видеть версию специально для FreeBSD!

### 10.4.1. Типичные проблемы

- Работать с диспетчером лицензий **FLEXlm** может быть непросто. Дополнительную документацию по нему можно найти на сайте <http://www.globetrotter.com/>.
- lmgrd очень требователен к файлу лицензии и выдает дамп памяти при выявлении любых проблем. Правильный файл лицензии должен иметь следующий вид:

```
# =====
# License File for UNIX Installations ("Pointer File")
# =====
SERVER chillig ANY
#USE_SERVER
VENDOR maplelmg

FEATURE Maple maplelmg 2000.0831 permanent 1 XXXXXXXXXXXX \
    PLATFORMS=i86_r ISSUER="Waterloo Maple Inc." \
    ISSUED=11-may-2000 NOTICE=" Technische Universitat Wien" \
    SN=XXXXXXXXXX
```

**Замечание:** Серийный номер и ключ забиты символами 'X'. chillig — имя хоста.

Редактирование файла лицензий возможно, если только не трогать строку 'FEATURE' (которая защищена лицензионным ключом).

## 10.5. Установка MATLAB®

Этот документ описывает процесс установки Linux-версии **MATLAB® версии 6.5** на систему FreeBSD. Эта программа работает вполне нормально, за исключением **Java Virtual Machine™** (см. Разд. 10.5.3).

Linux-версию **MATLAB** можно приобрести непосредственно в компании The MathWorks на сайте <http://www.mathworks.com>. Не забудьте получить файл лицензии или инструкции по его созданию. По ходу дела дайте знать производителю, что хотели бы увидеть версию их ПО специально для FreeBSD.

### 10.5.1. Процесс установки MATLAB

Для установки **MATLAB** выполните следующие шаги:

1. Вставьте инсталляционный CD и смонтируйте его. Станьте пользователем `root`, как рекомендует скрипт установки. Для запуска скрипта установки наберите:

```
# /compat/linux/bin/sh /cdrom/install
```

**Подсказка:** Запускается графический инсталлятор. Если вы получаете сообщения о невозможности открыть дисплей, наберите `setenv HOME ~USER`, где `USER` — пользователь, от имени которого выполнена команда `su(1)`.

2. При запросе корневого каталога **MATLAB**, наберите: `/compat/linux/usr/local/matlab`.

**Подсказка:** Чтобы упростить набор остальных команд в ходе установки, выполните в командном интерпретаторе следующую команду: `set MATLAB=/compat/linux/usr/local/matlab`

3. Отредактируйте файл лицензии в соответствии с инструкциями, в полученной лицензии **MATLAB**.

**Подсказка:** Этот файл можно подготовить заранее с помощью любого текстового редактора и скопировать его в `$MATLAB/license.dat` до того, как инсталлятор попросит его отредактировать.

4. Завершите процесс установки.

В этот момент ваша установка **MATLAB** завершена. Следующие шаги позволяют "связать" эту программу с вашей системой FreeBSD.

### 10.5.2. Запуск диспетчера лицензий

1. Создайте символические ссылки для скриптов диспетчера лицензий:



```
# ln -s $MATLAB/etc/lmboot /usr/local/etc/lmboot_TMW
# ln -s $MATLAB/etc/lmdown /usr/local/etc/lmdown_TMW
```

2. Создайте файл запуска /usr/local/etc/rc.d/flexlm.sh. Представленный ниже пример – измененная версия входящего в дистрибутив файла \$MATLAB/etc/rc.lm.glnx86. Изменены местонахождения файлов и диспетчер лицензий запускается под эмулятором Linux.

```
#!/bin/sh
case "$1" in
  start)
    if [ -f /usr/local/etc/lmboot_TMW ]; then
      /compat/linux/bin/sh /usr/local/etc/lmboot_TMW -u username && echo 'MATLAB_lmgrd'
    fi
    ;;
  stop)
    if [ -f /usr/local/etc/lmdown_TMW ]; then
      /compat/linux/bin/sh /usr/local/etc/lmdown_TMW > /dev/null 2>&1
    fi
    ;;
  *)
    echo "Usage: $0 {start|stop}"
    exit 1
    ;;
esac

exit 0
```

**Важно:** Этот файл надо сделать выполняемым:

```
# chmod +x /usr/local/etc/rc.d/flexlm.sh
```

Вы также должны заменить *username* именем пользователя в вашей системе (но не root).

3. Запустите диспетчер лицензий с помощью команды:

```
# /usr/local/etc/rc.d/flexlm.sh start
```

### 10.5.3. Связь с Java Runtime Environment

Измените ссылку **Java Runtime Environment (JRE)** так, чтобы он ссылалась на версию, работающую в FreeBSD:

```
# cd $MATLAB/sys/java/jre/glnx86/
# unlink jre; ln -s ./jre1.1.8 ./jre
```

### 10.5.4. Создание скрипта запуска MATLAB

1. Поместите следующий скрипт запуска в файл /usr/local/bin/matlab:

```
#!/bin/sh
```

```
/compat/linux/bin/sh /compat/linux/usr/local/matlab/bin/matlab "$@"
```

2. Затем выполните команду `chmod +x /usr/local/bin/matlab`.

**Подсказка:** В зависимости от версии `emulators/linux_base`, при выполнении этого скрипта могут быть выданы сообщения об ошибках. Чтобы избежать этого, отредактируйте файл `/compat/linux/usr/local/matlab/bin/matlab` и измените строку вида:

```
if [ `expr "$lsrmdir" : '.*->.*'` -ne 0 ]; then
```

(в версии 13.0.1 это строка 410) на следующую строку:

```
if test -L $newbase; then
```

### 10.5.5. Создание скрипта остановки MATLAB

Следующие действия необходимы для решения проблемы с некорректным завершением работы MATLAB.

1. Создайте файл `$MATLAB/toolbox/local/finish.m` и поместите в него одну строку:

```
! $MATLAB/bin/finish.sh
```

**Замечание:** `$MATLAB` - литерал.

**Подсказка:** В том же каталоге находятся файлы `finishsav.m` и `finishdlg.m`, которые позволяют сохранять рабочее пространство перед выходом. Если вы используете любой из них, вставьте представленную выше строку сразу после команды `save`.

2. Создайте файл `$MATLAB/bin/finish.sh`, который будет содержать следующий скрипт:

```
#!/usr/compat/linux/bin/sh
(sleep 5; killall -1 matlab_helper) &
exit 0
```

3. Сделайте этот файл выполняемым:

```
# chmod +x $MATLAB/bin/finish.sh
```

### 10.5.6. Использование MATLAB

В этот момент все готово для выполнения команды `matlab` и начала использования этой программы.

## 10.6. Установка Oracle®

*Предоставил Marcel Moolenaar.*

### 10.6.1. Введение

Ниже описан процесс установки **Oracle 8.0.5** и **Oracle 8.0.5.1 Enterprise Edition** для Linux на систему FreeBSD.

### 10.6.2. Установка Linux-среды

Удостоверьтесь, что порты `emulators/linux_base` и `devel/linux_devtools` установлены на вашей системе. Если у вас возникнут трудности с этими портами, воспользуйтесь пакетами или более ранними их версиями из коллекции портов.

Если вы хотите использовать интеллектуальный агент (intelligent agent), придется также установить пакет TCL от Red Hat: `tcl-8.0.3-20.i386.rpm`. Универсальная команда для установки пакетов с помощью официального порта **RPM** (`archivers/rpm`):

```
# rpm -i --ignoreos --root /compat/linux --dbpath /var/lib/rpm пакет
```

Установка этого *пакета* должна пройти без каких-либо ошибок.

### 10.6.3. Создание среды Oracle

Прежде чем вы сможете установить **Oracle**, необходимо настроить соответствующую среду. В этом документе описано, что *специально* нужно сделать, чтобы запустить **Oracle** для Linux под FreeBSD – это не пересказ официального руководства по установке **Oracle**.

#### 10.6.3.1. Настройка ядра

Как описано в руководстве по установке **Oracle**, необходимо установить максимальный размер совместно используемой (shared) памяти. Не используйте SHMMAX под FreeBSD. SHMMAX просто вычисляется, исходя из SHMMAXPGS и PGSIZE. Следовательно, нужно задавать SHMMAXPGS. За информацией о прочих опциях обратитесь к официальному руководству. Пример настроек:

```
options SHMMAXPGS=10000
options SHMMNI=100
options SHMSEG=10
options SEMMNS=200
options SEMMNI=70
options SEMMSL=61
```

Установите эти опции в зависимости от того, как и для чего вы будете использовать **Oracle**.

Не забудьте добавить следующие строки в файл конфигурации ядра:

```
options SYSVSHM # совместно используемая память SysV
options SYSVSEM # семафоры SysV
options SYSVMSG # межпроцессное взаимодействие SysV
```

### 10.6.3.2. Учетная запись Oracle

Создайте специальную учетную запись `oracle`, как и любую другую учетную запись. Единственное отличие в том, что для `oracle` необходимо указать командный интерпретатор Linux. Добавьте `/compat/linux/bin/bash` в `/etc/shells` и установите для `oracle` командный интерпретатор `/compat/linux/bin/bash`.

### 10.6.3.3. Переменные среды

Кроме стандартных переменных среды **Oracle**, таких как `ORACLE_HOME` и `ORACLE_SID`, вам нужно будет установить следующие переменные среды:

Переменная	Значение
<code>LD_LIBRARY_PATH</code>	<code>\$ORACLE_HOME/lib</code>
<code>CLASSPATH</code>	<code>\$ORACLE_HOME/jdbc/lib/classes111.zip</code>
<code>PATH</code>	<code>/compat/linux/bin /compat/linux/sbin</code> <code>/compat/linux/usr/bin /compat/linux/usr/sbin /bin /sbin</code> <code>/usr/bin /usr/sbin /usr/local/bin \$ORACLE_HOME/bin</code>

Желательно устанавливать все переменные среды в файле `.profile`. Вот реальный пример:

```
ORACLE_BASE=/oracle; export ORACLE_BASE
ORACLE_HOME=/oracle; export ORACLE_HOME
LD_LIBRARY_PATH=$ORACLE_HOME/lib
export LD_LIBRARY_PATH
ORACLE_SID=ORCL; export ORACLE_SID
ORACLE_TERM=386x; export ORACLE_TERM
CLASSPATH=$ORACLE_HOME/jdbc/lib/classes111.zip
export CLASSPATH
PATH=/compat/linux/bin:/compat/linux/sbin:/compat/linux/usr/bin
PATH=$PATH:/compat/linux/usr/sbin:/bin:/sbin:/usr/bin:/usr/sbin
PATH=$PATH:/usr/local/bin:$ORACLE_HOME/bin
export PATH
```

### 10.6.4. Установка Oracle

Из-за небольшой несовместимости с Linux-эмулятором, вам нужно будет создать подкаталог `.oracle` в каталоге `/var/tmp` прежде, чем можно будет начать установку. Либо сделайте ее владельцем пользователя `oracle`, либо поставьте права на запись для всех. Если вы все сделали правильно, то установка **Oracle** должна пройти без проблем. Если какие-либо трудности все же возникли, проверьте еще раз все конфигурационные файлы и/или целостность дистрибутива **Oracle**. После окончания установки **Oracle** примените патчи, описанные в следующих двух подразделах.

Одна из часто возникающих проблем — неправильно установленный адаптер TCP-протокола. В результате, невозможно запустить процессы прослушивания TCP. Вот решение проблемы:

```
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk ntcontab.o
```

```
# cd $ORACLE_HOME/lib
# ar r libnetwork.a ntcontab.o
# cd $ORACLE_HOME/network/lib
# make -f ins_network.mk install
```

Не забудьте повторно запустить `root.sh`!

#### 10.6.4.1. Изменение `root.sh`

При установке **Oracle** необходимо выполнить некоторые действия от имени пользователя `root`. Они записаны в скрипте командного интерпретатора `root.sh`, который находится в каталоге `orainst`. Перед запуском, примените к нему следующий патч (исправляет местонахождение утилиты `chown`), либо запускайте его в командном интерпретаторе Linux.

```
*** orainst/root.sh.orig Tue Oct 6 21:57:33 1998
--- orainst/root.sh Mon Dec 28 15:58:53 1998
*****
*** 31,37 ****
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
! CHOWN=/bin/chown
#
# Define variables to be used in this script
--- 31,37 ----
# This is the default value for CHOWN
# It will redefined later in this script for those ports
# which have it conditionally defined in ss_install.h
! CHOWN=/usr/sbin/chown
#
# Define variables to be used in this script
```

Если вы устанавливаете **Oracle** не с компакт-диска, можно изменить исходный файл `root.sh`. Он называется `rthd.sh` и находится в каталоге `orainst`.

#### 10.6.4.2. Изменение `genclntsh`

Скрипт `genclntsh` используется для того, чтобы создать единую совместно используемую клиентскую библиотеку, которая используется для создания демонстраций. Примените следующий патч, чтобы закомментировать определение переменной `PATH`:

```
*** bin/genclntsh.orig Wed Sep 30 07:37:19 1998
--- bin/genclntsh Tue Dec 22 15:36:49 1998
*****
*** 32,38 ****
#
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
--- 32,38 ----
```

```
#
# Explicit path to ensure that we're using the correct commands
#PATH=/usr/bin:/usr/ccs/bin export PATH
! #PATH=/usr/local/bin:/bin:/usr/bin:/usr/X11R6/bin export PATH
#
# each product MUST provide a $PRODUCT/admin/shrept.lst
```

### 10.6.5. Запуск Oracle

Теперь, если вы все сделали правильно, то можете использовать **Oracle** так же, как и в системе Linux.

## 10.7. Установка SAP® R/3®

*Предоставил Holger Kipp. Исходную версию преобразовал в SGML Valentino Vaschetto.*

Инсталляции систем **SAP** на базе FreeBSD не будут поддерживаться службой поддержки SAP — они предоставляют поддержку только на сертифицированных платформах.

### 10.7.1. Предисловие

Этот документ описывает возможный способ установки системы **SAP R/3** с **СУБД Oracle Database** на машине с ОС FreeBSD, включая установку FreeBSD и **Oracle**. Будут описаны две разные конфигурации:

- **SAP R/3 4.6B (IDES)** с **Oracle 8.0.5** на FreeBSD 4.3-STABLE
- **SAP R/3 4.6C** с **Oracle 8.1.7** на FreeBSD 4.5-STABLE

Хотя в этом документе мы пытаемся подробно описать все важные шаги, он не заменяет руководства по установке **Oracle** и **SAP R/3**.

По специфическим вопросам **SAP** и **Oracle** обратитесь к документации, поставляемой в составе **SAP R/3 Linux edition**, а также к другим источникам информации об **Oracle** и **SAP OSS**.

### 10.7.2. Программное обеспечение

Для установки **SAP** были использованы следующие диски CD-ROM:

#### 10.7.2.1. SAP R/3 4.6B, Oracle 8.0.5

Имя	Номер	Описание
KERNEL	51009113	SAP Kernel Oracle / Installation / AIX, Linux, Solaris
RDBMS	51007558	Oracle / RDBMS 8.0.5.X / Linux
EXPORT1	51010208	IDES / DB-Export / Диск 1 из 6
EXPORT2	51010209	IDES / DB-Export / Диск 2 из 6

Имя	Номер	Описание
EXPORT3	51010210	IDES / DB-Export / Диск 3 из 6
EXPORT4	51010211	IDES / DB-Export / Диск 4 из 6
EXPORT5	51010212	IDES / DB-Export / Диск 5 из 6
EXPORT6	51010213	IDES / DB-Export / Диск 6 из 6

Кроме того, мы использовали CD **Oracle 8 Server** (Опытная версия 8.0.5 для Linux, ядро версии 2.0.33), который не обязательно понадобится, и FreeBSD 4.3-STABLE (она вышла всего через несколько дней после 4.3 RELEASE).

#### 10.7.2.2. SAP R/3 4.6C SR2, Oracle 8.1.7

Имя	Номер	Описание
KERNEL	51014004	SAP Kernel Oracle / SAP Kernel Version 4.6D / DEC, Linux
RDBMS	51012930	Oracle 8.1.7/ RDBMS / Linux
EXPORT1	51013953	Release 4.6C SR2 / Export / Диск 1 из 4
EXPORT1	51013953	Release 4.6C SR2 / Export / Диск 2 из 4
EXPORT1	51013953	Release 4.6C SR2 / Export / Диск 3 bp 4
EXPORT1	51013953	Release 4.6C SR2 / Export / Диск 4 из 4
LANG1	51013954	Release 4.6C SR2 / Language / DE, EN, FR / Диск 1 из 3

В зависимости от языков, которые необходимо установить, могут потребоваться дополнительные CD. Здесь мы использовали только немецкий и английский языки, поэтому потребовался только первый языковой CD. Обратите внимание, что номера всех EXPORT CD идентичны. Все три языковых CD тоже имеют один номер (это отличается от нумерации CD в версии 4.6B IDES). На момент написания этого раздела (20.03.2002) установленное ПО работало на FreeBSD 4.5-STABLE.

### 10.7.3. Примечания по SAP

Прочитайте следующие документы перед установкой **SAP R/3** — они пригодятся в ходе установки:

#### 10.7.3.1. SAP R/3 4.6B, Oracle 8.0.5

Номер	Название
0171356	SAP Software on Linux: Essential Comments
0201147	INST: 4.6C R/3 Inst. on UNIX - Oracle
0373203	Update / Migration Oracle 8.0.5 --> 8.0.6/8.1.6 LINUX

Номер	Название
0072984	Release of Digital UNIX 4.0B for Oracle
0130581	R3SETUP step DIPGNTAB terminates
0144978	Your system has not been installed correctly
0162266	Questions and tips for R3SETUP on Windows NT / W2K

### 10.7.3.2. SAP R/3 4.6C, Oracle 8.1.7

Номер	Название
0015023	Initializing table TCPDB (RSXP0004) (EBCDIC)
0045619	R/3 with several languages or typefaces
0171356	SAP Software on Linux: Essential Comments
0195603	RedHat 6.1 Enterprise version: Known problems
0212876	The new archiving tool SAPCAR
0300900	Linux: Released DELL Hardware
0377187	RedHat 6.2: important remarks
0387074	INST: R/3 4.6C SR2 Installation on UNIX
0387077	INST: R/3 4.6C SR2 Inst. on UNIX - Oracle
0387078	SAP Software on UNIX: OS Dependencies 4.6C SR2

### 10.7.4. Требования к аппаратному обеспечению

Следующего оборудования достаточно для установки **SAP R/3 System**. Для производственного использования необходима более точная оценка параметров:

Компонент	4.6B	4.6C
Процессор	2 x 800MHz Pentium III	2 x 800MHz Pentium III
Память	1GB ECC	2GB ECC
Объем дисков	50-60GB (IDES)	50-60GB (IDES)

Для производственного использования рекомендуются процессоры Xeon с большим кешем, высокоскоростной доступ к дискам (SCSI, аппаратный RAID-контроллер), USV и ECC-RAM. Большой объем дискового пространства связан с заранее сконфигурированной системой IDEs, которая создает 27 Гбайт файлов базы данных в ходе установки. Этого пространства также достаточно для исходных производственных систем и прикладных данных.

#### 10.7.4.1. SAP R/3 4.6B, Oracle 8.0.5

Было использовано следующее стандартное оборудование: двухпроцессорная материнская плата с двумя процессорами Pentium III 800 MHz, SCSI-контроллером Adaptec® 29160 Ultra160 (для работы с 40/80 Гбайт стримером DLT и приводом CDRom), Mylex® AcceleRAID™ (2 канала, firmware 6.00-1-00 с 32 Мбайт RAM). К RAID-контроллеру Mylex подключены два диска по 17 Гбайт (зеркалированы) и



четыре диска по 36 Гбайт (RAID уровня 5).

#### 10.7.4.2. SAP R/3 4.6C, Oracle 8.1.7

Для этой установки был использован Dell™ PowerEdge™ 2500: двухпроцессорная плата с двумя процессорами Pentium III 1000 MHz (256 Кбайт кеш), 2 Гбайта PC133 ECC SDRAM, PERC/3 DC PCI RAID-контроллер со 128 Мбайтами и приводом EIDE DVD-ROM. К RAID-контроллеру подключены два диска по 18 Гбайт (зеркалированы) и четыре диска по 36 Гбайт (RAID уровня 5).

### 10.7.5. Установка FreeBSD

Сначала надо установить FreeBSD. Есть несколько способов сделать это (FreeBSD 4.3 была установлена по FTP, FreeBSD 4.5 - непосредственно с RELEASE CD); подробнее см. Разд. 2.13.

#### 10.7.5.1. Компоновка дисков

Для простоты диски при установке **SAP R/3 46B** и **SAP R/3 46C SR2** использовались одинаково. Изменились только имена устройств, поскольку установка выполнялась на разное оборудование (/dev/da и /dev/amr, соответственно, так что при использовании AMI MegaRAID® будут задействованы устройства /dev/amr0s1a вместо /dev/da0s1a):

Файловая система	Размер (в блока по 1 Кбайту)	Размер (Гбайт)	Смонтирована в
/dev/da0s1a	1.016.303	1	/
/dev/da0s1b		6	swap
/dev/da0s1e	2.032.623	2	/var
/dev/da0s1f	8.205.339	8	/usr
/dev/da1s1e	45.734.361	45	/compat/linux/oracle
/dev/da1s1f	2.032.623	2	/compat/linux/sapmnt
/dev/da1s1g	2.032.623	2	/compat/linux/usr/sap

Конфигурируем и инициализируем заранее два логических диска с помощью ПО RAID Mylex или PERC/3. Программы конфигурирования можно запустить в ходе загрузки BIOS.

Обратите внимание, что использованная компоновка дисков немного отличается от рекомендованной SAP, поскольку SAP рекомендует монтировать подкаталоги **Oracle** (и некоторые другие) отдельно — мы решили просто создать подкаталоги.

#### 10.7.5.2. make world и новое ядро

Загрузите последние исходные коды ветки -STABLE. Пересоздайте систему и ваше специализированное ядро после необходимых изменений в файле конфигурации ядра. В него надо включить параметры ядра, требуемые для **SAP R/3** и **Oracle**.

## 10.7.6. Установка среды Linux

### 10.7.6.1. Установка базовой системы Linux

Сначала необходимо установить порт `linux_base` (от имени пользователя `root`):

```
# cd /usr/ports/emulators/linux_base
# make install distclean
```

### 10.7.6.2. Установка среды разработки Linux

Среда разработки Linux необходима, если вы хотите установить **Oracle** в ОС FreeBSD, как описано в Разд. 10.6:

```
# cd /usr/ports/devel/linux_devtools
# make install distclean
```

Среда разработки Linux была установлена только для **SAP R/3 46B IDES**. Она не нужна, если сервер **Oracle** не перекомпоновывается в системе FreeBSD. Именно так и происходит, если вы используете tar-архив (tarball) **Oracle** с Linux-системы.

### 10.7.6.3. Установка необходимых пакетов RPM

Для запуска программы `R3SETUP` необходима поддержка модулей PAM. В ходе первой установки **SAP** на ОС FreeBSD 4.3-STABLE мы попытались установить PAM со всеми необходимыми пакетами и, в конце концов, принудительно установили пакет PAM, что и сработало. Для **SAP R/3 4.6C SR2** мы сразу принудительно установили PAM RPM, что тоже сработало, так что похоже, что пакеты, от которых декларирована зависимость, не нужны:

```
# rpm -i --ignoreos --nodeps --root /compat/linux --dbpath /var/lib/rpm \
pam-0.68-7.i386.rpm
```

Чтобы **Oracle 8.0.5** мог запустить интеллектуального агента, нам также пришлось установить пакет RedHat Tcl `tcl-8.0.5-30.i386.rpm` (иначе перекомпоновка в ходе установки **Oracle** не выполнится). Есть и другие проблемы с перекомпоновкой **Oracle**, но это проблема **Oracle** на Linux, не связанная с особенностями FreeBSD.

### 10.7.6.4. Дополнительные советы

Может также иметь смысл добавить `linprocfs` в `/etc/fstab`; подробнее об этом см. на странице справочника `the linprocfs(5)`. Еще можно установить параметр `kern.fallback_elf_brand=3` в файле `/etc/sysctl.conf`.

## 10.7.7. Создание среды SAP R/3

### 10.7.7.1. Создание необходимых файловых систем и точек монтирования

Для простой установки достаточно создать следующие файловые системы:

точка монтирования	размер в Гбайтах
/compat/linux/oracle	45 GB
/compat/linux/sapmnt	2 GB
/compat/linux/usr/sap	2 GB

Также необходимо создать несколько ссылок. В противном случае, инсталлятор **SAP** будет выдавать сообщения об ошибках, поскольку он проверяет созданные ссылки:

```
# ln -s /compat/linux/oracle /oracle
# ln -s /compat/linux/sapmnt /sapmnt
# ln -s /compat/linux/usr/sap /usr/sap
```

Возможные сообщения об ошибках в ходе установки (в данном случае, для установки System *PRD* и **SAP R/3 4.6C SR2**):

```
INFO 2002-03-19 16:45:36 R3LINKS_IND_IND SyLinkCreate:200
    Checking existence of symbolic link /usr/sap/PRD/SYS/exe/dbg to
    /sapmnt/PRD/exe. Creating if it does not exist...

WARNING 2002-03-19 16:45:36 R3LINKS_IND_IND SyLinkCreate:400
    Link /usr/sap/PRD/SYS/exe/dbg exists but it points to file
    /compat/linux/sapmnt/PRD/exe instead of /sapmnt/PRD/exe. The
    program cannot go on as long as this link exists at this
    location. Move the link to another location.

ERROR 2002-03-19 16:45:36 R3LINKS_IND_IND Ins_SetupLinks:0
    can not setup link '/usr/sap/PRD/SYS/exe/dbg' with content
    '/sapmnt/PRD/exe'
```

### 10.7.7.2. Создание пользователей и каталогов

Для системы **SAP R/3** необходимы два пользователя и три группы. Имена пользователей зависят от идентификатора системы **SAP** (SID), который состоит из трех букв. Некоторые из этих идентификаторов зарезервированы **SAP** (например, SAP и NIX. Полный список см. в документации **SAP**). Для установки **IDES** мы использовали IDS, а для установки 4.6C SR2 - PRD, поскольку эта система предназначалась для промышленного использования. Поэтому нам понадобились следующие группы (идентификаторы групп могут отличаться, мы просто указали наши значения, использованные при установке):

идентификатор группы	имя группы	описание
100	dba	Администратор базы данных
101	sapsys	Система SAP

идентификатор группы	имя группы	описание
102	oper	Оператор базы данных

Для стандартной установки **Oracle** используется только группа dba. В качестве группы oper используется та же группа dba (подробнее об этом см. в документации **Oracle** и **SAP**).

Нам также нужны следующие пользователи:

идентификатор пользователя	имя пользователя	общий вид имени	группа	дополнительное описание группы
1000	idsadm/prdadm	<i>sidadm</i>	sapsys	oper Администратор SAP
1002	oraid/oraprd	<i>orasid</i>	dba	oper Администратор Oracle

Добавление пользователей с помощью `adduser(8)` требует следующих параметров для "Администратора SAP" (обратите внимание на командный интерпретатор и начальный каталог):

```
Name: sidadm
Password: *****
Fullname: SAP Administrator SID
Uid: 1000
Gid: 101 (sapsys)
Class:
Groups: sapsys dba
HOME: /home/sidadm
Shell: bash (/compat/linux/bin/bash)
```

а для "Администратора Oracle":

```
Name: orasid
Password: *****
Fullname: Oracle Administrator SID
Uid: 1002
Gid: 100 (dba)
Class:
Groups: dba
HOME: /oracle/sid
Shell: bash (/compat/linux/bin/bash)
```

Для него также надо указать группу oper, если вы используете обе группы, dba и oper.

### 10.7.7.3. Создание каталогов

Эти каталоги обычно создаются в отдельных файловых системах. Впрочем, все зависит от ваших требований. Мы решили создавать их как обычные каталоги, поскольку в любом случае они находятся на одном массиве RAID 5:

Сначала мы установим владельцев и права для ряда каталогов (от имени пользователя root):

```
# chmod 775 /oracle
# chmod 777 /sapmnt
```

```
# chown root:dba /oracle
# chown sidadm:sapsys /compat/linux/usr/sap
# chmod 775 /compat/linux/usr/sap
```

Затем, мы создадим каталоги от имени пользователя *orasid*. Все они будут размещены в каталоге */oracle/SID*:

```
# su - orasid
# cd /oracle/SID
# mkdir mirrlogA mirrlogB origlogA origlogB
# mkdir sapdata1 sapdata2 sapdata3 sapdata4 sapdata5 sapdata6
# mkdir saparch sapreorg
# exit
```

Для установки **Oracle 8.1.7** потребуется несколько дополнительных каталогов:

```
# su - orasid
# cd /oracle
# mkdir 805_32
# mkdir client stage
# mkdir client/80x_32
# mkdir stage/817_32
# cd /oracle/SID
# mkdir 817_32
```

**Замечание:** Каталог *client/80x\_32* должен иметь точно такое имя. Не заменяйте *x* числом или другим символом.

На третьем шаге мы создаем каталоги от имени пользователя *sidadm*:

```
# su - sidadm
# cd /usr/sap
# mkdir SID
# mkdir trans
# exit
```

#### 10.7.7.4. Записи в файле */etc/services*

Для системы **SAP R/3** необходим ряд записей в файле */etc/services*, которые не будут правильно созданы при установке под FreeBSD. Пожалуйста, добавьте следующие записи (потребуется, по крайней мере, записи, соответствующие номеру экземпляра — в данном случае, 00. Не повредит добавить все записи, от 00 до 99, для *dp*, *gw*, *sr* и *ms*). Если вы собираетесь использовать **SAProuter** или вам необходим доступ к **SAP OSS**, потребуется также запись для 99, поскольку порт 3299 обычно используется для процесса **SAProuter** в целевой системе:

```
sapdp00  3200/tcp # SAP Dispatcher.      3200 + Instance-Number
sapgw00  3300/tcp # SAP Gateway.        3300 + Instance-Number
sapsp00  3400/tcp #                      3400 + Instance-Number
sapms00  3500/tcp #                      3500 + Instance-Number
sapmsSID 3600/tcp # SAP Message Server. 3600 + Instance-Number
```

```
sapgw00s 4800/tcp # SAP Secure Gateway 4800 + Instance-Number
```

#### 10.7.7.5. Необходимые локали

Системе **SAP** нужны, по крайней мере, две локали, не входящие в стандартную установку RedHat. SAP предлагает необходимые пакеты RPM для загрузки со своего FTP-сервера (который доступен только для клиентов с доступом к OSS). См. список необходимых пакетов RPM в заметке 0171356.

Можно также просто создать соответствующие ссылки (например, с *de\_DE* и *en\_US*), но мы не рекомендуем это для производственной системы (хотя это и сработало для системы IDES безо всяких проблем). Необходимы следующие локали:

```
de_DE.ISO-8859-1
en_US.ISO-8859-1
```

Создайте ссылки следующим образом:

```
# cd /compat/linux/usr/share/locale
# ln -s de_DE de_DE.ISO-8859-1
# ln -s en_US en_US.ISO-8859-1
```

Если их не будет, в ходе установки возникнет ряд проблем. Если их просто проигнорировать (установив STATUS для соответствующих шагов равным OK в файле CENTRDB.R3S), нельзя будет зарегистрироваться в системе **SAP** без дополнительных усилий.

#### 10.7.7.6. Настройка ядра

Системам **SAP R/3** надо много ресурсов. Поэтому мы добавили следующие параметры в файл конфигурации ядра:

```
# Для пожирателей памяти (SAP и Oracle):
options MAXDSIZ="(1024*1024*1024)"
options DFLDSIZ="(1024*1024*1024)"
# Необходимые опции System V.
options SYSVSHM #совместно используемая память в стиле SYSV
options SHMMAXPGS=262144 #макс. количество страниц совместно используемой
#options SHMMAXPGS=393216 #use this for the 46C inst.parameters
options SHMMNI=256 #макс. количество идентификаторов совместно используемой
options SHMSEG=100 #макс. количество сегментов разд. памяти на процесс
options SYSVMSG #очереди сообщений в стиле SYSV
options MSGSEG=32767 #макс. количество сегментов сообщений в системе
options MSGSSZ=32 #размер сегмента сообщений. ДОЛЖЕН быть степенью 2
options MSGMNB=65535 #макс. символов на очередь сообщений
options MSGTQL=2046 #макс. количество сообщений в системе
options SYSVSEM #семафоры в стиле SYSV
options SEMMNU=256 #количество структур UNDO семафоров
options SEMMNS=1024 #количество семафоров в системе
options SEMMNI=520 #количество идентификаторов семафоров
options SEMUME=100 #количество ключей UNDO
```

Минимальные значения указаны в документации, поставляемой вместе с SAP. Поскольку описания для Linux нет, дополнительную информацию см. в разделе HP-UX (32-bit). Поскольку в системе, на которую устанавливалась версия 4.6C SR2, оперативной памяти больше, сегменты совместно используемой можно сделать больше как для **SAP**, так и для **Oracle**, поэтому задайте большее количество страниц совместно используемой.

**Замечание:** При стандартной установке FreeBSD 4.5 на i386, задайте значения MAXDSIZ и DFLDSIZ не более 1 Гбайта. В противном случае могут выдаваться странные ошибки вроде ORA-27102: out of memory и Linux Error: 12: Cannot allocate memory.

## 10.7.8. Установка SAP R/3

### 10.7.8.1. Подготовка дисков CDROM SAP

В ходе установки придется монтировать и демонтировать много дисков CDROM. При наличии достаточного количества приводов CDROM, можно смонтировать их все. Мы же решили скопировать содержимое дисков CDROM в соответствующие каталоги:

```
/oracle/SID/sapreorg/имя_cd
```

где *имя\_cd* - одно из следующих KERNEL, RDBMS, EXPORT1, EXPORT2, EXPORT3, EXPORT4, EXPORT5 и EXPORT6 для установки 4.6B/IDES, и KERNEL, RDBMS, DISK1, DISK2, DISK3, DISK4 и LANG для установки 4.6C SR2. Все имена файлов на смонтированных дисках должны быть в верхнем регистре, в противном случае, используйте при монтировании опцию -g. Поэтому используйте следующие команды:

```
# mount_cd9660 -g /dev/cd0a /mnt
# cp -R /mnt/* /oracle/SID/sapreorg/имя_cd
# umount /mnt
```

### 10.7.8.2. Запуск скрипта установки

Сначала надо подготовить каталог install:

```
# cd /oracle/SID/sapreorg
# mkdir install
# cd install
```

Затем запускается скрипт установки, который копирует почти все необходимые файлы в каталог install:

```
# /oracle/SID/sapreorg/KERNEL/UNIX/INSTTOOL.SH
```

Дистрибутив IDES (4.6B) включает полностью настроенную демонстрационную систему SAP R/3, поэтому он включает шесть дисков EXPORT CD, а не три. В этот момент шаблон установки CENTRDB.R3S предназначен для установки стандартного центрального экземпляра (**R/3** и базы данных), а не для установки центрального экземпляра IDES, поэтому необходимо скопировать

соответствующий файл CENTRDB.R3S из каталога EXPORT1, иначе команда R3SETUP запросит только три диска EXPORT CDs.

Более новый релиз **SAP 4.6C SR2** включает четыре диска EXPORT CD. Шаги установки определяет файл параметров CENTRAL.R3S. В отличие от прежних релизов, больше нет отдельных шаблонов установки для центрального экземпляра с базой данных или без нее. **SAP** использует отдельный шаблон для установки базы данных. Для перезапуска установки в дальнейшем, однако, достаточно перезапустить исходный файл.

В ходе установки и после нее **SAP** требует, чтобы команда `hostname` возвращала только имя компьютера, не уточнённое именем домена. Поэтому либо задайте имя хоста в соответствии с этим требованием, либо настройте псевдоним с помощью команды `alias hostname='hostname -s'` для пользователей `orasisd` и `sidadm` (и для пользователя `root`, про крайней мере, в ходе шагов по установке, выполняемых от имени `root`). Можно также изменить файлы `.profile` и `.login` для обоих пользователей, которые создаются в ходе установки **SAP**.

### 10.7.8.3. Запуск R3SETUP 4.6B

Проверьте, что переменная среды `LD_LIBRARY_PATH` установлена правильно:

```
# export LD_LIBRARY_PATH=/oracle/IDS/lib:/sapmnt/IDS/exe:/oracle/805_32/lib
```

Выполните команду `R3SETUP` от имени пользователя `root` из каталога установки:

```
# cd /oracle/IDS/sapreorg/install
# ./R3SETUP -f CENTRDB.R3S
```

Скрипт затем задает ряд вопросов (стандартные ответы даны в скобках, а затем представлены реальные ответы):

Вопрос	Стандартное значение	Ответ
Enter SAP System ID	[C11]	IDSEnter
Enter SAP Instance Number	[00]	Enter
Enter SAPMOUNT Directory	[/sapmnt]	Enter
Enter name of SAP central host	[troubadix.domain.de]	Enter
Enter name of SAP db host	[troubadix]	Enter
Select character set	[1] (WE8DEC)	Enter
Enter Oracle server version (1) Oracle 8.0.5, (2) Oracle 8.0.6, (3) Oracle 8.1.5, (4) Oracle 8.1.6		1Enter
Extract Oracle Client archive	[1] (Yes, extract)	Enter
Enter path to KERNEL CD	[/sapcd]	/oracle/IDS/sapreorg/KERNEL
Enter path to RDBMS CD	[/sapcd]	/oracle/IDS/sapreorg/RDBMS
Enter path to EXPORT1 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT1
Directory to copy EXPORT1 CD	[/oracle/IDS/sapreorg/CD4_DIR]	Enter
Enter path to EXPORT2 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT2
Directory to copy EXPORT2 CD	[/oracle/IDS/sapreorg/CD5_DIR]	Enter



Вопрос	Стандартное значение	Ответ
Enter path to EXPORT3 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT3
Directory to copy EXPORT3 CD	[/oracle/IDS/sapreorg/CD6_DIR]	<b>Enter</b>
Enter path to EXPORT4 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT4
Directory to copy EXPORT4 CD	[/oracle/IDS/sapreorg/CD7_DIR]	<b>Enter</b>
Enter path to EXPORT5 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT5
Directory to copy EXPORT5 CD	[/oracle/IDS/sapreorg/CD8_DIR]	<b>Enter</b>
Enter path to EXPORT6 CD	[/sapcd]	/oracle/IDS/sapreorg/EXPORT6
Directory to copy EXPORT6 CD	[/oracle/IDS/sapreorg/CD9_DIR]	<b>Enter</b>
Enter amount of RAM for SAP + DB		850 <b>Enter</b> (in Megabytes)
Service Entry Message Server	[3600]	<b>Enter</b>
Enter Group-ID of sapsys	[101]	<b>Enter</b>
Enter Group-ID of oper	[102]	<b>Enter</b>
Enter Group-ID of dba	[100]	<b>Enter</b>
Enter User-ID of <i>sidadm</i>	[1000]	<b>Enter</b>
Enter User-ID of <i>orasid</i>	[1002]	<b>Enter</b>
Number of parallel procs	[2]	<b>Enter</b>

Если вы не скопировали диски в разные каталоги, инсталлятор **SAP** не сможет найти необходимые CD (идентифицируемые файлом LABEL.ASC на диске) и попросит затем вставить и смонтировать CD, и подтвердить или ввести точку его монтирования.

Файл CENTRDB.R3S может содержать ошибки. В нашем случае, он снова запросил EXPORT4 CD, но указал корректный ключ (6\_LOCATION, затем 7\_LOCATION и т.д.), так что, можно просто продолжать вводить корректные значения.

За исключением нескольких упомянутых ниже проблем, все должно идти нормально до момента, когда придется устанавливать программное обеспечение для работы с базой данных Oracle.

#### 10.7.8.4. Запуск R3SETUP 4.6C SR2

Проверьте, что переменная среды LD\_LIBRARY\_PATH установлена правильно. Это значение отличается от использованного при установке версии 4.6B с **Oracle 8.0.5**:

```
# export LD_LIBRARY_PATH=/sapmnt/PRD/exe:/oracle/PRD/817_32/lib
```

Выполните команду R3SETUP от имени пользователя root из каталога установки:

```
# cd /oracle/PRD/sapreorg/install
# ./R3SETUP -f CENTRAL.R3S
```

Скрипт затем задаст ряд вопросов (стандартные значения даны в скобках, а затем идут реальные ответы):

Вопрос	Стандартное значение	Ответ
Enter SAP System ID	[C11]	PRD <b>Enter</b>
Enter SAP Instance Number	[00]	<b>Enter</b>

Вопрос	Стандартное значение	Ответ
Enter SAPMOUNT Directory	[/sapmnt]	<b>Enter</b>
Enter name of SAP central host	[majestix]	<b>Enter</b>
Enter Database System ID	[PRD]	<b>PRDEnter</b>
Enter name of SAP db host	[majestix]	<b>Enter</b>
Select character set	[1] (WE8DEC)	<b>Enter</b>
Enter Oracle server version (2) Oracle 8.1.7		<b>2Enter</b>
Extract Oracle Client archive	[1] (Yes, extract)	<b>Enter</b>
Enter path to KERNEL CD	[/sapcd]	<b>/oracle/PRD/sapreorg/KERNEL</b>
Enter amount of RAM for SAP + DB	2044	<b>1800Enter</b> (in Megabytes)
Service Entry Message Server	[3600]	<b>Enter</b>
Enter Group-ID of sapsys	[100]	<b>Enter</b>
Enter Group-ID of oper	[101]	<b>Enter</b>
Enter Group-ID of dba	[102]	<b>Enter</b>
Enter User-ID of oraprd	[1002]	<b>Enter</b>
Enter User-ID of prdadm	[1000]	<b>Enter</b>
LDAP support		<b>3Enter</b> (no support)
Installation step completed	[1] (continue)	<b>Enter</b>
Choose installation service	[1] (DB inst, fi le)	<b>Enter</b>

Пока создание пользователей дает сообщение об ошибке при установке на стадиях OSUSERDBSID\_IND\_ORA (создание пользователя *orasid*) и OSUSERSIDADM\_IND\_ORA (создание пользователя *sidadm*).

За исключением некоторых упомянутых далее проблем, все должно идти нормально до момента, когда придется устанавливать программное обеспечение для работы с базой данных Oracle.

### 10.7.9. Установка Oracle 8.0.5

Описания возможных проблем с Linux и **Сервером Oracle** см. в соответствующих файлах SAP Notes и Oracle Readme. Большинство, если не все проблемы, связаны с несовместимыми библиотеками.

Подробнее об установке **Oracle** см. в главе Установка Oracle.

#### 10.7.9.1. Установка Oracle 8.0.5 с помощью *orainst*

Если надо использовать **Oracle 8.0.5**, для успешной перекомпоновки понадобится несколько дополнительных библиотек, поскольку **Oracle 8.0.5** был скомпонован со старой версией glibc (RedHat 6.0), но уже RedHat 6.1 использует новую библиотеку glibc. Так что, для успешной перекомпоновки нужно установить следующие дополнительные пакеты:

```
compat-libs-5.2-2.i386.rpm
```

```
compat-glibc-5.2-2.0.7.2.i386.rpm
```

```
compat-egcs-5.2-1.0.3a.1.i386.rpm
```

```
compat-egcs-c++-5.2-1.0.3a.1.i386.rpm
```

```
compat-binutils-5.2-2.9.1.0.23.1.i386.rpm
```

Дополнительную информацию см. в файлах SAP Notes или Oracle Readme. Если установить эти пакеты не представляется возможным (на момент установки у нас не было времени, чтобы это проверить), можно использовать исходные двоичные модули или перекомпилированные двоичные модули с исходной системы RedHat.

Для компиляции интеллектуального агента должен быть установлен пакет RedHat Tcl. Если вы не можете найти пакет `tcl-8.0.3-20.i386.rpm`, подойдет и более новый, вроде `tcl-8.0.5-30.i386.rpm` для RedHat 6.1.

За исключением перекомпиловки, установка выполняется просто:

```
# su - oraids
# export TERM=xterm
# export ORACLE_TERM=xterm
# export ORACLE_HOME=/oracle/IDS
# cd $ORACLE_HOME/orainst_sap
# ./orainst
```

Нажимайте на всех экранах клавишу **Enter**, пока программное обеспечение не будет установлено, убрав только пометку выбора с *Oracle On-Line Text Viewer*, поскольку этого компонента для Linux сейчас нет. **Oracle** затем захочет перекомпилировать модули с помощью `i386-glibc20-linux-gcc` вместо имеющихся `gcc`, `egcs` или `i386-redhat-linux-gcc`.

Из-за нехватки времени мы решили использовать двоичные модули из версии **Oracle 8.0.5 PreProduction** после того, как первая попытка заставить работать версию с RDBMS CD провалилась, — попытки найти и загрузить требуемые пакеты RPM нам показались настоящим кошмаром.

### 10.7.9.2. Установка Oracle 8.0.5 Pre-production Release для Linux (ядро 2.0.33)

Эту установку выполнить очень легко. Монтируем CD, запускаем инсталлятор. Затем он запрашивает местонахождение начального каталога Oracle и копирует туда двоичные модули. Мы, однако, не удаляли остатки прежних попыток установить RDBMS.

В конечном итоге, базу данных **Oracle** удалось запустить без проблем.

### 10.7.10. Установка tar-архива Oracle 8.1.7 для Linux

Создайте tar-архив `oracle81732.tgz` каталога установки на Linux-системе и разархивируйте его в каталог `/oracle/SID/817_32/`.

### 10.7.11. Продолжение установки SAP R/3

Сначала проверьте настройку среды для пользователей `idsamd` (`sidadm`) и `oraids` (`orasid`). У них обоих должны теперь быть файлы `.profile`, `.login` и `.cshrc`, использующие `hostname`. Если имя хоста в системе полностью уточнено, надо заменить `hostname` командой `hostname -s` во всех трех файлах.

### 10.7.11.1. Загрузка базы данных

Потом команду R3SETUP можно либо перезапустить, либо продолжить (в зависимости от того, была ли завершена ее работа). R3SETUP затем создает табличные пространства и загружает данные (для 46B IDES - с дисков от EXPORT1 до EXPORT6, для 46C - с дисков от DISK1 до DISK4) в базу данных с помощью утилиты R3load.

После завершения загрузки базы данных (это может занять несколько часов), будет запрошено несколько паролей. Для тестовых установок можно использовать хорошо известные стандартные пароли (но если защита важна - используйте другие!):

Вопрос	Ответ
Enter Password for sapr3	sapEnter
Confi rum Password for sapr3	sapEnter
Enter Password for sys	change_on_installEnter
Confi rm Password for sys	change_on_installEnter
Enter Password for system	managerEnter
Confi rm Password for system	managerEnter

Мы столкнулись с несколькими проблемами с dipgntab при установке 4.6B.

### 10.7.11.2. Процесс прослушивания

Запустите процесс прослушивания (**Oracle Listener**) от имени пользователя *orasid* следующим образом:

```
% umask 0; lsnrctl start
```

В противном случае, вы можете получить сообщение об ошибке ORA-12546, поскольку у сокетов будут неправильные права доступа. См. SAP Note 072984.

### 10.7.11.3. Обновление таблиц MNLS

Если вы планируете использовать в системе **SAP** языки, для которых не подходит кодировка Latin-1, придется изменить таблицы Multi National Language Support. Эта процедура описана в SAP OSS Notes 15023 и 45619. Если же нет, можете пропустить этот вопрос в ходе установки **SAP**.

**Замечание:** Если вам не нужна поддержка MNLS, все равно необходимо проверить таблицу TCPDB и инициализировать ее, если это еще не было сделано. Дополнительную информацию см. в SAP Note 0015023 и 0045619.

## 10.7.12. Шаги после установки

### 10.7.12.1. Запрос лицензионного ключа SAP R/3

Вы должны запросить ваш лицензионный ключ **SAP R/3**. Это необходимо, поскольку временная лицензия, использованная в ходе установки, действительна только четыре недели. Сначала получите ключ оборудования. Зарегистрируйтесь как пользователь `idsadm` и вызовите команду `saplicense`:

```
# /sapmnt/IDS/exe/saplicense -get
```

При вызове команды `saplicense` без параметров будет выдан список опций. После получения лицензионного ключа, его можно установить с помощью команды:

```
# /sapmnt/IDS/exe/saplicense -install
```

Затем вас попросят ввести следующие значения:

```
SAP SYSTEM ID      = SID, 3 символа
CUSTOMER KEY       = ключ оборудования, 11 символов
INSTALLATION NO    = установка, 10 цифр
EXPIRATION DATE    = yyyymmdd, обычно - "99991231"
LICENSE KEY        = лицензионный ключ, 24 символа
```

### 10.7.12.2. Создание пользователей

Создайте пользователя в клиенте 000 (некоторые задачи обязательно надо выполнять из клиента 000, от имени пользователя, отличающегося от `sap*` и `ddic`). В качестве имени пользователя мы обычно выбираем `wartung` (или `service`, по английски). Требуются профили `sap_new` и `sap_all`. Для дополнительной защиты надо изменить пароли стандартных пользователей на всех клиентах (в том числе, пользователей `sap*` и `ddic`).

### 10.7.12.3. Конфигурирование системы передачи, профиля, режимов работы и т.п.

В клиенте 000, от имени пользователя, отличающегося от `ddic` и `sap*`, выполните, как минимум, следующее:

Задача	Транзакция
Сконфигурируйте систему передачи, например, как <i>Stand-Alone Transport Domain Entity</i>	STMS
Создайте/Отредактируйте профиль для системы	RZ10
Сконфигурируйте режимы работы и экземпляры	RZ04

Эти и другие шаги, которые надо выполнить после установки, подробно описаны в руководствах по установке **SAP**.

### 10.7.12.4. Редактирование `initSID.sap` (`initIDS.sap`)

Файл `/oracle/IDS/dbs/initIDS.sap` содержит профиль резервного копирования **SAP**. Здесь надо

задать размер используемой ленты, тип сжатия и т.д. Чтобы можно было использовать `sapdba / brbackup`, мы изменили следующие значения:

```
compress = hardware
archive_function = copy_delete_save
cpio_flags = "-ov --format=newc --block-size=128 --quiet"
cpio_in_flags = "-iuv --block-size=128 --quiet"
tape_size = 38000M
tape_address = /dev/nsa0
tape_address_rew = /dev/sa0
```

Объяснения:

`compress`: мы использовали ленту HP DLT1, которая поддерживает аппаратное сжатие.

`archive_function`: этот параметр задает стандартное поведение для сохранения архивных журналов Oracle: новые журнальные файлы сохраняются на ленту, уже сохраненные файлы журнала сохраняются еще раз, а затем удаляются. Это предотвращает многочисленные проблемы, если потребуется восстановить базу данных, а одна из архивных лент окажется сбойной.

`cpio_flags`: по умолчанию используется `-b`, что устанавливает размер блока 5120 байт. Для лент DLT компания HP рекомендует размер блока не меньше 32 Кбайт, поэтому мы использовали значение `--block-size=128` для задания размера блока 64 Кбайта. Опция `--format=newc` необходима, поскольку у нас есть индексные дескрипторы (inodes) с номерами больше 65535. Последняя опция, `--quiet` необходима потому, что иначе команда `brbackup` выдает сообщение об ошибке, как только команда `cpio` выдаст количество сохраненных блоков.

`cpio_in_flags`: флаги, необходимые для загрузки данных с ленты. Формат распознается автоматически.

`tape_size`: обычно этот параметр задает реальную ёмкость ленты. Из соображений надежности (мы используем аппаратное сжатие), задано значение несколько меньше фактического.

`tape_address`: устройство без перемотки для использования в команде `cpio`.

`tape_address_rew`: устройство с перемоткой для использования в команде `cpio`.

#### 10.7.12.5. Проблемы конфигурирования после установки

Следующие параметры **SAP** надо настроить после установки (примеры для IDES 46B, 1 Гбайт памяти):

Имя	Значение
ztta/roll_extension	250000000
abap/heap_area_dia	300000000
abap/heap_area_nondia	400000000
em/initial_size_MB	256
em/blocksize_kB	1024
ipc/shm_psize_40	70000000

SAP Note 0013026:

Имя	Значение
-----	----------

Имя	Значение
ztta/dynpro_area	2500000

SAP Note 0157246:

Имя	Значение
rdisp/ROLL_MAXFS	16000
rdisp/PG_MAXFS	30000

**Замечание:** При указанных выше параметрах в системе с 1 Гбайт памяти можно обнаружить примерно следующее использование памяти:

Mem: 547M Active, 305M Inact, 109M Wired, 40M Cache, 112M Buf, 3492K Free

### 10.7.13. Проблемы в ходе установки

#### 10.7.13.1. Перезапуск R3SETUP после устранения проблемы

R3SETUP останавливается при выявлении ошибки. Если вы просмотрели соответствующие журнальные файлы и исправили ошибку, придется запускать R3SETUP снова, обычно выбирая REPEAT как опцию для последнего шага, на котором команда R3SETUP выдала сообщение об ошибке.

Для перезапуска команды R3SETUP просто запустите её с соответствующим файлом R3S:

```
# ./R3SETUP -f CENTRDB.R3S
```

для 4.6B или с файлом

```
# ./R3SETUP -f CENTRAL.R3S
```

для 4.6C, независимо от того, произошла ли ошибка при работе с файлом CENTRAL.R3S или DATABASE.R3S.

**Замечание:** На некоторых стадиях команда R3SETUP предполагает, что запущены и работают процессы как сервера базы данных, так и **SAP** (поскольку эти шаги уже выполнены). Если возникнут ошибки и, например, запустить сервер базы данных не получится, придется вручную запускать сервер базы данных и **SAP** после исправления ошибок и до повторного запуска R3SETUP.

Не забудьте также снова запустить процесс прослушивания **Oracle** (как пользователь `orasis` с помощью команды `umask 0; lsnrctl start`), если он тоже был остановлен (например, из-за необходимой перезагрузки системы).

**10.7.13.2. OSUSERSIDADM\_IND\_ORА в ходе R3SETUP**

Если R3SETUP выдает сообщения об ошибках на этом этапе, отредактируйте используемый при этом файл шаблона R3SETUP (CENTRDB.R3S (4.6B), либо CENTRAL.R3S или DATABASE.R3S (4.6C)). Найдите раздел [OSUSERSIDADM\_IND\_ORА] или поищите единственную запись STATUS=ERROR и отредактируйте следующие значения:

```
HOME=/home/sidadm (было пусто)
STATUS=OK (был статус ERROR)
```

Затем надо снова перезапустить R3SETUP.

**10.7.13.3. OSUSERDBSID\_IND\_ORА в ходе R3SETUP**

Возможно, команда R3SETUP также выдаст сообщения об ошибке на этой стадии. Ошибка здесь аналогична возникающей на стадии OSUSERSIDADM\_IND\_ORА. Просто отредактируйте используемый файл шаблона R3SETUP (CENTRDB.R3S (4.6B), либо CENTRAL.R3S или DATABASE.R3S (4.6C)). Найдите раздел [OSUSERDBSID\_IND\_ORА] или поищите единственную запись STATUS=ERROR и отредактируйте следующее значение в этом разделе:

```
STATUS=OK
```

Затем перезапустите R3SETUP.

**10.7.13.4. oraview.vrf FILE NOT FOUND в ходе установки Oracle**

Вы не сняли выбор с *Oracle On-Line Text Viewer* перед началом установки. Он помечен для установки, хотя этот продукт и не доступен сейчас для Linux. Снимите пометку с этого продукта в меню установки **Oracle** и перезапустите установку.

**10.7.13.5. TEXTENV\_INVALID в ходе R3SETUP, RFC или запуска SAPgui**

Если возникает эта ошибка, не найдена нужная локаль. SAP Note 0171356 перечисляет необходимые пакеты RPM, которые надо установить (например, saplocales-1.0-3, saposcheck-1.0-1 для RedHat 6.1). Если игнорировать все ошибки и менять STATUS соответствующих шагов с ERROR на OK (в файле CENTRDB.R3S) каждый раз, когда R3SETUP сообщает об ошибке и просто перезапускать R3SETUP, система **SAP** не будет правильно сконфигурирована, и вы затем не сможете подключиться к системе с помощью **SAPgui**, хотя запустить систему и получится. Попытка подключения с помощью старой Linux-версии **SAPgui** приведет к выдаче следующих сообщений:

```
Sat May 5 14:23:14 2001
*** ERROR => no valid userarea given [trgmsggo. 0401]
Sat May 5 14:23:22 2001
*** ERROR => ERROR NR 24 occurred [trgmsggi. 0410]
*** ERROR => Error when generating text environment. [trgmsggi. 0435]
*** ERROR => function failed [trgmsggi. 0447]
*** ERROR => no socket operation allowed [trxio.c 3363]
Speicherzugriffsfehler
```



Это связано с тем, что система **SAP R/3** не может корректно назначить локаль и сама не была надлежащим образом сконфигурирована (не хватает записей в некоторых таблицах базы данных). Чтобы можно было подключиться к **SAP**, добавьте следующие записи в файл `DEFAULT.PFL` (см. Note 0043288):

```
abap/set_etct_env_at_new_mode = 0
install/collate/active = 0
rscp/TCP0B = TCP0B
```

Перезапустите систему **SAP**. Теперь вы можете подключиться к системе, хотя специфические для страны языковые установки могут работать не так, как предполагалось. После исправления настроек страны (и добавления соответствующих локалей) эти записи можно удалить из файла `DEFAULT.PFL` и перезапустить систему **SAP**.

#### 10.7.13.6. ORA-00001

Эта ошибка возникает только с **Oracle 8.1.7** на FreeBSD 4.5. Причина в том, что сервер **Oracle** не может правильно проинициализироваться и аварийно завершает работу, оставляя не освобожденными в системе семафоры и совместно используемую память. При следующей попытке запустить сервер базы данных выдается ошибка ORA-00001.

Найдите оставшиеся семафоры и сегменты памяти с помощью команды `ipcs -a` и удалите с помощью `ipcrm`.

#### 10.7.13.7. ORA-00445 (фоновый процесс PMON не запущен)

Эта ошибка произошла с **Oracle 8.1.7**. Она выдается, если сервер был запущен с помощью обычного скрипта `startsap` (например, `startsap_majestix_00`) от имени пользователя `prdadm`.

Возможный способ обхода – запускать сервер базы данных от имени пользователя `oraprd` с помощью `svrmgrl`:

```
% svrmgrl
SVRMGR> connect internal;
SVRMGR> startup;
SVRMGR> exit
```

#### 10.7.13.8. ORA-12546 (запускайте процесс прослушивания с правильными правами)

Запускайте процесс прослушивания **Oracle** от имени пользователя `oraids` следующими командами:

```
# umask 0; lsnrctl start
```

В противном случае, вы можете получить сообщение об ошибке ORA-12546, поскольку сокеты не будут иметь нужных прав доступа. См. SAP Note 0072984.

### 10.7.13.9. ORA-27102 (не хватает памяти)

Эта ошибка произошла при попытке использовать значения MAXDSIZ и DFLDSIZ больше 1 Гбайта (1024x1024x1024). Кроме того, мы получили Linux Error 12: Cannot allocate memory.

### 10.7.13.10. [DIPGNTAB\_IND\_IND] в ходе R3SETUP

В общем случае, см. SAP Note 0130581 (прекращается работа R3SETUP на шаге DIPGNTAB). В ходе установки IDES-версии по каким-то причинам процесс установки использовал вместо правильного имени системы **SAP**, **IDS**, пустую строку, **" "**. Это к небольшим проблемам при доступе к каталогам, поскольку пути генерируются динамически на базе *SID* (в данном случае, **IDS**). Поэтому вместо обращения к:

```
/usr/sap/IDS/SYS/...
/usr/sap/IDS/DVMGS00
```

используются следующие пути:

```
/usr/sap//SYS/...
/usr/sap/D00
```

Чтобы продолжить установку мы создали ссылку и дополнительный каталог:

```
# pwd
/compat/linux/usr/sap
# ls -l
total 4
drwxr-xr-x 3 idsadm sapsys 512 May 5 11:20 D00
drwxr-x--x 5 idsadm sapsys 512 May 5 11:35 IDS
lrwxr-xr-x 1 root sapsys 7 May 5 11:35 SYS -> IDS/SYS
drwxrwxr-x 2 idsadm sapsys 512 May 5 13:00 tmp
drwxrwxr-x 11 idsadm sapsys 512 May 4 14:20 trans
```

Мы также нашли документы SAP Notes (0029227 и 0008401), описывающие это поведение. Мы не столкнулись с подобными проблемами при установке **SAP 4.6C**.

### 10.7.13.11. [RFCRSWBOINI\_IND\_IND] в ходе R3SETUP

В ходе установки **SAP 4.6C**, эта ошибка возникла в результате другой ошибки, произошедшей ранее по ходу установки. В данном случае придется просмотреть соответствующие журнальные файлы и устранить исходную проблему.

Если после просмотра журнальных файлов выявлена только эта ошибка (проверьте SAP Notes), можно поменять STATUS соответствующего шага с ERROR на OK (в файле CENTRDB.R3S) и перезапустить R3SETUP. После установки надо выполнить отчет RSWBOINS из транзакции SE38. Дополнительную информацию о стадиях RFCRSWBOINI и RFCRADBDIF см. в SAP Note 0162266.

### 10.7.13.12. [RFCRADBDIF\_IND\_IND] в ходе R3SETUP

Здесь применяются те же ограничения: проверьте путем просмотра журнальных файлов, что эта ошибка не вызвана какими-то предыдущими проблемами.

Если подтверждается, что применим документ SAP Note 0162266, просто поменяйте STATUS соответствующего шага с ERROR на OK (в файле CENTRDB.R3S) и перезапустите R3SETUP. После установки надо выполнить отчет RADDDBDIF из транзакции SE38.

### 10.7.13.13. sigaction sig31: File size limit exceeded

Это сообщение об ошибке выдается в ходе запуска процессов **SAP** *disp+work*. Если **SAP** запускается скриптом *startsap*, запускаются отдельные подпроцессы, выполняющие грязную работу по запуску всех остальных процессов **SAP**. В результате, сам скрипт не получит уведомления, если что-то пойдет не так.

Чтобы проверить, нормально ли запустились процессы **SAP**, посмотрите на состояние процессов с помощью команды `ps ax | grep SID`, которая выдаст список всех процессов **Oracle** и **SAP**. Если похоже, что некоторых процессов не хватает или вы не можете подключиться к системе **SAP**, просмотрите соответствующие журнальные файлы, которые можно найти в каталоге `/usr/sap/SID/DVEBMGSnr/work/`. Надо просматривать файлы `dev_ms` и `dev_disp`.

Сигнал 31 выдается, если объем памяти, совместно используемой **Oracle** и **SAP**, превосходит заданный в файле конфигурации ядра, и от него можно избавиться, указав большее значение:

```
# большее значение для производственных систем 46C:
options SHMMAXPGS=393216
# меньшее значение, достаточное для 46B:
#options SHMMAXPGS=262144
```

### 10.7.13.14. Сбой при запуске saposcol

Есть ряд проблем с программой *saposcol* (версии 4.6D). Система **SAP** использует *saposcol* для сбора данных о производительности системы. Эта программа не нужна для использования системы **SAP**, так что проблему можно отнести к несерьезным. Более старые версии (4.6B) работают, но собирают не все данные (многие вызовы просто возвращают 0, например, для использования процессора).

## 10.8. Дополнительные сведения

Если вы интересуетесь, как обеспечивается двоичная совместимость с Linux, этот раздел для вас. Большинство материала взято из электронного письма, адресованного в Список рассылки, посвященный неформальным беседам о FreeBSD (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-chat>) by Terry Lambert <tlambert@primenet.com> (ID письма: <199906020108.SAA07001@usr09.primenet.com>).

### 10.8.1. Как все это устроено?

FreeBSD поддерживает абстракцию, называемую "загрузчик выполняемых классов". Фактически, он является первой стадией системного вызова `execve(2)`.

На самом деле, FreeBSD имеет список загрузчиков вместо одного, завершающийся загрузчиком `#!` для запуска любых командных интерпретаторов и скриптов.

Исторически сложилось, что единственный загрузчик в UNIX системах проверял "магическое число" (чаще всего первые 4 или 8 байт файла), чтобы определить, известен ли формат двоичного файла системе, и если да, то вызвал соответствующий загрузчик.

Если файл не опознавался системой как двоичный, системный вызов `execve(2)` возвращал ошибку, и текущий командный интерпретатор начинал выполнять файл как скрипт.

По умолчанию скрипт выполнялся "текущим командным интерпретатором".

Позднее, `sh(1)` был модифицирован, так, чтобы проверять первые два символа в файле, и если они оказывались `:\n`, то файл выполнялся как сценарий для `csh(1)` (утверждается, что SCO были первыми, кто сделал эту модификацию).

FreeBSD сейчас ведет себя по-другому: пробегает по списку загрузчиков, включающему специальный загрузчик `#!`, который вызывает нужный интерпретатор, указанный после этих символов до следующего пробела, или `/bin/sh`, если не нашел подходящего.

Для поддержки Linux ABI FreeBSD ищет магическое число, соответствующее двоичному файлу ELF (на этой стадии не различаются FreeBSD, Solaris, Linux или любая другая ОС поддерживающая формат ELF).

Далее, ELF-загрузчик определяет "марку" (brand) двоичного файла ELF (специальный комментарий в ELF-файле, отсутствующий в двоичных файлах ELF SVR4/Solaris).

Соответственно, Linux программы должны быть "маркированы" для Linux (например, с помощью утилиты `brandelf(1)`):

```
# brandelf -t Linux file
```

Когда это сделано, загрузчик ELF выявит марку Linux в файле.

Когда ELF-загрузчик находит "марку" Linux, он заменяет соответствующий указатель в структуре `proc`. Все системные вызовы индексируются через этот указатель (в традиционной UNIX системе это массив структур `sysent[ ]`, содержащий системные вызовы). Кроме того, процесс помечается для специальной обработки вектора обработчиков сигналов, а также ряда других (небольших) исправлений, которые осуществляются специальным модулем ядра для поддержки Linux.

Вектор системных вызовов Linux содержит, среди прочего, список записей `sysent[ ]`, адреса которых находятся в модуле ядра.

При выполнении системного вызова из двоичного файла Linux, код обработчика разыменовывает указатель на функцию системного вызова из структуры `proc`, и получает точки входа системных вызовов Linux, а не FreeBSD.

Плюс ко всему, в Linux-режиме динамически "изменяется корень" файловой системы при поиске файлов; фактически так же, как и параметр `union` при монтировании файловых систем (не путать с `unionfs`!). Сперва, файл ищется в каталоге `/compat/linux/исходное_полное_имя` и только затем, в случае неудачи, в `/исходное_полное_имя`. Это гарантирует, что программы, которым требуются другие программы, смогут работать (например, весь набор инструментальных средств Linux сможет работать в среде поддержки Linux ABI). Это также дает возможность Linux программам выполнять FreeBSD команды, если не найдется соответствующих Linux команд. Например, можно скопировать FreeBSD `uname(1)` в дерево каталогов `/compat/linux`, и Linux-программы не смогут разобраться, что они работают не в Linux.

Фактически, имеется ядро Linux в ядре FreeBSD; различные базовые функции, реализующие все услуги ядра, идентичны как в записях таблицы системных вызовов FreeBSD, так и в записях таблицы

системных вызовов Linux: операции с файловой системой, виртуальная память, средства доставки сигналов, System V IPC ... Единственное отличие в том, что FreeBSD-программы получают *интерфейсные* функции FreeBSD, а Linux-программы получают *интерфейсные* функции Linux (в большинстве более старых ОС есть только их собственные интерфейсные функции: функции берутся из статического глобального массива структур `sysent[]`, а не из массива, полученного разыменованием динамически проинициализированного указателя в структуре `proc` процесса, выполняющего вызов).

Какая же реализация ABI для FreeBSD "родная"? Это не имеет значения. Единственное различие (на данный момент, в будущем все может и, вероятно, изменится), пожалуй, в том, что функции системных вызовов FreeBSD зашиты в ядро, а для Linux они могут быть либо статически скомпонованы в ядро, либо получаться через модуль ядра.

Да, но можно ли назвать это эмуляцией? Нет. Это реализация ABI, а не эмуляция. Как таковой, эмулятор (или симулятор) отсутствует.

В таком случае, почему же иногда говорят об "эмуляции Linux"? Чтобы "насолить" FreeBSD!

Фактически, причина в том, что на момент первой реализации не существовало слова, которое бы точнее описывало этот процесс. Нельзя было сказать, что FreeBSD запускает приложения Linux (без перекомпиляции или загрузки соответствующего модуля ядра это невозможно). Но надо было как-то описать, что загружается—отсюда и "эмулятор Linux".

# III. Системное Администрирование

Оставшиеся главы Руководства охватывают все аспекты администрирования FreeBSD системы. Каждая глава начинается с описания того, что вы сможете изучить в результате прочтения этой главы.

Эти главы спланированы там, что вы можете прочитать их когда вам нужно узнать какую-либо информацию. Вам не нужно читать их в определенном порядке, и не нужно прочитать их все перед тем, как начать пользоваться FreeBSD.

# Глава 11. Настройка и оптимизация

*Написал Chern Lee. Основывается на учебнике, написанном Mike Smith. и на tuning(7), написанном Matt Dillon. Перевод на русский язык: Андрей Мельник, Денис Пеплин, Валерий Кравчук.*

## 11.1. Введение

Один из важных аспектов FreeBSD это настройка системы. Правильная настройка системы поможет избежать головной боли при последующих обновлениях. Эта глава описывает большую часть процесса настройки FreeBSD, включая некоторые параметры, которые можно установить для оптимизации системы FreeBSD.

После прочтения этой главы вы узнаете:

- Как эффективно работать с файловыми системами и разделами подкачки.
- Основы настройки `rc.conf` и системы запуска приложений `/usr/local/etc/rc.d`.
- Как настроить и протестировать сетевую карту.
- Как настроить виртуальные хосты на сетевых устройствах.
- Как использовать различные файлы конфигурации в `/etc`.
- Как оптимизировать FreeBSD, используя переменные `sysctl`.
- Как увеличить скорость работы дисков и изменить ограничения, накладываемые ядром.

Перед прочтением этой главы вам следует:

- Понять основы UNIX и FreeBSD (Гл. 3).
- Ознакомиться с основами конфигурации/компиляции ядра (Гл. 8).

## 11.2. Начальное конфигурирование

### 11.2.1. Разделы диска

#### 11.2.1.1. Основы построения разделов

Во время разметки жёсткого диска с помощью `disklabel(8)` или `sysinstall(8)`, важно помнить, что скорость чтения и записи данных уменьшается от внешних к внутренним трекам диска. Самые маленькие и самые часто используемые файловые системы (корневую и раздел подкачки) должны быть расположены в начале диска, в то время как самые большие, такие, как `/usr`, в конце. Самым оптимальным считается следующий порядок расположения файловых систем: `root`, `swap`, `/var`, `/usr`.

Размер файловой системы `/var` определяется предназначением машины. `/var` используется для хранения почтовых ящиков, очередей печати и лог файлов. Размер почтовых ящиков и лог файлов может расти неограниченно в зависимости от количества пользователей системы и от того, как долго

хранятся лог-файлы. Большинству пользователей никогда не потребуется гигабайт, но помните, что `/var/tmp` должен быть достаточно большим для пакетов.

В разделе `/usr` содержится большинство файлов, необходимых для поддержки системы, `ports(7)` (порты, рекомендуется) и исходные тексты (опционально). Оба эти каталога опциональны при установке. Для этого раздела рекомендуется как минимум 2 гигабайта.

При установке размера разделов, не забудьте принять во внимание рост размера требуемого системе дискового пространства. Переполнение одного раздела даже при наличии свободного места на другом может вызвать затруднения.

**Замечание:** Многие пользователи обнаружили, что размер разделов, предлагаемый `sysinstall(8)`’ом по умолчанию, иногда меньше подходящего для разделов `/var` и `/`. Тщательно планируйте размер разделов и не жалейте места.

### 11.2.1.2. Раздел подкачки

Как правило, размер раздела подкачки должен быть равен удвоенному размеру оперативной памяти. Например, если на машине установлено 128 мегабайт памяти, раздел подкачки должен быть 256 мегабайт. Системы с меньшим количеством памяти могут работать лучше с большим объёмом раздела подкачки. Не рекомендуется устанавливать размер раздела подкачки меньше 256 мегабайт, необходимо также принять во внимание возможное наращивание объема установленной на машине памяти. Алгоритмы кэширования VM настроены на максимальное быстроедействие, когда размер раздела подкачки равен как минимум удвоенному размеру памяти. Заниженный размер раздела подкачки может привести к неэффективной работе постраничного сканирования VM и вызвать проблемы при увеличении объёма памяти.

На больших системах с несколькими SCSI дисками (или несколькими IDE дисками, находящимися на разных контроллерах), рекомендуется создавать раздел подкачки на каждом диске (до четырёх дисков). Разделы подкачки должны быть примерно одного размера. Ядро не накладывает ограничений на размер раздела подкачки, но внутренние структуры позволяют иметь общий размер разделов подкачки, равный наибольшему, умноженному на четыре. Выделение под разделы подкачки примерно одинакового места позволить ядру оптимально расположить разделы подкачки. Установка размера подкачки больше требуемого нормальна, даже если этот объем не используется. В этих условиях может быть проще восстановиться после зависания программы перед тем, как возникнет необходимость перезагрузки.

### 11.2.1.3. Зачем нужны разделы?

Некоторые пользователи считают, что лучше использовать один большой раздел, но есть несколько причин, по которым этого лучше не делать. Во-первых, у каждого раздела свои характеристики, и отделяя их, можно выполнить соответствующие настройки. Например, корневая и файловая система и `/usr` в основном предназначены для чтения, без большого объема записи. В то же время множество операций чтения и записи выполняется в `/var` и `/var/tmp`.

При правильном размещении и выборе размера разделов системы, фрагментация в более маленьких разделах, куда часто записываются данные, не перенесётся на остальные разделы. Размещение самых часто используемых разделов ближе к началу диска увеличит скорость ввода/вывода там, где она



нужна больше всего. Хотя производительность важна и для больших дисков, передвижение их ближе к концу диска не повлечёт значительного уменьшения быстродействия по сравнению с перемещением ближе к концу диска `/var`. И, наконец, разделы существуют и из соображений безопасности. Наличие маленького аккуратного корневого раздела, доступного только для чтения даёт значительные шансы на "выживание" после краха системы.

## 11.3. Основные настройки

Основные настройки системы располагаются в `/etc/rc.conf`. Этот файл вмещает широкий спектр конфигурационной информации, используемой при загрузке системы. Имя этого файла прямо отражает его назначение, это файл настройки для файлов `rc*`.

Администратор должен сделать записи в `rc.conf` чтобы переопределить строки по умолчанию из `/etc/defaults/rc.conf`. Файлы по умолчанию нельзя копировать в `/etc` - они вмещают значения по умолчанию, а не примеры значений. Все специфичные для данной системы изменения должны быть сделаны в файле `rc.conf`.

Существует несколько методов для отделения общей конфигурации для группы систем от конкретной для данной системы в целях уменьшения объема работы администратора. Рекомендуемый метод - прописать общую конфигурацию в отдельный файл, например, в `/etc/rc.conf.site`, и включить его название в `/etc/rc.conf`, который вмещает только специфичную для данной системы информацию.

Поскольку `rc.conf` читается `sh(1)`, есть тривиальный способ сделать это. Например:

- `rc.conf`:

```
. rc.conf.site
hostname="node15.example.com"
network_interfaces="fxp0 lo0"
ifconfig_fxp0="inet 10.1.1.1"
```

- `rc.conf.site`:

```
defaultrouter="10.1.1.254"
saver="daemon"
blanktime="100"
```

Файл `rc.conf.site` может быть распространён на все системы, используя `rsync` или подобную ей программу, в то время, как `rc.conf` должен остаться только на одной машине.

Обновление системы с помощью `sysinstall(8)` или `make world` не повлекут за собой перезапись `rc.conf`. Вся информация в этом файле сохранится.

## 11.4. Настройка приложений

Обычно, установленные приложения имеют свои конфигурационные файлы, со своим собственным синтаксисом. Важно хранить эти файлы отдельно от файлов основной системы, чтобы их можно было легко администрировать с помощью средств управления пакетами.

Обычно эти файлы устанавливаются в `/usr/local/etc`. В случае, если приложению нужно большое количество конфигурационных файлов, для их хранения будет создан подкаталог.

Обычно, вместе с установкой портов и пакетов, устанавливаются и примеры конфигурационных файлов. Обычно они имеют расширение `.default`. Если не существует конфигурационных файлов для этого приложения, они будут созданы путём копирования `.default` файлов.

Например, `/usr/local/etc/apache`:

```
-rw-r--r--  1 root  wheel   2184 May  20   1998 access.conf
-rw-r--r--  1 root  wheel   2184 May  20   1998 access.conf.default
-rw-r--r--  1 root  wheel   9555 May  20   1998 httpd.conf
-rw-r--r--  1 root  wheel   9555 May  20   1998 httpd.conf.default
-rw-r--r--  1 root  wheel  12205 May  20   1998 magic
-rw-r--r--  1 root  wheel  12205 May  20   1998 magic.default
-rw-r--r--  1 root  wheel   2700 May  20   1998 mime.types
-rw-r--r--  1 root  wheel   2700 May  20   1998 mime.types.default
-rw-r--r--  1 root  wheel   7980 May  20   1998 srm.conf
-rw-r--r--  1 root  wheel   7933 May  20   1998 srm.conf.default
```

Размеры файлов показывают, что только файл `srm.conf` был изменён. При следующем обновлении **Apache** этот файл уже не будет перезаписан.

## 11.5. Запуск сервисов

*Предоставил Tom Rhodes.*

Многие пользователи предпочитают устанавливать программы сторонних производителей в FreeBSD из набора портов. В подобных случаях может потребоваться сконфигурировать программы так, чтобы они запускались при инициализации системы. Сервисы, такие как `mail/postfix` или `www/apache13`, — это лишь два примера множества программных пакетов, которые можно запускать при инициализации системы. В этом разделе описывается процедура, предназначенная для запуска программ сторонних разработчиков.

Большинство входящих в FreeBSD сервисов, таких как `cron(8)`, запускается с помощью стартовых скриптов системы. Эти скрипты могут различаться в зависимости от версии FreeBSD или ее производителя; однако важнее всего учитывать, что их начальную конфигурацию можно задать с помощью простых стартовых скриптов.

До появления `rcNG` приложения должны были помещать простой стартовый скрипт в каталог `/usr/local/etc/rc.d`, который затем читался скриптами инициализации системы. Эти скрипты затем выполнялись в ходе последующих стадий запуска системы.

Хотя много разработчиков потратили часы на попытки внедрить старый стиль конфигурирования в новую систему, остаётся фактом, что для некоторых утилит сторонних производителей по-прежнему необходим скрипт, помещённый в указанный выше каталог. Незначительные различия в скриптах

зависят от того, используется ли rcNG. До версии FreeBSD 5.1 использовались скрипты в старом стиле, и почти во всех случаях скрипты в новом стиле должны подойти так же хорошо.

Хотя каждый скрипт должен соответствовать некоторым минимальным требованиям, в большинстве случаев эти требования не зависят от версии FreeBSD. Каждый скрипт должен иметь в конце расширение `.sh` и каждый скрипт должен быть выполняемым. Последнее требование может быть выполнено путем установки командой `chmod` уникальных прав доступа 755. Также, как минимум, должна быть опция `start` для запуска приложения и опция `stop` для его остановки.

Простейший стартовый скрипт, пожалуй, будет похож на следующий:

```
#!/bin/sh
echo -n ' utility'

case "$1" in
start)
    /usr/local/bin/utility
    ;;
stop)
    kill -9 `cat /var/run/utility.pid`
    ;;
*)
    echo "Usage: `basename $0` {start|stop}" >&2
    exit 64
    ;;
esac

exit 0
```

Этот скрипт поддерживает опции `stop` и `start` для приложения, которое мы здесь называем просто — `utility`. Для этого приложения затем можно поместить следующую строку в файл `/etc/rc.conf`:

```
utility_enable="YES"
```

А можно запускать его и вручную, с помощью команды:

```
# /usr/local/etc/rc.d/utility.sh start
```

Хотя и не все программы сторонних производителей требуют добавления строки в файл `rc.conf`, практически каждый день очередной новый порт меняется так, чтобы поддерживать подобную конфигурацию. Поищите в результатах, выдаваемых после установки более детальную информацию по конкретному приложению. Некоторые программы сторонних производителей будут включать стартовые скрипты, позволяющие использовать приложение с rcNG; но это мы еще обсудим в следующем разделе.

### 11.5.1. Расширенное конфигурирование приложения

Теперь, когда FreeBSD включает rcNG, конфигурирование запуска приложений стало более оптимальным; фактически, оно стало более тщательным. С помощью ключевых слов, рассмотренных в разделе rcNG, приложения теперь можно настроить для запуска после других заданных сервисов, например, DNS; можно разрешить передачу дополнительных флагов через `rc.conf` вместо жесткого задания флагов в стартовых скриптах, и т.д. Простой скрипт может иметь следующий вид:

```
#!/bin/sh
#
# PROVIDE: utility
# REQUIRE: DAEMON
# BEFORE: LOGIN
# KEYWORD: FreeBSD shutdown

#
# НЕ МЕНЯЙТЕ ЗДЕСЬ ЭТИ СТАНДАРТНЫЕ ЗНАЧЕНИЯ
# ЗАДАВАЙТЕ ИХ В ФАЙЛЕ /etc/rc.conf

#

utility_enable=${utility_enable-"NO"}
utility_flags=${utility_flags-""}
utility_pidfile=${utility_pidfile-"/var/run/utility.pid"}

. /etc/rc.subr

name="utility"
rcvar='set_rcvar'
command="/usr/local/sbin/utility"

load_rc_config $name

pidfile="${utility_pidfile}"

start_cmd="echo \"Starting ${name}.\"; /usr/bin/nice -5 ${command} ${utility_flags} ${command_args}"

run_rc_command "$1"
```

Этот скрипт будет гарантировать, что указанное приложение **utility** будет запущено перед сервисом **login**, но после сервиса **daemon**. Он также предоставляет метод для создания и отслеживания файла идентификатора процесса, **PID**.

Этот новый метод также позволяет легко работать с аргументами командной строки, включать стандартные функции из файла `/etc/rc.subr`, обеспечивает совместимость с утилитой `rcorder(8)` и упрощает конфигурирование с помощью файла `rc.conf`. По сути, этот сценарий можно даже поместить в каталог `/etc/rc.d`. Это, однако, потенциально может сбить с толку утилиту `mergemaster(8)` при обновлениях программного обеспечения.

### 11.5.2. Использование сервисов для запуска сервисов

Другие сервисы, такие как демоны сервера POP3, IMAP, и т.п. могут быть запущены с помощью `inetd(8)`. Для этого необходимо установить сервисную утилиту из набора портов и добавить соответствующую строчку конфигурации в файл `/etc/inetd.conf` или раскомментировать подходящую строку конфигурации из уже имеющихся. Работа с демоном **inetd** и его конфигурирование подробно описано в разделе `inetd`.

В некоторых случаях использование для запуска системных служб демона `cron(8)` может оказаться более приемлемым. Этот подход имеет несколько преимуществ, поскольку демон `cron` запускает эти

процессы от имени владельца файла `crontab`. Это позволяет обычным пользователям запускать и поддерживать некоторые приложения.

Утилита `cron` поддерживает уникальную возможность, `@reboot`, — это значение можно использовать вместо спецификации времени. В результате, задание будет выполнено при запуске `cron(8)`, обычно — в ходе инициализации системы.

## 11.6. Настройка утилиты `cron`

*Предоставил Tom Rhodes.*

Одна из наиболее полезных утилит FreeBSD это `cron(8)`. Утилита `cron` работает в фоновом режиме и постоянно проверяет файл `/etc/crontab`. Утилита `cron` проверяет также каталог `/var/cron/tabs` в поиске новых файлов `crontab`. Файлы `crontab` содержат информацию об определенных функциях, которые `cron` выполняет в указанное время.

Утилита `cron` использует два разных типа конфигурационных файлов, системный и пользовательский. Все различие между этими двумя форматами заключается в шестом поле. В системном файле шестое поле это имя пользователя, с правами которого будет запущена команда. Это позволяет запускать команды из системного `crontab` от любого пользователя. В пользовательском файле шестое поле указывает запускаемую команду, и все команды запускаются от пользователя, который создал `crontab`; это важно для безопасности.

**Замечание:** Пользовательские `crontab` позволяют индивидуальным пользователям планировать задачи без привилегий суперпользователя (`root`). Команды из `crontab` пользователя запускаются с привилегиями этого пользователя.

Пользователь `root` может использовать собственный `crontab`, как и любой другой пользователь. Он будет отличаться от системного `crontab` `/etc/crontab`. Поскольку существует системный `crontab`, обычно не требуется создавать пользовательский `crontab` для `root`.

Давайте заглянем в файл `/etc/crontab` (системный `crontab`):

```
# /etc/crontab - root's crontab for FreeBSD
#
# $FreeBSD: src/etc/crontab,v 1.32 2002/11/22 16:13:39 tom Exp $
# ❶
#
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin ❷
HOME=/var/log
#
#
#minute hour mday month wday who command ❸
#
#
*/5 * * * * root /usr/libexec/atrun ❹
```

- ❶ Как и в большинстве файлов настройки FreeBSD, символы `"#"` означают комментарии. Комментарии нужны для напоминания о том, что означает строка и зачем она добавлена. Комментарии не могут находиться на той же строке, что и команда, или они будут восприняты как часть команды; располагайте их на новой строке. Пустые строки игнорируются.
- ❷ Сначала должны быть заданы переменные окружения. Знак равно (=) используется для задания переменных окружения, в этом примере `SHELL`, `PATH`, и `HOME`. Если переменная для оболочки не задана, `cron` использует оболочку по умолчанию, `sh`. Если не задана переменная `PATH`, значение по умолчанию не устанавливается и пути к файлам должны быть полными. Если не задана переменная `HOME`, `cron` будет использовать домашний каталог соответствующего пользователя.
- ❸ В строке всего семь полей. Их значения `minute`, `hour`, `mday`, `month`, `wday`, `who` (кто), и `command`. Значение полей почти очевидно. `minute` это время в минутах, когда будет запущена команда. `hour` означает то же самое для часов. `mday` означает день месяца. `month`, это то же самое, что час и минута, но для месяцев. Параметр `wday` это день недели. Все эти поля должны быть в числовом формате, время в двадцатичетырехчасовом исчислении. Поле `who` имеет специальное значение, и присутствует только в файле `/etc/crontab`. Это поле определяет пользователя, с правами которого должна быть запущена команда. Когда пользователь устанавливает собственный файл `crontab`, он не указывает этот параметр. Последний параметр `command`. Он указывает команду, которая должна быть запущена.
- ❹ Последняя строка определяет параметры, описанные выше. Здесь задано значение `*/5`, и несколько символов `*`. Эти символы `*` означают "первый-последний", и могут быть интерпретированы как *каждый*. Таким образом, для этой строки соответствующая команда `atrun` вызывается под пользователем `root` каждые пять минут независимо от дня или месяца. За дополнительной информацией по команде `atrun` обращайтесь к странице справочника `atrun(8)`.  
  
Команды могут принимать любое количество параметров; однако команды, состоящие из нескольких строк, должны быть объединены символом `"\"`.

Этот формат одинаков для каждого файла `crontab`, за исключением одной детали. Шестое поле, где указано имя пользователя, присутствует только в файле `/etc/crontab`. Это поле должно быть исключено из `crontab` файлов пользователей.

### 11.6.1. Установка `crontab`

**Важно:** Вы не должны использовать процедуру, описанную здесь, для установки системного `crontab`. Просто используйте свой любимый текстовый редактор: утилита `cron` узнает о том, что файл изменился и сразу начнет использовать обновленную версию. Обратитесь к этой части FAQ ([http://www.FreeBSD.org/doc/ru\\_RU.KOI8-R/books/faq/admin.html#ROOT-NOT-FOUND-CRON-ERRORS](http://www.FreeBSD.org/doc/ru_RU.KOI8-R/books/faq/admin.html#ROOT-NOT-FOUND-CRON-ERRORS)) за дальнейшей информацией.

Для установки готового `crontab` пользователя, сначала создайте в вашем любимом редакторе файл соответствующего формата, а затем воспользуйтесь утилитой `crontab`. Обычно она запускается так:

```
% crontab crontab-file
```

В этом примере, `crontab-file` это имя файла `crontab`, который только что был создан.

Существует также параметр для просмотра установленных файлов crontab: задайте crontab параметр -l.

Для пользователей, составляющих crontab вручную, без временного файла, существует параметр crontab -e. Она вызовет редактор с пустым файлом. Когда файл будет сохранен, crontab автоматически установит его.

Если позднее вы захотите полностью удалить свой crontab, используйте crontab с параметром -r.

## 11.7. Использование rc в FreeBSD 5.X

*Предоставил Tom Rhodes.*

В FreeBSD недавно была интегрирована из NetBSD система rc.d, используемая для старта системы. Многие из файлов в каталоге /etc/rc.d предназначены для основных сервисов, они могут управляться параметрами start, stop, и restart. Например, sshd(8) может быть перезапущен следующей командой:

```
# /etc/rc.d/sshd restart
```

Эта процедура похожа для других сервисов. Конечно, сервисы обычно запускаются автоматически, как указано в rc.conf(5). Например, включение демона Network Address Translation при запуске выполняется простым добавлением следующей строки в /etc/rc.conf:

```
natd_enable="YES"
```

Если natd\_enable="NO" уже присутствует, просто измените NO на YES. Скрипты rc автоматически загрузят все другие зависимые сервисы, как описано ниже.

Поскольку система rc.d в основном предназначена для запуска/отключения сервисов во время запуска/отключения системы, стандартные параметры start, stop и restart будут работать только если установлена соответствующая переменная в /etc/rc.conf. Например, команда выше sshd restart будет работать только если переменная sshd\_enable в файле /etc/rc.conf установлена в YES. Для выполнения скриптов независимо от установок в /etc/rc.conf, параметры start, stop или restart необходимо задавать с префиксом 'force'. Например, для перезапуска sshd независимо от установок в /etc/rc.conf, выполните следующую команду:

```
# /etc/rc.d/sshd forcerestart
```

Проверить состояние переменной в файле /etc/rc.conf легко: запустите соответствующий скрипт из rc.d с параметром rcvar. Проверка переменной для sshd выполняется следующей командой:

```
# /etc/rc.d/sshd rcvar
# sshd
$sshd_enable=YES
```

**Замечание:** Вторая строка (# sshd) это вывод команды sshd, а не консоль root.

Чтобы определить, запущен ли сервис, существует параметр `status`. Например для проверки того, запущен ли `sshd`, выполните:

```
# /etc/rc.d/sshd status
sshd is running as pid 433.
```

Возможна также перегрузка (`reload`) сервиса. Скрипт, запущенный с этим параметром, попытается отправить сервису сигнал, вызывающий перезагрузку файлов настройки. В большинстве случаев это означает отправку сервису сигнала `SIGHUP`.

Структура **rcNG** используется не только для сетевых серверов, она отвечает также за большую часть инициализации системы. Рассмотрим, к примеру, файл `bgfsck`. Во время выполнения этот скрипт выводит следующее сообщение:

```
Starting background file system checks in 60 seconds.
```

Следовательно, этот файл используется для фоновой проверки файловых систем, которая выполняется только в процессе инициализации системы.

Функционирование многих сервисов системы зависит от корректной работы других сервисов. Например, `NIS` и другие основанные на `RPC` сервисы могут не запуститься, пока не загрузится `rpcbind` (`portmapper`). Для разрешения этой проблемы, в начале каждого скрипта в комментарии включаются информация о зависимостях и другие метаданные. Программа `rcorder(8)` для разбора этих комментариев во время старта системы для определения порядка, в котором должны вызываться системные сервисы в соответствии с зависимостями. В начало каждого стартового файла должны быть включены следующие строки:

- **PROVIDE:** Задаёт имя сервиса, предоставляемого этим файлом.
- **REQUIRE:** Список сервисов, необходимых этому сервису. Этот файл будет запущен *после* указанных сервисов.
- **BEFORE:** Список сервисов, зависящих от этого сервиса. Этот файл будет запущен *до* указанных сервисов.
- **KEYWORD:** `FreeBSD` или `NetBSD`. Используется для функций, зависящих от версии `*BSD`.

Используя этот метод, администратор может легко контролировать системные сервисы без использования "уровней запуска", как в некоторых других операционных системах `UNIX`.

Дополнительную информацию о системе `rc.d` `FreeBSD 5.X` можно найти на страницах справочника `rc(8)` и `and rc.subr(8)`.

## 11.8. Настройка карт сетевых интерфейсов

*Предоставил Marc Fonvieille.*

В наши дни мы не представляем себе компьютера без сетевого подключения. Добавление и настройка сетевой карты это обычная задача любого администратора `FreeBSD`.



### 11.8.1. Поиск подходящего драйвера

В первую очередь определите тип используемой карты (PCI или ISA), модель карты и используемый в ней чип. FreeBSD поддерживает многие PCI и ISA карты. Обратитесь к Списку поддерживаемого оборудования вашего релиза чтобы узнать, поддерживается ли карта.

Как только вы убедились, что карта поддерживается, потребуется определить подходящий драйвер. В файле `/usr/src/sys/i386/conf/LINT` находится список драйверов сетевых интерфейсов с информацией о поддерживаемых чипсетах/картах. Если вы сомневаетесь в том, какой драйвер подойдет, прочтите страницу справочника к драйверу. Страница справочника содержит больше информации о поддерживаемом оборудовании и даже о проблемах, которые могут возникнуть.

Если ваша карта широко распространена, вам скорее всего не потребуется долго искать драйвер. Драйверы для широко распространенных карт представлены в ядре `GENERIC`, так что ваша карта должна определиться при загрузке, примерно так:

```
dc0: <82c169 PNIC 10/100BaseTX> port 0xa000-0xa0ff mem 0xd3800000-0xd38000ff irq 15 at device 11.0 on pci0
dc0: Ethernet address: 00:a0:cc:da:da:da
miibus0: <MII bus> on dc0
ukphy0: <Generic IEEE 802.3u media interface> on miibus0
ukphy0: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
dc1: <82c169 PNIC 10/100BaseTX> port 0x9800-0x98ff mem 0xd3000000-0xd30000ff irq 11 at device 12.0 on pci0
dc1: Ethernet address: 00:a0:cc:da:da:db
miibus1: <MII bus> on dc1
ukphy1: <Generic IEEE 802.3u media interface> on miibus1
ukphy1: 10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
```

В этом примере две карты используют имеющийся в системе драйвер `dc(4)`.

Для использования сетевой карты потребуется загрузить подходящий драйвер. Это можно сделать двумя способами. Самый простой способ это загрузка модуля ядра для сетевой карты с помощью `kldload(8)`. Не для каждой сетевой карты есть модуль (например ISA карты и карты, использующие драйвер `ed(4)`). В качестве альтернативы, вы можете статически добавить поддержку сетевой карты в ядро. Проверьте `/usr/src/sys/i386/conf/LINT` и страницу справочника драйвера, чтобы узнать, что добавить в файл конфигурации ядра. За дополнительной информацией о пересборке ядра обращайтесь к Гл. 8. Если ваша карта была обнаружена ядром (`GENERIC`) во время загрузки, собирать новое ядро не требуется.

### 11.8.2. Настройка сетевой карты

Как только для сетевой карты загружен подходящий драйвер, ее потребуется настроить. Как и многое другое, сетевая карта может быть настроена во время установки с помощью **`sysinstall`**.

Для вывода информации о настройке сетевых интерфейсов системы, введите следующую команду:

```
% ifconfig
dc0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet 192.168.1.3 netmask 0xfffff00 broadcast 192.168.1.255
    ether 00:a0:cc:da:da:da
    media: Ethernet autoselect (100baseTX <full-duplex>)
    status: active
```

```

dc1: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet 10.0.0.1 netmask 0xffffffff00 broadcast 10.0.0.255
    ether 00:a0:cc:da:da:db
    media: Ethernet 10baseT/UTP
    status: no carrier
lp0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    inet 127.0.0.1 netmask 0xff000000
tun0: flags=8010<POINTOPOINT,MULTICAST> mtu 1500

```

**Замечание:** Старые версии FreeBSD могут потребовать запуска `ifconfig(8)` с параметром `-a`, за более подробным описанием синтаксиса `ifconfig(8)` обращайтесь к странице справочника. Учтите также, что строки, относящиеся к IPv6 (`inet6` и т.п.) убраны из этого примера.

В этом примере были показаны следующие устройства:

- `dc0`: первый Ethernet интерфейс
- `dc1`: второй Ethernet интерфейс
- `lp0`: интерфейс параллельного порта
- `lo0`: устройство `loopback`
- `tun0`: туннельное устройство, используемое **ppp**

Для присвоения имени сетевой карте FreeBSD использует имя драйвера и порядковый номер, в котором карта обнаруживается при инициализации устройств. Например, `sis2` это третья сетевая карта, использующая драйвер `sis(4)`.

В этом примере, устройство `dc0` включено и работает. Ключевые признаки таковы:

1. UP означает, что карта настроена и готова.
2. У карты есть интернет (`inet`) адрес (в данном случае `192.168.1.3`).
3. Установлена маска подсети (`netmask`; `0xffffffff00`, то же, что и `255.255.255.0`).
4. Широковещательный адрес (в данном случае, `192.168.1.255`).
5. Значение MAC адреса карты (`ether`) `00:a0:cc:da:da:da`
6. Выбор физической среды передачи данных в режиме автовыбора (`media: Ethernet autoselect (100baseTX <full-duplex>)`). Мы видим, что `dc1` была настроена для работы с `10baseT/UTP`. За более подробной информацией о доступных драйверу типах среды обращайтесь к странице справочника.
7. Статус соединения (`status`) `active`, т.е. несущая обнаружена. Для `dc1`, мы видим `status: no carrier`. Это нормально, когда Ethernet кабель не подключен к карте.

Если `ifconfig(8)` показывает примерно следующее:

```

dc0: flags=8843<BROADCAST,SIMPLEX,MULTICAST> mtu 1500
    ether 00:a0:cc:da:da:da

```

это означает, что карта не была настроена.

Для настройки карты вам потребуются привилегии пользователя `root`. Настройка сетевой карты может быть выполнена из командной строки с помощью `ifconfig g(8)`, но вам потребуется делать это после каждой перезагрузки системы. Подходящее место для настройки сетевых карт это файл `/etc/rc.conf`.

Откройте `/etc/rc.conf` в текстовом редакторе. Вам потребуется добавить строку для каждой сетевой карты, имеющейся в системе, например, в нашем случае, было добавлено две строки:

```
ifconfig_dc0="inet 192.168.1.3 netmask 255.255.255.0"
ifconfig_dc1="inet 10.0.0.1 netmask 255.255.255.0 media 10baseT/UTP"
```

Замените `dc0`, `dc1`, и так далее на соответствующие имена ваших карт, подставьте соответствующие адреса. Обратитесь к страницам справочника сетевой карты и `ifconfig g(8)`, за подробной информацией о доступных опциях и к странице справочника `rc.conf(5)` за дополнительной информацией о синтаксисе `/etc/rc.conf`.

Если вы настроили сетевую карту в процессе установки системы, некоторые строки, касающиеся сетевой карты, могут уже присутствовать. Внимательно проверьте `/etc/rc.conf` перед добавлением каких-либо строк.

Отредактируйте также файл `/etc/hosts` для добавления имен и IP адресов различных компьютеров сети, если их еще там нет. За дополнительной информацией обращайтесь к `man.hosts.5`; и к `/usr/share/examples/etc/hosts`.

### 11.8.3. Тестирование и решение проблем

Как только вы внесете необходимые изменения в `/etc/rc.conf`, перезагрузите компьютер. Изменения настроек интерфейсов будут применены, кроме того будет проверена правильность настроек.

Как только система перезагрузится, проверьте сетевые интерфейсы.

#### 11.8.3.1. Проверка Ethernet карты

Для проверки правильности настройки сетевой карты, попробуйте выполнить `ping` для самого интерфейса, а затем для другой машины в локальной сети.

Сначала проверьте локальный интерфейс:

```
% ping -c5 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=0.082 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.074 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.108 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.076 ms

--- 192.168.1.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.074/0.083/0.108/0.013 ms
```

Затем проверьте другую машину в локальной сети:

```
% ping -c5 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=0.726 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.766 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.700 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.747 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.704 ms

--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.700/0.729/0.766/0.025 ms
```

Вы можете также использовать имя машины вместо 192.168.1.2, если настроен файл `/etc/hosts`.

### 11.8.3.2. Решение проблем

Решение проблем с аппаратным и программным обеспечением всегда вызывает сложности, которые можно уменьшить, проверив сначала самые простые варианты. Подключен ли сетевой кабель?

Правильно ли настроены сетевые сервисы? Правильно ли настроен брандмауэр? Поддерживается ли используемая карта в FreeBSD? Всегда проверяйте информацию об оборудовании перед отправкой сообщения об ошибке. Обновите FreeBSD до последней версии STABLE. Просмотрите архивы списков рассылки, или поищите информацию в интернет.

Если карта работает, но производительность низка, может помочь чтение страницы справочника `tuning(7)`. Проверьте также настройки сети, поскольку неправильные настройки могут стать причиной низкой скорости соединения.

Некоторые пользователи встречаются с несколькими `device timeouts`, что нормально для некоторых сетевых карт. Если это продолжается и надоедает, убедитесь, что устройство не конфликтует с другим устройством. Внимательно проверьте подключение кабеля. Возможно также, что вам просто надо установить другую карту.

Время от времени, пользователи видят несколько ошибок `watchdog timeout`. Первое, что требуется сделать, это проверить сетевой кабель. Многие карты требуют поддержки Bus Mastering слотом PCI. На некоторых старых материнских платах, только один PCI слот имеет такую поддержку (обычно слот 0). Сверьтесь с документацией на сетевую карту и материнскую плату, чтобы определить, может ли это быть проблемой.

Сообщение `No route to host` появляются, если система не в состоянии доставить пакеты к хосту назначения. Это может случиться, если не определен маршрут по умолчанию, или кабель не подключен. Проверьте вывод команды `netstat -rn` и убедитесь, что к соответствующему хосту есть работающий маршрут. Если это не так, прочтите Гл. 24.

Сообщения `ping: sendto: Permission denied` зачастую появляются при неправильно настроенном брандмауэре. Если `ipfw` включен в ядре, но правила не определены, правило по умолчанию блокирует весь трафик, даже запросы `ping`! Прочтите Разд. 14.9 с более подробной информацией.

Иногда карты недостаточна, или ниже среднего. В этих случаях лучше всего изменить режим выбора типа подключения с `autoselect` на правильный тип. Обычно это работает для большинства оборудования, но не может решить проблему во всех случаях. Проверьте еще раз настройки сети и прочтите страницу руководства `tuning(7)`.

## 11.9. Настройка виртуальных серверов

Очень часто FreeBSD используется для размещения сайтов, когда один сервер работает в сети как несколько серверов. Это достигается присвоением нескольких сетевых адресов одному интерфейсу.

У сетевого интерфейса всегда есть один "настоящий" адрес, хотя он может иметь любое количество "синонимов" (alias). Эти синонимы обычно добавляются путём помещения соответствующих записей в `/etc/rc.conf`.

Синоним для интерфейса `fxp0` выглядит следующим образом:

```
ifconfig_fxp0_alias0="inet xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx"
```

Заметьте, что записи синонимов должны начинаться с `alias0` и идти далее в определенном порядке (например, `_alias1`, `_alias2`, и т.д.). Конфигурационный процесс остановится на первом по порядку отсутствующем числе.

Определение маски подсети для синонима очень важно, но к счастью, так же просто. Для каждого интерфейса должен быть один адрес с истинной маской подсети. Любой другой адрес в сети должен иметь маску подсети, состоящую из всех единиц (что выражается как `255.255.255.255` или как `0xffffffff`).

Например, рассмотрим случай, когда интерфейс `fxp0` подключён к двум сетям, к сети `10.1.1.0` с маской подсети `255.255.255.0` и к сети `202.0.75.16` с маской `255.255.255.240`. Мы хотим, чтобы система была видна по IP, начиная с `10.1.1.1` по `10.1.1.5` и с `202.0.75.17` по `202.0.75.20`. Как было сказано выше, только первый адрес в заданном диапазоне (в данном случае, `10.1.1.1` и `202.0.75.17`) должен иметь реальную маску сети; все остальные (с `10.1.1.2` по `10.1.1.5` и с `202.0.75.18` по `202.0.75.20`) должны быть сконфигурированы с маской сети `255.255.255.255`.

Для этого должны быть внесены следующие записи:

```
ifconfig_fxp0="inet 10.1.1.1 netmask 255.255.255.0"
ifconfig_fxp0_alias0="inet 10.1.1.2 netmask 255.255.255.255"
ifconfig_fxp0_alias1="inet 10.1.1.3 netmask 255.255.255.255"
ifconfig_fxp0_alias2="inet 10.1.1.4 netmask 255.255.255.255"
ifconfig_fxp0_alias3="inet 10.1.1.5 netmask 255.255.255.255"
ifconfig_fxp0_alias4="inet 202.0.75.17 netmask 255.255.255.240"
ifconfig_fxp0_alias5="inet 202.0.75.18 netmask 255.255.255.255"
ifconfig_fxp0_alias6="inet 202.0.75.19 netmask 255.255.255.255"
ifconfig_fxp0_alias7="inet 202.0.75.20 netmask 255.255.255.255"
```

## 11.10. Файлы настройки

### 11.10.1. Каталог `/etc`

Во FreeBSD определён ряд директорий, предназначенных для хранения конфигурационных файлов. Это:

<code>/etc</code>	Основные файлы конфигурации системы. Тут размещены системно-зависимые данные.
<code>/etc/defaults</code>	Версии системных конфигурационных файлов по умолчанию.

<code>/etc/mail</code>	Дополнительные конфигурационные файлы <code>sendmail(8)</code> остальные конфигурационные файлы МТА.
<code>/etc/ppp</code>	Настройка для <code>user-</code> и <code>kernel-ppp</code> программ.
<code>/etc/namedb</code>	Основное место расположения данных <code>named(8)</code> . Обычно <code>named.conf</code> и файлы зон расположены здесь.
<code>/usr/local/etc</code>	Конфигурационные файлы установленных приложений. Могут содержать подкаталоги приложений.
<code>/usr/local/etc/rc.d</code>	Скрипты запуска/остановки установленных приложений.
<code>/var/db</code>	Автоматически генерируемые системно-специфичные файлы баз данных, такие как база данных пакетов, и так далее

## 11.10.2. Имена хостов

### 11.10.2.1. `/etc/resolv.conf`

`/etc/resolv.conf` определяет, как резолвер (resolver) FreeBSD получает доступ к Системе Доменных Имен (DNS).

Основные записи `resolv.conf`:

<code>nameserver</code>	IP адрес сервера имён. Сервера опрашиваются в порядке описания. Максимальное количество адресов - три.
<code>search</code>	Список доменов для поиска с помощью <code>hostname lookup</code> . Обычно определяется доменом, в котором находится компьютер.
<code>domain</code>	Домен, в котором находится компьютер.

Типичный вид `resolv.conf`:

```
search example.com
nameserver 147.11.1.11
nameserver 147.11.100.30
```

**Замечание:** Опции `search` и `domain` нельзя использовать совместно.

Если вы используете DHCP, `dhclient(8)` обычно перезаписывает `resolv.conf` информацией, полученной от серверов DHCP.

### 11.10.2.2. `/etc/hosts`

`/etc/hosts` - простая текстовая база данных, напоминающая старый Интернет. Она работает совместно с DNS и NIS, сопоставляя доменные имена IP адресу. Отдельные компьютеры, соединённые с помощью локальной сети могут быть записаны тут вместо `named(8)` сервера с целью упрощения. Кроме

того, `/etc/hosts` используется для записи IP адресов и соответствующих им доменов, избавляя от внешнего трафика, используемого для запросов к DNS серверам.

```
# $FreeBSD$
#
# Host Database
# This file should contain the addresses and aliases
# for local hosts that share this file.
# In the presence of the domain name service or NIS, this file may
# not be consulted at all; see /etc/nsswitch.conf for the resolution order.
#
#
::1                localhost localhost.my.domain myname.my.domain
127.0.0.1          localhost localhost.my.domain myname.my.domain

#
# Imaginary network.
#10.0.0.2          myname.my.domain myname
#10.0.0.3          myfriend.my.domain myfriend
#
# According to RFC 1918, you can use the following IP networks for
# private nets which will never be connected to the Internet:
#
#      10.0.0.0      -   10.255.255.255
#      172.16.0.0    -   172.31.255.255
#      192.168.0.0   -   192.168.255.255
#
# In case you want to be able to connect to the Internet, you need
# real official assigned numbers. PLEASE PLEASE PLEASE do not try
# to invent your own network numbers but instead get one from your
# network provider (if any) or from the Internet Registry (ftp to
# rs.internic.net, directory '/templates').
#
```

Формат `/etc/hosts`:

```
[IP адрес в Интернете] [имя компьютера] [alias1] [alias2] ...
```

Например:

```
10.0.0.1 myRealHostname.example.com myRealHostname foobar1 foobar2
```

За дополнительной информацией обращайтесь к `hosts(5)`.

### 11.10.3. Настройка лог файлов

#### 11.10.3.1. `syslog.conf`

`syslog.conf` является файлом конфигурации для `syslogd(8)`. В нём указываются, типы сообщений генерируемые `syslog`, и лог файлы, в которые они записываются.

```

# $FreeBSD$
#
#      Spaces ARE valid field separators in this file. However,
#      other *nix-like systems still insist on using tabs as field
#      separators. If you are sharing this file between systems, you
#      may want to use only tabs as field separators here.
#      Consult the syslog.conf(5) manual page.
*.err;kern.debug;auth.notice;mail.crit      /dev/console
*.notice;kern.debug;lpr.info;mail.crit;news.err /var/log/messages
security.*                                   /var/log/security
mail.info                                    /var/log/maillog
lpr.info                                     /var/log/lpd-errs
cron.*                                       /var/log/cron
*.err                                        root
*.notice;news.err                          root
*.alert                                     root
*.emerg                                     *
# uncomment this to log all writes to /dev/console to /var/log/console.log
#console.info                               /var/log/console.log
# uncomment this to enable logging of all log messages to /var/log/all.log
#*. *                                        /var/log/all.log
# uncomment this to enable logging to a remote log host named loghost
#*. *                                        @loghost
# uncomment these if you're running inn
# news.crit                                 /var/log/news/news.crit
# news.err                                  /var/log/news/news.err
# news.notice                              /var/log/news/news.notice
!startslip
*. *                                        /var/log/slip.log
!ppp
*. *                                        /var/log/ppp.log

```

За более полной информацией обратитесь к `syslog.conf(5)`.

### 11.10.3.2. newsyslog.conf

`newsyslog.conf` - конфигурационный файл `newsyslog(8)`, программы, обычно контролируемой `cron(8)`. `newsyslog(8)` определяет, когда лог-файлы нуждаются в архивировании и перегруппировке. `logfile` перемещается в `logfile.0`, `logfile.0` перемещается в `logfile.1`, и так далее. Другое именование получится при архивировании с помощью `gzip(1)`: `logfile.0.gz`, `logfile.1.gz`, и т.д.

`newsyslog.conf` показывает, какие лог файлы должны быть проинспектированы, сколько их должно быть сохранено, и когда они должны быть пересмотрены. Лог файлы могут быть перегруппированы и/или заархивированы, когда они либо достигнут определённого размера, либо при достижении определённых даты/времени.

```

# configuration file for newsyslog
# $FreeBSD$
#
# filename           [owner:group]      mode count size when [ZB] [/pid_file] [sig_num]
/var/log/cron        600 3      100 *      Z
/var/log/amd.log     644 7      100 *      Z

```



/var/log/kerberos.log	644	7	100	*	Z
/var/log/lpd-errs	644	7	100	*	Z
/var/log/maillog	644	7	*	@T00	Z
/var/log/sendmail.st	644	10	*	168	B
/var/log/messages	644	5	100	*	Z
/var/log/all.log	600	7	*	@T00	Z
/var/log/slip.log	600	3	100	*	Z
/var/log/ppp.log	600	3	100	*	Z
/var/log/security	600	10	100	*	Z
/var/log/wtmp	644	3	*	@01T05	B
/var/log/daily.log	640	7	*	@T00	Z
/var/log/weekly.log	640	5	1	\$W6D0	Z
/var/log/monthly.log	640	12	*	\$M1D0	Z
/var/log/console.log	640	5	100	*	Z

За дополнительной информацией обращайтесь к `newsyslog(8)`.

#### 11.10.4. `sysctl.conf`

`sysctl.conf` очень похож на `rc.conf`. Значения устанавливаются в виде `variable=value`. Указанные значения устанавливаются после перевода системы в многопользовательский режим. Однако не все переменные могут быть установлены в этом режиме.

Пример `sysctl.conf` настроенной для выключения нотирования фатальных ошибок и разрешения Linux-программам определять, что они запускаются под FreeBSD:

```
kern.logsigexit=0      # Do not log fatal signal exits (e.g. sig 11)
compat.linux.osname=FreeBSD
compat.linux.osrelease=4.3-STABLE
```

### 11.11. Настройка с помощью `sysctl`

`sysctl(8)` - это интерфейс, позволяющий вам вносить изменения в работающую систему FreeBSD. Эти изменения могут касаться многих опций стека TCP/IP и виртуальной памяти и могут облегчить опытному администратору жизнь. Более пяти тысяч системных переменных могут быть прочитаны и записаны с помощью `sysctl(8)`.

По своей сути, `sysctl(8)` выполняет две функции: чтение и изменение настроек системы.

Для просмотра всех доступных для чтения переменных::

```
% sysctl -a
```

Чтобы прочесть определённую переменную, например, `kern.maxproc`, введите:

```
% sysctl kern.maxproc
kern.maxproc: 1044
```

Для присвоения значения переменной, используйте выражение вида *переменная=значение*:

```
# sysctl kern.maxfiles=5000
kern.maxfiles: 2088 -> 5000
```

Изменяемые с помощью `sysctl` переменные обычно принимают значения либо строкового, либо целого, либо булевого типа. Переменные булевого типа могут принимать два значения (1 (истина) и 0 (ложь)).

Если вы хотите устанавливать некоторые переменные автоматически при каждой загрузке компьютера, добавьте их в файл `/etc/sysctl.conf`. За дополнительной информацией обращайтесь к странице справочника `sysctl.conf(5)` и к Разд. 11.10.4.

### 11.11.1. Переменные `sysctl(8)` только для чтения

*Предоставил Tom Rhodes.*

В некоторых случаях желательно изменить переменные `sysctl(8)` только для чтения. Хотя это не рекомендуется, иногда другого способа решить проблему нет.

Например, на некоторых моделях ноутбуков диапазон памяти устройства `cardbus(4)` не определяется и выдается приблизительно такая ошибка:

```
cbb0: Could not map register memory
device_probe_and_attach: cbb0 attach returned 12
```

Ситуации, похожие на эту, требуют изменения некоторых значений `sysctl(8)`, модификация которых запрещена. Для разрешения этой ситуации пользователь может поместить `sysctl(8)` "OID" в файл `/boot/loader.conf`. Значения по умолчанию хранятся в файле `/boot/defaults/loader.conf`.

Решение проблемы, приведенной выше, потребует помещения строки

`hw.pci.allow_unsupported_io_range=1` в вышеупомянутый файл. Теперь `cardbus(4)` будет работать нормально.

## 11.12. Оптимизация дисков

### 11.12.1. Переменные `Sysctl`

#### 11.12.1.1. `vfs.vmiodirenable`

Значением переменной `vfs.vmiodirenable` может быть установлено в 0 (выключено) или 1 (включено); по умолчанию 1. Эта переменная отвечает за метод кэширования каталогов. Размер большинства каталогов невелик. Они могут поместиться в одном фрагменте (обычно 1К), и могут занимать ещё меньше места (обычно 512 байт) в кэше буфера. При отключении этой переменной (при установке значения 0) буфер прокэширует только заданное число каталогов даже если у вас много памяти. При включении (при установке значения 1) эта переменная `sysctl` позволит использовать страничное кэширование VM, делая доступным для кэширования каталогов весь объём памяти. Однако, минимальный объём памяти, используемой для кэширования каталогов стал равен объёму страницы (обычно 4 К) вместо 512 байт. Мы рекомендуем оставлять эту опцию включенной, если ваш компьютер исполняет программы, манипулирующие значительным количеством файлов. Примером таких программ могут быть кэширующие прокси-серверы, большие почтовые серверы и серверы

новостей. Обычно включение этой опции не понижает производительности, однако лучше поэкспериментировать, чтобы узнать оптимальное значение для вашей машины.

#### 11.12.1.2. `vfs.write_behind`

Переменная `sysctl vfs.write_behind` по умолчанию установлена в 1 (включено). Она указывает системе выполнять запись на носитель по кластерам, что обычно делается для больших файлов. Идея в том, чтобы избежать заполнения кэша неполными буферами, когда это не увеличивает производительность. Однако, это может заблокировать процессы и в некоторых случаях вам может понадобиться отключить этот параметр.

#### 11.12.1.3. `vfs.hirunningspace`

Переменная `sysctl vfs.hirunningspace` определяет число запросов записи на диск, которые могут быть поставлены в очередь. Значение по умолчанию обычно подходит, но на компьютерах с большим количеством дисков вы можете увеличить его до четырех или пяти *мегабайт*. Учтите, что установка слишком большого значения (превышающего размер буфера записи) может привести к очень значительному падению общей производительности. Не делайте это значение произвольно большим! Большие значения могут привести к задержкам чтения, выполняемого в то же время

Есть много других переменных `sysctl`, относящихся к кэшированию в буфер и страничному кэшированию VM. Мы не рекомендуем изменять эти значения. Начиная FreeBSD 4.3, система VM делает отличную работу по автоматической самонастройке.

#### 11.12.1.4. `vm.swap_idle_enabled`

Переменная `sysctl vm.swap_idle_enabled` полезна в больших многопользовательских системах, где есть много пользователей, входящих и выходящих из системы, и множество ожидающих процессов. Такие системы обычно генерируют большое количество запросов на выделение памяти. Включение этой переменной и настройка задержки выгрузки (`swapout hysteresis`, в секундах) установкой переменных `vm.swap_idle_threshold1` и `vm.swap_idle_threshold2` позволит освобождать страницы памяти, занятые ожидающими процессами, более быстро, чем при нормальном алгоритме выгрузки. Это помогает даемону выгрузки страниц. Не включайте этот параметр, пока он на самом деле вам не понадобится, поскольку его действие в сущности заключается в более ранней выгрузке страниц из памяти; это повышает нагрузку на подкачку и диск. В малых системах эффект от включения этого параметра предсказуем, но в больших системах нагруженной на подкачку этот параметр позволяет системе VM проще загружать и выгружать процессы из памяти.

#### 11.12.1.5. `hw.ata.wc`

Во FreeBSD 4.3 кэширование записи на IDE диски было отключено. Это понижало производительность IDE дисков в тестах, но было необходимо для лучшей сохранности данных. Проблема состоит в том, что IDE диски неправильно указывают время завершения записи на диск. При включенном кэшировании IDE диски могут не только записать данные в неправильном порядке – при большой нагрузке на диск некоторые блоки могут задержаться до бесконечности. Сбой, или отключение питания могут стать причиной серьёзных повреждений в файловой системе. Поэтому для безопасности системы значение по умолчанию этого параметра было изменено. К сожалению,

результатом этого стало столь значительная потеря производительности, что после выхода релиза значение этого параметра было возвращено в первоначальное состояние. Вам следует проверить значение переменной `sysctl hw.ata.ws` на вашей машине. Если кэширование выключено - вы можете включить его, установив значение переменной ядра, равное 1. Это должно быть сделано при помощи загрузчика при загрузке. Если вы сделаете это позже - изменения не будут иметь силы.

За более подробной информацией обращайтесь к `ata(4)`.

#### 11.12.1.6. `SCSI_DELAY` (`kern.cam.scsi_delay`)

Параметр настройки ядра `SCSI_DELAY` может использоваться для уменьшения времени загрузки системы. Значение по умолчанию велико и может составлять более 15 секунд в процессе загрузки. Уменьшение его до 5 секунд обычно работает (особенно с современными дисками). В новых версиях FreeBSD (5.0 и выше) должен использоваться параметр `kern.cam.scsi_delay`, настраиваемый во время загрузки. Этот параметр и параметр настройки ядра принимают значения в *миллисекундах*, а не в *секундах*.

### 11.12.2. Soft Updates

Программа `tunefs(8)` используется для настройки файловой системы. Эта программа может принимать большое количество параметров, но мы рассмотрим лишь один из них - включение и выключение Soft Updates, что может быть достигнуто следующим образом:

```
# tunefs -n enable /filesystem
# tunefs -n disable /filesystem
```

Нельзя изменять файловую систему с помощью `tunefs(8)` когда она смонтирована. Самое подходящее время для включения "Soft Updates" - перед монтированием разделов, в однопользовательском режиме.

**Замечание:** Начиная с FreeBSD 4.5, можно включить Soft Updates во время создания файловой системы, используя `newfs(8)` с параметром `-U`.

Soft Updates существенно увеличивают скорость создания и удаления файлов путём использования кэширования. Мы рекомендуем использовать Soft Updates на всех ваших файловых системах. Однако у Soft Updates есть и обратные стороны: во-первых, Soft Updates гарантирует целостность файловой системы в случае сбоя, но может наблюдаться задержка в несколько секунд (или даже минуту!) перед записью на жесткий диск. Если система зависнет - вы можете потерять больше, чем, если бы вы не включили Soft Updates. Во-вторых, Soft Updates задерживает освобождение блоков файловой системы. Если ваша файловая система заполнена, выполнение значительного обновления, например, `make installworld`, может вызвать переполнение.

#### 11.12.2.1. Дополнительная информация о Soft Updates

Есть два традиционных способа записи метаданных файловых систем на диск. (Пример метаданных: индексные дескрипторы и каталоги.)

Исторически, поведение по умолчанию заключается в синхронном обновлении метаданных. Если каталог был изменен, система ждет, пока изменение не будет физически записано на диск. Содержимое файлов проходит через кэш и записывается на диск асинхронно. Преимущество этого способа в его надежности. При сбое во время обновления метаданные остаются в нормальном состоянии. Файл либо создается целиком, либо вообще не создается. Если блоки данных не были записаны в файл из буфера во время сбоя, `fsck(8)` сможет определить это и восстановить файловую систему, установив длину файла в 0. Кроме того, реализация этого способа проста и понятна. Недостаток в том, что обновление метаданных занимает много времени. Команда `rm -r`, например, последовательно удаляет все файлы в каталоге, и каждое изменение в каталоге (удаление файла) будет синхронно записано на диск. Сюда включаются обновления самого каталога, таблицы индексных дескрипторов, и возможно блоков, занятых файлом. Те же соглашения работают при распаковке больших иерархий (`tar -x`).

Другой вариант это асинхронное обновление метаданных. Это поведение по умолчанию для `Linux/ext2fs` и `*BSD ufs` с параметром `mount -o async`. Все обновления метаданных просто пропускаются через кэш буфера, как и содержимое файлов. Преимущество этой реализации в том, что нет необходимости ждать каждый раз, пока метаданные будут записаны на диск, поэтому все операции с большим объемом обновления метаданных будут происходить гораздо быстрее чем при синхронном обновлении. Кроме того, реализация все еще проста и понятна, поэтому риск появления ошибок в коде невелик. Недостаток в том, что нет никаких гарантий исправности файловой системы. Если во время во время обновления большого объема метаданных произойдет сбой (например, отключение питания, или нажатие кнопки `reset`), файловая система останется в непредсказуемом состоянии. Нет возможности определить состояние файловой системы после такого сбоя; блоки данных файла могут быть уже записаны на диск, а обновления таблицы индексных дескрипторов нет. Невозможно реализовать `fsck`, которая могла бы исправить получившийся хаос (поскольку необходимой информации нет на диске). Если файловая система была уничтожена во время восстановления, единственный способ восстановления — запустить `newfs(8)` и воспользоваться резервной копией.

Обычное решение этой проблемы состояло в реализации *протоколировании проблемной области* (*dirty region logging*), известном как *журналирование*, хотя этот термин использовался неправильно и порой также применялся к другим формам протоколирования транзакций. Обновление метаданных как и прежде происходит синхронно, но в отдельную область диска. Позже они перемещаются туда, где должны быть. Поскольку область протоколирования это небольшая, последовательная область диска, головкам жесткого диска не приходится перемещаться на большие расстояния даже во время значительных обновлений, поэтому такой способ быстрее, чем синхронные обновления. Кроме того, сложность реализации довольно ограничена, поэтому риск внесения ошибок невелик. Недостаток в том, что все обновления метаданных записываются дважды (один раз в область протоколирования и один раз окончательно), поэтому при обычной работе производительность может понизиться. С другой стороны, в случае сбоя все незаконченные действия с метаданными могут быть быстро отменены, или завершены после загрузки системы, поэтому система после сбоя загружается быстрее.

Kirk McKusick, разработчик Berkeley FFS, решил эту проблему с помощью Soft Updates: все незавершенные обновления метаданных находятся в памяти и записываются на диск в упорядоченном виде ('упорядоченное обновления метаданных'). При значительных обновлениях метаданных более поздние обновления 'присоединяются' к предыдущим, если они все еще находятся в памяти и еще не записаны на диск. Поэтому все операции, скажем, над каталогом, обычно выполняются в памяти перед записью обновления на диск (блоки данных сортируются в соответствии с их положением, так что они не будут записаны на диск до метаданных. При крахе операционной системы выполняется 'откат': считается, что все операции, не записанные на диск, никогда не происходили. Файловая система находится в том состоянии, в котором она была за 30–60 секунд до сбоя. Используемый алгоритм гарантирует, что все используемые ресурсы маркированы соответствующим образом в своих областях:

блоки и индексные дескрипторы. После сбоя могут остаться только ошибки, выделения ресурсов, они помечаются как "используемые", хотя на самом деле "свободны". `fsck(8)` разбирается в ситуации и освобождает более не используемые ресурсы. После сбоя система может быть безопасно смонтирована с опцией `mount -f`. Для освобождения ресурсов, которые могут не использоваться, в дальнейшем потребуется запустить `fsck(8)`. Эта идея лежит в основе *background (фоновая) fsck*: во время запуска системы записывается только *снимок* файловой системы. Все системы могут быть смонтированы в "грязном" состоянии, и система загружается в многопользовательский режим. Затем, фоновые `fsck` ставятся в очередь для всех систем, где это требуется, чтобы освободить неиспользуемые ресурсы. (Файловые системы, где не используются Soft Updates, все еще требуют запуска `fsck` в обычном режиме).

Преимущество этого способа в том, что обновления метаданных происходят почти так же быстро, как при асинхронных обновлениях (т.е. быстрее, чем при *журналировании*, когда метаданные записываются дважды). Недостаток в сложности кода (подразумевающим больший риск появления ошибок в области, где вероятность потери данных пользователя особенно высока) и в более высоких требованиях к объему памяти. К тому же могут возникнуть некоторые странные на первый взгляд ситуации. После сбоя состояние файловой системы несколько более "старое". В ситуации, когда стандартный способ синхронизации оставит несколько файлов нулевой длины после выполнения `fsck`, в файловой системе с Soft Updates их не останется вовсе, поскольку ни метаданные, ни содержимое файлов не были записаны на диск. Дискосое пространство не будет освобождено пока обновления не будут записаны на диск, что может занять некоторое время после выполнения `rm`. Это может повлечь проблемы при установке большого количества файлов на файловую систему, где не хватает места для помещения всех файлов дважды.

## 11.13. Изменение ограничений, накладываемых ядром

### 11.13.1. Ограничения на Файлы/Процессы

#### 11.13.1.1. `kern.maxfiles`

Значение `kern.maxfiles` может быть увеличено или уменьшено в зависимости от потребностей вашей системы. Эта переменная определяет максимальное число дескрипторов файлов. Когда таблица дескрипторов файлов полна, в очереди системных сообщений появится сообщение `file: table is full`. Это сообщение может быть прочитано с помощью команды `dmesg`.

Каждый открытый файл, сокет или буфер использует дескриптор файла. Широкомасштабному серверу может понадобиться много тысяч дескрипторов файлов, в зависимости от количества программ, одновременно выполняемых на сервере.

Стандартное значение `kern.maxfile` определяется переменной `MAXUSERS` в вашем файле конфигурации ядра. Значение `kern.maxfiles` увеличивается пропорционально значению `MAXUSERS`. При компилировании ядра, нужно установить эту переменную согласно потребностям вашей системы. Исходя из значения этой переменной, ядро устанавливает значения большинства предопределённых переменных. Даже если предполагается, что к компьютеру не будут одновременно подсоединяться 256 пользователей, требуемые ресурсы могут быть такими же, как у крупномасштабного сервера.

**Замечание:** Начиная с FreeBSD 4.5, установка значения `MAXUSERS` в 0 в файле конфигурации ядра выберет подходящее значение по умолчанию, основанное на объеме оперативной памяти системы.

### 11.13.1.2. `kern.ipc.somaxconn`

Переменная `sysctl kern.ipc.somaxconn` ограничивает размер очереди для приема новых TCP соединений. Значение по умолчанию 128 слишком мало для надежной обработки новых соединений для нагруженного web сервера. Для такого сервера рекомендуется увеличить это значение до 1024 или выше. Демон сервиса может сам ограничивать очередь приема новых соединений (например, `sendmail(8)`, или **Apache**), но обычно в файле настройки демона есть директива для настройки длины очереди. Более длинная очередь также помогает избежать атак Denial of Service (DoS).

## 11.13.2. Сетевые Ограничения

Опция ядра `NMBCLUSTERS` обуславливает количество Mbuf, доступных на машине. На сервере с большим трафиком и маленьким Mbuf производительность будет пониженной. Каждый кластер представлен двумя килобайтами памяти, поэтому значение 1024 означает 2 мегабайта памяти ядра, зарезервированной для сетевых буферов. Для определения оптимального значения необходимо провести простые вычисления. Если у вас веб сервер, который может обслуживать 1000 одновременных соединений, и каждое соединение "съедает" 16 К буфера приема и 16 К буфера отправки, вам потребуется 32 МВ памяти под буферы. Хорошее правило — умножение этого значения на 2,  $2 \times 32 \text{ МВ} / 2 \text{ КВ} = 64 \text{ МВ} / 2 \text{ КВ} = 32768$ . Мы рекомендуем значения между 4096 и 32768 для машин с большим объемом памяти. Не указывайте произвольно большое значение параметра, это может привести к падению системы при загрузке. Используйте `netstat(1) -m` для определения количества используемых сетевых кластеров.

Для настройки в процессе загрузки используйте в loader переменную `kern.ipc.nmbclusters`. Только в старых версиях FreeBSD потребуется пересобрать ядро (`config(8)`) с измененным параметром `NMBCLUSTERS`.

Для нагруженных серверов, интенсивно использующих системный вызов `sendfile(2)`, может потребоваться увеличения буферов `sendfile(2)` с помощью параметра конфигурации ядра `NSFBUFS`, или изменения значения путем установки переменной в `/boot/loader.conf` (обратитесь к `loader(8)` за подробностями). Общий признак того, что параметр требуется изменить — состояние процессов `sfbufo`. Переменная `sysctl kern.ipc.nsfbufs` установлена только для чтения. Этот параметр увеличивается вместе с `kern.maxusers`, хотя может потребоваться увеличить его отдельно.

**Важно:** Даже если сокет помечен как неблокирующий, вызов `sendfile(2)` на неблокирующем сокете может вызвать блокирование `sendfile(2)`, пока не станет доступным достаточное количество `struct sf_buf`.

### 11.13.2.1. `net.inet.ip.portrange.*`

Переменные `sysctl net.inet.ip.portrange.*` контролируют диапазоны номеров портов, автоматически привязываемых к TCP и UDP сокетам. Есть три диапазона: нижний диапазон, диапазон

по умолчанию и верхний диапазон. Большинство сетевых программ используют диапазон по умолчанию, контролируемый `net.inet.ip.portrange.first` и `net.inet.ip.portrange.last`, установленными соответственно в 1024 и 5000. Диапазоны портов привязки используются исходящих соединений и при некоторых условиях портов может не хватить. Это чаще всего происходит на сильно загруженном прокси сервере. Диапазон портов не становится проблемой при работе серверов, которые обрабатывают в основном входящие соединения, или с небольшим количеством исходящих соединений, например mail relay. Для ситуаций, когда возможен недостаток портов, рекомендуется немного увеличить `net.inet.ip.portrange.last`. Может подойти значение 10000, 20000, или 30000. Учтите также возможное влияние брандмауэра при изменении диапазона портов. Некоторые могут блокировать большие диапазоны портов (обычно с небольшими номерами) и вынуждают использовать более высокие диапазоны для исходящих соединений. По этой причине рекомендуется настроить значение `net.inet.ip.portrange.first`.

### 11.13.2.2. TCP Bandwidth Delay Product

TCP Bandwidth Delay Product Limiting похоже на TCP/Vegas в NetBSD. Оно может быть включено установкой переменной `sysctl net.inet.tcp.inflight_enable` в 1. Система попытается вычислить задержку пакетов для каждого соединения и ограничить объем данных в очереди сети до значения, требуемого для поддержания оптимальной пропускной способности.

Эта возможность полезна при передаче данных через модемы, Gigabit Ethernet, или даже через высокоскоростные WAN соединения (или любые другие соединения с большой задержкой передачи), особенно если вы также используете изменение размера окна или настроили большое окно передачи. Если вы включили этот параметр, убедитесь также, что переменная `net.inet.tcp.inflight_debug` установлена в 0 (отладка выключена), а для использования в реальных может понадобиться установка переменной `net.inet.tcp.inflight_min` к значению как минимум 6144. Но учтите, что установка большого значения этой переменной может фактически отключить ограничение в зависимости от вида соединения. Ограничение уменьшает количество данных на определенном маршруте и управляет очередью пакетов, как и уменьшает общее количество данных в очереди локального интерфейса хоста. С меньшим количеством пакетов в очереди двусторонние интерактивные соединения, особенно на медленных линиях, могут проходить быстрее. Но имейте в виду, что эта функция работает только при передаче данных (передача данных / сторона сервера). Она не работает при получении данных (загрузке).

Изменение значения переменной `net.inet.tcp.inflight_stab` не рекомендуется. Этот параметр по умолчанию равен 20, что означает добавление 2 пакетов к вычислению задержки передачи. Дополнительное окно требуется для стабилизации алгоритма и улучшения ответной реакции на изменение условий, но также приводит к большему времени `ping` на медленных соединениях (задержка все же гораздо меньше, чем без алгоритма `inflight`). Вы можете попробовать уменьшить этот параметр до 15, 10 или 5; а также уменьшить `net.inet.tcp.inflight_min` (например, до 3500) для получения желаемого эффекта. Уменьшение значений этих параметров может использоваться только как крайняя мера.

## 11.14. Увеличение объема подкачки

Вне зависимости от того, что вы планировали, иногда система ведет себя неожиданно. Если вам



потребовался дополнительный объем подкачки, его довольно просто добавить. Есть три способа увеличения объема подкачки: добавить новый жесткий диск, включить подкачку по NFS, или создать файл подкачки на существующем разделе.

### 11.14.1. Подкачка на новом жестком диске

Лучший способ добавить подкачку, конечно, использовать еще один жесткий диск. Вы можете сделать это в любой момент. Если такой способ подходит, прочтите еще раз информацию по пространству подкачки в разделе Разд. 11.2 Руководства, где рассказывается о наилучшем способе организации раздела подкачки.

### 11.14.2. Подкачка через NFS

Подкачка через NFS рекомендуется только в том случае, если в системе отсутствует жесткий диск. Подкачка через NFS медленна и неэффективна в версиях FreeBSD до 4.X. Она довольно быстра и эффективна в 4.0-RELEASE и выше. Но даже в новых версиях FreeBSD, подкачка через NFS ограничена скоростью сетевого подключения и к тому же дополнительно нагружает NFS сервер.

### 11.14.3. Файлы подкачки

Вы можете создать файл определенного размера и использовать его как файл подкачки. В нашем примере будет использован файл `/usr/swap0` размером 64MB. Конечно, вы можете использовать любое имя.

#### Пример 11-1. Создание файла подкачки в FreeBSD 4.X

1. Убедитесь, что ядре включен драйвер `vn`. Он *не*включен в последних версиях GENERIC.  

```
pseudo-device  vn 1  #Vnode driver (turns a file into a device)
```
2. Создайте устройство `vn`:  

```
# cd /dev
# sh MAKEDEV vn0
```
3. Создайте файл подкачки (`/usr/swap0`):  

```
# dd if=/dev/zero of=/usr/swap0 bs=1024k count=64
```
4. Установите подходящие права на (`/usr/swap0`):  

```
# chmod 0600 /usr/swap0
```
5. Включите файл подкачки в `/etc/rc.conf`:  

```
swapfile="/usr/swap0"  # Set to name of swapfile if aux swapfile desired.
```
6. Перезагрузите компьютер, или для включения подкачки прямо сейчас выполните:  

```
# vnconfig -e /dev/vn0b /usr/swap0 swap
```

**Пример 11-2. Создание файла подкачки в FreeBSD 5.X**

1. Убедитесь, что в файле настройки ядра присутствует драйвер виртуального диска (md(4)). Он есть в ядре GENERIC.  
`device md # Memory "disks"`
2. Создайте файл подкачки (/usr/swap0):  
`# dd if=/dev/zero of=/usr/swap0 bs=1024k count=64`
3. Установите подходящие права на (/usr/swap0):  
`# chmod 0600 /usr/swap0`
4. Включите файл подкачки в /etc/rc.conf:  
`swapfile="/usr/swap0" # Set to name of swapfile if aux swapfile desired.`
5. Перегрузите компьютер или для включения подкачки прямо сейчас введите:  
`# mdconfig -a -t vnode -f /usr/swap0 -u 0 && swapon /dev/md0`

**11.15. Управление питанием и ресурсами**

*Написал Hiten Pandya, Tom Rhodes.*

Очень важно использовать аппаратные ресурсы эффективно. До того, как появился ACPI, управление потреблением питания и температурными характеристиками системы было очень сложной для операционной системы задачей. Аппаратное обеспечение контролировалось одним из видов встроенного интерфейса BIOS, таким как: *Plug and Play BIOS (PNPBIOS)*, *Advanced Power Management (APM)* и так далее. Управление питанием и ресурсами это один из ключевых компонентов современной операционной системы. Например, вам может потребоваться, чтобы операционная система следила за температурными ограничениями и возможно, предупреждала при неожиданном росте температуры.

В этом разделе Руководства FreeBSD, мы предоставим исчерпывающую информацию о ACPI. В конце раздела есть ссылки для дальнейшего чтения. Учтите, что ACPI есть только в FreeBSD 5.X и выше в качестве стандартного модуля ядра. В FreeBSD 4.9 ACPI можно включить добавлением строки `device acpi` к файлу настройки ядра и его пересборкой.

**11.15.1. Что такое ACPI?**

Advanced Configuration and Power Interface (ACPI) это стандарт, написанный объединением поставщиков в целях предоставления стандартного интерфейса для аппаратных ресурсов и управления питанием (отсюда и название). Это ключевой элемент *Operating System-directed configuration and Power Management*, т.е.: он предоставляет операционной системе (OS) больше контроля и более универсален. Современные системы вышли за пределы ограничений существующих Plug and Play интерфейсов (таких как APM, использовавшийся в FreeBSD 4.X), до появления ACPI. ACPI это прямой наследник APM (Advanced Power Management).

### 11.15.2. Недостатки Advanced Power Management (APM)

Средства *Advanced Power Management (APM)* управляют энергопотреблением системы в зависимости от нагрузки. APM BIOS предоставляется поставщиком системы и специфичен для данной аппаратной платформы. Драйвер APM в OS обеспечивает доступ к *APM Software Interface*, который позволяет управлять уровнями потребления питания.

В APM имеется четыре основных проблемы. Во-первых, управление энергопотреблением осуществляется через зависимый от поставщика BIOS, и OS ничего не знает нем. Один пример: когда пользователь устанавливает время ожидания для жесткого диска в APM BIOS, и это время истекает, BIOS останавливает жесткий диск без согласования с OS. Во-вторых, алгоритм APM встроен в BIOS, и все действия происходят вне контроля OS. Это означает, что пользователи могут решить проблемы с APM BIOS только путем перепрошивки его ROM; это очень опасная процедура, и если она завершится неудачно, система может оказаться в невосстановимом состоянии. В-третьих, реализация технологии APM зависит от поставщика, что означает дублирование усилий и если в BIOS одного из поставщиков будет найдена и исправлена ошибка, ее могли не исправить другие поставщики. Наконец, объем APM BIOS недостаточно велик для реализации сложной политики управления питанием, или такой политики, которая может хорошо адаптироваться к потребностям компьютера.

*Plug and Play BIOS (PNPBIOS)* был неудобен во многих ситуациях. PNPBIOS это 16-битная технология, поэтому OS требовалось использовать 16-битную эмуляцию для "взаимодействия" с методами PNPBIOS.

FreeBSD драйвер APM документирован в странице справочника `apm(4)`.

### 11.15.3. Настройка ACPI

`loader(8)` загружает драйвер `acpi.ko` по умолчанию, его *не* надо встраивать в ядро. Причина в том, что с модулями проще работать, например переключиться на другой `acpi.ko` без пересборки ядра. Преимущество в упрощении тестирования. Другая причина в том, что запуск ACPI после старта системы не очень полезен и при некоторых условиях может приводить к краху. Если вы сомневаетесь, отключите ACPI совсем. Драйвер не должен и не может быть выгружен, поскольку системная шина используется для различных взаимодействий оборудования. ACPI может быть выключен с помощью утилиты `acpicnf(8)`. Фактически большинство взаимодействий с ACPI может быть выполнено через `acpicnf(8)`. В основном это означает, что если в выводе `dmesg(8)` есть что-то об ACPI, он скорее всего работает.

**Замечание:** ACPI и APM не могут сосуществовать и должны использоваться отдельно. Каждый из них прервет загрузку, если обнаружит загруженный драйвер другого.

В простейшей форме, ACPI может использоваться для перевода системы в спящий режим с помощью `acpicnf(8)`, с флагом `-s` и параметром 1–5. Большинству пользователей нужен только параметр 1. Параметр 5 делает "мягкое" завершение работы, так же как и:

```
# halt -p
```

Доступны и другие параметры. Обратитесь к странице справочника `acpicnf(8)` за дополнительной информацией.

## 11.16. Использование и отладка FreeBSD ACPI

Написал Nate Lawson. При помощи Peter Schultz, Tom Rhodes.

ACPI это фундаментально новый способ обнаружения устройств, управления энергопотреблением и предоставления стандартизированного доступа к различному оборудованию, ранее управлявшемуся BIOS. Был достигнут определенный прогресс в приспособлении ACPI к работе со всеми системами, но все еще встречаются ошибки в байткоде *ACPI Machine Language* (AML) некоторых материнских плат, незавершенные участки кода в подсистемах ядра FreeBSD и ошибки в интерпретаторе Intel ACPI-CA.

Этот раздел предназначен для того, чтобы упростить ваше содействие разработчикам FreeBSD ACPI в определении причин наблюдаемых вами проблем, выполнении отладки и выработке решения. Спасибо за помощь и надеемся, что мы сможем помочь в решении проблем вашей системы.

### 11.16.1. Отправка отладочной информации

**Замечание:** Перед отправкой сообщения об ошибке убедитесь, что у вас последняя версия BIOS, и, если доступна, последняя версия *firmware* встроенного контроллера.

Те из вас, кто желает составить сообщение о проблеме прямо сейчас, могут воспользоваться адресом [freebsd-acpi@FreeBSD.org](mailto:freebsd-acpi@FreeBSD.org) (<mailto:freebsd-acpi@FreeBSD.org>), отправив на него следующую информацию:

- Описание неправильного поведения, включая тип системы, модель и все, что приводит к появлению ошибки. Кроме того, сообщите настолько точно, насколько возможно, когда появилась ошибка, если ранее вы ее не видели.
- Вывод `dmesg(8)` после `"boot -v"`, включая все сообщения, появившиеся при изучении ошибки.
- Вывод `dmesg(8)` после `"boot -v"` с выключенным ACPI, если его отключение помогает решить проблему.
- Вывод `sysctl hw.acpi`. Это также хороший способ получения списка возможностей системы.
- URL где можно найти ваш *ACPI Source Language* (ASL). Не отправляйте ASL непосредственно в список рассылки, поскольку он может быть очень большим. Копия ASL может быть создана командой:

```
# acpidump -t -d > name-system.asl
```

(Замените вашим логином `name` и производителем/моделью `system`. Пример: `njl-FooCo6000.asl`)

Большинство разработчиков читают Список рассылки, посвященный обсуждению FreeBSD-CURRENT (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-current>), но для уверенности, что проблему увидят, отправьте ее в `freebsd-acpi` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi>). Будьте терпеливы, все мы заняты полный рабочий день где-то еще. Если ваше сообщение не заметили сразу, мы возможно попросим вас отправить PR (сообщение о проблеме) через `send-pr(1)`. При вводе PR, включайте ту же информацию, что запрошена выше. Это поможет нам отследить проблему и решить ее. Не отправляйте PR без предварительной отправки письма в `freebsd-acpi` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi>), поскольку мы используем PR в качестве

напоминаний о существующих проблемах, а не как механизм сообщений об ошибках. Вероятно, о вашей проблеме кто-то уже сообщал ранее.

### 11.16.2. Общие сведения

ACPI представлен во всех современных компьютерах, соответствующих архитектурам ia32 (x86), ia64 (Itanium) и amd64 (AMD). Полный стандарт включает множество возможностей, в том числе управление производительностью CPU, уровнем питания, температурой, различными системами аккумуляторов, встроенными контроллерами и опросом шины. В большинстве систем стандарт реализован не полностью. Например, настольные системы обычно реализуют только опрос шины, а портативные компьютеры кроме того могут поддерживать управление охлаждением и энергопотреблением. Они также поддерживают приостановку и последующий запуск системы различного уровня сложности.

ACPI-совместимые системы состоят из различных компонентов. Производители BIOS и чипсетов предоставляют различные жестко заданные таблицы, (например, FADT), которые определяют функции вроде карты APIC (используется для SMP), регистры настройки и простые значения параметров. Кроме того, предоставляется таблица байткода (*Differentiated System Description Table*, DSDT), определяющая древоподобное пространство имен устройств и методов.

Драйвер ACPI должен прочесть заданные таблицы, реализовать интерпретатор для байткода, модифицировать драйвера устройств и ядро для приема информации от подсистемы ACPI. Для FreeBSD Intel предоставила интерпретатор (ACPI-CA), тот же что для Linux и NetBSD. Исходный код ACPI-CA находится в каталоге `src/sys/contrib/dev/acpica`. Код для приспособления ACPI-CA к работе в FreeBSD, находится в `src/sys/dev/acpica/Osd`. Наконец, драйвера, реализующие различные ACPI устройства, находятся в `src/sys/dev/acpica`.

### 11.16.3. Часто встречающиеся проблемы

Для правильной работы ACPI все ее части должны работать правильно. Вот некоторые часто встречающиеся проблемы, в порядке частоты появления, и некоторые обходные пути или исправления.

#### 11.16.3.1. Приостановка/возобновление работы

ACPI поддерживает три состояния приостановки в RAM (STR), S1-S3, и одно состояние приостановки на диск (STD), называемое S4. S5 это "мягкое выключение" и это нормальное состояние системы, когда она подключена к сети, но не включена. S4 может быть реализован двумя различными путями. S4BIOS это BIOS-поддерживаемая приостановка на диск. S4OS реализуется полностью операционной системой.

Начните с проверки переменных `sysctl hw.acpi`, относящихся к приостановке (suspend). Вот результат для Thinkpad:

```
hw.acpi.supported_sleep_state: S3 S4 S5
hw.acpi.s4bios: 0
```

Это означает, что мы можем использовать `acpicnf -s` для тестирования S3, S4OS, и S5. Если `s4bios` был единицей (1), это означает поддержку S4BIOS вместо S4OS.

При тестировании приостановки/возобновления работы, начните с S1, если этот режим поддерживается. Это состояние скорее всего поддерживается, поскольку не требует слишком

серьезной поддержки со стороны драйвера. Никто не реализовал S2, который похож на S1. Следующий режим для тестирования это S3. Это наиболее глубокое STR состояние, оно требует существенной поддержки со стороны драйвера, чтобы правильно реинициализировать оборудование. Если у вас возникли проблемы при выходе из этого состояния, отправьте письмо в рассылку `freebsd-acpi` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi>), но не ждите, что проблема будет обязательно решена, поскольку существует множество драйверов/оборудования, нуждающихся в дальнейшем тестировании и разработке.

Для изоляции проблемы удалите из ядра столько драйверов, сколько возможно. Если это работает, вы можете выяснить, какой драйвер вызывает проблему путем загрузки драйверов до тех пор, пока опять не произойдет сбой. Обычно бинарные драйвера, такие как `nvidia.ko`, драйвера дисплея X11 и USB вызывают большинство проблем, а драйвера Ethernet интерфейсов как правило работают отлично. Если вы можете нормально загрузить/выгрузить драйвера, автоматизируйте этот процесс, поместив соответствующие команды в `/etc/rc.suspend` и `/etc/rc.resume`. Это закомментированные примеры выгрузки и загрузки драйверов. Попробуйте установить параметр `hw.acpi.reset_video` в нуль (0), если ваш дисплей не включается после возобновления работы. Попробуйте установить большие или меньшие значения для `hw.acpi.sleep_delay`, чтобы проверить, поможет ли это.

Другой способ, который можно попробовать, это запуск последнего дистрибутива Linux с поддержкой ACPI и протестировать поддержку остановки/возобновления работы на том же оборудовании. Если она работает на Linux, проблема скорее всего в драйверах FreeBSD и поиск драйвера, вызывающего проблему, поможет разрешить ситуацию. Имейте ввиду, что разработчики ACPI обычно не поддерживают другие драйверы (звук, АТА, и т.п.), так что все результаты работы по поиску проблемы возможно необходимо отправить в список рассылки `freebsd-current` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-current>) и человеку, поддерживающему драйвер. Если вы решитесь заняться отладкой, поместите соответствующий код (`printf(3)`) в вызывающий проблему драйвер для обнаружения места, где прерывается функция восстановления.

Наконец, попробуйте отключить ACPI и включить APM. Если приостановка/возобновление работает с APM, вам возможно лучше подойдет APM, особенно на старом оборудовании (до 2000). Включение корректной поддержки ACPI поставщиками оборудования требует времени и вероятно в старом оборудовании поддержка ACPI в BIOS была некорректна.

### 11.16.3.2. Система останавливается (временно или постоянно)

Большинство систем останавливаются в результате потери прерываний или "шторма" прерываний. В чипсетах существует много проблем, связанных с тем, как BIOS настраивает прерывания перед загрузкой, правильностью таблицы APIC (MADT), и маршрутизации *System Control Interrupt* (SCI).

"Шторм" прерываний может быть обнаружен по потерянными прерываниям путем проверки вывода строки с `acpi0` команды `vmstat -i`. Если счетчик увеличивается более, чем несколько раз в секунду, это "шторм" прерываний. Если система останавливается, попробуйте войти в DDB (**CTRL+ALT+ESC** на консоли) и ввести `show interrupts`.

Наиболее надежный способ избавиться от проблемы с прерываниями, это отключение поддержки APIC с помощью параметра `loader.conf hint.apic.0.disabled="1"`.

### 11.16.3.3. Паника

Паника, связанная с ACPI, случается довольно редко и имеет наибольший приоритет исправления.

Первый шаг это изоляция действий, приводящих к панике (если это возможно) и получение отладки. Следуйте инструкции по включению `options DDB` и настройке последовательной консоли (смотрите Разд. 20.6.5.3) или настройке раздела `dump(8)`. Вы можете получить отладочную информацию DDB с помощью `tr`. Если вы записываете отладку вручную, убедитесь, что переписали как минимум пять (5) строк снизу и пять (5) строк сверху.

Затем попробуйте изолировать проблему, загрузившись с выключенным ACPI. Если это работает, вы можете изолировать подсистему ACPI, используя различные параметры `debug.acpi.disable`. Обратитесь к странице справочника `acpi(4)` за примерами.

#### 11.16.3.4. Система включается после приостановки или завершения работы

Во-первых, попробуйте установить в `loader.conf(5)` параметр `hw.acpi.disable_on_poweroff="0"`. Это предотвращает отключение различных событий в ACPI во время завершения работы. В некоторых системах этот параметр необходимо установить в 1 (по умолчанию) по тем же причинам. Обычно это решает проблему, если система неожиданно включается после приостановки или отключения питания.

#### 11.16.3.5. Другие проблемы

Если вы наблюдаете другие проблемы с ACPI (работа с внешним оборудованием, проблемы с обнаружением устройств, и т.д.), отправьте описание проблемы в список рассылки; однако, некоторые из этих проблем могут относиться к незавершенным частям подсистемы ACPI, поэтому может потребоваться время на их реализацию. Будьте терпеливы, и подготовьтесь к тестированию исправлений, которые мы можем вам выслать.

### 11.16.4. ASL, `aspidump`, и IASL

Наиболее часто встречается проблема, связанная с предоставлением поставщиками BIOS некорректного (или полностью ошибочного!) байткода. Это обычно проявляется появлением консольных сообщений ядра, подобных этому:

```
ACPI-1287: *** Error: Method execution failed [\\_SB_.PCI0.LPC0.FIGD._STA] \\
(Node 0xc3f6d160), AE_NOT_FOUND
```

Зачастую вы можете разрешить эти проблемы путем обновления BIOS до последней ревизии. Большинство консольных сообщений безвредны, но если существуют другие проблемы, такие как не работающий статус батареи, возможно существуют проблемы в AML. Байткод, известный как AML, компилируется из исходного текста на языке ASL. AML находится в таблице, известной как DSDT. Для получения копии ASL, используйте `aspidump(8)`. Вы можете использовать оба параметра `-t` (показывать содержимое постоянных таблиц) и `-d` (дизассемблировать AML в ASL). Обратитесь к разделу Отправка отладочной информации за примером синтаксиса.

Простейшая первая проверка, которую вы можете провести, это перекомпиляция ASL для поиска ошибок. Предупреждения обычно могут быть проигнорированы, но ошибки обычно не позволяют ACPI работать правильно. Для перекомпиляции ASL, выполните следующую команду:

```
# iasl your.asl
```

### 11.16.5. Исправление ASL

В дальней перспективе, наша задача состоит в том, чтобы обеспечить поддержку ACPI практически для каждой системы без вмешательства пользователя. Однако, на данный момент мы все еще разрабатываем обходные пути для ошибок, которые часто делают поставщики BIOS. Интерпретатор Microsoft (`acpi.sys` и `acpiec.sys`) не занимается проверкой четкости соблюдения стандартов, поэтому многие поставщики BIOS, проверяющие ACPI только под Windows, никогда не исправляют ASL. Мы надеемся продолжать обнаружение и документацию нестандартных поведений, позволяемых интерпретатором Microsoft, и воспроизводить их, чтобы FreeBSD могла работать без необходимости исправления ASL пользователями. В качестве обходного пути для обнаружения неправильного поведения, вы можете исправить ASL вручную. Если исправления будут работать, пожалуйста отправьте `diff(1)` между старым и новым ASL, чтобы мы могли реализовать обходной путь для неправильного поведения ACPI-CA, чтобы исправление вручную больше не требовалось.

Вот список наиболее часто встречающихся проблем, их причин и способы исправления:

#### 11.16.5.1. OS зависимости

Некоторые AML предполагают, что мир состоит из различных версий Windows. Вы можете настроить FreeBSD, чтобы она сообщала любое другое имя OS и посмотреть, исправит ли это имеющуюся проблему. Простой способ указания другого имени системы это установка переменной `/boot/loader.conf hw.acpi.osname="Windows 2001"` или в другое подобное значение, имеющееся в ASL.

#### 11.16.5.2. Отсутствие возврата значения

Некоторые методы не возвращают значение явно, как того требует стандарт. Хотя ACPI-CA не обрабатывает эту ситуацию, в FreeBSD существует обходной путь, позволяющий ей явно возвращать значение. Вы можете также добавить явные операторы `Return` (возврат) там, где требуется, если знаете, что значение должно быть возвращено. Для принудительного компилирования ASL командой `iasl`, используйте флаг `-f`.

#### 11.16.5.3. Перезапись AML по умолчанию

После настройки `your.asl` для компиляции запустите:

```
# iasl your.asl
```

Вы можете добавить флаг `-f` для создания AML даже при наличии ошибок компиляции. Помните, что некоторые ошибки (например, отсутствующие операторы `Return`), автоматически обходятся интерпретатором.

Файл `DSDT.aml` используется `iasl` по умолчанию. Вы можете загрузить его вместо ошибочной копии BIOS (которая остается в постоянной памяти) путем редактирования `/boot/loader.conf`:

```
acpi_dsdt_load="YES"
acpi_dsdt_name="/boot/DSDT.aml"
```

Убедитесь, что скопировали `DSDT.aml` в каталог `/boot`.



### 11.16.6. Получение отладочной информации ACPI

Возможности отладки драйвера ACPI очень гибкие. Они позволяют вам указывать набор подсистем, а также уровень отладки. Подсистемы, которые вы хотите отлаживать, указываются как "слой", и подразделяются на компоненты ACPI-CA (ACPI\_ALL\_COMPONENTS) и поддержку оборудования ACPI (ACPI\_ALL\_DRIVERS). Уровень отладки варьируется от ACPI\_LV\_ERROR (только сообщать об ошибках) до ACPI\_LV\_VERBOSE (все сообщения). Уровень отладки представляет собой битовую маску, поэтому возможна одновременная установка нескольких параметров, разделенных пробелами. На практике, при использовании для получения отладочной информации последовательной консоли, слишком большое количество информации может переполнить буфер консоли. Полный список отдельных слоев и уровней можно найти на странице справочника acpi(4).

Вывод отладочной информации по умолчанию не включен. Для его включения добавьте параметр `options ACPI_DEBUG` к файлу настройки ядра, если ACPI встроен в ядро. Вы можете добавить параметр `ACPI_DEBUG=1` в файл `/etc/make.conf` для глобального включения этого параметра. Если вы используете модуль `acpi.ko`, его можно пересобрать индивидуально:

```
# cd /sys/modules/acpi/acpi
&& make clean && make
ACPI_DEBUG=1
```

Установите `acpi.ko` в `/boot/kernel` и добавьте предпочитаемый уровень и слой к `loader.conf`. Этот пример включает отладочные сообщения для всех компонентов ACPI-CA и всех драйверов оборудования ACPI (CPU, LID и т.д.). Будут выводиться только сообщения об ошибках, наименьший уровень отладки.

```
debug.acpi.layer="ACPI_ALL_COMPONENTS ACPI_ALL_DRIVERS"
debug.acpi.level="ACPI_LV_ERROR"
```

Если требуемая информация получается в результате определенного события (скажем, приостановка и восстановление), вы можете не изменять `loader.conf` и использовать для указания слоя и уровня `sysctl` после загрузки и подготовки системы к определенному событию. Имена переменных `sysctl` те же, что и имена параметров настройки в `loader.conf`.

### 11.16.7. Ссылки

Дальнейшую информацию о ACPI можно найти по следующим ссылкам:

- Список рассылки FreeBSD ACPI (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi>)
- Архивы списка рассылки ACPI <http://lists.freebsd.org/pipermail/freebsd-acpi/>
- Старые архивы списка рассылки ACPI <http://home.jp.FreeBSD.org/mail-list/acpi-jp/>
- Спецификация ACPI 2.0 <http://acpi.info/spec.htm>
- Страницы справочника FreeBSD: `acpi(4)`, `acpi_thermal(4)`, `acpidump(8)`, `iasl(8)`, `acpidb(8)`
- Ресурс по отладке DSDT ([http://www.cpqlinux.com/acpi-howto.html#fix\\_broken\\_dsdt](http://www.cpqlinux.com/acpi-howto.html#fix_broken_dsdt)). (Использует в качестве примера Compaq, но обычно полезен.)

# Глава 12. Процесс загрузки FreeBSD

*Перевод на русский язык: Андрей Захватов.*

## 12.1. Описание

Процесс включения компьютера и загрузки операционной системы называется "процессом первоначальной загрузки", или просто "загрузкой". Процесс загрузки FreeBSD предоставляет большие возможности по гибкой настройке того, что происходит при запуске системы, позволяя вам выбирать из различных операционных систем, установленных на одном и том же компьютере, или даже из различных версий той же самой операционной системы или установленного ядра.

Эта глава подробно описывает параметры, которые вы можете изменить для настройки процесса загрузки FreeBSD. Под этим подразумевается все, что происходит до начала работы ядра FreeBSD, обнаружения устройств и запуска `init(8)`. Если вы не совсем уверены, то это происходит, когда выводимый текст меняет цвет с ярко-белого на серый.

После чтения этой главы вы будете знать:

- Из каких частей состоит система начальной загрузки FreeBSD, и как эти части взаимодействуют.
- Параметры, которые вы можете передать компонентам начальной загрузки FreeBSD для управления этим процессом.
- Основы работы `device.hints(5)`

**Только для x86:** Эта глава описывает процесс загрузки FreeBSD только для систем на основе архитектуры Intel x86.

## 12.2. Проблема загрузки

Включение компьютера и запуск операционной системы приводят к интересной дилемме. По определению до запуска операционной системы компьютер не умеет ничего. В том числе и не знает, как запускать программы с диска. Так что компьютер не может запустить программу с диска без операционной системы, но программы операционной системы находятся на диске, но как запустить операционную систему?

Эта проблема имеет параллели с одной проблемой из книги *Приключения барона Мюнхгаузена*. Герой провалился в болото, и вытащил сам себя, ухватив за волосы и потянув. В эпоху начала компьютеризации термин *начальная загрузка* применялся к механизму, используемому для загрузки операционной системы, и затем был сокращен до просто "загрузки".

На оборудовании архитектуры x86 за загрузку операционной системы отвечает BIOS (Basic Input/Output System). Для этого BIOS ищет на жестком диске MBR (Master Boot Record), которая должна располагаться в определенном месте на диске. BIOS может загрузить и запустить MBR, и предполагается, что MBR может взять на себя остальную работу, связанную с загрузкой операционной системой.

Если на вашем диске установлена только одна операционная система, то стандартной MBR будет достаточно. Такая MBR выполняет поиск на диске первого загрузочного слайса, после чего запускает с этого слайса код загрузки оставшейся части операционной системы.

Если на ваших дисках установлено несколько операционных систем, то вы можете установить другую MBR, ту, что может выдать список различных операционных систем и позволит вам выбрать одну из них для загрузки. FreeBSD поставляется с одной из такой MBR, которую можно установить; другие производители операционных систем также предоставляют свои MBR.

Оставшаяся часть системы начальной загрузки FreeBSD разделяется на три этапа. Первый этап запускается из MBR, и он знает достаточно для перевода компьютера в особое состояние и загрузки второго этапа. Второй этап может делать несколько больше до запуска третьего этапа. Третий этап заканчивает работу по загрузке операционной системы. Работа разделена на эти три этапа, потому что стандарты ПК ограничивают размеры программ, которые могут быть запущены на первом и втором этапах. Последовательное выполнение работ позволяет FreeBSD получить более гибкий загрузчик.

Затем стартует ядро, которое начинает опознавать устройства и выполняет их инициализацию. После завершения процесса своей загрузки, ядро передает управление пользовательскому процессу с именем `init(8)`, который выполняет проверку дисков на возможность использования. Затем `init(8)` запускает пользовательский процесс настройки ресурсов, который монтирует файловые системы, выполняет настройку сетевых адаптеров для работы в сети и вообще осуществляет запуск всех процессов, обычно выполняемых в системе FreeBSD при загрузке.

## 12.3. MBR и первый, второй и третий этапы загрузки

### 12.3.1. MBR, `/boot/boot0`

MBR для FreeBSD находится в `/boot/boot0`. Это *копия* MBR, так как настоящая MBR должна располагаться в специальном месте диска, вне области FreeBSD.

`boot0` очень прост, так как программа в MBR может иметь размер, не превышающий 512 байт. Если вы установили MBR FreeBSD и несколько операционных систем на ваш жесткий диск, то во время загрузки вы увидите нечто похожее на следующее:

#### Пример 12-1. Образец экрана `boot0`

```
F1 DOS
F2 FreeBSD
F3 Linux
F4 ??
F5 Drive 1
```

```
Default: F2
```

Известно, что другие операционные системы, в частности, Windows 95, записывают поверх существующей MBR свою собственную. Если так случилось в вашем случае, или же вы хотите заменить существующую MBR на MBR от FreeBSD, то воспользуйтесь следующей командой:

```
# fdisk -B -b /boot/boot0 device
```

Здесь *device* является устройством, с которого вы загружаетесь, таким, как `ad0` в случае первого диска IDE, `ad2` в случае первого диска IDE на втором контроллере IDE, `da0` для первого диска SCSI и так далее.

Однако если вы используете Linux и предпочитаете, чтобы процесс загрузки управлялся через **LILO**, вы можете отредактировать файл `/etc/lilo.conf` для FreeBSD или выбрать `Leave The Master Boot Record Untouched` в процессе установки FreeBSD. Если вы установили менеджер загрузки FreeBSD, то вы можете снова загрузить Linux и изменить конфигурационный файл `/etc/lilo.conf` для **LILO**, добавив следующий параметр:

```
other=/dev/hdXY
table=/dev/hdb
loader=/boot/chain.b
label=FreeBSD
```

который позволит загружать FreeBSD и Linux посредством **LILO**. В нашем примере мы используем *XY* для обозначения номера диска и раздела. Если вы используете диск SCSI, то вам может потребоваться заменить `/dev/hdXY` на что-то типа `/dev/sdXY`, где снова используется обозначение *XY*. Строка `loader=/boot/chain.b` может быть опущена, если обе операционные системы располагаются на одном и том же диске. Вы можете теперь запустить `/sbin/lilo -v` для того, чтобы ваши изменения были восприняты системой, что должно быть подтверждено сообщениями на экране.

### 12.3.2. Этап первый, `/boot/boot1`, и этап второй, `/boot/boot2`

Концептуально первый и второй этапы загрузки являются частями одной и той же программы, в той же самой области диска. Из-за ограничений на объем дискового пространства они были разделены на две, но вы всегда должны устанавливать их вместе.

Они располагаются в загрузочном секторе загрузочного слайса, то есть там, где `boot0` или любая другая программа из MBR ожидает найти программу, которую следует запустить для продолжения процесса загрузки. Файлы в каталоге `/boot` являются копиями реальных файлов, которые хранятся вне файловой системы FreeBSD.

`boot1` очень прост, так как он тоже может иметь размер, не превышающий 512 байт, и знает достаточно о *метке диска* FreeBSD, хранящей информацию о слайсе, для того, чтобы найти и запустить `boot2`.

`boot2` устроен несколько более сложно, и умеет работать с файловой системой FreeBSD в объёме, достаточном для нахождения в ней файлов, и может предоставлять простой интерфейс для выбора и передачи управления ядру или загрузчику.

Так как загрузчик устроен гораздо более сложно, и дает удобный и простой способ настройки процесса загрузки, `boot2` обычно запускает его, однако раньше его задачей был запуск непосредственно самого ядра.

#### Пример 12-2. Образец экрана `boot2`

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/kernel
boot:
```

Если вам когда-либо понадобится заменить установленные `boot1` и `boot2`, то используйте утилиту `disklabel(8)`:

```
# disklabel -B diskslice
```

Здесь `diskslice` являются диском и слайсом, с которых вы загружаетесь, такие, как `ad0s1` в случае первого слайса на первом диске IDE.

**Режим Dangerously Dedicated:** Если вы используете только имя диска, к примеру, `ad0`, в команде `disklabel(8)` вы создадите диск в режиме эксклюзивного использования, без слайсов. Это, скорее всего, вовсе не то, что вы хотите сделать, так что дважды проверьте параметры команды `disklabel(8)`, прежде, чем нажать **Return**.

### 12.3.3. Третий этап, `/boot/loader`

Передача управления загрузчику является последним, третьим этапом в процессе начальной загрузки, а сам загрузчик находится в файловой системе, обычно как `/boot/loader`.

Загрузчик являет собой удобный в использовании инструмент для настройки при помощи простого набора команд, управляемого более мощным интерпретатором с более сложным набором команд.

#### 12.3.3.1. Процесс работы загрузчика

Во время инициализации загрузчик пытается произвести поиск консоли, дисков и определить, с какого диска он был запущен. Соответствующим образом он задаёт значения переменных и запускает интерпретатор, которому могут передаваться пользовательские команды как из скрипта, так и в интерактивном режиме.

Затем загрузчик читает файл `/boot/loader.rc`, который по умолчанию использует файл `/boot/defaults/loader.conf`, устанавливающий подходящие значения по умолчанию для переменных и читает файл `/boot/loader.conf` для изменения в этих переменных. Затем с этими переменными работает `loader.rc`, загружающий выбранные модули и ядро.

И наконец, по умолчанию загрузчик выдерживает 10-секундную паузу, ожидая нажатия клавиши, и загружает ядро, если этого не произошло. Если ожидание было прервано, пользователю выдается приглашение, которое воспринимает простой набор команд, в помощью которых пользователь может изменить значения переменных, выгрузить все модули, загрузить модули и окончательно продолжить процесс загрузки или перезагрузить машину.

#### 12.3.3.2. Встроенные команды загрузчика

Далее следуют наиболее часто используемые команды загрузчика. Полное описание всех имеющихся команд можно найти на странице справки о команде `loader(8)`.

**autoboot** *секунды*

Продолжает загрузку ядра, если не будет прерван в течение указанного в секундах промежутка времени. Он выводит счетчик, и по умолчанию выдерживается интервал в 10 секунд.

**boot** [*-параметры*] [*имя ядра*]

Продолжить процесс загрузки указанного ядра, если оно было указано, и с указанными параметрами, если они были указаны.

**boot-conf**

Повторно провести тот же самый процесс автоматической настройки модулей на основе переменных, что был произведен при загрузке. Это имеет смысл, если до этого вы выполнили команду `unload`, изменили некоторые переменные, например, наиболее часто меняемую `kernel`.

**help** [*тема*]

Вывод сообщений подсказки из файла `/boot/loader.help`. Если в качестве темы указано слово `index`, то выводится список имеющихся тем.

**include** *имя файла* ...

Выполнить файл с указанным именем. Файл считывается и его содержимое интерпретируется строчка за строчкой. Ошибка приводит к немедленному прекращению выполнения команды `include`.

**load** [*-t тип*] *имя файла*

Загружает ядро, модуль ядра или файл указанного типа с указанным именем. Все аргументы после имени файла передаются в файл.

**ls** [*-l*] [*маршрут*]

Выводит список файлов по указанному маршруту или в корневом каталоге, если маршрут не был указан. Если указан параметр `-l`, будут выводиться и размеры файлов.

**lsdev** [*-v*]

Выводится список всех устройств, с которых могут быть загружены модули. Если указан параметр `-v`, выводится дополнительная информация.

**lsmod** [*-v*]

Выводит список загруженных модулей. Если указан параметр `-v`, то выводится дополнительная информация.

**more** *имя файла*

Вывод указанного файла с паузой при выводе каждой строки `LINES`.

**reboot**

Выполнить немедленную перезагрузку машины.

**set** *переменная*

**set** *переменная=значение*

Задаёт значения переменных окружения загрузчика.

**unload**

Удаление из памяти всех загруженных модулей.

### 12.3.3.3. Примеры использования загрузчика

Вот несколько примеров практического использования загрузчика:

- Чтобы просто загрузить ваше ядро обычным образом, но в однопользовательском режиме:

```
boot -s
```

- Для выгрузки обычных ядра и модулей, а потом просто загрузить ваше старое (или другое) ядро:

```
unload  
load kernel.old
```

Вы можете использовать `kernel.GENERIC` для обозначения стандартного ядра, поставляемого на установочном диске, или `kernel.old` для обращения к ранее установленному ядру (после того, как, например, вы обновили или отконфигурировали новое ядро).

**Замечание:** Для загрузки ваших обычных модулей с другим ядром используйте такие команды:

```
unload  
set kernel="kernel.old"  
boot-conf
```

- Для загрузки скрипта конфигурации ядра (автоматизированный скрипт, который выполняет то, что вы обычно делаете в конфигураторе ядра во время загрузки):

```
load -t userconfig_script /boot/kernel.conf
```

## 12.4. Взаимодействие с ядром во время загрузки

Как только ядро окажется загруженным при помощи загрузчика (обычный способ) или `boot2` (минуя загрузчик), оно проверяет флаги загрузки, если они есть, и действует соответствующим образом.

### 12.4.1. Флаги загрузки ядра

Вот наиболее часто используемые флаги загрузки:

**-a**

во время инициализации ядра запрашивать устройство для его монтирования в качестве корневой файловой системы.

**-c**

загрузка с компакт-диска.

-c

запустить UserConfig для конфигурации ядра во время загрузки

-s

после загрузки перейти в однопользовательский режим

-v

во время запуска ядра выводить более подробную информацию

**Замечание:** Есть и другие флаги загрузки, обратитесь к странице справочника по boot(8) для выяснения подробной информации по ним.

## 12.5. Хинты устройств

*Текст предоставил Tom Rhodes.*

**Замечание:** Эта функция присутствует только во FreeBSD 5.0 и последующих версиях, но не в более ранних.

Во время начального запуска системы загрузчик loader(8) производит чтение файла device.hints(5). В этом файле хранится необходимая для загрузки ядра информация, задаваемая в виде переменных, которую иногда называют хинтами для устройств ("device hints"). Эти "хинты устройств" используются драйверами устройств для их конфигурации.

Хинты для устройств могут быть также заданы в приглашении начального загрузчика Стадии 3. Переменные могут быть добавлены при помощи команды set, удалены посредством unset и просмотрены командой show. В этот момент могут быть также переопределены переменные, заданные в файле /boot/device.hints. Хинты для устройств, введённые в начальном загрузчике, не сохраняются, и при следующей перезагрузке будут утеряны.

После загрузки системы для выдачи значений всех переменных можно воспользоваться командой kenv(1).

Синтаксически в файле /boot/device.hints в каждой строке определяется по одной переменной, в качестве метки начала комментария используется стандартный символ "#". Строки строятся следующим образом:

```
hint.driver.unit.keyword="value"
```

Синтаксис для начального загрузчика Стадии 3 таков:

```
set hint.driver.unit.keyword=value
```

driver определяет имя драйвера устройства, unit соответствует порядковому номеру модуля устройства, а keyword является ключевым словом хинта. В качестве ключевых слов могут применяться следующие опции:



- `at`: задаёт шину, к которой подключено устройство.
- `port`: задаёт начальный адрес используемого диапазона ввода/вывода (I/O).
- `irq`: задаёт используемый номер запроса на прерывание.
- `drq`: задаёт номер канала DMA.
- `maddr`: задаёт физический адрес памяти, занимаемый устройством.
- `flags`: устанавливает различные битовые флаги для устройства.
- `disabled`: если установлено в значение "1", то устройство не используется.

Драйверы устройств могут поддерживать (и даже требовать) другие хинты, здесь не перечисленные, поэтому рекомендуется просматривать справочные страницы по этим драйверам. Для получения дополнительной информации обратитесь к страницам справки по `device.hints(5)`, `kenv(1)`, `loader.conf(5)` и `loader(8)`.

## 12.6. Init: инициализация управления процессами

После того, как ядро завершит загрузку, оно передает управление пользовательскому процессу `init(8)`, который расположен в файле `/sbin/init` или в файле, маршрут к которому указан в переменной `init_path` загрузчика.

### 12.6.1. Процесс автоматической перезагрузки

Процесс автоматической перезагрузки проверяет целостность имеющихся файловых систем. Если это не так, и утилита `fsck(8)` не может исправить положение, то `init(8)` переводит систему в однопользовательский режим для того, чтобы системный администратор сам разобрался с возникающими проблемами.

### 12.6.2. Однопользовательский режим

В этот режим можно перейти во время процесса автоматической перезагрузки, при ручной загрузке с параметром `-s` или заданием переменной `boot_single` для программы `loader`.

Этот режим может быть также вызван запуском программы `shutdown(8)` без параметров перезагрузки (`-r`) или останова (`-h`) из многопользовательского режима.

Если режим доступа к системной консоли `console` установлен в файле `/etc/ttys` в `insecure`, то система выведет запрос на ввод пароля пользователя `root` перед переходом в однопользовательский режим.

#### Пример 12-3. Незащищённая консоль в `/etc/ttys`

```
# name getty type status comments
#
# Если консоль помечена как "insecure", то init будет запрашивать пароль
# пользователя root при переходе в однопользовательский режим.
console none                unknown off insecure
```

**Замечание:** Обозначение консоли как `insecure` означает, что вы считаете физический доступ к консоли незащищённым, и хотите, чтобы только тот, кто знает пароль пользователя `root`, мог воспользоваться однопользовательским режимом, но это не значит, что вы хотите работать с консолью небезопасным способом. Таким образом, если вы хотите добиться защищённости, указывайте `insecure`, а не `secure`.

### 12.6.3. Многопользовательский режим

Если `init(8)` определит, что ваши файловые системы находятся в полном порядке, или после того, как пользователь выйдет из однопользовательского режима, система перейдет в многопользовательский режим, работа в котором начинается с настройки ресурсов системы.

#### 12.6.3.1. Настройка ресурсов (rc)

Система настройки ресурсов считывает настройки, применяемые по умолчанию, из файла `/etc/defaults/rc.conf`, а настройки, специфичные для конкретной системы, из `/etc/rc.conf`, после чего осуществляется монтирование файловых систем, перечисленных в файле `/etc/fstab`, запуск сетевых служб, различных системных демонов и, наконец, выполнение скриптов запуска дополнительно установленных пакетов.

Страница справочника по `rc(8)` является хорошим источником информации о системе настройки ресурсов. так же, как и самостоятельное изучение скриптов.

## 12.7. Процесс остановки системы

Во время контролируемого процесса остановки системы через утилиту `shutdown(8)` программа `init(8)` будет пытаться запустить скрипт `/etc/rc.shutdown`, после чего будет посылать всем процессам сигнал `TERM`, а затем и `KILL` тем процессам, которые ещё не завершили свою работу.

Для выключения машины с FreeBSD на аппаратных платформах и системах, которые поддерживают управление электропитанием, просто воспользуйтесь командой `shutdown -p now` для немедленного отключения электропитания. Чтобы просто перезагрузить систему FreeBSD, воспользуйтесь командой `shutdown -r now`. Для запуска команды `shutdown(8)` вам необходимо быть пользователем `root` или членом группы `operator`. Кроме того, можно также воспользоваться командами `halt(8)` и `reboot(8)`, пожалуйста, обратитесь к соответствующим страницам справки и справочной странице по команде `shutdown(8)` для получения дополнительной информации.

**Замечание:** Для управления электропитанием требуется наличие поддержки `acpi(4)` в ядре или в виде загруженного модуля при использовании FreeBSD 5.X, а для FreeBSD 4.X необходима поддержка `apm(4)`.

# Глава 13. Пользователи и основы управления учетными записями

*Предоставил Neil Blakey-Milner. Перевод на русский язык: Денис Пеплин.*

## 13.1. Краткий обзор

FreeBSD позволяет одновременную работу множества пользователей на одном компьютере. Разумеется, только один пользователь может сидеть за клавиатурой и перед экраном в один момент времени <sup>1</sup>, но любое количество пользователей может выполнять работу через сеть. Для использования системы у каждого пользователя должна быть учетная запись.

После прочтения этой главы вы будете знать:

- Различия между разнообразными учетными записями в системе FreeBSD.
- Как добавлять учетные записи пользователей.
- Как удалять учетные записи пользователей.
- Как изменять данные учетных записей пользователей, такие как полное имя пользователя, или предпочитаемую оболочку.
- Как устанавливать ограничения на использование ресурсов, например на использование памяти и времени CPU для учетных записей или групп учетных записей.
- Как использовать группы для упрощения управлением учетными записями.

Перед прочтением этой главы вам потребуется:

- Понимание основ UNIX и FreeBSD (Гл. 3).

## 13.2. Введение

Весь доступ к системе осуществляется через учетные записи, и все процессы запускаются пользователями, так что управление пользователями и учетными записями в системах FreeBSD имеет всеобъемлющее значение.

С каждой учетной записью в системе FreeBSD связана определенная идентификационная информация.

Имя пользователя

Имя пользователя в том виде, в каком оно вводится в приглашение `login:`. Имена пользователей должны быть уникальны в пределах одного компьютера; не может быть двух пользователей с одинаковым именем пользователя. Существует множество правил для создания правильных имен пользователей, документированных в `passwd(5)`; вы как правило будете использовать имена пользователей, состоящие из восьми или меньшего количества символов, все символы в нижнем регистре.

## Пароль

С каждой учетной записью связан пароль. Пароль может быть пустым, в этом случае для доступа к системе не нужен пароль. Обычно это очень плохая идея; у каждой учетной записи должен быть пароль.

## ID пользователя (User ID, UID)

The UID это номер, традиционно от 0 до 65535<sup>2</sup>, используемый для однозначной идентификации пользователя в системе. Сама система FreeBSD для идентификации пользователей использует UID — любая команда FreeBSD, позволяющая вам указывать имя пользователя, первым делом преобразует его к UID. Это означает, что вы можете создать несколько учетных записей с различными именами пользователей, но с одним UID. FreeBSD будет воспринимать эти учетные записи как одного пользователя. Маловероятно, что вам когда-либо это понадобится.

## ID группы (Group ID, GID)

GID это номер, традиционно от 0 до 65535<sup>2</sup>, используемый для однозначной идентификации главной группы, к которой принадлежит пользователь. Группы это механизм для контроля доступа к ресурсам на основе GID пользователя вместо его UID. Это может значительно уменьшить размер некоторых файлов настройки. Кроме того, пользователь может быть включен более чем в одну группу.

## Класс логина

Классы логинов это расширение к механизму групп, позволяющее системе более гибко управлять различными пользователями.

## Время изменения пароля

По умолчанию FreeBSD не принуждает пользователей периодически менять пароли. Вы можете включить эту функцию для определенных пользователей, заставив некоторых или всех пользователей менять пароли по прошествии определенного времени.

## Время истечения действия учетной записи

По умолчанию в FreeBSD время действия учетных записей не ограничено. Если вы создаете учетные записи, продолжительность жизни которых ограничена, например в учетные записи для студентов в школе, вы можете определить время истечения действия учетной записи. После наступления этого времени учетная запись не может использоваться для входа в систему, хотя каталоги и файлы этой учетной записи останутся нетронутыми.

## Полное имя пользователя

Имя пользователя является уникальным идентификатором учетной записи в FreeBSD, но недостаточно для сопоставления с реальным именем пользователя. Эта информация может быть добавлена в учетную запись.

## Домашний каталог

Домашний каталог это полный путь к каталогу в системе, в котором пользователь начнет работать после входа в систему. По общепринятому соглашению все домашние каталоги пользователей помещаются в `/home/username` или `/usr/home/username`. Пользователи хранят личные файлы в домашнем каталоге и в любых подкаталогах, создаваемых внутри домашнего каталога.

## Оболочка пользователя

Оболочка необходима пользователям как средство взаимодействия с системой по умолчанию. Существует множество различных видов оболочек, опытные пользователи работают с собственными настройками, которые могут быть отражены в установках их учетных записей.

Существует три основных типа учетных записей: суперпользователь, системные пользователи, и учетные записи пользователей. Учетная запись суперпользователя, обычно называемая `root`, используется для управления системой без ограничения привилегий. Системные пользователи запускают сервисы. Наконец, учетные записи пользователей необходимы обычным людям для входа в систему, чтения почты, и так далее.

### 13.3. Учетная запись суперпользователя

Учетная запись суперпользователя, обычно называемая `root`, существует в системе изначально для целей системного администрирования, и не должна использоваться для повседневных задач, таких как получение и отправка почты, общее исследование системы или программирование.

Причина в том, что суперпользователь, в отличие от обычных пользователей, может работать без ограничений и неправильное использование учетной записи суперпользователя может привести к полному уничтожению системы. Учетные записи пользователей не способны уничтожить систему вследствие ошибки, поэтому обычно лучше использовать учетные записи обычных пользователей везде, где это возможно, пока вам не потребуются дополнительные привилегии для какой-то определенной задачи.

Вы всегда должны дважды и трижды проверять команды, выполняемые под учетной записью суперпользователя, поскольку даже один лишний пробел или отсутствующий символ может привести к безвозвратной потере данных.

Таким образом, первое, что вам необходимо сделать после прочтения этой главы, это создать непривилегированную учетную запись пользователя для повседневного использования (если вы еще этого не сделали). Это необходимо сделать независимо от того, работаете ли вы на многопользовательском или однопользовательском компьютере. Позже в этой главе мы обсудим как создать дополнительные учетные записи, и как менять уровень привилегий между нормальным пользователем и суперпользователем.

### 13.4. Системные учетные записи

Системные пользователи предназначены для запуска сервисов, таких как DNS, почта, веб серверы и так далее. Это необходимо по соображениям безопасности; если все сервисы работают от суперпользователя, они могут действовать без ограничений.

Примеры системных пользователей это `daemon`, `operator`, `bind` (для Domain Name Service, DNS), и `news`. Зачастую системные администраторы создают `httpd` для запуска устанавливаемых веб серверов.

`nobody` это классический непривилегированный системный пользователь. Тем не менее, необходимо помнить, что чем больше сервисов используют `nobody`, тем больше файлов и процессов ассоциировано с этим пользователем, и следовательно тем больше прав появляется у этого пользователя.

## 13.5. Учетные записи пользователей

Учетные записи пользователей в основном означают доступ в систему для обычных людей, и эти учетные записи отделяют пользователя и его рабочую среду, предотвращая повреждение пользователем системы или данных других пользователей, и позволяя пользователям настраивать свою рабочую среду без влияния на других пользователей.

Каждая персона, получающая доступ к вашей системе, должна получить уникальную учетную запись пользователя. Это позволит вам выяснить кто что делает, предотвращая сбивание одним пользователем настроек других пользователей, чтение чужой почты и так далее.

Каждый пользователь может настраивать свою собственную рабочую среду для приспособления системы под свои нужды с помощью альтернативных оболочек, редакторов, привязки клавиш и настроек языка.

## 13.6. Изменение учетных записей

В среде UNIX существуют различные команды для работы с учетными записями пользователей. Наиболее часто используемые команды приведены в таблице, ниже находятся более детальные примеры их использования.

Команда	Краткое описание
<code>adduser(8)</code>	Рекомендуемое приложение командной строки для добавления новых пользователей.
<code>rmuser(8)</code>	Рекомендуемое приложение командной строки для удаления пользователей.
<code>chpass(1)</code>	Гибкий инструмент для изменения информации в базе данных пользователей.
<code>passwd(1)</code>	Простой инструмент командной строки для изменения паролей пользователей.
<code>pw(8)</code>	Мощный и гибкий инструмент для изменения любой информации, связанной с учетными записями пользователей.

### 13.6.1. `adduser`

`adduser(8)` это простая программа для добавления новых пользователей. Она создает записи в системных файлах `passwd` и `group`. Она также создает домашний каталог для нового пользователя, копируя файлы настройки по умолчанию (`"dotfiles"`, файлы имя которых начинается с символа `.`) из `/usr/share/skel` и опционально может отправлять новому пользователю приветственное сообщение.

В FreeBSD 5.0, скрипт `adduser(8)` был переписан с языка Perl на язык shell, работающий в качестве оболочки к `pw(8)`, так что использование этого скрипта в FreeBSD 4.X немного отличается использования в FreeBSD 5.X.

Для создания файла настройки используйте `adduser -s -config_create`.<sup>3</sup> Далее, мы настроим параметры `adduser(8)` по умолчанию и создадим нашу первую учетную запись пользователя, поскольку повседневное использование `root` не приемлемо.

**Пример 13-1. Настройка adduser и добавление пользователя в FreeBSD 4.X**

```
# adduser -v
Use option "-silent" if you don't want to see all warnings and questions.
Check /etc/shells
Check /etc/master.passwd
Check /etc/group
Enter your default shell: csh date no sh tcsh zsh [sh]: zsh
Your default shell is: zsh -> /usr/local/bin/zsh
Enter your default HOME partition: [/home]:
Copy dotfiles from: /usr/share/skel no [/usr/share/skel]:
Send message from file: /etc/adduser.message no
[/etc/adduser.message]: no
Do not send message
Use passwords (y/n) [y]: y

Write your changes to /etc/adduser.conf? (y/n) [n]: y

Ok, let's go.
Don't worry about mistakes. I will give you the chance later to correct any input.
Enter username [a-z0-9_-]: jru
Enter full name []: J. Random User
Enter shell csh date no sh tcsh zsh [zsh]:
Enter home directory (full path) [/home/jru]:
Uid [1001]:
Enter login class: default []:
Login group jru [jru]:
Login group is "jru". Invite jru into other groups: guest no
[no]: wheel
Enter password []:
Enter password again []:

Name: jru
Password: ****
Fullname: J. Random User
Uid: 1001
Gid: 1001 (jru)
Class:
Groups: jru wheel
HOME: /home/jru
Shell: /usr/local/bin/zsh
OK? (y/n) [y]: y
Added user "jru"
Copy files from /usr/share/skel to /home/jru
Add another user? (y/n) [y]: n
Goodbye!
#
```

В результате мы изменили оболочку по умолчанию на **zsh** (дополнительная оболочка, находящаяся в коллекции портов), и отключили отправку добавляемым пользователям письма с приветствием. Затем мы сохранили настройки, создали учетную запись для **jru**, и убедились, что **jru** находится в группе **wheel** (теперь этот пользователь может получить привилегии **root** с помощью команды **su(1)**).

**Замечание:** Пароль, который вы вводите, не отображается, "звездочки" при вводе пароля также не отображаются. Убедитесь, что вы не ошиблись при вводе пароля.

**Замечание:** Теперь просто используйте `adduser(8)` без аргументов, и вам не потребуется изменять настройки по умолчанию. Если программа будет запрашивать изменение настроек по умолчанию, попробуйте параметр `-s`.

### Пример 13-2. Добавление пользователя в FreeBSD 5.X

```
# adduser
Username: jru
Full name: J. Random User
Uid (Leave empty for default):
Login group [jru]:
Login group is jru. Invite jru into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh zsh nologin) [sh]: zsh
Home directory [/home/jru]:
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : jru
Password   : ****
Full Name  : J. Random User
Uid        : 1001
Class      :
Groups     : jru wheel
Home       : /home/jru
Shell      : /usr/local/bin/zsh
Locked     : no
OK? (yes/no): yes
adduser: INFO: Successfully added (jru) to the user database.
Add another user? (yes/no): no
Goodbye!
#
```

### 13.6.2. rmuser

Для полного удаления пользователя из системы вы можете использовать `rmuser(8)`. Эта программа выполняет следующие действия:

1. Удаление записи пользователя из `crontab(1)` (если она присутствует).
2. Удаляет задачи `at(1)`, принадлежащие пользователю.



3. Уничтожает все процессы, принадлежащие пользователю.
4. Удаляет пользователя из локального файла паролей.
5. Удаляет домашний каталог пользователя (если он принадлежит пользователю).
6. Удаляет принадлежащую пользователю входящую почту из /var/mail.
7. Удаляет все файлы, принадлежащие пользователю, из каталогов с временными файлами, например /tmp.
8. Наконец, удаляет имя пользователя из всех групп, которым оно принадлежит, в /etc/group.

**Замечание:** Если после этого удаления группа остается пустой и имя группы совпадает с именем пользователя, она удаляется; Это необходимо для удаления пользовательских уникальных групп, создаваемых adduser(8).

rmuser(8) не может использоваться для удаления учетной записи суперпользователя, поскольку это почти всегда означает разрушение системы.

По умолчанию используется интерактивный режим, программа пытается убедиться, что вы уверены в своих действиях.

### Пример 13-3. Интерактивное удаление учетной записи с помощью rmuser

```
# rmuser jru
Matching password entry:
jru:*:1001:1001::0:0:J. Random User:/home/jru:/usr/local/bin/zsh
Is this the entry you wish to remove? y
Remove user's home directory (/home/jru)? y
Updating password file, updating databases, done.
Updating group file: trusted (removing group jru -- personal group is empty) done.
Removing user's incoming mail file /var/mail/jru: done.
Removing files belonging to jru from /tmp: done.
Removing files belonging to jru from /var/tmp: done.
Removing files belonging to jru from /var/tmp/vi.recover: done.
#
```

### 13.6.3. chpass

chpass(1) изменяет информацию в базе данных пользователей: пароли, оболочки, персональную информацию.

Только системные администраторы с правами суперпользователя могут изменять информацию и пароли других пользователей с помощью chpass(1).

При запуске без параметров (кроме опционального имени пользователя), chpass(1) вызывает редактор, содержащий информацию о пользователе. Когда пользователь выходит из редактора, база данных пользователей обновляется этой информацией.

**Замечание:** В FreeBSD 5.X, после выхода из редактора будет запрошен пароль (если вы не суперпользователь).

#### Пример 13-4. Интерактивная работа с `chpass` суперпользователя

```
#Changing user database information for jru.
Login: jru
Password: *
Uid [#]: 1001
Gid [# or name]: 1001
Change [month day year]:
Expire [month day year]:
Class:
Home directory: /home/jru
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

Обычные пользователи могут изменять лишь небольшую часть этой информации, и только для своей учетной записи.

#### Пример 13-5. Интерактивная работа с `chpass` обычного пользователя

```
#Changing user database information for jru.
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

**Замечание:** `chfn(1)` и `chsh(1)` это всего лишь ссылки на `chpass(1)`, как и `upchpass(1)`, `upchfn(1)` и `upchsh(1)`. NIS поддерживается автоматически, так что указание `ur` перед командой не обязательно. Если это непонятно, не беспокойтесь, NIS будет рассмотрен в Гл. 23.

### 13.6.4. `passwd`

`passwd(1)` это обычный способ изменения собственного пароля пользователя, или пароля другого пользователя суперпользователем.

**Замечание:** Для предотвращения случайного или неавторизованного изменения, перед установкой нового пароля необходимо ввести старый.

**Пример 13-6. Изменение пароля**

```
% passwd
Changing local password for jru.
Old password:
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

**Пример 13-7. Изменение пароля другого пользователя суперпользователем**

```
# passwd jru
Changing local password for jru.
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

**Замечание:** Как и с `chpass(1)`, `upasswd(1)` это всего лишь ссылка на `passwd(1)`, так что NIS работает с обеими командами.

**13.6.5. pw**

`pw(8)` это утилита командной строки для создания, удаления, модифицирования и отображения пользователей и групп. Она функционирует как внешний интерфейс к системным файлам пользователей и групп. У `pw(8)` очень мощный набор параметров командной строки, что делает это программу подходящей для использования в shell скриптах, но новым пользователям она может показаться более сложной, чем другие представленные здесь команды.

**13.7. Ограничение пользователей**

Если у вас есть пользователи, может появиться мысль о возможности ограничения нагрузки на систему. FreeBSD предоставляет администратору несколько способов ограничения объема занимаемых пользователями системных ресурсов. Существует два вида ограничений: дисковые квоты и другие ограничения ресурсов.

Дисковые квоты ограничивают объем пространства, занимаемого пользователями, это способ быстрой проверки объема без вычисления его каждый раз. Квоты обсуждаются в Разд. 16.14.

Другие ограничения ресурсов включают способы ограничения использования CPU, памяти и других ресурсов, которые могут потребляться пользователем. Ограничения накладываются с помощью классов учетных записей и обсуждаются в этом разделе.

Классы учетных записей определяются в `/etc/login.conf`. Описание полной семантики выходит за пределы обсуждаемого здесь материала, она детально описана в странице справочника `login.conf(5)`.

Достаточно сказать, что каждому пользователю присвоен класс (default по умолчанию), и каждому классу присвоен набор характеристик. Характеристика определяется в виде пары *имя=значение*, где *имя* это определенный идентификатор, а *значение* это произвольная строка, обрабатываемая в зависимости от имени. Настройка классов и характеристик довольно проста и также описана в `login.conf(5)`.

**Замечание:** Система не читает настройки в `/etc/login.conf` непосредственно, она обращается к файлу базы данных `/etc/login.conf.db`. Для создания `/etc/login.conf.db` из `/etc/login.conf`, выполните следующую команду:

```
# cap_mkdb /etc/login.conf
```

Ограничения на ресурсы отличаются от обычных характеристик: во-первых, для каждого ограничения существует "мягкое" (текущее) и "жесткое" ограничение. Мягкое ограничение может настраиваться пользователем или приложением, но не может превышать жесткое ограничение. Последнее может быть уменьшено пользователем, но никогда не увеличено. Во-вторых, большинство ограничений ресурсов применяются к процессам определенного пользователя, а не к пользователю вообще. Обратите внимание, что эти различия реализуются специфической обработкой лимитов, а не реализацией структуры характеристик учетных записей (т.е. это не *настоящий* специальный случай характеристик).

Ниже приведен список наиболее часто используемых ограничений на ресурсы (остальные, вместе с другими характеристиками можно найти в `login.conf(5)`).

`coredumpsize`

Ограничение на размер файла core, генерируемого программой, по очевидным причинам подчиняющееся другим ограничениям на используемое дисковое пространство (например, `filesize`, или дисковые квоты). Тем не менее, оно часто используется как менее строгий метод контролирования потребления дискового пространства: поскольку пользователь не создает файлы core самостоятельно, и зачастую не удаляет их, установка этого параметра может предохранить его от выхода за пределы дисковых квот, если большая программа (например, **emacs**) создаст core файл.

`cputime`

Это максимальное количество времени CPU, потребляемого пользователем. Превысившие это время процессы будут уничтожены ядром.

**Замечание:** Это ограничение потребляемого *времени* CPU, а не процентов использования CPU, которые отображаются в некоторых полях `top(1)` и `ps(1)`. Ограничения на них на время написания этого материала невозможны и такие ограничения практически бесполезны: компилятор — вполне законное приложение — иногда может легко использовать почти 100% CPU.

`filesize`

Это максимальный размер файла, который может обрабатываться пользователем. В отличие от дисковых квот, это ограничение применяется к отдельным файлам, а не ко всему набору принадлежащих пользователю файлов.

`maxproc`

Это максимальное число процессов, которые могут быть запущены пользователем. В это число включаются и консольные и фоновые процессы. По очевидным причинам, они не могут быть больше, чем системное ограничение, указываемое через переменную `sysctl(8) kern.maxproc`. Имейте ввиду, что установка слишком жестких ограничений может стать помехой работе пользователя: зачастую полезно входить в систему с нескольких консолей или использовать каналы. Некоторые задачи, такие как компиляция большой программы, также порождают множество процессов (например, `make(1)`, `cc(1)` и другие препроцессоры).

`memorylocked`

Это максимальный объем памяти, блокировка которого может быть запрошена процессом (см. например `mlock(2)`). Некоторые критически важные для системы программы, такие как `amd(8)`, блокируют память так, что при выгрузке они не создают системе дополнительных проблем.

`memoryuse`

Это максимальный объем памяти, которая может быть занята процессами. Он включает основную и основную память и использование подкачки. Это ограничение не снимает все вопросы, связанные с использованием памяти, но для начала это подходящее ограничение.

`openfiles`

Это максимальное количество файлов, которые могут быть открыты процессами. В FreeBSD, файлы также используются для представления сокетов и каналов IPC; не устанавливайте слишком маленькое значение. Ограничение этого параметра, устанавливаемое для всей системы, определяется переменной `sysctl(8) kern.maxfiles`.

sbsize

Это ограничение потребляемого пользователем объема сетевой памяти, т.е. mbufs. Оно было введено как ответ на старые DoS атаки, при которых создавалось множество сокетов, но обычно может быть использовано и для ограничения сетевых соединений.

stacksize

Это максимальный размер, до которого может вырасти стек процесса. Сам по себе этот параметр не может ограничить размер используемой программой памяти, следовательно, его необходимо использовать вместе с другими ограничениями.

Существуют несколько других аспектов, которые необходимо учитывать при установке ограничений ресурсов. Ниже приведены некоторые общие подсказки, советы и различные комментарии.

- Процессам, загружаемым при старте системы скриптами `/etc/rc` присваивается класс `daemon`.
- Хотя `/etc/login.conf`, поставляемый с системой, это хороший источник подходящих значений для большинства ограничений, только вы, администратор, можете знать подходящие значения для вашей системы. Установка слишком слабых ограничений может повлечь злоупотребления системой, а установка слишком сильных ограничений может стать помехой производительности.
- Пользователи X Window System (X11) возможно должны получить больше ресурсов, чем другие пользователи. X11 сама по себе потребляет много ресурсов, а также провоцирует пользователей на одновременный запуск большего количества программ.
- Помните, что многие ограничения применяются к отдельным процессам, а не к пользователю вообще. Например, установка `openfiles` в 50 означает, что каждый процесс, запущенный пользователем, может открывать до 50 файлов. Таким образом, общее количество файлов, которые могут быть открыты пользователем, вычисляется как `openfiles`, помноженное на `maxproc`. Это также применимо к потребляемой памяти.

За дальнейшей информацией по ограничениям на ресурсы, классам учетных записей и характеристикам, обращайтесь к соответствующим страницам справочника.

## 13.8. Персонализация пользователей

Локализация это окружение, настраиваемое системным администратором или пользователем для работы с различными языками, наборами символов, стандартами даты и времени, и так далее. Это обсуждается в главе локализация.

## 13.9. Группы

Группа это просто список пользователей. Группа идентифицируется по имени и GID (Group ID, идентификатор группы). В FreeBSD (и большинстве других UNIX-подобных системах) ядро для

определения прав процесса использует два фактора: его ID пользователя и список групп, которым он принадлежит. Когда вы слышите что-то о "group ID" пользователя или процесса, это обычно означает только первую группу из списка.

Имена групп связываются с ID групп в файле `/etc/group`. Это текстовый файл с четырьмя разделенными двоеточием полями. Первое поле это имя группы, второе это зашифрованный пароль, третье это ID группы, а четвертое это разделенный запятыми список членов группы. Этот файл может быть безопасно отредактирован вручную (предполагается, конечно, что вы не сделаете синтаксических ошибок!). За более полным описанием синтаксиса обратитесь к странице справочника `group(5)`.

Если вы не хотите редактировать `/etc/group` вручную, используйте команду `pw(8)` для добавления и редактирования групп. Например, для добавления группы, называемой `teamtwo`, и проверки ее существования вы можете использовать:

### Пример 13-8. Добавление группы с использованием `pw(8)`

```
# pw groupadd teamtwo
# pw groupshow teamtwo
teamtwo:*:1100:
```

Число 1100 это ID группы `teamtwo`. На данный момент в `teamtwo` нет членов, и поэтому она практически бесполезна. Давайте изменим эту ситуацию, добавив `jru` в группу `teamtwo`.

### Пример 13-9. Добавление пользователя в группу с использованием `pw(8)`

```
# pw groupmod teamtwo -M jru
# pw groupshow teamtwo
teamtwo:*:1100:jru
```

Аргумент к параметру `-m` это разделенный запятыми список пользователей, являющихся членами группы. Из предыдущих разделов мы знаем, что файл паролей также указывает группу для каждого пользователя. Пользователь автоматически добавляется системой к списку групп; пользователь не будет показан как член группы при использовании `pw(8) groupshow`, но эта информация будет показана при использовании `id(1)` или похожего инструмента. Другими словами, с этим параметром программа `pw(8)` работает только с файлом `/etc/group`; она никогда не будет пытаться получить дополнительную информацию из файла `/etc/passwd`.

### Пример 13-10. Использование `id(1)` для определения принадлежности к группам

```
% id jru
uid=1001(jru) gid=1001(jru) groups=1001(jru), 1100(teamtwo)
```

Как вы можете видеть, `jru` является членом групп `jru` и `teamtwo`.

За дальнейшей информацией о `pw(8)`, обратитесь к ее странице справочника, а за дополнительной информацией о формате файла `/etc/group` к странице справочника `group(5)`.

## Примечания

1. Конечно, пока не используется множество терминалов, но мы оставим эту тему для Гл. 20.

2. Возможно использование UID/GID вплоть до 4294967295, но эти ID могут вызвать серьезные проблемы с программами, делающими предположения о значениях ID.
3. С параметром `-s adduser(8)` не будет выводиться информацию. Позже для изменения настроек по умолчанию мы используем параметр `-v`.



# Глава 14. Безопасность

*Большая часть этой главы была взята из страницы справочника security(7) которую написал Matthew Dillon. Перевод на русский язык: Денис Пеплин.*

## 14.1. Краткое описание

Эта глава представляет введение в основные концепции безопасности системы, некоторые эмпирические правила и более подробно обращается к отдельным темам, касающимся FreeBSD. Большая часть затрагиваемых тем может быть применена к безопасности системы и безопасности в интернет вообще. Интернет больше не то "дружественное" место, где каждый хочет быть вам добрым соседом. Защита системы необходима для сохранения ваших данных, интеллектуальной собственности, времени и всего остального от хакеров и им подобных.

FreeBSD предоставляет массу утилит и механизмов для обеспечения целостности и безопасности системы и сети.

После прочтения этой главы вы узнаете:

- Основные концепции безопасности системы, специфику FreeBSD.
- О различных механизмах шифрования в FreeBSD, таких как DES и MD5.
- Как настроить аутентификацию с использованием одноразовых паролей.
- Как настроить TCP Wrappers для использования с inetd.
- Как настроить **KerberosIV** в релизах FreeBSD до 5.0.
- Как настроить **Kerberos5** в релизах FreeBSD после 5.0.
- Как создать межсетевые экраны с помощью IPFW.
- Как настроить IPsec и создать VPN между компьютерами на FreeBSD/Windows.
- Как настроить и использовать **OpenSSH**, реализацию SSH в FreeBSD.
- Что такое ACL и как их использовать.
- Как работать с сообщениями безопасности FreeBSD.

Перед чтением этой главы вам потребуется:

- Понимание основных концепций FreeBSD и интернет.

## 14.2. Введение

Безопасность это первая и основная функция системного администратора. Хотя все многопользовательские системы BSD UNIX уже снабжены некоторой защитой, работа по созданию и поддержке дополнительных механизмов безопасности, обеспечивающих защищенную работу пользователей, это одна из самых серьезных задач системного администратора. Компьютеры безопасны настолько, насколько вы сделаете их безопасными и требования безопасности всегда находятся в противоречии с удобством работы пользователей. Системы UNIX способны одновременно

работать с огромным количеством процессов и многие из этих процессов серверные — это означает, что с ними могут взаимодействовать внешние программы. Сегодня десктопы заменили мини-компьютеры и мэйнфреймы, и поскольку компьютеры в наши дни подключены к сети интернет, безопасность важна как никогда.

Наилучшая реализация системы безопасности представима в виде "послойной" системы. Вообще говоря все, что нужно сделать, это создать столько слоев безопасности, сколько необходимо и затем внимательно следить за вторжениями в систему. Не переусердствуйте в настройке системы безопасности, иначе она сделает невозможной обнаружение вторжений, являющееся одним из наиболее важных аспектов механизма безопасности. Например, нет большого смысла в установке флага `schg` (`chflags(1)`) на каждый исполняемый файл системы, поскольку хотя таким способом можно временно защитить исполняемые файлы, это мешает обнаружению факта взлома системы.

Безопасность системы также относится к различным формам атак, имеющих своей целью вызвать крах системы, или сделать систему недоступной другим способом, но не пытающихся получить доступ к учетной записи `root` ("break root"). Угрозы безопасности могут быть поделены на несколько категорий:

1. Отказ в обслуживании (Denial of service, DoS).
2. Взлом пользовательских учетных записей.
3. Взлом учетной записи `root` через доступные сервисы.
4. Взлом учетной записи `root` через учетные записи пользователей.
5. Создание backdoor.

Атака "отказ в обслуживании" отбирает у машины необходимые ресурсы. Обычно DoS атаки используют грубую силу, чтобы попытаться обрушить систему или сделать ее недоступной другим способом, превысив лимиты ее сервисов или сетевого стека. Некоторые DoS атаки пытаются использовать ошибки в сетевом стеке для обрушения системы одним пакетом. Эту проблему можно решить только исправив ядро системы. Атаки зачастую можно предотвратить правильной установкой параметров, ограничивающих нагрузку на систему в неблагоприятных условиях. С атаками, использующими грубую силу, бороться сложно. Например, атака с использованием пакетов с поддельными адресами, которую почти невозможно остановить, может быстро отключить вашу систему от интернет. Возможно, она не приведет к отказу системы, но сможет переполнить соединение с интернет.

Взлом учетной записи пользователя обычно встречается чаще, чем DoS атаки. Многие системные администраторы все еще используют стандартные сервисы **telnetd**, **rlogind** и **ftpd** на своих серверах. Эти сервисы по умолчанию не работают с зашифрованными соединениям. В результате при среднем количестве пользователей пароль одного или нескольких пользователей, входящих в систему через внешнее соединение (это обычный и наиболее удобный способ входа в систему), будет перехвачен. Внимательный системный администратор должен анализировать логи удаленного доступа на предмет подозрительных адресов пользователей даже в случае успешного входа.

Кто-то может предположить, что атакующий при наличии доступа к учетной записи пользователя может взломать учетную запись `root`. Однако, реальность такова, что в хорошо защищенной и поддерживаемой системе доступ к учетной записи пользователя не обязательно даст атакующему доступ к `root`. Разница между доступом к обычной учетной записи и к `root` важна, поскольку без доступа к `root` атакующий обычно не способен скрыть свои действия, и в худшем случае сможет лишь испортить файлы пользователя или вызвать крах системы. Взлом пользовательских учетных записей

встречается очень часто, поскольку пользователи заботятся о безопасности так, как системные администраторы.

Системные администраторы должны помнить, что существует множество потенциальных способов взлома учетной записи `root`. Атакующий может узнать пароль `root`, найти ошибку в сервисе, работающем с привилегиями и взломать учетную запись `root` через сетевое соединение с этим сервисом, или узнать об ошибке в `suid-root` программе, позволяющей атакующему взлом `root` с помощью взломанной учетной записи пользователя. Если атакующий нашел способ взлома `root`, ему может не понадобиться установка `backdoor`. Многие из обнаруженных и закрытых на сегодняшний день брешей в системе, позволяющие взлом `root`, требуют от атакующего серьезной работы по заметанию следов, поэтому большинство атакующих устанавливают `backdoor`. `Backdoor` предоставляет атакующему простой способ восстановления доступа к системе с привилегиями `root`, но также дает системному администратору удобный способ обнаружения вторжения. Устранение возможности установки `backdoor` возможно повредит безопасности системы, поскольку это не устраним брешь, позволившую проникнуть в систему.

Меры безопасности всегда должны быть реализованы многоуровнево, и могут быть классифицированы следующим образом:

1. Защита `root` и служебных учетных записей.
2. Защита работающих под `root` сервисов и `suid/sgid` исполняемых файлов.
3. Защита учетных записей пользователей.
4. Защита файла паролей.
5. Защита ядра, `raw` устройств и файловых систем.
6. Быстрое обнаружение несанкционированных изменений в системе.
7. Паранойя.

В следующем разделе этой главы эти темы изложены более подробно.

## 14.3. Защита FreeBSD

**Команда и протокол:** В этом документе мы будем использовать **выделенный** текст, упоминая приложение, и моноширинный шрифт, упоминая определенные команды. Для протоколов используется обычный шрифт. Это типографическое отличие полезно для таких случаев, как `ssh`, поскольку это и команда и протокол.

В последующем разделе будут рассмотрены методы защиты системы FreeBSD, упомянутые в предыдущем разделе этой главы.

### 14.3.1. Защита учетной записи `root` и служебных учетных записей

Во-первых, не беспокойтесь о защите служебных учетных записей, если не защищена учетная запись `root`. В большинстве систем у учетной записи `root` есть пароль. Использование пароля `root` опасно *всегда*. Это не означает, что вы должны удалить пароль. Пароль почти всегда необходим для доступа по консоли. Но это означает, что вы должны сделать невозможным использование пароля не из

консоли или может быть даже с помощью команды `su(1)`. Например, убедитесь, что псевдотерминалы в файле `/etc/ttys` перечислены с параметром `insecure`, что делает невозможным вход на них под `root` напрямую с помощью `telnet` или `rlogin`. При использовании других средств входа, таких как **sshd**, убедитесь что вход под `root` напрямую отключен и в них. Сделайте это, открыв файл `/etc/ssh/sshd_config`, и убедившись, что параметр `PermitRootLogin` установлен в `NO`. Проверьте каждый метод доступа — сервис `FTP` и ему подобные часто подвержены взлому. Прямой вход под `root` должен быть разрешен только с системной консоли.

Конечно, как системный администратор вы должны иметь доступ `root`, поэтому потребуется открыть несколько “лазеек”. Но убедитесь, что для доступа к ним необходим дополнительный пароль. Одним из способов доступа к `root` является добавление соответствующих учетных записей к группе `wheel` (в файле `/etc/group`). Это позволяет использовать `su` для доступа к `root`. Вы никогда не должны давать таким учетным записям доступ к `wheel` непосредственно, помещая их в группу `wheel` в файле паролей. Служебные учетные записи должны помещаться в группу `staff`, а затем добавляться к группе `wheel` в файле `/etc/group`. Только те члены группы `staff`, которым действительно нужен доступ к `root`, должны быть помещены в группу `wheel`. При работе с такими методами аутентификации как `Kerberos`, возможно также использование файла `.k5login` в каталоге пользователя `root` для доступа к учетной записи `root` с помощью `ksu(1)` без помещения кого-либо в группу `wheel`. Это решение возможно лучше, поскольку механизм `wheel` все еще позволяет взлом `root`, если злоумышленник получил копию файла паролей и смог взломать служебную учетную запись. Хотя использование механизма `wheel` лучше, чем работа через `root` напрямую, это не обязательно самый безопасный способ.

Непрямой способ защиты служебных учетных записей и конечно `root` это использование альтернативных методов доступа и замена зашифрованных паролей на символ `“*”`. Используя команду `vipw(8)`, замените каждый зашифрованный пароль служебных учетных записей на этот символ для запрета входа с аутентификацией по паролю. Эта команда обновит файл `/etc/master.passwd` и базу данных пользователей/паролей.

Служебная учетная запись вроде этой:

```
foobar:R9DT/Fal/LV9U:1000:1000::0:0:Foo Bar:/home/foobar:/usr/local/bin/tcsh
```

Должна быть заменена на такую:

```
foobar:*:1000:1000::0:0:Foo Bar:/home/foobar:/usr/local/bin/tcsh
```

Это изменение предотвратит обычный вход, поскольку зашифрованный пароль никогда не совпадет с `“*”`. После этого члены группы `staff` должны использовать другой механизм аутентификации, например `kerberos(1)` или `ssh(1)` с парой ключей: публичным и приватным. При использовании такой системы как `Kerberos`, потребуется защитить сервер `Kerberos` и рабочую станцию. При использовании пары публичного/приватного ключей с `ssh`, потребуется защитить компьютер, с которого происходит вход (обычно это рабочая станция). Дополнительный слой защиты может быть добавлен путем защиты пары ключей при создании их с помощью `ssh-keygen(1)`. Возможность заменить пароли служебных учетных записей на `“*”` гарантирует также, что вход может быть осуществлен только через защищенные методы доступа, которые вы настроили. Это принуждает всех членов `staff` использовать защищенные, шифрованные соединения для всех входов, что закрывает большую брешь, используемую многими нарушителями: перехват паролей с другого, слабо защищенного компьютера.

Более не прямой механизм безопасности предполагает, что вы входите с более защищенного сервера на менее защищенный. Например, если главный сервер работает со всеми сервисами, рабочая станция не должна работать ни с одним. Для поднятия уровня безопасности до приемлемого уровня, число

запущенных на ней сервисов необходимо сократить до минимума, вплоть до отключения их всех, кроме того необходимо использовать защищенный паролем хранитель экрана. Конечно, при наличии физического доступа к рабочей станции атакующий может взломать любую систему безопасности. Это определенно проблема, которую вы должны учитывать, но учтите также тот факт, что большинство взломов совершаются удаленно, через сеть, людьми, которые не имеют физического доступа к вашим рабочим станциям или серверам.

Использование такой системы как Kerberos дает возможность заблокировать или изменить пароль в одном месте, что сразу отразится на всех компьютерах, где существует служебная учетная запись. Если эта учетная запись будет взломана, возможность немедленно изменить пароль на всех компьютерах нельзя недооценивать. Без этой возможности изменение паролей на N машинах может стать проблемой. Вы можете также наложить ограничения на смену паролей с помощью Kerberos: не только установить значения timeout в Kerberos, но и добавить требование смены пароля пользователем после определенного периода времени (скажем, раз в месяц).

### 14.3.2. Защита работающих под root сервисов и suid/sgid исполняемых файлов

Предусмотрительный системный администратор запускает только те сервисы, в которых нуждается, ни больше ни меньше. Учитывайте, что сервисы сторонних разработчиков наиболее подвержены ошибкам. К примеру, работа со старыми версиями **imapd** или **popper** это все равно что раздача доступа root всему миру. Никогда не запускайте сервисы, которые вы не проверили достаточно внимательно. Многим сервисам не требуется работа под root. Например, демоны **ntalk**, **comsat**, и **finger** могут быть запущены в так называемых *песочницах* (*sandboxes*). Песочница это не идеальное решение, поскольку вызывает много проблем, но она подходит под модель послойной безопасности: если кто-то сможет взломать сервис, работающий в песочнице, ему потребуется взломать еще и саму песочницу. Чем больше уровней ("слоев") потребуется пройти атакующему, тем меньше вероятность его успеха. Ошибки, позволяющие получать root доступ, находили фактически во всех сервисах, запускаемых под root, включая основные системные сервисы. Если вы обслуживаете машину, на которую входят только через **sshd** и никогда не входят через **telnetd**, **rshd** или **rlogind**, отключите эти сервисы!

В FreeBSD сервисы **ntalkd**, **comsat** и **finger** теперь по умолчанию работают в "песочнице". Другая программа, которая может быть кандидатом на запуск в "песочнице" это **named(8)**. `/etc/defaults/rc.conf` включает необходимые для запуска **named** в "песочнице" аргументы в закомментированной форме. В зависимости от того, устанавливаете ли вы новую систему, или обновляете старую, учетные записи пользователей, используемые этими "песочницами" могут не быть созданы. Предусмотрительный системный администратор должен узнать о "песочницах" для сервисов и установить их если есть возможность.

Есть множество других сервисов, которые обычно не работают в "песочницах": **sendmail**, **popper**, **imapd**, **ftpd**, и другие. Некоторым из этих сервисов есть альтернативы, но их установка может потребовать больше работы, чем вы готовы выполнить (фактор удобства). Вы можете запустить эти сервисы под root и положиться на другие механизмы обнаружения вторжений, которые могут пройти через них.

Другая большая потенциальная root брешь в системе это suid-root и sgid исполняемые файлы. Большинство этих исполняемых файлов, таких как **rlogin**, установлены в `/bin`, `/sbin`, `/usr/bin`, или `/usr/sbin`. Хотя ничто не может быть безопасно на 100%, находящиеся по умолчанию в системе suid и sgid исполняемые файлы могут быть признаны достаточно безопасными. Но root бреши все еще обнаруживаются в этих исполняемых файлах. root брешь, обнаруженная в xlib в 1998 делала **xterm** (который обычно suid) подверженным взлому. Лучше сразу принять меры предосторожности, чем

сожалеть потом. Предусмотрительный системный администратор ограничит права запуска `suid` исполняемых файлов, которые должны запускаться пользователями группы `staff`, только этой группой, а также запретит доступ (`chmod 000`) к тем исполняемым файлам `suid`, которые никем не используются. Серверу без монитора обычно не требуется исполняемый файл **xterm**. Исполняемые `sgid` исполняемые файлы могут быть почти так же опасны. Если нарушитель сможет взломать `sgid-kmem` исполняемый файл, он возможно сможет прочесть `/dev/kmem` и таким образом получить файл зашифрованных паролей, что потенциально делает возможным взлом любой защищенной паролем учетной записи. Аналогично нарушитель, проникший в группу `kmem`, может отслеживать последовательности клавиш, отправленные через псевдотерминалы, включая псевдотерминалы, используемые через безопасные соединения. Нарушитель, вошедший в группу `tty` может сделать вывод почти на любой пользовательский терминал. Если пользователь работает с терминальной программой или эмулятором с возможностью эмуляции клавиатуры, взломщик может потенциально стенировать поток данных, который заставит терминал пользователя ввести команду, и она будет запущена с правами этого пользователя.

### 14.3.3. Защита учетных записей пользователей

Учетные записи пользователей обычно сложнее всего защитить. Вы можете ввести драконовские ограничения доступа к служебным учетным записям, заменив их пароли на символ `"*"`, но возможно не сможете сделать то же с обычными учетными записями пользователей. Если есть такая возможность, вы возможно сможете защитить учетные записи пользователей соответствующим образом. Если нет, просто более бдительно отслеживайте эти учетные записи. Использование `ssh` и `Kerberos` для учетных записей пользователей более проблематично, поскольку требует дополнительной административной работы и технической поддержки, но все же это решение лучше, чем файл с шифрованными паролями.

### 14.3.4. Защита файла паролей

Единственный абсолютно надежный способ это замена на `*` максимально возможного количества паролей и использование `ssh` или `Kerberos` для доступа к таким учетным записям. Хотя файл с шифрованными паролями (`/etc/spwd.db`) доступен для чтения только `root`, возможно, что нарушитель сможет получить доступ на чтение к этому файлу, даже если не получит права `root` на запись.

Ваши скрипты безопасности должны всегда проверять и составлять отчет об изменениях файла паролей (обратитесь к разделу Проверка целостности файлов ниже по тексту).

### 14.3.5. Защита ядра, `raw` устройств и файловых систем

Если атакующий взламывает `root`, он сможет сделать практически все, но есть способы усложнить его задачу. Например, в большинстве современных ядер встроено устройство перехвата пакетов. В `FreeBSD` оно называется `brf`. Нарушитель обычно пытается запустить перехват пакетов на взломанной машине. Вы не должны предоставлять ему такой возможности, на большинстве систем устройство `brf` не должно быть встроено в ядро.

Но даже если вы выключите устройство `brf`, все еще остаются проблемы, связанные с устройствами `/dev/mem` и `/dev/kmem`. Нарушитель все еще может писать на дисковые `raw` устройства. Есть также другая возможность ядра, загрузка модулей, `kldload(8)`. Активный нарушитель может использовать

KLD модуль для установки собственного устройства `brf` или другого перехватывающего устройства на работающее ядро. Для решения этих проблем запускайте ядро с большим уровнем безопасности, как минимум 1. Уровень безопасности может быть установлен с помощью `sysctl` через переменную `kern.securelevel`. После установки уровня безопасности в 1 доступ на запись в `raw` устройства будет запрещена и полностью заработают специальные флаги `chflags`, такие как `schg`. Убедитесь также, что флаг `schg` установлен на критически важных загрузочных исполняемых файлах, каталогах и файлах скриптов — на всем, что запускается до установки уровня безопасности. Это требует большого объема работы, и обновление системы на более высоком уровне безопасности может стать гораздо сложнее. Вы можете пойти на компромисс и запускать систему на высоком уровне безопасности, но не устанавливать флаг `schg` для каждого существующего системного файла и каталога. Другая возможность состоит в монтировании `/` и `/usr` только для чтения. Необходимо заметить, что такие правила слишком жесткие и могут помешать обнаружению вторжения.

### 14.3.6. Проверка целостности файлов: исполняемые, конфигурационные файлы и т.д.

Вы можете защищать только ядро, файлы настройки и управления системой только до тех пор, пока эта защита не вступит в конфликт с удобством работы в системе. Например, использование `chflags` для установки бита `schg` на большинство файлов в `/` вероятно может только навредить, поскольку хотя и может защитить файлы, препятствует обнаружению. Последний слой системы безопасности, возможно, наиболее важный — обнаружение. Остальные меры безопасности практически бесполезны (или, что еще хуже, могут дать вам ложное ощущение безопасности) если вы не обнаружите потенциальное вторжение. Половина функций системы безопасности направлена на замедление атакующего, а не на его остановку, для того, чтобы дать системе обнаружения возможность поймать нарушителя на месте преступления.

Лучший способ обнаружения вторжения — отслеживание измененных, отсутствующих, или неожиданно появившихся файлов. Для наблюдения за измененными файлами лучше всего использовать другую (зачастую централизованную) систему с ограниченным доступом. Добавление написанных вами скриптов к этой дополнительно защищенной системе с ограниченным доступом делает ее практически невидимой для потенциальных взломщиков, и это важно. В целях достижения максимального эффекта вам может потребоваться предоставить этой системе доступ к другим машинам в сети, обычно с помощью NFS экспорта только для чтения или сгенерировав пары ключей `ssh` для доступа к другим машинам по `ssh`. Помимо большого объема сетевого трафика, NFS более скрытый метод — он позволяет контролировать файловые системы на каждом клиентском компьютере практически незаметно. Если ваш сервер с ограниченным доступом подключен к клиентским компьютерам через коммутатор, NFS метод это зачастую лучший выбор. При соединении через концентратор, или через несколько маршрутизаторов, NFS метод может стать слишком небезопасным и использование `ssh` может стать лучшим выбором даже несмотря на то, что `ssh` оставляет следы своей работы.

Как только у вас появился сервер с ограниченным доступом, и как минимум доступ на чтение в клиентских системах, потребуется написать скрипты для выполнения мониторинга. При наличии доступа по NFS вы можете написать скрипты с помощью простых системных утилит, таких как `find(1)` и `md5(1)`. Лучше всего подсчитывать `md5` файлов на клиентском компьютере как минимум один раз в день, а файлы, контролируемые запуск из `/etc` и `/usr/local/etc` даже более часто. При обнаружении расхождений в `md5`, контролирующий компьютер должен просигнализировать системному администратору проверить изменившиеся файлы. Хороший скрипт безопасности проверит также

наличие несоответствующих исполняемых `suid` файлов и новых или измененных файлов в системных разделах `/` и `/usr`.

При использовании `ssh` вместо `NFS`, написать скрипты безопасности гораздо сложнее. Вам обязательно потребуется скопировать (`scp`) скрипты на клиентский компьютер, сделать их невидимыми, и для безопасности потребуется также скопировать исполняемые файлы (такие как `find`), которые будут использоваться скриптом. Приложение `ssh` на клиентском компьютере может быть уже взломано. В конечном итоге, без `ssh` не обойтись при работе через небезопасные соединения, но его гораздо сложнее использовать.

Хороший скрипт безопасности проверит также изменения в файлах настройки, работающих при подключении пользователей и служебных учетных записей: `.rhosts`, `.shosts`, `.ssh/authorized_keys` и так далее... файлы, которые могли не попасть в область проверки MD5.

Если для пользователей выделен большой объем дискового пространства, проверка каждого файла на таких разделах может занять слишком много времени. В таком случае установка флагов монтирования для запрета `suid` исполняемых файлов и устройств на таких разделах это хорошая идея. Примените параметры `mount(8)` `nodev` и `nosuid`. Проверяйте эти разделы в любом случае, хотя бы раз в неделю, поскольку необходимо обнаруживать попытки взлома, независимо от того, эффективны они или нет.

Учет процессов (`accton(8)`) это относительно несложная возможность операционной системы, которая может помочь как механизм обнаружения состоявшихся вторжений. Она особенно полезна для обнаружения пути проникновения нарушителя в систему, если файл не был затронут проникновением.

Наконец, скрипты безопасности должны обработать лог файлы, которые необходимо создавать настолько защищенным способом, насколько это возможно — подключение `syslog` удаленно может быть очень полезным. Злоумышленник попытается уничтожить следы взлома, и лог файлы критически важны для системного администратора, пытающегося отследить время и метод первого проникновения. Один из надежных способов получения лог файлов является подключение системной консоли к последовательному порту и постоянный сбор информации через защищенную машину, отслеживающую консоли.

### 14.3.7. Паранойя

Немного паранойи никогда не повредит. Как правило, системный администратор может добавлять элементы безопасности в любом количестве, пока это не влияет на удобство, а также некоторое количество элементов безопасности, *вливающих* на удобство. Что даже более важно, системный администратор должен немного изменить их — если вы используете рекомендации, например те, что даны в этом документе, они становятся известны атакующему, который также имеет доступ к этому документу. *prospective attacker who also has access to this document.*

### 14.3.8. Атаки DoS

Этот раздел охватывает DoS атаки. DoS атаки это обычно пакетные атаки. Хотя против современной атаки с подделкой пакетов, которая перегружает сеть, мало что можно сделать, вы можете ограничить повреждения, убедившись, что атака не может обрушить ваши сервера.

1. Ограничение количества порождаемых процессов.



2. Уменьшение последствий springboard атак (ICMP ответ, широковещательный ping и т.д.).

3. Кэш маршрутизации ядра.

Обычная DoS атака против порождающего процессы сервера пытается исчерпать ресурсы сервера по процессам, файловым дескрипторам и памяти до тех пор, пока машина не "подвиснет". У **inetd** (обратитесь к `inetd(8)`) есть несколько параметров, позволяющих ограничить такие атаки. Необходимо учесть, что хотя можно предотвратить падение системы, в общем случае невозможно предотвратить прекращение работы сервиса. Внимательно прочтите страницу справочника и обратите особое внимание на параметры `-s`, `-S`, и `-R`. Учтите, что параметр `-s` не работает в случае атак с использованием поддельных IP пакетов, поэтому как правило необходимо использование комбинации параметров. Некоторые standalone сервисы используют собственные параметры, ограничивающие порождение процессов.

У **Sendmail** есть собственный параметр `-OMaxDaemonChildren`, которая работает гораздо лучше, чем параметр `sendmail`, ограничивающий нагрузку. Вам необходимо задать параметр запуска **sendmail** `MaxDaemonChildren` достаточно большим, чтобы обслуживать ожидаемую нагрузку, но так, чтобы компьютер мог обслужить такое количество приложений **sendmail** без падения системы. Хорошей мерой является запуск `sendmail` в режиме очереди (`-ODeliveryMode=queued`) и запуск даемона (`sendmail -bd`) отдельно от очереди (`sendmail -q15m`). Если вы все же хотите организовать доставку в режиме реального времени, запускайте очередь с меньшим интервалом `-q1m`, но убедитесь в правильной установке параметра `sendmail MaxDaemonChildren` для предотвращения ошибок.

**Syslogd** может быть атакован непосредственно, настоятельно рекомендуется использовать параметр `-s` если это возможно и параметр `-a` в остальных случаях.

Вы также должны быть очень осторожны с сервисами, совершающими обратное подключение, такими как **tcpwrapper** с `reverse-identd`, который может быть атакован непосредственно. По этой причине возможность **tcpwrappers** `reverse-ident` обычно не следует использовать.

Правильным будет запрет доступа к внутренним сервисам из внешней сети путем соответствующей настройки межсетевого экрана на внешнем маршрутизаторе. Идея в том, чтобы предотвратить перегрузку сервисов атаками из внешней сети, а кроме того защитить `root` от взлома через сеть. Всегда настраивайте исключающий межсетевой экран, т.е. "закрыть все *кроме* портов A, B, C, D, и M-Z". Этим способом вы можете закрыть все порты нижнего диапазона, кроме явно указанных, таких как **named** (если вы поддерживаете интернет-зону), **ntalkd**, **sendmail**, и других сервисов, доступных из интернет. Если вы попытаетесь настроить межсетевой экран другим способом — включающий, или разрешающий межсетевой экран, есть большой шанс забыть "закрыть" пару сервисов, или добавить новый внутрисетевой сервис и забыть обновить межсетевой экран. Вы можете открыть диапазон портов с большими номерами для обычных приложений без угрозы портам нижнего диапазона. Учтите также, что FreeBSD позволяет вам контролировать диапазоны портов, используемые для динамической привязки через различные переменные `sysctl net.inet.ip.portrange` (`sysctl -a | fgrep portrange`), что позволяет упростить настройку межсетевого экрана. Например, вы можете использовать обычный диапазон портов со значениями от 4000 до 5000, и диапазон портов с большими номерами от 49152 до 65535, а затем заблокировать все до 4000 порта (конечно оставив доступ из интернет к определенным портам).

Другой распространенный тип DoS атак называется springboard — сервер атакуется таким образом, что генерируемые ответы перегружают его, локальную сеть или какие-то другие компьютеры.

Наиболее распространенная атака этого вида это *широковещательная ICMP ping атака*. Атакующий подделывает пакеты ping, подставляя IP адрес машины, которую он намеревается атаковать, и отправляет их на широковещательный адрес вашей локальной сети. Если ваш внешний

маршрутизатор не настроен на отбрасывание пакетов `ping` на широковещательные адреса, ваша сеть начинает генерировать соответствующие ответы на поддельный адрес, что приводит к перегрузке хоста-жертвы, особенно если атакующий использует этот же трюк с множеством широковещательных адресов в множестве сетей одновременно. Были зарегистрированы широковещательные атаки свыше ста двадцати мегабит. Другая распространенная `springboard` атака направлена на ICMP систему сообщения об ошибках. Конструируя пакеты, вызывающие ICMP сообщения об ошибках, атакующий может нагрузить входящее соединение сервера и вынудить сервер нагрузить исходящее соединение ICMP ответами. Этот тип атаки может также обрушить сервер, когда тот исчерпает `mbuf`, обычно если сервер не может ограничить число ответов ICMP, когда они генерируются слишком быстро. В ядре FreeBSD есть новая опция сборки, `ICMP_BANDLIM`, которая ограничивает эффективность этого типа атак. Последний основной класс `springboard` атак относится к определенным внутренним сервисам `inetd`, таким как сервис `udp echo`. Атакующий просто подделывает адрес источника и адрес назначения UDP пакетов, устанавливая в их качестве соответственно `echo` порт сервера А и В, оба этих сервера принадлежат вашей локальной сети. Эти два сервера начинают перебрасываться этим пакетом друг с другом. Атакующий может вызвать перегрузку обоих серверов и их сетей, просто отправив несколько пакетов таким способом. Аналогичные проблемы существуют с портом **chargen**. Компетентный системный администратор должен отключить эти тестовые сервисы `inetd`.

Атаки с поддельными пакетами могут также использоваться для переполнения кэша маршрутизации ядра. Обратитесь к параметрам `sysctl net.inet.ip.rtexpire, rtminexpire, и rtmaxcache`. Атака с поддельными пакетами, использующая произвольный IP адрес источника, заставит ядро сгенерировать временный кэшированный маршрут в таблице маршрутизации, который можно увидеть с помощью `netstat -rna | fgrep w3`. Эти маршруты обычно удаляются через 1600 секунд или около того. Если ядро определит, что кэшированная маршрутная таблица стала слишком большой, оно динамически уменьшит `rtexpire`, но никогда не станет делать его меньше чем `rtminexpire`. С этим связаны две проблемы:

1. Ядро не отреагирует достаточно быстро, когда легко нагруженный сервер будет внезапно атакован.
2. Значение `rtminexpire` недостаточно мало для поддержки работоспособности в условиях продолжительной атаки.

Если ваши серверы подключены к интернет через линию T3 или более быструю, предусмотрительно будет изменить оба значения `rtexpire` и `rtminexpire` с помощью `sysctl(8)`. Никогда не устанавливайте ни один из этих параметров в нуль (если только вы не хотите обрушить систему). Установка обоих параметров в значение 2 секунды должна предотвратить таблицу маршрутизации от атак.

### 14.3.9. Проблемы, связанные с доступом к Kerberos и SSH

При использовании Kerberos и ssh необходимо учесть несколько возможных проблем. Kerberos V это отличный протокол аутентификации, но в адаптированных к нему приложениях **telnet** и **rlogin** есть несколько ошибок, которые могут сделать их непригодными к работе с бинарными потоками. К тому же, по умолчанию Kerberos не шифрует сессию, если вы не используете параметр `-x`. **ssh** шифрует все по умолчанию.

ssh работает очень хорошо во всех ситуациях, но пересылает ключи по умолчанию. Это означает, что если вы работаете с защищенной рабочей станции, ключи на которой дают доступ к остальной сети, и заходите по ssh на незащищенный компьютер, эти ключи могут быть использованы для взлома.

Атакующему не удастся получить сами ключи, но поскольку ssh открывает порт во время входа в систему, то если на незащищенной машине взломан root, эти ключи могут быть использованы для доступа к другим компьютерам, на которых они действуют.

Мы рекомендуем использовать ssh в комбинации с Kerberos для служебных учетных записей если это возможно. **ssh** может быть собран с поддержкой Kerberos. Это уменьшает зависимость от потенциально подверженных взлому ssh ключей, и в то же время защищает пароли через Kerberos. Ключи ssh должны использоваться только для работы скриптов на защищенных компьютерах (там, где Kerberos использовать не получится). Мы также рекомендуем или выключить передачу ключей в настройках ssh, или использовать параметр `from=IP/DOMAIN`, поддерживаемый ssh в файле `authorized_keys`, который позволяет использовать ключи только с определенных компьютеров.

## 14.4. DES, MD5, и шифрование

*Частично переписал и обновил Bill Swingle.*

У каждого пользователя UNIX системы есть пароль, связанный с его учетной записью. Очевидно, что эти пароли должны быть известны только пользователю и соответствующей операционной системе. Для защиты паролей они шифруются способом, известным как "односторонний хэш", то есть их можно легко зашифровать, но нельзя расшифровать. Другими словами, то, что мы сказали чуть раньше было очевидно, но не совсем верно: операционной системе *сам пароль* неизвестен. Ей известен только пароль в *зашифрованной* форме. Единственный способ получить "обычный" пароль это простой перебор всех возможных паролей.

К сожалению, единственный способ шифрования пароля при появлении UNIX был основан на DES, Data Encryption Standard. Это не было проблемой для пользователей, живущих в США, но поскольку исходный код DES нельзя было экспортировать из США, FreeBSD нашла способ одновременно не нарушать законов США и сохранить совместимость со всеми другими вариантами UNIX, где все еще использовался DES.

Решение было в разделении библиотек шифрования, чтобы пользователи в США могли устанавливать и использовать библиотеки DES, а у остальных пользователей был метод шифрования, разрешенный к экспорту. Так FreeBSD пришла к использованию MD5 в качестве метода шифрования по умолчанию. MD5 считается более безопасным, чем DES, поэтому установка DES рекомендуется в основном из соображений совместимости.

### 14.4.1. Определения механизма шифрования

До FreeBSD 4.4 `libcrypt.a` была символической ссылкой на библиотеку, используемую для шифрования. В FreeBSD 4.4 `libcrypt.a` была изменена для предоставления настраиваемой библиотеки аутентификации по хэшу пароля. На данный момент библиотека поддерживает хэши DES, MD5 и Blowfish. По умолчанию FreeBSD использует для шифрования паролей MD5.

Довольно легко определить какой метод шифрования используется в FreeBSD. Один из способов это проверка файла `/etc/master.passwd`. Пароли, зашифрованные в хэш MD5 длиннее, чем те, что зашифрованы с помощью DES и начинаются с символов `$1$`. Пароли, начинающиеся с символов `$2a$` зашифрованы с помощью Blowfish. Пароли, зашифрованные DES не содержат каких-то определенных идентифицирующих символов, но они короче, чем пароли MD5 и закодированы в 64-символьном

алфавите, не содержащем символа \$, поэтому относительно короткая строка, не начинающаяся с этого символа это скорее всего DES пароль.

Формат паролей, используемых для новых паролей, определяется параметром `passwd_format` в `/etc/login.conf`, которое может принимать значения `des`, `md5` или `blf`. Обратитесь к странице справочника `login.conf(5)` за дополнительной информацией о параметрах `login`.

## 14.5. Одноразовые пароли

S/Key это схема с одноразовыми паролями, основанная на одностороннем хэше. FreeBSD использует хэш MD4 для совместимости, но другие системы используют MD5 и DES-MAC. S/Key была частью базовой системы FreeBSD начиная с версии 1.1.5 и используется также во все большем числе операционных систем. S/Key это зарегистрированная торговая марка Bell Communications Research, Inc.

Начиная с FreeBSD версии 5.0, S/Key была замещена на функциональный эквивалент — OPIE (One-time Passwords In Everything). OPIE по умолчанию использует MD5.

Есть три различных вида паролей, о которых мы поговорим ниже. Первый вид это ваш обычный пароль UNIX или пароль Kerberos; мы будем называть его "пароль UNIX". Второй вид это одноразовый пароль, сгенерированный программой S/Key `key` или программой OPIE `opiekey(1)` и принимаемый командами `keyinit` или `opiepasswd(1)` и в приглашении `login`; мы будем называть их "одноразовыми паролями". Последний вид паролей это защищенные пароли, которые вы передаете программам `key/opiekey` (и иногда программам `keyinit/opiepasswd`), и которые эти программы используют для создания одноразовых паролей; мы будем называть его "защищенными паролями" или просто "паролями".

Защищенный пароль не имеет никакого отношения к вашему паролю UNIX; они могут быть одинаковыми, но это не рекомендуется. Защищенные пароли S/Key и OPIE не ограничены 8-ю символами, как старые UNIX пароли<sup>1</sup>, они могут быть настолько длинными, насколько вы захотите. Очень часто используются пароли длиной в шесть или семь символов. По большей части система S/Key или OPIE работает полностью независимо от системы паролей UNIX.

Помимо паролей, есть два других вида данных, важных для S/Key и OPIE. Первый, известный как "seed" или "ключ", состоит из двух букв и пяти цифр. Другой, называемый "счетчиком цикла", это номер от 1 до 100. S/Key создает одноразовый пароль, соединяя ключ и защищенный пароль, а затем применяя MD4/MD5 столько раз, сколько указано счетчиком цикла и выдает результат в виде шести коротких слов на английском. Эти шесть слов на английском и есть ваш одноразовый пароль. Система аутентификации (как правило PAM) хранит последний использованный одноразовый пароль, и пользователь аутентифицируется если хэш вводимого пользователем пароля совпадает с предыдущим паролем. Поскольку используется односторонний хэш, невозможно сгенерировать следующий одноразовый пароль если получен предыдущий; счетчик цикла уменьшается после каждого успешного входа для поддержки синхронизации пользователя с программой `login`. Когда счетчик цикла уменьшается до 1, S/Key и OPIE должны быть переинициализированы.

В каждой из обсуждаемых ниже систем задействованы три программы. Программы `key` и `opiekey` получают счетчик цикла, ключ и защищенный пароль и создают одноразовый пароль или последовательный список одноразовых паролей. Программы `keyinit` и `opiepasswd` используются для инициализации S/Key и OPIE соответственно, и для смены паролей, счетчиков цикла, или ключей; они принимают защищенный пароль или счетчик цикла, ключ и одноразовый пароль. Программы `keyinfo`

и `opieinfo` проверяют соответствующие файлы (`/etc/skeykeys` или `/etc/opiekeys`) и печатают текущий счетчик цикла и ключ вызывающего пользователя.

Мы рассмотрим четыре вида операций. Первая это использование `keyinit` или `opiepasswd` через защищенное соединение для первоначальной настройки системы одноразовых паролей, или для изменения пароля или ключа. Вторая операция это использование в тех же целях `keyinit` или `opiepasswd` через незащищенное соединение, в сочетании с `key` или `opiekey` через защищенное соединение. Третья это использование `key/opiekey` для входа через незащищенное соединение. Четвертая это использование `key` или `opiekey` для генерации набора ключей, которые могут быть записаны или распечатаны для соединения из места, где защищенное соединение недоступно.

### 14.5.1. Защищенная установка соединения

Для первоначальной настройки S/Key, измените ваш пароль или ключ при входе через защищенное соединение (например, с консоли компьютера или через **ssh**), используйте команду `keyinit` без параметров при входе под своим именем:

```
% keyinit
Adding unfurl:
Reminder - Only use this method if you are directly connected.
If you are using telnet or rlogin exit with no password and use keyinit -s.
Enter secret password:
Again secret password:

ID unfurl s/key is 99 to17757
DEFY CLUB PRO NASH LACE SOFT
```

Для OPIE, вместо этого используется `opiepasswd`:

```
% opiepasswd -c
[grimreaper] ~ $ opiepasswd -f -c
Adding unfurl:
Only use this method from the console; NEVER from remote. If you are using
telnet, xterm, or a dial-in, type ^C now or exit with no password.
Then run opiepasswd without the -c parameter.
Using MD5 to compute responses.
Enter new secret pass phrase:
Again new secret pass phrase:
ID unfurl OTP key is 499 to4268
MOS MALL GOAT ARM AVID COED
```

В приглашениях `Enter new secret pass phrase:` или `Enter secret password:`, введите пароль или фразу. Запомните, это не тот пароль, с которым вы будете входить, он используется для генерации одноразовых паролей. Строка `ID` содержит информацию для вашего конкретного случая: имя пользователя, счетчик цикла и ключ. При входе система запомнит эти параметры и отправит их вам, поэтому их не надо запоминать. В последней строке находится одноразовый пароль, соответствующий этим параметрам и секретному паролю; если вы войдете в систему сразу, используйте этот одноразовый пароль.

## 14.5.2. Незащищенная установка соединения

Для инициализации или изменения защищенного пароля через незащищенное соединение, вам потребуется существующее защищенное соединение куда-то, где вы сможете запустить `key` или `opiekey`; это может быть средство доступа Macintosh или shell на компьютере, которому вы доверяете. Вам потребуется также установить значение счетчика цикла (100 возможно подойдет), и задать ключ или использовать сгенерированный. Через незащищенное соединение (к компьютеру, на котором производится настройка), используйте команду `keyinit -s`:

```
% keyinit -s
Updating unfurl:
Old key: tol17758
Reminder you need the 6 English words from the key command.
Enter sequence count from 1 to 9999: 100
Enter new key [default tol17759]:
s/key 100 to 17759
s/key access password:
s/key access password:CURE MIKE BANE HIM RACY GORE
```

Для OPIE, используйте `opiepasswd`:

```
% opiepasswd

Updating unfurl:
You need the response from an OTP generator.
Old secret pass phrase:
    otp-md5 498 to4268 ext
    Response: GAME GAG WELT OUT DOWN CHAT
New secret pass phrase:
    otp-md5 499 to4269
    Response: LINE PAP MILK NELL BUOY TROY

ID mark OTP key is 499 gr4269
LINE PAP MILK NELL BUOY TROY
```

Чтобы принять ключ по умолчанию нажмите **Enter**. Затем, перед вводом пароля доступа введите те же параметры в вашем защищенном соединении или средстве доступа S/Key:

```
% key 100 tol17759
Reminder - Do not use this program while logged in via telnet or rlogin.
Enter secret password: <secret password>
CURE MIKE BANE HIM RACY GORE
```

Или для OPIE:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Теперь переключитесь на незащищенное соединение и скопируйте одноразовый пароль, сгенерированный соответствующей программой.

### 14.5.3. Создание одного одноразового пароля

Как только вы настроите S/Key или OPIE, во время входа появится приглашение вроде этого:

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.

FreeBSD/i386 (example.com) (ttya)

login: <username>
s/key 97 fw13894
Password:
```

Или для OPIE:

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.

FreeBSD/i386 (example.com) (ttya)

login: <username>
otp-md5 498 gr4269 ext
Password:
```

Кроме того, у S/Key и OPIE есть полезная особенность (не показанная здесь): если вы нажмете **Enter** в приглашении на ввод пароля, включится эхо, и вы сможете увидеть то, что вводите. Это может быть очень полезно, если вы пытаетесь ввести пароль вручную, например с распечатки.

В этот момент вам потребуется сгенерировать одноразовый пароль, чтобы ввести его в приглашение. Это должно быть выполнено на защищенной системе, в которой вы можете запустить key или opiekey (есть версии для DOS, Windows и Mac OS). Им требуются значения счетчика цикла и ключ в качестве параметров командной строки. Вы можете скопировать и вставить их прямо из приглашения login компьютера, на который входите.

В защищенной системе:

```
% key 97 fw13894
Reminder - Do not use this program while logged in via telnet or rlogin.
Enter secret password:
WELD LIP ACTS ENDS ME HAAG
```

Для OPIE:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Теперь, когда у вас есть одноразовый пароль, можете продолжить вход в систему:

```
login: <username>
s/key 97 fw13894
Password: <return to enable echo>
s/key 97 fw13894
Password [echo on]: WELD LIP ACTS ENDS ME HAAG
Last login: Tue Mar 21 11:56:41 from 10.0.0.2 ...
```

#### 14.5.4. Создание нескольких одноразовых паролей

Иногда вы отправляетесь туда, где нет доступа к защищенному компьютеру или защищенному соединению. В этом случае, можно использовать команды `key` и `opiekey` для создания нескольких одноразовых паролей, которые вы сможете распечатать и забрать с собой. Например:

```
% key -n 5 30 zz99999
Reminder - Do not use this program while logged in via telnet or rlogin.
Enter secret password: <secret password>
26: SODA RUDE LEA LIND BUDD SILT
27: JILT SPY DUTY GLOW COWL ROT
28: THEM OW COLA RUNT BONG SCOT
29: COT MASH BARR BRIM NAN FLAG
30: CAN KNEE CAST NAME FOLK BILK
```

Или для OPIE:

```
% opiekey -n 5 30 zz99999
Using the MD5 algorithm to compute response.
Reminder: Don't use opiekey from telnet or dial-in sessions.
Enter secret pass phrase: <secret password>
26: JOAN BORE FOSS DES NAY QUIT
27: LATE BIAS SLAY FOLK MUCH TRIG
28: SALT TIN ANTI LOON NEAL USE
29: RIO ODIN GO BYE FURY TIC
30: GREW JIVE SAN GIRD BOIL PHI
```

Параметр `-n 5` запрашивает пять паролей, `30` указывает значение последнего счетчика цикла. Обратите внимание, что пароли печатаются в *обратном* по сравнению с обычным использованием порядке. Если вы действительно параноик, перепишите результат вручную; иначе скопируйте и передайте его `lpr`. Обратите внимание, что каждая линия содержит как счетчик цикла, так и одноразовый пароль; вам может показаться удобным отрывать пароль после использования.

#### 14.5.5. Ограничение использования UNIX® паролей

S/Key может наложить ограничения на использование UNIX паролей на основе имени хоста, имени пользователя, порта терминала или IP адреса сессии. Эти ограничения можно найти в файле настройки `/etc/skey.access`. Страница справочника `skey.access(5)` содержит дополнительную информацию о полном формате файла а также детали о некоторых предосторожностях, которые должны быть предприняты перед тем, как положиться в вопросах безопасности на этот файл.

Если файла `/etc/skey.access` нет (это ситуация по умолчанию в системах FreeBSD 4.X), всем пользователям будет разрешено входить с паролями UNIX. Если файл существует, использование



S/Key станет обязательно для всех, если только параметры настройки в файле `skey.access` не указывают иначе. В любом случае, пароли UNIX разрешены при входе с консоли.

Вот пример файла настройки `skey.access`, иллюстрирующий три наиболее распространенных вида параметров настройки:

```
permit internet 192.168.0.0 255.255.0.0
permit user fnord
permit port ttyd0
```

Первая строка (`permit internet`) разрешает пользователям, чей IP адрес (который подвержен подделке) соответствует заданному значению и маске, входить с использованием паролей UNIX. Это должно рассматриваться не как механизм безопасности, а как напоминание пользователям, что они работают через небезопасное соединение и должны использовать для аутентификации S/Key.

Вторая строка (`permit user`) позволяет определенным пользователям, в данном случае `fnord`, всегда использовать пароли UNIX. Вообще говоря, это должно использоваться только для тех, кто не может использовать программу `key`, например если они работают с простых терминалов или необучаемы.

Третья строка (`permit port`) позволяет всем пользователям, вошедшим с определенного терминала использовать пароли UNIX; этот параметр должен использоваться для подключений по `dial-up`.

ОПЕ может ограничивать использование паролей UNIX на основе IP адреса как и S/Key.

Соответствующий файл называется `/etc/opeaccess`, он существует по умолчанию в FreeBSD 5.0 и более современных системах. Обратитесь к `opeaccess(5)` за более подробной информацией об этом файле и о предосторожностях, которые вы должны предпринять при использовании этого файла.

Вот пример файла `opeaccess`:

```
permit 192.168.0.0 255.255.0.0
```

Эта строка позволяет пользователям, чей IP адрес (который подвержен подделке) соответствует указанному значению и маске, входить с паролем UNIX.

Если ни одно из правил в `opeaccess` не сработало, поведением по умолчанию является запрет всех не-ОПЕ входов.

## 14.6. TCP Wrappers

*Написал: Tom Rhodes.*

Каждый, кто знаком с `inetd(8)`, возможно когда-то слышал о TCP Wrappers. Но немногие полностью понимают их полезность в сетевой среде: большинство используют брандмауэр. Хотя его применимость очень широка, есть вещи, с которыми брандмауэр не может работать, такие как отправка текста обратной вызывающей стороне. Программное обеспечение уровня TCP может делать это и многое другое. В следующих нескольких разделах обсуждаются многие возможности TCP Wrappers, и, когда это необходимо, даются примеры настроек.

Программное обеспечение TCP Wrappers расширяет возможность `inetd` по поддержке каждого демона. С ним становится возможным протоколирование, возврат сообщений вызывающей стороне, ограничение подключений внутренней сетью и т.п. Хотя некоторые из этих возможностей могут быть

реализованы брандмауэром, TCP Wrappers не только предоставляют дополнительный уровень защиты, но и дают больше контроля над системой, чем это возможно с брандмауэром.

Поскольку рассматривается расширение к настройкам `inetd`, предполагается, что читатель ознакомился с разделом о настройке `inetd`.

**Замечание:** Хотя программы, запускаемые из `inetd(8)`, на самом деле не соответствуют термину "даемоны", существует традиция называть их именно так. Этот термин и используется в данном разделе.

### 14.6.1. Начальная настройка

Единственное требование для использования TCP Wrappers в FreeBSD это наличие в `rc.conf` параметров запуска `inetd -Ww`; это настройки по умолчанию. Конечно, ожидается также наличие правильной настройки `/etc/hosts.allow`, но `syslogd(8)` отправит сообщения в системный протокол если что-то не так.

**Замечание:** В отличие от других реализаций TCP Wrappers, использование `hosts.deny` не поддерживается. Все параметры настройки должны быть помещены в `/etc/hosts.allow`.

В простейшей конфигурации, политика подключения сводится к разрешению или блокированию в зависимости от параметров в `/etc/hosts.allow`. Настройка в FreeBSD по умолчанию заключается в разрешении подключения к любому даемону, запущенному из `inetd`. Изменение этого поведения будет обсуждаться только после рассмотрения базовой настройки.

Базовая настройка обычно принимает форму `daemon : address : action`, где `daemon` это имя даемона, который запускается `inetd`. В поле `address` может находиться имя хоста, IP адрес, или IPv6 адрес, заключенный в квадратные скобки ( [ ] ). Поле `action` может принимать значения `allow` или `deny`, чтобы соответственно разрешать или запрещать доступ. Помните, что поиск правил производится до первого совпадения. При обнаружении совпадения применяется соответствующее правило и поиск прерывается.

Существуют и другие параметры, но они будут описаны в следующих разделах. Простая конфигурация может быть, например, такой: для разрешения соединений по протоколу POP3 к даемону `mail/qpopper`, в `hosts.allow` необходимо добавить следующие строки:

```
# This line is required for POP3 connections:
qpopper : ALL : allow
```

После добавления этой строки, `inetd` необходимо перезапустить. Это можно выполнить командой `kill(1)` или скриптом `/etc/rc.d/inetd` с параметром `restart`.

### 14.6.2. Расширенная конфигурация

У TCP Wrappers имеются дополнительные параметры; они дают дополнительные возможности контроля над соединениями. Иногда бывает полезно возвращать комментарий определенным хостам или при подключении к определенным дамонам. В других случаях может быть необходимо добавить запись в лог файл, или отправить письмо администратору. В определенных ситуациях сервис должен

использоваться только для локальных соединений. Все это возможно с использованием параметров с шаблонами, символами подстановки и путем выполнения внешних команд. Следующие два раздела посвящены этим типам настроек.

#### 14.6.2.1. Внешние команды

Предположим ситуацию, в которой соединение должно быть запрещено, а о причине необходимо сообщить вызывающей стороне. Как это можно сделать? Соответствующую возможность предоставляет параметр `twist`. При попытке подключения выполняется команда или скрипт, заданный этим параметром. Пример дан в файле `hosts.allow`:

```
# The rest of the daemons are protected.
ALL : ALL \
    : severity auth.info \
    : twist /bin/echo "You are not welcome to use %d from %h."
```

В этом примере сообщение, "You are not allowed to use daemon from hostname." будет возвращено от всех демонов, которые не были предварительно настроены в файле доступа. Обратите внимание, что возвращаемое сообщение *должно* быть заключено в кавычки; из этого правила нет исключений.

**Внимание:** Возможна реализация DoS атаки, когда группа атакующих производит множество запросов на подключение.

Возможно также использование параметра `spawn`. Как и параметр `twist`, параметр `spawn` подразумевает запрет соединения и может использоваться для запуска команд или скриптов. В отличие от `twist`, `spawn` не отправляет ответ вызывающей стороне. Например, следующая конфигурация:

```
# We do not allow connections from example.com:
ALL : .example.com \
    : spawn (/bin/echo %a from %h attempted to access %d >> \
    /var/log/connections.log) \
    : deny
```

отклонит все попытки соединения из домена `*.example.com`; имя хоста, IP адрес и демон протоколируются в файл `/var/log/connections.log`.

Помимо приведенных выше символов подстановки, например `%a`, существует еще несколько символов. Обратитесь к странице `hosts_access(5)` справочной системы за полным списком.

#### 14.6.2.2. Параметры – шаблоны

До этого момента в примерах использовался шаблон `ALL`. Существуют и другие параметры, функциональность которых в дальнейшем может быть расширена. `ALL` соответствует любому демону, домену или IP адресу. Другой доступный шаблон это `PARANOID`, который соответствует хосту, IP адрес которого может быть подделан. Другими словами, `paranoid` может быть использован для определения действия с хостами, IP адрес которых не соответствует имени хоста. Вот пример применения этого параметра:

```
# Block possibly spoofed requests to sendmail:
sendmail : PARANOID : deny
```

В этом примере все запросы на подключения к `sendmail` от хостов, IP адрес которых не соответствует имени хоста, будут отклонены.

**Предостережение:** Использование `PARANOID` невозможно, если у клиента или сервера неправильно настроен DNS. В таких случаях необходимо вмешательство администратора.

Более подробная информация о шаблонах и их возможностях дана на странице `hosts_access(5)` справочной системы.

Для того, чтобы любая выбранная конфигурация заработала, в `hosts.allow` необходимо закомментировать первую строку настройки. В начале раздела об этом не упоминалось.

## 14.7. KerberosIV

*Предоставил Mark Murray. Оригинальный текст предоставил Mark Dapoz.*

Kerberos это сетевая дополнительная система/протокол, которая делает возможной аутентификацию пользователей через сервисы на защищенном сервере. Такие сервисы, как удаленный вход, удаленное копирование, защищенное копирование файлов между системами и другие задачи с высоким риском становятся допустимо безопасными и более контролируемыми.

Последующие инструкции могут использоваться в качестве руководства по настройке поставляемого с FreeBSD Kerberos. Тем не менее, вам могут потребоваться страницы справочника полного дистрибутива.

### 14.7.1. Установка KerberosIV

Kerberos это опциональный компонент FreeBSD. Простейший способ установки этой программы это выбор `krb4` или `krb5` из `sysinstall` во время первой установки FreeBSD. Будет установлен "eBones" (KerberosIV) или "Heimdal" (Kerberos5) вариант Kerberos. Включение этих реализаций объясняется тем, что они разработаны вне США/Канады и доступны вне этих стран, поскольку на них не влияют ограничения на экспорт криптографического кода из США.

Кроме того, реализация MIT Kerberos доступна из коллекции портов в виде пакета `security/krb5`.

### 14.7.2. Создание базы данных

Это необходимо сделать только на сервере Kerberos. Во-первых, убедитесь что не осталось старой базы данных Kerberos. Войдите в каталог `/etc/kerberosIV` и убедитесь, что в нем находятся только эти файлы:

```
# cd /etc/kerberosIV
# ls
README krb.conf          krb.realms
```

Если присутствуют еще какие-то файлы (такие как `principal.*` или `master_key`), используйте команду `kdb_destroy` для удаления старой базы данных Kerberos, или, если Kerberos не запущен, просто удалите эти файлы.

Затем отредактируйте файлы `krb.conf` и `krb.realms`, введя ваши данные. В этом примере уникальный идентификатор `EXAMPLE.COM`, сервер `grunt.example.com`. Отредактируем или создадим файл `krb.conf`:

```
# cat krb.conf
EXAMPLE.COM
EXAMPLE.COM grunt.example.com admin server
CS.BERKELEY.EDU okeeffe.berkeley.edu
ATHENA.MIT.EDU kerberos.mit.edu
ATHENA.MIT.EDU kerberos-1.mit.edu
ATHENA.MIT.EDU kerberos-2.mit.edu
ATHENA.MIT.EDU kerberos-3.mit.edu
LCS.MIT.EDU kerberos.lcs.mit.edu
TELECOM.MIT.EDU bitsy.mit.edu
ARC.NASA.GOV trident.arc.nasa.gov
```

В этом примере другие идентификаторы введены для иллюстрации настройки с несколькими хостами. С целью упрощения настройки вы можете не включать их.

Первая строка содержит идентификатор, под которым работает эта система. Остальные строки связывают идентификаторы с именами хостов. Сначала указывается идентификатор, затем хост под этим идентификатором, работающий как "центр распространения ключей". Слова `admin server` с последующим именем хоста означают, что этот хост также является сервером администрирования базы данных. За дальнейшей информацией об этих терминах обратитесь к страницам справочника по Kerberos.

Мы добавили `grunt.example.com` к идентификатору `EXAMPLE.COM` и кроме того сопоставили всем хостам в домене `.example.com` идентификатор `EXAMPLE.COM`. Файл `krb.realms` будет выглядеть так:

```
# cat krb.realms
grunt.example.com EXAMPLE.COM
.example.com EXAMPLE.COM
.berkeley.edu CS.BERKELEY.EDU
.MIT.EDU ATHENA.MIT.EDU
.mit.edu ATHENA.MIT.EDU
```

Как и в предыдущем примере, другие идентификаторы добавлены только для примера. С целью упрощения настройки вы можете не включать их.

В первой строке *определенная* система сопоставляется с идентификатором. В остальных строках показано, сопоставить идентификатору остальные системы определенного поддомена.

Теперь мы готовы к созданию базы данных. Потребуется всего лишь запустить сервер Kerberos (или центр распространения ключей). Используйте для этого `kdb_init`:

```
# kdb_init
Realm name [default  ATHENA.MIT.EDU ]: EXAMPLE.COM
You will be prompted for the database Master Password.
It is important that you NOT FORGET this password.
```

Введите главный ключ Kerberos:

Теперь мы должны сохранить ключ, чтобы сервера на локальных компьютерах могли его взять. Используйте для этого команду `kstash`:

```
# kstash
```

```
Enter Kerberos master key:
```

```
Current Kerberos master key version is 1.
```

```
Master key entered. BEWARE!
```

Этой командой зашифрованный главный пароль сохранен в `/etc/kerberosIV/master_key`.

### 14.7.3. Запуск Kerberos

Для каждой системы, защищаемой Kerberos, в базу данных должны быть добавлены две записи. Это `krasswd` и `rcmd`. Они добавляются вместе с именем системы.

Эти даemons, **`kpasswd`** и **`rcmd`** позволяют другим системам изменять пароли Kerberos и запускать такие команды как `rsp(1)`, `rlogin(1)`, `rsh(1)`.

Теперь добавим эти записи:

```
# kdb_edit
```

```
Opening database...
```

```
Enter Kerberos master key:
```

```
Current Kerberos master key version is 1.
```

```
Master key entered. BEWARE!
```

```
Previous or default values are in [brackets] ,  
enter return to leave the same, or new value.
```

```
Principal name: passwd
```

```
Instance: grunt
```

```
<Not found>, Create [y] ? y
```

```
Principal: passwd, Instance: grunt, kdc_key_ver: 1
```

```
New Password: <---- enter RANDOM here
```

```
Verifying password
```

```
New Password: <---- enter RANDOM here
```

```
Random password [y] ? y
```

```
Principal's new key version = 1
```

```
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 ] ?
```

```
Max ticket lifetime (*5 minutes) [ 255 ] ?
```

```
Attributes [ 0 ] ?
```

```

Edit O.K.
Principal name: rcmd
Instance: grunt

<Not found>, Create [y] ?

Principal: rcmd, Instance: grunt, kdc_key_ver: 1
New Password: <---- enter RANDOM here
Verifying password

New Password:          <---- enter RANDOM here

Random password [y] ?

Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 ] ?
Max ticket lifetime (*5 minutes) [ 255 ] ?
Attributes [ 0 ] ?
Edit O.K.
Principal name:          <---- null entry here will cause an exit

```

#### 14.7.4. Создание файла настройки сервера

Теперь необходимо создать все записи сервисов, которые были определены для каждого компьютера. Используем для этого команду `ext_srvtab`. Будет создан файл, который должен быть скопирован или перемещен *безопасным способом* в каталог `/etc/kerberosIV` каждого Kerberos клиента. Этот файл должен присутствовать на каждом сервере и клиенте, он необходим для работы Kerberos.

```

# ext_srvtab grunt
Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered. BEWARE!
Generating 'grunt-new-srvtab'....

```

Эта команда создаст временный файл, который должен быть переименован в `srvtab`, чтобы серверы смогли обратиться к нему. Используйте команду `mv(1)` для перемещения его в исходной системе:

```
# mv grunt-new-srvtab srvtab
```

Если файл предназначен для клиентской системы, и сеть не безопасна, скопируйте `client-new-srvtab` на съемный носитель и перенесите файл с его помощью. Убедитесь, что переименовали его в `srvtab` в каталоге `/etc/kerberosIV` клиента, и что режим доступа к нему 600:

```

# mv grumble-new-srvtab srvtab
# chmod 600 srvtab

```

### 14.7.5. Пополнение базы данных

Теперь необходимо добавить в базу данных пользователей. Во-первых, создадим запись для пользователя jane. Используйте команду kdb\_edit:

```
# kdb_edit
Opening database...

Enter Kerberos master key:

Current Kerberos master key version is 1.

Master key entered.  BEWARE!
Previous or default values are in [brackets] ,
enter return to leave the same, or new value.

Principal name: jane
Instance:

<Not found>, Create [y] ? y

Principal: jane, Instance: , kdc_key_ver: 1
New Password:          <---- enter a secure password here
Verifying password

New Password:          <---- re-enter the password here
Principal's new key version = 1
Expiration date (enter yyyy-mm-dd) [ 2000-01-01 ] ?
Max ticket lifetime (*5 minutes) [ 255 ] ?
Attributes [ 0 ] ?
Edit O.K.
Principal name:      <---- null entry here will cause an exit
```

### 14.7.6. Тестирование всей системы

Во-первых, запустите демоны Kerberos. При правильном редактировании файла /etc/rc.conf они запустятся автоматически при перезагрузке. Это необходимо только на сервере Kerberos. Клиенты Kerberos получают все необходимые данные из каталога /etc/kerberosIV.

```
# kerberos &
Kerberos server starting
Sleep forever on error
Log file is /var/log/kerberos.log
Current Kerberos master key version is 1.

Master key entered.  BEWARE!

Current Kerberos master key version is 1
Local realm: EXAMPLE.COM
# kadmin -n &
KADM Server KADM0.0A initializing
Please do not use 'kill -9' to kill this job, use a
```



```
regular kill instead
```

```
Current Kerberos master key version is 1.
```

```
Master key entered. BEWARE!
```

Теперь для получения доступа через созданного пользователя jane используйте kinit:

```
% kinit jane
```

```
MIT Project Athena (grunt.example.com)
```

```
Kerberos Initialization for "jane"
```

```
Password:
```

Попробуйте просмотреть имеющиеся данные с помощью klist:

```
% klist
```

```
Ticket file:      /tmp/tkt245
```

```
Principal:       jane@EXAMPLE.COM
```

Issued	Expires	Principal
Apr 30 11:23:22	Apr 30 19:23:22	krbtgt.EXAMPLE.COM@EXAMPLE.COM

Теперь попробуйте изменить пароль с помощью passwd(1), чтобы убедиться, что демон **kpasswd** может получить информацию из базы данных Kerberos:

```
% passwd
```

```
realm EXAMPLE.COM
```

```
Old password for jane:
```

```
New Password for jane:
```

```
Verifying password
```

```
New Password for jane:
```

```
Password changed.
```

### 14.7.7. Включение su

Kerberos позволяет назначить *каждому* пользователю, который нуждается в привилегиях root, свой *собственный* пароль su(1). Необходимо добавить учетную запись, которой разрешено получать root доступ через su(1). Это делается путем связывания учетной записи root с пользовательской учетной записью. Создадим в базе данных Kerberos запись jane.root с помощью kdb\_edit:

```
# kdb_edit
```

```
Opening database...
```

```
Enter Kerberos master key:
```

```
Current Kerberos master key version is 1.
```

```
Master key entered. BEWARE!
```

```
Previous or default values are in [brackets] ,  
enter return to leave the same, or new value.
```

```
Principal name: jane
```

Instance: **root**

<Not found>, Create [y] ? y

Principal: jane, Instance: root, kdc\_key\_ver: 1

New Password: <---- enter a SECURE password here

Verifying password

New Password: <---- re-enter the password here

Principal's new key version = 1

Expiration date (enter yyyy-mm-dd) [ 2000-01-01 ] ?

Max ticket lifetime (\*5 minutes) [ 255 ] ? **12** <--- Keep this short!

Attributes [ 0 ] ?

Edit O.K.

Principal name: <---- null entry here will cause an exit

Теперь проверим работоспособность этой записи:

```
# kinit jane.root
```

```
MIT Project Athena (grunt.example.com)
```

```
Kerberos Initialization for "jane.root"
```

```
Password:
```

Необходимо добавить пользователя к root файлу .klogin:

```
# cat /root/.klogin
```

```
jane.root@EXAMPLE.COM
```

Теперь попробуйте выполнить su(1):

```
% su
```

```
Password:
```

и посмотрите на имеющиеся данные:

```
# klist
```

```
Ticket file: /tmp/tkt_root_245
```

```
Principal: jane.root@EXAMPLE.COM
```

Issued	Expires	Principal
May 2 20:43:12	May 3 04:43:12	krbtgt.EXAMPLE.COM@EXAMPLE.COM

### 14.7.8. Использование других команд

В примере выше мы создали запись (principal) jane с доступом к root (instance). Она основана на пользователе с таким же именем, как и идентификатор, что принято Kerberos по умолчанию; <principal>.<instance> в форме <username>.root позволяет использовать su(1) для доступа к root, если соответствующие записи находятся в файле .klogin домашнего каталога root:

```
# cat /root/.klogin
```

```
jane.root@EXAMPLE.COM
```

Подобно этому, если в файле `.klogin` из домашнего каталога пользователя есть строки в форме:

```
% cat ~/.klogin
jane@EXAMPLE.COM
jack@EXAMPLE.COM
```

это позволит любому с идентификатором `EXAMPLE.COM`, кто аутентифицировался как `jane` или `jack` (с помощью команды `kinit`, см. выше) получить доступ к учетной пользователя `jane` или файлам этой системы (`grunt`) через `rlogin(1)`, `rsh(1)` или `rsp(1)`.

Например, `jane` может входить в другую систему используя `Kerberos`:

```
% kinit
MIT Project Athena (grunt.example.com)
Password:
% rlogin grunt
Last login: Mon May  1 21:14:47 from grumble
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
    The Regents of the University of California.  All rights reserved.

FreeBSD BUILT-19950429 (GR386) #0: Sat Apr 29 17:50:09 SAT 1995
```

Или `jack` входит в учетную запись `jane`'s на этом же компьютере (файл `.klogin` `jane` настроен как показано выше, и в `Kerberos` настроена учетная запись `jack`):

```
% kinit
% rlogin grunt -l jane
MIT Project Athena (grunt.example.com)
Password:
Last login: Mon May  1 21:16:55 from grumble
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
    The Regents of the University of California.  All rights reserved.

FreeBSD BUILT-19950429 (GR386) #0: Sat Apr 29 17:50:09 SAT 1995
```

## 14.8. Kerberos5

*Предоставил Tillman Hodgson. Оригинальный материал предоставил Mark Murray.*

Все релизы FreeBSD после FreeBSD-5.1 включают поддержку только **Kerberos5**. Таким образом, **Kerberos5** это единственная включаемая в поставку версия и его конфигурация похожа на **KerberosIV** во многих аспектах. Эта информация применима только к **Kerberos5** из релизов после FreeBSD-5.0. Пользователи, желающие использовать пакет **KerberosIV**, могут установить его из порта `security/krb4`.

**Kerberos** это дополнительная сетевая система/протокол, позволяющая пользователям авторизоваться через защищенные сервисы на защищенном сервере. Такие сервисы как удаленный вход, удаленное копирование, защищенное копирование файлов между системами и другие задачи с высоким риском становятся допустимо безопасными и более контролируемыми.

**Kerberos** может быть описана как прокси система идентификации-проверки. Она также может быть описана как защищенная внешняя система аутентификации. **Kerberos** предоставляет только одну функцию — защищенную аутентификацию пользователей сети. Он не предоставляет функций авторизации (что разрешено делать пользователям) или функций аудита (какой пользователь что делает). После того, как клиент и сервер использовали **Kerberos** для идентификации, они могут зашифровать все соединения для гарантирования собственной безопасности и целостности данных.

Следовательно крайне рекомендуется использовать **Kerberos** с другими методами безопасности, предоставляющими сервисы авторизации и аудита.

Последующие инструкции могут использоваться в качестве руководства по настройке **Kerberos**, поставляемого с FreeBSD. Тем не менее, вам потребуется обратиться к соответствующим страницам справочника за полным описанием.

В целях демонстрации установки **Kerberos**, будут применены следующие обозначения:

- DNS домен ("зона") example.org.
- Уникальный идентификатор **Kerberos** EXAMPLE.ORG.

**Замечание:** Используйте действующие имена доменов при настройке **Kerberos** даже если вы будете использовать его во внутренней сети. Это позволит избежать проблем с DNS и гарантирует возможность связи с **Kerberos** под другими идентификаторами.

### 14.8.1. История

**Kerberos** был создан MIT в качестве решения проблем с безопасностью сети. Протокол **Kerberos** использует стойкую криптографию, так что клиент может идентифицироваться на сервере (и обратно) через незащищенное сетевое соединение.

**Kerberos** это и имя сетевого протокола аутентификации и общий термин для описания программ, где он реализован (например, **Kerberos telnet**). Текущая версия протокола 5 описана в RFC 1510.

Доступно несколько свободных реализаций этого протокола, работающих на множестве операционных систем. Massachusetts Institute of Technology (MIT), где **Kerberos** был первоначально разработан, продолжает разрабатывать собственный пакет **Kerberos**. Он обычно использовался в США как криптографический продукт, и в этом качестве попадал под действие ограничений на экспорт. MIT **Kerberos** доступен в виде порта (security/krb5). Heimdal **Kerberos** это другая реализация версии 5, которая разрабатывалась исключительно вне США для обхода экспортных ограничений (и поэтому часто включалась в некоммерческие реализации UNIX). Heimdal **Kerberos** доступен в виде порта (security/heimdal), его минимальный комплект включен в базовую установку FreeBSD.

В целях получения наибольшей аудитории, в этих инструкциях предполагается использование Heimdal включаемого в FreeBSD.

### 14.8.2. Настройка Heimdal KDC

Центр распространения ключей (Key Distribution Center, KDC) это централизованный сервис аутентификации, предоставляемый **Kerberos** — это компьютер, который предоставляет доступ через

**Kerberos.** KDC считается доверяемым всеми другими компьютерами с определенным идентификатором **Kerberos** и поэтому к нему предъявляются высокие требования безопасности.

Имейте ввиду, что хотя работа сервера **Kerberos** требует очень немного вычислительных ресурсов, из соображений безопасности для него рекомендуется отдельный компьютер, работающий только в качестве KDC.

Перед началом настройки KDC, убедитесь что в файле `/etc/rc.conf` содержатся правильные настройки для работы в качестве KDC (вам может потребоваться изменить пути в соответствии с собственной системой):

```
kerberos5_server_enable="YES"
kadmind5_server_enable="YES"
kerberos_stash="YES"
```

**Замечание:** Параметр `kerberos_stash` существует только в FreeBSD 4.X.

Затем приступим к редактированию файла настройки **Kerberos**, `/etc/krb5.conf`:

```
[libdefaults]
    default_realm = EXAMPLE.ORG
[realms]
    EXAMPLE.ORG = {
        kdc = kerberos.example.org
    }
[domain_realm]
    .example.org = EXAMPLE.ORG
```

Обратите внимание что в файле `/etc/krb5.conf` подразумевается наличие у KDC полного имени `kerberos.example.org`. Вам потребуется добавить CNAME (синоним) к файлу зоны, если у KDC другое имя.

**Замечание:** Для больших сетей с правильно настроенным сервером BIND DNS пример выше может быть урезан до:

```
[libdefaults]
    default_realm = EXAMPLE.ORG
```

Со следующими строками, добавленными в файл зоны `example.org`:

```
_kerberos._udp      IN  SRV      01 00 88 kerberos.example.org.
_kerberos._tcp      IN  SRV      01 00 88 kerberos.example.org.
_kpasswd._udp       IN  SRV      01 00 464 kerberos.example.org.
_kerberos-adm._tcp  IN  SRV      01 00 749 kerberos.example.org.
_kerberos           IN  TXT       EXAMPLE.ORG.
```

Создадим теперь базу данных **Kerberos**. Эта база данных содержит ключи всех основных хостов, зашифрованных с помощью главного пароля. Вам не требуется помнить этот пароль, он хранится в файле (`/var/heimdal/m-key`). Для создания главного ключа запустите `kstash` и введите пароль.

Как только будет создан главный ключ, вы можете инициализировать базу данных с помощью программы `kadmin` с ключом `-l` (означающим "local"). Этот ключ сообщает `kadmin` обращаться к файлам базы данных непосредственно вместо использования сетевого сервиса `kadmind`. Это помогает решить "проблему курицы и яйца", когда обращение идет к еще не созданной базе данных. Как только вы увидите приглашение `kadmin`, используйте команду `init` для создания базы данных идентификаторов.

Наконец, оставаясь в приглашении `kadmin`, создайте первую запись с помощью команды `add`. Оставьте неизменными параметры по умолчанию, вы всегда сможете изменить их позже с помощью команды `modify`. Обратите внимание, что вы всегда можете использовать команду `?` для просмотра доступных параметров.

Пример создания базы данных показан ниже:

```
# kstash
Master key: xxxxxxxx
Verifying password - Master key: xxxxxxxx

# kadmin -l
kadmin> init EXAMPLE.ORG
Realm max ticket life [unlimited]:
kadmin> add tillman
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Attributes []:
Password: xxxxxxxx
Verifying password - Password: xxxxxxxx
```

Теперь пришло время запустить сервисы KDC. Выполните команды `/etc/rc.d/kerberos start` и `/etc/rc.d/kadmind start` для запуска сервисов. Ни один из поддерживающих **Kerberos** демонов на этот момент запущен не будет, но у вас должна быть возможность убедиться в том, что KDC функционирует путем получения списка доступа для пользователя, которого вы только что самостоятельно создали из командной строки самого KDC:

```
% k5init tillman
tillman@EXAMPLE.ORG's Password:

% k5list
Credentials cache: FILE:/tmp/krb5cc_500
Principal: tillman@EXAMPLE.ORG

    Issued                Expires                Principal
Aug 27 15:37:58  Aug 28 01:37:58  krbtgt/EXAMPLE.ORG@EXAMPLE.ORG
```

### 14.8.3. Сервер Kerberos с сервисами Heimdal

Для начала нам потребуется копия файла настройки **Kerberos**, `/etc/krb5.conf`. Просто скопируйте его с KDC на клиентский компьютер безопасным способом (используя сетевые утилиты, такие как `scp(1)`, или физически, с помощью дискеты).

Затем вам понадобится файл `/etc/krb5.keytab`. Это основное различие между сервером, поддерживающим **Kerberos** и рабочими станциями — на сервере должен быть файл `keytab`. В этом

файле находится центральный ключ сервера, который позволяет KDC проверять все другие идентификаторы. Он должен быть помещен на сервер безопасным способом, поскольку безопасность сервера может быть нарушена, если ключ станет общедоступен. Это означает, что его передача через незашифрованный канал, такой как FTP — очень плохая идея.

Обычно перенос файла `keytab` на сервер производится с помощью программы `kadmin`. Это удобно, поскольку вам потребуется также создать запись хоста (KDC часть `krb5.keytab`) с помощью `kadmin`.

Обратите внимание, что должны быть уже зарегистрированы в системе и необходимо наличие прав на использование интерфейса `kadmin` в файле `kadmind.acl`. Обратитесь к разделу "Remote administration" в info страницах Heimdal (`info heimdal`) за деталями по составлению списка доступа. Если вы не хотите включать удаленный доступ `kadmin`, можете просто подключиться к KDC через защищенное соединение (локальную консоль, `ssh(1)` или **Kerberos telnet(1)**) и выполнять администрирование локально с помощью `kadmin -l`.

После добавления файла `/etc/krb5.conf`, вы можете использовать `kadmin` с сервера **Kerberos**. Команда `add --random-key` позволит вам добавить запись для сервера, а команда `ext` позволит перенести эту запись в собственный `keytab` файл сервера. Например:

```
# kadmin
kadmin> add --random-key host/myserver.example.org
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Attributes []:
kadmin> ext host/myserver.example.org
kadmin> exit
```

Обратите внимание, что команда `ext` (сокращение от "extract") сохраняет полученный ключ в файле `/etc/krb5.keytab` по умолчанию.

Если на KDC не запущен `kadmind` (возможно по соображениям безопасности) и вы не можете получить доступ к `kadmin` удаленно, возможно добавление записи хоста (`host/myserver.EXAMPLE.ORG`) непосредственно на KDC с последующим извлечением ее во временный файл (и перезаписью `/etc/krb5.keytab` на KDC) примерно так:

```
# kadmin
kadmin> ext --keytab=/tmp/example.keytab host/myserver.example.org
kadmin> exit
```

Затем вы можете скопировать `keytab` на сервер защищенным способом (например, используя `scp` или дискету). Убедитесь, что используемое имя `keytab` не совпадает с именем по умолчанию во избежание перезаписывания `keytab` на KDC.

Теперь ваш сервер может связываться с KDC (добавлен файл `krb5.conf`) и идентифицировать себя (добавлен файл `krb5.keytab`). Теперь вы готовы к включению некоторых сервисов **Kerberos**. В этом примере мы включим сервис `telnet`, поместив в `/etc/inetd.conf` нижеприведенную строку и перезапустив сервис `inetd(8)` командой `/etc/rc.d/inetd restart`:

```
telnet    stream  tcp      nowait  root    /usr/libexec/telnetd  telnetd -a user
```

Очень важно установить ключ `-a` (тип аутентификации) в `user`. Обратитесь к странице справочника `telnetd(8)` за подробной информацией.

#### 14.8.4. Клиент **Kerberos** с **Heimdal**

Настройка клиентского компьютера почти тривиально проста. Как только настройка **Kerberos** закончена, вам потребуется только файл настройки **Kerberos**, `/etc/krb5.conf`. Просто скопируйте его безопасным способом на клиентский компьютер с KDC.

Протестируйте клиентский компьютер, попытавшись использовать `kinit`, `klist`, и `kdestroy` для получения, отображения и удаления списка доступа. Соединитесь с **Kerberos** сервером используя клиент **Kerberos**, если соединение не работает и получение доступа является проблемой, это скорее всего проблема сервера, а не клиента или KDC.

При тестировании приложения вроде `telnet`, попробуйте использовать программу перехвата пакетов (такую как `tcpdump(1)`), чтобы убедиться, что ваш пароль не передается незашифрованным. Попробуйте использовать `telnet` с параметром `-x`, чтобы зашифровать весь поток данных (подобно `ssh`).

Основные клиентские приложения **Kerberos** (традиционно называемые `kinit`, `klist`, `kdestroy`, и `kpasswd`) находятся в базовой установке FreeBSD. Обратите внимание, что в FreeBSD версий до 5.0 они были переименованы в `k5init`, `k5list`, `k5destroy`, `k5passwd`, и `k5stash` (хотя их обычно использовали лишь однократно).

Различные неосновные клиентские приложения **Kerberos** также устанавливаются по умолчанию. Здесь проявляется "минимальность" базовой установки Heimdal: `telnet` это единственное приложение, поддерживающее **Kerberos**.

Порт Heimdal добавляет некоторые отсутствующие клиентские приложения: поддерживающие **Kerberos** версии `ftp`, `rsh`, `rsh`, `rlogin`, и некоторые другие реже используемые программы. Порт MIT также содержит полный пакет клиентских приложений **Kerberos**.

#### 14.8.5. Пользовательские файлы настройки: `.k5login` и `.k5users`

Учетные записи пользователя в **Kerberos** (например `tillman@EXAMPLE.ORG`) обычно связаны с локальными учетными записями (например с локальной учетной записью `tillman`). Клиентские приложения, такие как `telnet`, обычно не требуют указания имени пользователя или учетной записи.

Тем не менее, время от времени вам может потребоваться дать доступ к локальной учетной записи кому-то, у кого нет соответствующей учетной записи **Kerberos**. Например, пользователю `tillman@EXAMPLE.ORG` может потребоваться доступ к локальной учетной записи `webdevelopers`. Другим учетным записям также может потребоваться доступ к этой локальной учетной записи.

Файлы `.k5login` и `.k5users`, помещенные в домашний каталог пользователя, могут быть использованы подобно действенной комбинации `.hosts` и `.rhosts` для решения этой проблемы. Например, файл `.k5login` со следующим содержанием:

```
tillman@example.org
jdoe@example.org
```

помещен в домашний каталог локального пользователя `webdevelopers`, то обе упомянутые учетные записи получат доступ к этой учетной записи без необходимости наличия общего пароля.

Рекомендуется прочитать страницу справочника по этим командам. Обратите внимание, что страница справочника о `ksu` содержит информацию по `.k5users`.



### 14.8.6. Подсказки, советы и решение проблем с Kerberos

- При использовании портов как Heimdal так и MIT **Kerberos** убедитесь, что в PATH версии **Kerberos** клиентов указаны перед их версиями в базовой системе.
- Все ли компьютеры в пределах данного realm синхронизированы по времени? Если нет, аутентификация может завершиться неудачно. Разд. 23.11 описывает как синхронизировать часы с использованием NTP.
- MIT и Heimdal успешно взаимодействуют. За исключением kadmin, протокол для которого не стандартизован.
- Если вы изменяете hostname, потребуется также изменить учетную запись host/ и обновить keytab. Это также необходимо для специальных записей в keytab, таких как www/ запись модуля Apache www/mod\_auth\_kerb.
- Все хосты под общим идентификатором должны разрешаться DNS (прямое и обратное разрешение), или как минимум через /etc/hosts. Записи CNAME будут работать, но записи A и PTR должны быть корректны и находиться на своем месте. Сообщение об ошибке не всегда интуитивно понятно: Kerberos5 refuses authentication because Read req failed: Key table entry not found.
- Некоторые операционные системы, способные работать в качестве клиентов KDC не устанавливают права для ksu в setuid root. Это означает, что ksu не работает, что хорошо является хорошей идеей для безопасности, но неудобно. Это не ошибка KDC.
- С MIT **Kerberos**, если вы хотите продлить действие доступа до значения большего, чем десять часов по умолчанию, используйте команду modify\_principal в kadmin для изменения maxlife доступа к самой учетной записи и к учетной записи krbtgt. Затем возможно использование kinit с параметром -l для запроса доступа с большим временем действия.
- 

**Замечание:** Если вы запускаете перехватчик пакетов на KDC для разрешения проблем, а затем запускаете kinit с рабочей станции, то увидите, что TGT посылается непосредственно при запуске kinit — даже до того, как вы введете пароль! Объяснение в том, что сервер **Kerberos** свободно распространяет TGT (Ticket Granting Ticket) на каждый неавторизованный запрос; однако, каждый TGT зашифрован ключом, полученным из пароля пользователя. Следовательно, когда пользователь вводит свой пароль, он не отправляется на KDC, а используется для расшифровки TGT, который уже получен kinit. Если в процессе расшифровки получается правильный билет с правильным значением времени, у пользователя есть действующее "удостоверение". Это удостоверение содержит ключ сессии для установления безопасного соединения с сервером **Kerberos**, как и действующий TGT, зашифрованный ключом сервера **Kerberos**. Второй уровень шифрования недоступен пользователю, но позволяет серверу **Kerberos** проверять правильность каждого TGT.

- Если вы хотите установить большое время жизни доступа (например, неделю), и используете **OpenSSH** для соединения с компьютером, где хранится "билет", убедитесь, что параметр **Kerberos** TicketCleanup установлен в no в файле sshd\_config, или билеты будут уничтожены при выходе из сеанса.
- Помните, что время жизни билетов хостов больше. Если время жизни билета для учетной записи пользователя составляет неделю, а время жизни учетной записи хоста, к которому вы

подсоединяетесь девять часов, учетная запись хоста в кэше устареет и кэш билетов будет работать не так, как ожидается.

- При настройке файла `krb5.dict` на предотвращение использования определенных плохих паролей (страница справочника для `kadmin` кратко рассказывает об этом), запомните, что это применимо только к учетным записям, для которых действует политика паролей. Формат файла `krb5.dict` прост: одно слово на строку. Может помочь создание символической ссылки на `/usr/share/dict/words`.

### 14.8.7. Отличия от порта MIT

Основное различие между установками MIT и Heimdal относится к программе `kadmin`, которая имеет другой (но эквивалентный) набор команд и использует другой протокол. Если ваш KDC работает на MIT, вы не сможете использовать `kadmin` для удаленного администрирования KDC (и наоборот, по этой же причине).

Опции командной строки клиентов также могут немного отличаться для одинаковых задач. Рекомендуется следовать инструкциям на MIT **Kerberos** веб сайте (<http://web.mit.edu/Kerberos/www/>). Будьте внимательны при определении РАТН: порт MIT устанавливается по умолчанию в `/usr/local/`, и если в РАТН вначале указаны системные каталоги, вместо приложений MIT могут быть запущены системные приложения.

**Замечание:** С портом MIT `security/krb5`, предоставляемым FreeBSD, убедитесь что файл `/usr/local/share/doc/krb5/README.FreeBSD` установлен портом, если вы хотите понять почему вход через `telnetd` и `klogind` иногда происходит так странно. Наиболее важно, исправление "incorrect permissions on cache file" требует использования бинарного файла `login.krb5` для аутентификации, чтобы права на переданное удостоверение передавались правильно.

## 14.8.8. Преодоление ограничений, обнаруженных в Kerberos

### 14.8.8.1. Kerberos это все или ничего

Каждый сервис, работающий в сети, должен быть модифицирован для работы с **Kerberos** (или другим способом защищен от атак по сети) или удостоверения пользователей могут быть украдены или использованы повторно. В качестве примера может быть приведено использование **Kerberos** версий оболочек для удаленной работы (например через `rsh` и `telnet`), при наличии POP3 сервера, получающего пароли в незашифрованном виде.

### 14.8.8.2. Kerberos предназначен для однопользовательских рабочих станций

В многопользовательской среде **Kerberos** менее безопасен. Это потому, что он хранит билеты в каталоге `/tmp`, которая доступна для чтения всем. Если пользователь работает с несколькими другими пользователями одновременно на одном компьютере (т.е. в многопользовательской среде), возможна кража (копирование) билета другим пользователем.

Решить проблему можно с помощью параметра командной строки `-s` или (предпочтительно) с помощью переменной окружения `KRB5CCNAME`, но это делается редко. Для преодоления ограничения достаточно сохранять билет в домашнем каталоге пользователя и использовать простые ограничения на доступ к файлам.

#### 14.8.8.3. От KDC зависит вся система

Архитектура системы такова, что KDC должен быть максимально защищен, поскольку главный пароль базы данных содержится в нем. На KDC не должно быть запущено никаких других сервисов и он должен быть защищен физически. Опасность велика, поскольку **Kerberos** хранит все пароли зашифрованными одним ключом (‘главным’ ключом), который хранится в файле на KDC.

Хорошей новостью является то, что кража главного ключа не станет такой проблемой, как может показаться. Главный ключ используется только для шифрования базы данных **Kerberos** и в качестве seed для генератора случайных чисел. Поскольку доступ к KDC защищен, атакующий мало что сможет сделать с главным ключом.

Кроме того, если KDC станет недоступен (возможно по причине атак DoS или проблем в сети) сетевые сервисы будет невозможно использовать, поскольку аутентификация не может быть выполнена. Уменьшить последствия можно при наличии нескольких KDC (один главный и один или несколько резервных) и с аккуратно реализованной резервной аутентификацией (отлично подойдет PAM).

#### 14.8.8.4. Недостатки Kerberos

**Kerberos** позволяет пользователям, хостам и сервисам производить аутентификацию друг друга. В нем нет механизма аутентификации KDC для пользователей, хостов или сервисов. Это означает, что поддельный kinit (например) может записывать все имена пользователей и паролей. Помочь решить проблему может `security/tripwire` или другой инструмент проверки целостности файловой системы.

#### 14.8.9. Ресурсы и информация для дальнейшего изучения

- **Kerberos FAQ** (<http://www.faqs.org/faqs/Kerberos-faq/general/preamble.html>)
- Разработка системы аутентификации: диалог в четырех сценах (<http://web.mit.edu/Kerberos/www/dialogue.html>)
- RFC 1510, **Kerberos Network Authentication Service (V5)** (<http://www.ietf.org/rfc/rfc1510.txt?number=1510>)
- Домашняя страница MIT **Kerberos** (<http://web.mit.edu/Kerberos/www/>)
- Домашняя страница Heimdal **Kerberos** (<http://www.pdc.kth.se/heimdal/>)

## 14.9. Межсетевые экраны

*Предоставили Gary Palmer, Alex Nash.*

Интерес к межсетевым экранам (брандмауэр, firewall) со стороны людей, подключенных к интернет, все возрастает и появились даже приложения для локальной сети, предоставляющие повышенный уровень безопасности. В этом разделе мы надеемся изложить что такое межсетевые экраны, как их использовать, и как использовать возможности, предоставляемые ядром FreeBSD для их реализации.

**Замечание:** Люди часто думают, что наличие межсетевого экрана между внутренней сетью и "Большим плохим интернетом" решит все их проблемы безопасности. Это может помочь, но плохо настроенный межсетевой экран представляет более серьезную угрозу безопасности, чем его полное отсутствие. Межсетевой экран добавляет еще один уровень безопасности вашим системам, но не может остановить проникновение решительно настроенного взломщика в вашу сеть. Если вы снижаете внутреннюю безопасность системы, поскольку верите в надежность межсетевого экрана, это существенно упрощает работу взломщика.

### 14.9.1. Что такое межсетевой экран?

Есть два четко различающихся типа межсетевых экранов, повседневно используемых в современном интернет. Первый тип правильнее называть *маршрутизатор с фильтрацией пакетов*. Этот тип межсетевого экрана работает на машине, подключенной к нескольким сетям и применяет к каждому пакету набор правил, определяющий переправлять ли этот пакет или блокировать. Второй тип, известный как *прокси сервер*, реализован в виде демонов, выполняющих аутентификацию и пересылку пакетов, возможно на машине с несколькими сетевыми подключениями, где пересылка пакетов в ядре отключена.

Иногда эти два типа межсетевых экранов используются вместе, так что только определенной машине (известной как *защитный хост (bastion host)*) позволено отправлять пакеты через фильтрующий маршрутизатор во внутреннюю сеть. Прокси сервисы работают на защитном хосте, что обычно более безопасно, чем обычные механизмы аутентификации.

FreeBSD поставляется с встроенным в ядро фильтром пакетом (известным как IPFW), ему будет посвящена оставшаяся часть раздела. Прокси серверы могут быть собраны на FreeBSD из программного обеспечения сторонних разработчиков, но их слишком много и невозможно описать их в этом разделе.

#### 14.9.1.1. Маршрутизаторы с фильтрацией пакетов

Маршрутизатор это машина, пересылающая пакеты между двумя или несколькими сетями. Маршрутизатор с фильтрацией пакетов запрограммирован на сравнение каждого пакета со списком правил перед тем как решить, пересылать его или нет. Большинство современного программного обеспечения маршрутизации имеет возможности фильтрации, и по умолчанию пересылаются все пакеты. Для включения фильтров, вам потребуется определить набор правил.

Для определения того, должен ли быть пропущен пакет, межсетевой экран ищет в наборе правило, совпадающее с содержимым заголовков пакета. Как только совпадение найдено, выполняется действие, присвоенное данному правилу. Действие может заключаться в отбрасывании пакета, пересылке пакета, или даже в отправлении ICMP сообщения в адрес источника. Учитывается только первое совпадение, поскольку правила просматриваются в определенном порядке. Следовательно, список правил можно назвать "цепочкой правил".

Критерий отбора пакетов зависит от используемого программного обеспечения, но обычно вы можете определять правила, зависящие от IP адреса источника пакета, IP адреса назначения, номера порта источника пакета, номера порта назначения (для протоколов, поддерживающих порты), или даже от типа пакета (UDP, TCP, ICMP, и т.д.).

#### 14.9.1.2. Прокси серверы

Прокси серверы это компьютеры, где обычные системные демоны (**telnetd**, **ftpd**, и т.д.) заменены специальными серверами. Эти серверы называются *прокси серверами*, поскольку они обычно работают только с входящими соединениями. Это позволяет запускать (например) **telnet** прокси сервер на межсетевом экране, и делать возможным вход по **telnet** на межсетевой экран, прохождение механизма аутентификации, и получение доступа к внутренней сети (аналогично, прокси серверы могут быть использованы для выхода во внешнюю сеть).

Прокси серверы обычно лучше защищены, чем другие серверы, и зачастую имеют более широкий набор механизмов аутентификации, включая системы "одноразовых" паролей, так что даже если кто-то узнает, какой пароль вы использовали, он не сможет использовать его для получения доступа к системе, поскольку срок действия пароля истекает немедленно после его первого использования. Поскольку пароль не дает доступа непосредственно к компьютеру, на котором находится прокси-сервер, становится гораздо сложнее установить в систему backdoor.

Прокси серверы обычно имеют способ дополнительного ограничения доступа, так что только определенные хосты могут получить доступ к серверам. Большинство также позволяют администратору указывать, пользователей и компьютеры, к которым они могут обращаться. Опять же доступные возможности в основном зависят от используемого программного обеспечения.

#### 14.9.2. Что позволяет делать IPFW?

Программное обеспечение IPFW, поставляемое с FreeBSD, это система фильтрации и учета пакетов, находящаяся в ядре и снабженная пользовательской утилитой настройки, **ipfw(8)**. Вместе они позволяют определять и просматривать правила, используемые ядром при маршрутизации.

IPFW состоит из двух связанных частей. Межсетевой экран осуществляет фильтрацию пакетов. Часть, занимающаяся учетом IP пакетов, отслеживает использование маршрутизатора на основе правил подобных тем, что используются в части межсетевого экрана. Это позволяет администратору определять, например, объем трафика, полученного маршрутизатором от определенного компьютера, или объем пересылаемого WWW трафика.

Благодаря тому, как реализован IPFW, вы можете использовать его и на компьютерах, не являющихся маршрутизаторами для фильтрации входящих и исходящих соединений. Это особый случай более общего использования IPFW, и в этой ситуации используются те же команды и техника.

#### 14.9.3. Включение IPFW в FreeBSD

Поскольку основная часть системы IPFW находится в ядре, вам потребуется добавить один или несколько параметров в файл настройки ядра, в зависимости от требуемых возможностей, и пересобрать ядро. Обратитесь к главе о пересборке ядра (Гл. 8) за подробным описанием этой процедуры.

**Внимание:** Правилom IPFW по умолчанию является `deny ip from any to any`. Если вы не добавите других правил во время загрузки для разрешения доступа, то *заблокируете доступ* к серверу с включенным в ядро межсетевым экраном после перезагрузки. Мы предлагаем указать `firewall_type=open` в файле `/etc/rc.conf` при первоначальном добавлении межсетевого экрана, а затем, после тестирования его работоспособности, отредактировать правила в файле `/etc/rc.firewall`. Дополнительной предосторожностью может быть первоначальная настройка межсетевого экрана с локальной консоли, вместо входа через **ssh**. Кроме того, возможна сборка ядра с параметрами `IPFIREWALL` и `IPFIREWALL_DEFAULT_TO_ACCEPT`. В этом случае правило IPFW по умолчанию будет изменено на `allow ip from any to any`, что предотвратит возможную блокировку.

Существует четыре параметра настройки ядра, относящихся к IPFW:

`options IPFIREWALL`

Включает в ядро код для фильтрации пакетов.

`options IPFIREWALL_VERBOSE`

Включает протоколирование пакетов через `syslogd(8)`. Без этого параметра, даже если вы укажете в правилах фильтрации протоколировать пакеты, это не работает.

`options IPFIREWALL_VERBOSE_LIMIT=10`

Ограничивает число пакетов, протоколируемых каждым правилом через `syslogd(8)`. Вы можете использовать этот параметр если хотите протоколировать работу межсетевого экрана, но не хотите делать возможной DoS атаку путем переполнения `syslog`.

Когда для одного из правил в цепочке достигается определенный параметром предел, протоколирование для этого правила выключается. Для включения протоколирования, вам потребуется сбросить соответствующий счетчик с помощью утилиты `ipfw(8)`:

```
# ipfw zero 4500
```

где 4500 это номер правила, для которого вы хотите возобновить протоколирование.

`options IPFIREWALL_DEFAULT_TO_ACCEPT`

Изменяет правило по умолчанию с "deny" на "allow". Это предотвращает возможное блокирование, если ядро загружено с поддержкой `IPFIREWALL`, но межсетевой экран еще не настроен. Этот параметр также полезен, если вы используете `ipfw(8)` в качестве средства от определенных проблем по мере их возникновения. Тем не менее, используйте параметр с осторожностью, поскольку он открывает межсетевой экран и изменяет его поведение.

**Замечание:** Предыдущие версии FreeBSD содержали параметр `IPFIREWALL_ACCT`. Этот параметр устарел, поскольку код автоматически включает возможность учета.

## 14.9.4. Настройка IPFW

Настройка программного обеспечения IPFW выполняется с помощью утилиты `ipfw(8)`. Синтаксис этой команды выглядит очень сложным, но он становится относительно прост как только вы поймете его структуру.

В настоящее время утилита использует четыре различных категории команд: добавление/удаление (addition/deletion), просмотр (listing), сброс (flushing) и очистка (clearing). Добавление/удаление используется для создания правил, определяющих как пакеты принимаются, отбрасываются и протоколируются. Просмотр используется для определения содержимого набора правил (называемого еще цепочкой) и счетчиков пакетов (учет). Сброс используется для удаления всех правил цепочки. Очистка используется для обнуления одного или нескольких счетчиков.

### 14.9.4.1. Изменение правил IPFW

Синтаксис этой формы команды такой:

```
ipfw [-N] команда [номер] действие [log] протокол адреса [параметры]
```

При использовании этой формы команды доступен один флаг:

-N

Разрешение адресов и имен сервисов при отображении.

Задаваемая *команда* может быть сокращена до более короткой уникальной формы. Существующие *команды*:

add

Добавление правила к списку фильтрации/учета

delete

Удаление правила из списка фильтрации/учета

Предыдущие версии IPFW использовали отдельные записи для фильтрации и учета пакетов. Современные версии учитывают пакеты для каждого правила.

Если указано значение *номер*, оно используется для помещения правила на определенную позицию в цепочке. Иначе правило помещается в конец цепочки с номером на 100 больше, чем у предыдущего правила (сюда не включается правило по умолчанию с номером 65535).

С параметром `log` соответствующие правила выводят информацию на системную консоль, если ядро собрано с опцией `IPFIREWALL_VERBOSE`.

Существующие *действия*:

reject

Отбросить пакет и отправить в адрес источника ICMP пакет, сообщающий о недостижимости хоста или порта.

allow

Пропустить пакет как обычно. (синонимы: pass, permit, и accept)

deny

Отбросить пакет. Источнику не выдается ICMP сообщение (как если бы пакет вообще не достиг цели).

count

Обновить счетчик пакета, но не применять по отношению к нему правила allow/deny. Поиск продолжится со следующего правила в цепочке.

Каждое *действие* может быть записано в виде более короткого уникального префикса.

Могут быть определены следующие *протоколы*:

all

Соответствует всем IP пакетам

icmp

Соответствует ICMP пакетам

tcp

Соответствует TCP пакетам

udp

Соответствует UDP пакетам

Поле *адреса* формируется так:

источник *адрес/маска* [*порт*] цель *адрес/маска* [*порт*] [*via интерфейс*]

Вы можете указать *порт* только вместе с *протоколами*, поддерживающими порты (UDP и TCP).

Параметр *via* опционален и может содержать IP адрес или имя домена локального IP интерфейса, или имя интерфейса (например *ed0*), он настраивает правило на соответствие только тем пакетам, которые проходят через этот интерфейс. Номера интерфейсов могут быть заменены на опциональную маску. Например, *ppp\** будет соответствовать PPP интерфейсам ядра.

Синтаксис, используемый для указания *адреса/маски*:

*адрес*

или

*адрес / маска-биты*

или

*адрес : маска-шаблон*



Вместо IP адреса возможно указание существующего имени хоста. *маска-биты* это десятичный номер, указывающий количество бит, которые должны быть установлены в маске адреса. Например, 192.216.222.1/24 создаст маску, соответствующую всем адресам подсети класса C (в данном случае, 192.216.222). A valid hostname may be specified in place of the IP address. *маска-шаблон* это IP, который будет логически перемножен с заданным адресом. Ключевое слово *any* может использоваться для обозначения "любого IP адреса".

Номера портов указываются в следующем формате:

*порт* [, *порт* [, *порт* [...]]]

для указания одного порта или списка портов, или

*порт-порт*

для указания диапазона портов. Вы можете также комбинировать указание одного диапазона со списком портов, но диапазон всегда должен указываться первым.

Доступные *параметры*:

*frag*

Срабатывает, если пакет не является первым пакетом дейтаграммы.

*in*

Соответствует входящим пакетам.

*out*

Соответствует исходящим пакетам.

*ipoptions spec*

Срабатывает, если заголовок IP содержит перечисленный через запятую список параметров, указанных в *spec*. Поддерживаемые параметры IP: *ssrr* (strict source route), *lsrr* (loose source route), *rr* (record packet route), и *ts* (time stamp). Действие отдельных параметров может быть изменено путем указания префикса !.

*established*

Срабатывает, если пакет является частью уже установленного TCP соединения (т.е. если установлены биты RST или ACK). Вы можете поднять производительность межсетевого экрана, поместив правило с *established* близко к началу цепочки.

*setup*

Соответствует, если пакет является попыткой установки TCP соединения (установлен бит SYN, а бит ACK не установлен).

**tcpflags флаги**

Срабатывает, если заголовок TCP содержит список перечисленных через запятую *флагов*.  
Поддерживаемые флаги: `fin`, `syn`, `rst`, `psh`, `ack`, и `urg`. Действие правил по отдельным флагам может быть изменено указанием префикса `!`.

**icmptypes типы**

Срабатывает, если тип пакета ICMP находится в списке *типы*. Список может быть указан в виде любой комбинации диапазонов и/или отдельных типов, разделенных запятыми. Обычно используемые типы ICMP: 0 echo reply (ping reply), 3 destination unreachable, 5 redirect, 8 echo request (ping request), и 11 time exceeded (используется для обозначения истечения TTL, как с `traceroute(8)`).

**14.9.4.2. Просмотр правил IPFW**

Синтаксис этой формы команды такой:

```
ipfw [-a] [-c] [-d] [-e] [-t] [-N] [-S] list
```

Для этой формы команды существует семь флагов:

**-a**

Показывать значения счетчиков. Этот параметр — единственный путь для просмотра значений счетчиков.

**-c**

Просмотр правил в компактной форме.

**-d**

Показывать динамические правила в дополнение к статическим.

**-e**

Если определен параметр `-d`, показывать также динамические правила с истекшим сроком действия.

**-t**

Отображать последнее время срабатывания для каждого правила в цепочке. Этот список несовместим с синтаксисом, принимаемым `ipfw(8)`.

**-N**

Попытаться разрешить заданные адреса и имена сервисов.

**-S**

Отображать набор, к которому принадлежит каждое правило. Если этот флаг не указан, заблокированные правила не будут отображены.

### 14.9.4.3. Сброс правил IPFW

Синтаксис для сброса правил:

```
ipfw flush
```

Все правила в цепочке будут удалены, за исключением правила по умолчанию, устанавливаемого ядром (номер 65535). Будьте осторожны при сбросе правил; правило, отбрасывающее пакеты по умолчанию отключит систему от сети, пока разрешающие правила не будут добавлены в цепочку.

### 14.9.4.4. Очистка счетчиков пакетов IPFW

Синтаксис для очистки одного или нескольких счетчиков пакетов:

```
ipfw zero [index]
```

При использовании без аргумента *номер* будут очищены все счетчики пакетов. Если *index* указан, операция очистки применяется только к указанному правилу цепочки.

## 14.9.5. Примеры команд для ipfw

Следующая команда запретит все пакеты с хоста `evil.crackers.org` на `telnet` порт хоста `nice.people.org`:

```
# ipfw add deny tcp from evil.crackers.org to nice.people.org 23
```

Следующий пример запрещает и протоколирует весь TCP трафик из сети `crackers.org` (класса C) к компьютеру `nice.people.org` (на любой порт).

```
# ipfw add deny log tcp from evil.crackers.org/24 to nice.people.org
```

Если вы хотите запретить организацию X сессий в вашу сеть (часть сети класса C), следующая команда осуществит необходимую фильтрацию:

```
# ipfw add deny tcp from any to my.org/28 6000 setup
```

Для просмотра записей учета:

```
# ipfw -a list
```

или в краткой форме

```
# ipfw -a l
```

Вы можете также просмотреть время последнего срабатывания правил с помощью команды:

```
# ipfw -at 1
```

### 14.9.6. Создание межсетевого экрана с фильтрацией пакетов

**Замечание:** Следующие рекомендации означают только одно: рекомендации. Требования к каждому межсетевому экрану различаются, и мы не можем рассказать вам, как создать межсетевой экран, отвечающий вашим потребностям.

При первоначальной настройке межсетевого экрана, до тестирования производительности и введения сервера в строй, настоятельно рекомендуется использовать версии команд с протоколированием и включить протоколирование в ядре. Это позволит вам быстро выявить проблемные области и исправить настройку без больших усилий. Даже после завершения первоначальной настройки рекомендуется использовать протоколирование для `'deny'`, поскольку это позволяет отслеживать возможные атаки и изменять правила межсетевого экрана, если требования к нему изменятся.

**Замечание:** Если вы используете версию команды `accept` с протоколированием, будьте осторожны, поскольку она может создать *большой* объем протокольных данных. Будет произведено протоколирование каждого пакета, проходящего через межсетевой экран, поэтому большие объемы FTP/http и другого трафика существенно замедлят систему. Это также увеличит задержку таких пакетов, поскольку ядру требуется выполнить дополнительную работу перед тем, как пропустить пакет. **syslogd** также будет использовать гораздо больше времени процессора, поскольку он отправит все дополнительные данные на диск, и раздел `/var/log` может быть быстро заполнен.

Вам потребуется включить межсетевой экран в `/etc/rc.conf.local` или `/etc/rc.conf`.

Соответствующая страница справочника разъясняет что именно необходимо сделать и содержит примеры готовых настроек. Если вы не используете предустановленную настройку, команда `ipfw list` может поместить текущий набор правил в файл, откуда он может быть помещен в стартовые файлы системы. Если вы не используете `/etc/rc.conf.local` или `/etc/rc.conf` для включения межсетевого экрана, важно убедиться в том, что он включается после настройки интерфейсов.

Далее необходимо определить, *что именно* делает ваш межсетевой экран! Это в основном зависит от того, насколько широкий доступ вы хотите открыть снаружи к вашей сети. Вот несколько общих правил:

- Заблокируйте доступ снаружи к портам TCP с номерами ниже 1024. Здесь расположена большая часть критичных для безопасности сервисов, таких как `finger`, `SMTP` (почта) и `telnet`.
- Заблокируйте *весь* входящий трафик UDP. Есть очень немного полезных сервисов, работающих через UDP, но они обычно представляют угрозу безопасности (например, `Sun RPC` и `NFS` протоколы). У этого способа есть и недостатки, поскольку протокол UDP не поддерживает соединения, и запрещение входящих пакетов заблокирует также ответы на исходящий UDP трафик. Это может стать проблемой для тех, кто использует внешние серверы, работающие с UDP. Если вы хотите открыть доступ к этим сервисам, потребуется разрешить входящие пакеты с соответствующих портов. К примеру, для **ntp** вам может потребоваться разрешить пакеты, приходящие с порта 123.

- Заблокировать весь трафик снаружи к порту 6000. Порт 6000 используется для доступа к серверам X11, и может быть угрозой безопасности (особенно если у пользователей есть привычка выполнять на своих рабочих станциях команду `xhost +`). X11 может использовать диапазон портов, начинающийся с 6000, верхний предел определяется количеством X дисплеев, которые могут быть запущены на машине. Верхний предел, определенный RFC 1700 (Assigned Numbers), равен 6063.
- Проверьте порты, используемые внутренними сервисами (например, SQL серверами и т.п.). Возможно хорошей идеей является блокирование и этих портов, поскольку они обычно не попадают в диапазон 1-1024, указанный выше.

Еще один список для проверки настроек межсетевого экрана доступен на CERT по адресу [http://www.cert.org/tech\\_tips/packet\\_filtering.html](http://www.cert.org/tech_tips/packet_filtering.html)

Как сказано выше, все эти правила всего лишь *руководство*. Вы сами сможете решить, какие правила фильтрации будут использованы в межсетевом экране. Мы не можем нести НИКАКОЙ ответственности в случае взлома вашей сети, даже если вы следовали советам, представленным выше.

### 14.9.7. Накладные расходы и оптимизация IPFW

Многие пользователи хотят знать, как сильно IPFW нагружает систему. Ответ в основном зависит от набора правил и скорости процессора. При небольшом наборе правил для большинства приложений, работающих в Ethernet ответ "незначительно". Для тех, кому нужен более точный ответ, и предназначен этот раздел.

Последующие измерения были выполнены с 2.2.5-STABLE на 486-66. (Хотя IPFW немного изменился в последующих релизах FreeBSD, скорость осталась приблизительно той же.) IPFW был модифицирован для измерения времени, затраченного `ip_fw_chk`, с выводом на консоль результата после каждого 1000-го пакета.

Были протестированы два набора из 1000 правил. Первый был составлен для демонстрации плохого набора правил путем повторения правила:

```
# ipfw add deny tcp from any to any 55555
```

Этот набор правил плох, поскольку большая часть правил IPFW не соответствует проверяемым пакетам (из-за номера порта). После 999-й итерации этого правила следует правило `allow ip from any to any`.

Второй набор правил был разработан для быстрой проверки каждого правила:

```
# ipfw add deny ip from 1.2.3.4 to 1.2.3.4
```

Не совпадающий IP адрес источника в правиле выше приведет к очень быстрой проверке этих правил. Как и прежде, 1000-е правило `allow ip from any to any`.

Затраты на проверку пакета в первом случае приблизительно 2.703 мс/пакет, или приблизительно 2.7 микросекунд на правило. Теоретический предел скорости проверки около 370 пакетов в секунду. Предполагая подключение через 10 Mbps Ethernet и размер пакета приблизительно 1500 байт, получаем только 55.5% использования пропускной способности.

Во втором случае каждый пакет был проверен приблизительно за 1.172 мс, или приблизительно 1.2 микросекунд на правило. Теоретический предел скорости проверки около 853 пакетов в секунду, что делает возможным полное использование пропускной способности 10 Mbps Ethernet.

Чрезмерное количество проверяемых правил и их вид не позволяет составить картину близкую к обычным условиям — эти правила были использованы только для получения информации о времени проверки. Вот несколько рекомендаций, которые необходимо учесть для создания эффективного набора правил:

- Поместите правило `established` как можно раньше для обработки большей части TCP трафика. Не помещайте перед ним правила `allow tcp`.
- Помещайте часто используемые правила ближе к началу набора чем редко используемые (конечно же, *без изменения действия всего набора*). Вы можете определить наиболее часто используемые правила путем проверки счетчиков пакетов командой `ipfw -a 1`.

## 14.10. OpenSSL

*Написал: Tom Rhodes.*

Одной из программ, требующих особого внимания пользователей, является набор программ **OpenSSL**, включенный в FreeBSD. **OpenSSL** предоставляет уровень шифрования поверх обычных уровней соединения; следовательно, он может быть использован многими сетевыми приложениями и сервисами.

**OpenSSL** может использоваться для шифрования соединений почтовых клиентов, транзакций через интернет, например для кредитных карт, и многого другого. Многие порты, такие как `www/apache13-ssl` и `mail/sylpheed-claws` собираются с **OpenSSL**.

**Замечание:** В большинстве случаев в коллекции портов собирается порт `security/openssl`, если только переменная `WITH_OPENSSL_BASE` не установлена явно в "yes".

Версия **OpenSSL**, включаемая в FreeBSD, поддерживает сетевые протоколы безопасности Secure Sockets Layer v2/v3 (SSLv2/SSLv3), Transport Layer Security v1 (TLSv1) и может быть использована в качестве основной криптографической библиотеки для приложений.

**Замечание:** Хотя **OpenSSL** поддерживает алгоритм IDEA, по умолчанию он отключен из-за патентных ограничений Соединенных Штатов. Для его использования необходимо ознакомиться с лицензией, и, если ограничения приемлемы, установить в `make.conf` переменную `MAKE_IDEA`.

Возможно наиболее часто **OpenSSL** используется для предоставления сертификатов программным пакетам. Эти сертификаты подтверждают, что данные компании или частного лица верны и не подделаны. Если рассматриваемый сертификат не был проверен одним из нескольких Certificate Authorities (CA), обычно выводится предупреждение. Certificate Authority это компания, такая как VeriSign, которая подписывает сертификаты для подтверждения данных частных лиц или компаний. Эта процедура не бесплатна и не является абсолютно необходимой для использования сертификатов; однако может успокоить некоторых особо осторожных пользователей.

### 14.10.1. Генерирование сертификатов

Для генерирования сертификатов доступна следующая команда:

```
# openssl req -new -nodes -out req.pem -keyout cert.pem
Generating a 1024 bit RSA private key
.....+++++
.....+++++
writing new private key to 'cert.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:PA
Locality Name (eg, city) []:Pittsburgh
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:Systems Administrator
Common Name (eg, YOUR name) []:localhost.example.org
Email Address []:trhodes@FreeBSD.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:SOME PASSWORD
An optional company name []:Another Name
```

Ввод после приглашения "Common Name" содержит имя домена. Здесь вводится имя сервера для верификации; помещение в это поле чего-либо кроме этого имени приведет к созданию бесполезного сертификата. Доступны и другие параметры, например срок действия, альтернативные алгоритмы шифрования и т.д. Полный список находится на странице справочного руководства openssl(1).

Файл cert.pem должен находиться в каталоге, из которого была выполнена вышеупомянутая команда. Этот сертификат может быть отправлен одному из многих СА для подписи.

Когда подпись СА не требуется, может быть создан самоподписанный сертификат. Сначала создайте ключ СА:

```
# openssl gendsa -des3 -out \
mysa.key 1024
```

Используйте этот ключ при создании сертификата:

```
# openssl req -new -x509 -days 365 -key \
mysa.key -out new.crt
```

В каталоге должно появиться два новых файла: подпись сертификата, mysa.key и сам сертификат, new.crt. Они должны быть помещены в каталог, доступный для чтения только root, желательно внутри /etc. Права на каталог можно изменить chmod с параметрами 0600.

### 14.10.2. Использование сертификатов, пример

Итак, что могут сделать эти файлы? Хорошим применением может стать шифрование соединений для **Sendmail** MTA. Это сделает ненужным использование простой текстовой аутентификации для тех, кто отправляет почту через локальный MTA.

**Замечание:** Это не лучшее из возможных использований, поскольку некоторые MUA выдадут ошибку, если сертификат не установлен локально. Обратитесь к поставляемой с программой документации за информацией по установке сертификата.

Следующие строки должны быть помещены в локальный файл `.mc`:

```
dnl SSL Options
define(`confCACERT_PATH', `/etc/certs')dnl
define(`confCACERT', `/etc/certs/new.crt')dnl
define(`confSERVER_CERT', `/etc/certs/new.crt')dnl
define(`confSERVER_KEY', `/etc/certs/myca.key')dnl
define(`confTLS_SRV_OPTIONS', `V')dnl
```

Где `/etc/certs/` это каталог для локального хранения сертификата и ключей. После настройки необходимо собрать локальный файл `.cf`. Это легко сделать, набрав `make install` в каталоге `/etc/mail`. Затем выполните команду `make restart`, которая должна запустить демон **Sendmail**.

Если все пройдет нормально, в файле `/var/log/maillog` не появятся сообщения об ошибках и запустится процесс **Sendmail**.

Для проведения простого теста подключитесь к почтовому серверу программой `telnet(1)`:

```
# telnet example.com 25
Trying 192.0.34.166...
Connected to example.com.
Escape character is '^]'.
220 example.com ESMTP Sendmail 8.12.10/8.12.10; Tue, 31 Aug 2004 03:41:22 -0400 (EDT)
ehlo example.com
250-example.com Hello example.com [192.0.34.166], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH LOGIN PLAIN
250-STARTTLS
250-DELIVERBY
250 HELP
quit
221 2.0.0 example.com closing connection
Connection closed by foreign host.
```

Если в выводе появилась строка `'STARTTLS'`, все работает правильно.



## 14.11. VPN через IPsec

Написал Nik Clayton.

Создание VPN между двумя сетями, соединенными через интернет, с использованием шлюзов FreeBSD.

### 14.11.1. Принципы работы IPsec

Написал Hiten M. Pandya.

Этот раздел послужит вам руководством по настройке IPsec и его использованию в среде FreeBSD и **Microsoft Windows 2000/XP**, соединяемых безопасным способом. Для настройки IPsec необходимо ознакомиться с процессом сборки ядра (Гл. 8).

*IPsec* это протокол, расположенный поверх слоя Internet Protocol (IP). Он позволяет двум или более хостам связываться защищенным способом (отсюда и название протокола). "Сетевой стек" FreeBSD IPsec основан на реализации KAME (<http://www.kame.net/>), поддерживающей оба семейства протоколов, IPv4 и IPv6.

**Замечание:** FreeBSD 5.X содержит "аппаратно поддерживаемый" стек IPsec, известный как "Fast IPsec", заимствованный из OpenBSD. Для оптимизации производительности IPsec он задействует криптографическое оборудование (когда оно доступно) через подсистему `crypto(4)`. Это новая подсистема и она не поддерживает всех возможностей, доступных в KAME версии IPsec. Для включения IPsec с аппаратной поддержкой необходимо добавить в файл настройки ядра следующий параметр:

```
options    FAST_IPSEC    # new IPsec (cannot define w/ IPSEC)
```

Обратите внимание, что на данный момент невозможно использовать подсистему "Fast IPsec" вместе с KAME реализацией IPsec. Обратитесь к странице справочника `fast_ipsec(4)` за дальнейшей информацией.

IPsec состоит из двух субпротоколов:

- *Encapsulated Security Payload (ESP)*, защищающей данные IP пакета от вмешательства третьей стороны путем шифрования содержимого с помощью симметричных криптографических алгоритмов (таких как Blowfi sh,3DES).
- *Authentication Header (AH)*, защищающий заголовок IP пакета от вмешательства третьей стороны и подделки путем вычисления криптографической контрольной суммы и хеширования полей заголовка IP пакета защищенной функцией хеширования. К пакету добавляется дополнительный заголовок с хешем, позволяющий аутентификацию информации пакета.

ESP и AH могут быть использованы вместе или по отдельности, в зависимости от обстоятельств.

IPsec может быть использован или для непосредственного шифрования трафика между двумя хостами (*транспортный режим*); или для построения "виртуальных туннелей" между двумя подсетями, которые могут быть использованы для защиты соединений между двумя корпоративными сетями (*туннельный режим*). Последний обычно называют *виртуальной частной сетью* (Virtual Private Network, VPN). За детальной информацией о подсистеме IPsec в FreeBSD обратитесь к странице справочника `ipsec(4)`.

Для включения поддержки IPsec в ядре, добавьте следующие параметры к файлу настройки ядра:

```
options    IPSEC          #IP security
options    IPSEC_ESP      #IP security (crypto; define w/ IPSEC)
```

Если желательна поддержка отладки IPsec, должна быть также добавлена следующая строка:

```
options    IPSEC_DEBUG    #debug for IP security
```

### 14.11.2. Проблема

Не существует стандарта VPN. Они могут быть реализованы множеством различных технологий, каждая из которых имеет свои сильные и слабые стороны. Этот раздел представляет сценарий и стратегию реализации VPN для этого сценария.

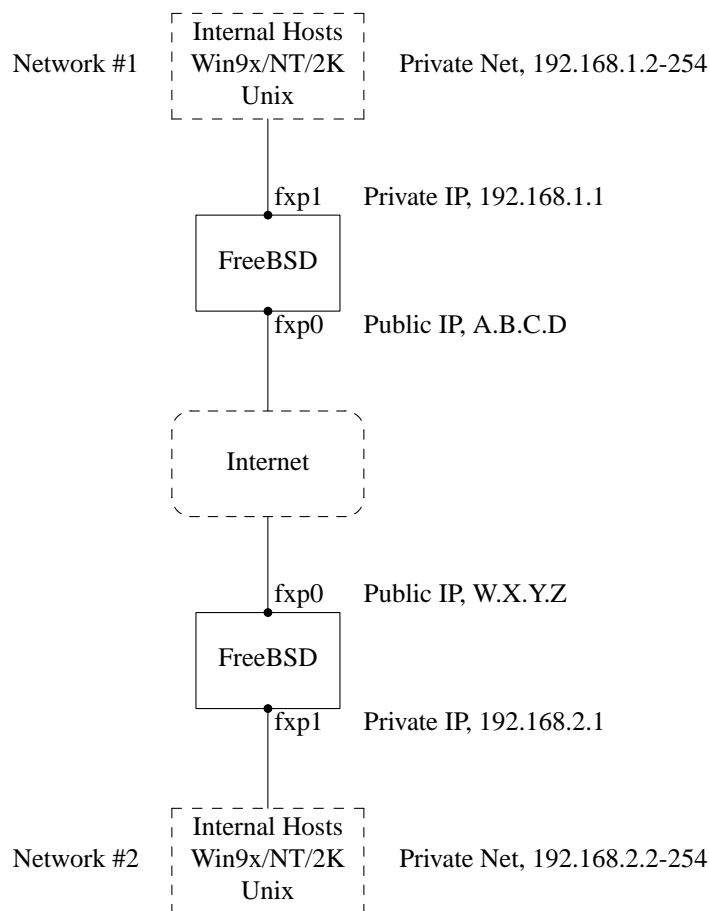
### 14.11.3. Сценарий: Две сети, подключенных к интернет, работающие как одна

Исходные условия таковы:

- Существует как минимум две сети
- Внутри обеих сетей используется IP
- Обе сети соединены через интернет через шлюз, работающий на FreeBSD.
- У шлюза каждой из сетей есть как минимум один публичный IP адрес.
- Внутренние IP адреса двух сетей могут быть публичными или приватными, не имеет значения. На шлюзе может работать NAT, если это необходимо.
- Внутренние IP адреса двух сетей *не должны пересекаться*. Хотя вероятно теоретически возможно использование комбинации VPN технологии и NAT для настройки такой конфигурации, эта конфигурация будет кошмарна.

Если две сети, которые вы пытаетесь соединить, используют один и тот же диапазон приватных адресов (например, обе используют 192.168.1.x), номера в одной из сетей необходимо изменить.

Топология сети может выглядеть примерно так:



Здесь два публичных IP адреса. Для упоминания их в дальнейшем будут использоваться буквы. Если вы увидите эти буквы, замените их на свои публичные IP адреса. Также обратите внимание, что у обоих шлюзов внутренний адрес заканчивается на .1 и диапазоны частных адресов двух сетей различны (192.168.1.x и 192.168.2.x соответственно). Все компьютеры локальных сетей настроены на использование в качестве шлюза по умолчанию компьютера с адресом, оканчивающимся на .1.

С сетевой точки зрения замысел в том, чтобы каждая сеть видела компьютеры из другой сети так, как если бы они были непосредственно подключены к тому же самому маршрутизатору — хотя и немного медленному маршрутизатору, иногда теряющему пакеты.

Это означает, что (например) компьютер 192.168.1.20 может запустить

```
ping 192.168.2.34
```

и это будет прозрачно работать. Компьютеры с Windows должны видеть компьютеры в другой сети, просматривать сетевые ресурсы, и так далее, точно так же, как и для компьютеров в локальной сети.

И все это безопасным способом. Это означает, что трафик между сетями зашифрован.

Создание VPN между этими двумя сетями это многошаговый процесс. Этапы создания VPN таковы:

1. Создание "виртуального" сетевого подключения между двумя сетями через интернет.  
Тестирование подключения с помощью таких инструментов как `ping(8)`, чтобы убедиться, что оно работает.

2. Применение политики безопасности чтобы убедиться, что трафик между двумя сетями прозрачно шифруется и расшифровывается если необходимо. Тестирование с помощью таких инструментов как `tcpdump(1)`, чтобы убедиться, что трафик шифруется.
3. Настройка дополнительных программ на шлюзах FreeBSD, чтобы компьютеры Windows из одной сети видели компьютеры в другой через VPN.

#### 14.11.3.1. Шаг 1: Создание и тестирование "виртуального" сетевого подключения

Предположим, что вы работаете на шлюзе сети #1 (с публичным адресом A.B.C.D, приватным адресом 192.168.1.1) и запускаете `ping 192.168.2.1`, т.е. на приватный адрес машины с IP адресом W.X.Y.Z. Что должно произойти, чтобы это сработало?

1. Шлюз должен знать, как достичь 192.168.2.1. Другими словами, у него должен быть маршрут к 192.168.2.1.
2. Приватные IP адреса, такие как диапазон 192.168.x не адресуются в интернет. Каждый пакет, отправляемый на 192.168.2.1 должен быть "завернут" в другой пакет. Исходным адресом пакета должен быть A.B.C.D, а адресом назначения W.X.Y.Z. Этот процесс называется *инкапсуляцией*.
3. Как только этот пакет достигнет W.X.Y.Z, необходимо будет "разинкапсулировать" его и доставить к 192.168.2.1.

Как вы можете увидеть, это требует "туннеля" между двумя сетями. Два конца "туннеля" это IP адреса A.B.C.D и W.X.Y.Z. Туннель используется для передачи трафика с приватными IP адресами через интернет.

В FreeBSD этот туннель создается с помощью устройства `generic interface`, или `gif`. Как вы можете догадаться, интерфейс `gif` на каждом хосте должен быть настроен с четырьмя IP адресами; два для публичных IP адресов и два для приватных IP адресов.

В ядро обеих компьютеров FreeBSD должна быть встроена поддержка устройства `gif`. Вы можете сделать это, добавив строку:

```
pseudo-device gif
```

к файлу настройки ядра на обеих компьютерах, с последующей компиляцией, установкой и перезагрузкой.

Настройка туннеля это двухшаговый процесс. Во-первых, необходимо задать сведения о внешнем (или публичном) IP адресе с помощью `gifconfi g(8)`. Затем о приватном IP адресе с помощью `ifconfi g(8)`.

**Замечание:** В FreeBSD 5.X функциональность, предоставляемая утилитой `gifconfig(8)`, была внесена в `ifconfig(8)`.

На шлюзе сети #1 для настройки туннеля вам потребуется запустить следующие две команды.

```
gifconfig gif0 A.B.C.D W.X.Y.Z
ifconfig gif0 inet 192.168.1.1 192.168.2.1 netmask 0xffffffff
```

На другом шлюзе подобные команды, но с IP адресами в обратном порядке.

```
gifconfig gif0 W.X.Y.Z A.B.C.D
ifconfig gif0 inet 192.168.2.1 192.168.1.1 netmask 0xffffffff
```

Затем вы можете запустить:

```
gifconfig gif0
```

для просмотра настройки. Например, на шлюзе сети #1 вы увидите:

```
# gifconfig gif0
gif0: flags=8011<UP,POINTTOPOINT,MULTICAST> mtu 1280
inet 192.168.1.1 --> 192.168.2.1 netmask 0xffffffff
physical address inet A.B.C.D --> W.X.Y.Z
```

Как вы можете видеть, был создан туннель между физическими адресами A.B.C.D и W.X.Y.Z, для туннелирования разрешен трафик между 192.168.1.1 и 192.168.2.1.

Это также добавляет запись к таблице маршрутизации на обеих машинах, вы можете проверить запись командой `netstat -rn`. Вот вывод этой команды на шлюзе сети #1.

```
# netstat -rn
Routing tables

Internet:
Destination      Gateway          Flags    Refs      Use    Netif  Expire
...
192.168.2.1      192.168.1.1    UH        0         0     gif0
...
```

Как показывает значение поля 'Flags', это маршрут к хосту, что означает, что каждый шлюз знает, как достичь другого шлюза, но не знает как достичь остальной части соответствующей сети. Эта проблема будет быстро решена.

Вероятно, на обеих машинах запущен межсетевой экран. VPN должен обходить его. Вы можете разрешить весь трафик между двумя сетями, или включить правила, защищающие каждый конец соединения от другого.

Это сильно упрощает тестирование настройки межсетевого экрана, если вы разрешаете весь трафик через VPN. Вы всегда можете усилить защиту позже. Если вы используете на шлюзах `ipfw(8)`, команда вроде этой

```
ipfw add 1 allow ip from any to any via gif0
```

разрешит весь трафик между двумя концами VPN без влияния на другие правила межсетевого экрана. Очевидно, вам потребуется запустить эту команду на обеих шлюзах.

Этого достаточно для включения `ping` с одного шлюза на другой. На 192.168.1.1, вы сможете запустить

```
ping 192.168.2.1
```

и получить ответ, и аналогично на другом шлюзе.

Однако, машины в другой сети пока недоступны. Это из-за маршрутизации — хотя шлюзы знают, как связаться друг с другом, они не знают, как связаться с сетью за другим шлюзом.

Для решения этой проблемы вы должны добавить статический маршрут на каждом шлюзе. Команда на первом шлюзе будет выглядеть так:

```
route add 192.168.2.0 192.168.2.1 netmask 0xffffffff00
```

Она говорит "Для достижения хостов в сети 192.168.2.0, отправляйте пакеты хосту 192.168.2.1". Вам потребуется запустить похожую команду на другом шлюзе, но с адресами 192.168.1.x.

IP трафик с хостов в одной сети теперь может достичь хосты в другой сети.

Теперь создано две трети VPN между двумя сетями, поскольку это "виртуальная (virtual)" "сеть (network)". Она еще не приватная (private). Вы можете протестировать ее с помощью ping(8) и tcpdump(1). Войдите на шлюз и запустите

```
tcpdump dst host 192.168.2.1
```

В другой сессии на этом же хосте запустите

```
ping 192.168.2.1
```

Вы увидите примерно такие строки:

```
16:10:24.018080 192.168.1.1 > 192.168.2.1: icmp: echo request
16:10:24.018109 192.168.1.1 > 192.168.2.1: icmp: echo reply
16:10:25.018814 192.168.1.1 > 192.168.2.1: icmp: echo request
16:10:25.018847 192.168.1.1 > 192.168.2.1: icmp: echo reply
16:10:26.028896 192.168.1.1 > 192.168.2.1: icmp: echo request
16:10:26.029112 192.168.1.1 > 192.168.2.1: icmp: echo reply
```

Как вы видите, ICMP сообщения пересылаются вперед и назад незашифрованными. Если вы использовали с tcpdump(1) параметр -s для получения большего объема данных пакета, то увидите больше информации.

Конечно же это неприемлемо. В следующем разделе мы обсудим защиту соединения между двумя сетями, так что весь трафик будет автоматически шифроваться.

### Резюме:

- Настройте оба ядра с "pseudo-device gif".
- Отредактируйте /etc/rc.conf на шлюзе #1 и добавьте следующие строки (подставляя IP адреса где необходимо).

```
gifconfig_gif0="A.B.C.D W.X.Y.Z"
ifconfig_gif0="inet 192.168.1.1 192.168.2.1 netmask 0xffffffff"
static_routes="vpn"
route_vpn="192.168.2.0 192.168.2.1 netmask 0xffffffff00"
```

- Отредактируйте скрипт межсетевого экрана (`/etc/rc.firewall`, или подобный) на обоих хостах и добавьте  

```
ipfw add 1 allow ip from any to any via gif0
```
- Выполните соответствующие изменения в `/etc/rc.conf` на шлюзе #2, меняя порядок IP адресов.

### 14.11.3.2. Шаг 2: Защита соединения

Для защиты соединения мы будем использовать IPsec. IPsec предоставляет хостам механизм определения ключа для шифрования и для последующего использования этого ключа для шифрования данных между двумя хостами.

Здесь будут рассмотрены два аспекта настройки.

1. У хостов должен быть способ согласования используемого алгоритма шифрования. Как только хосты договорятся об этом, можно говорить об установленном между ними "безопасном соединении".
2. Должен быть механизм определения, какой трафик необходимо шифровать. Конечно, вам не требуется шифровать весь исходящий трафик — достаточно шифровать только трафик, идущий через VPN. Правила, определяющие то, какой трафик необходимо шифровать, называются "политикой безопасности".

Безопасное соединение и политика безопасности поддерживаются ядром, и могут быть изменены программами пользователя. Однако перед тем, как вы сможете сделать это, необходимо настроить поддержку протоколов IPsec и Encapsulated Security Payload (ESP) в ядре. Это делается добавлением в настройку ядра параметров:

```
options IPSEC
options IPSEC_ESP
```

с последующим перекомпилированием, переустановкой и перезагрузкой. Как и прежде вам потребуется сделать это с ядрами на обоих шлюзах.

При настройке параметров безопасности (security associations) у вас есть два варианта. Вы можете настроить их вручную для обоих хостов, задав алгоритм шифрования, ключи для шифрования и так далее, или использовать демоны, реализующие Internet Key Exchange protocol (IKE), который делает это за вас.

Рекомендуется последнее. Помимо прочего, этот способ более прост.

Редактирование и отображение политики безопасности выполняется с помощью `setkey(8)`. По аналогии, `setkey` используется для настройки таблиц политики безопасности ядра так же, как `route(8)` используется для настройки таблиц маршрутизации ядра. `setkey` также может отображать текущие параметры безопасности, и продолжая аналогию дальше, это соответствует `netstat -r`.

Существует множество демонов для управления параметрами безопасности в FreeBSD. Здесь будет описано использование одного из них, `gasoon`. `gasoon` находится в категории `security/` коллекции портов FreeBSD и устанавливается обычным способом.

гасооп должен работать на обеих шлюзах. На каждом из хостов он настраивается с IP адресом другого конца VPN, и секретным ключом (по вашему выбору, должен быть одним и тем же на обеих шлюзах).

Эти два демона подключаются друг к другу, подтверждают, что они именно те, за кого себя выдают (используя секретный ключ, заданный вами). Затем демоны генерируют новый секретный ключ и используют его для шифрования трафика через VPN. Они периодически изменяют этот ключ, так что даже если атакующий сломает один из ключей (что теоретически почти невозможно) это не даст ему слишком много — он сломал ключ, который два демона уже сменили на другой.

Настройки гасооп сохраняются в `#{PREFIX}/etc/racoon`. Этот файл не требует слишком больших изменений. Другим компонентом настройки гасооп, который потребует изменить, является "предварительный ключ".

В настройке по умолчанию гасооп ищет его в файле `#{PREFIX}/etc/racoon/psk.txt`. Необходимо отметить, что предварительный ключ *не* используется для шифрования трафика через VPN соединение это просто маркер, позволяющий управляющим ключами дамонам доверять друг другу.

`psk.txt` содержит строку для каждого удаленного сервера, с которым происходит соединение. В этом примере два сервера, каждый файл `psk.txt` будет содержать одну строку (каждый конец VPN общается только с другим концом).

На шлюзе #1 эта строка будет выглядеть примерно так:

```
W.X.Y.Z                secret
```

То есть *публичный* IP адрес удаленной стороны, пробел и текстовая строка, секретная фраза.

На шлюзе #2 строка будет выглядеть примерно так:

```
A.B.C.D                secret
```

То есть публичный IP адрес удаленной стороны и та же секретная фраза. Перед запуском гасооп режим доступа к файлу `psk.txt` должен быть установлен в 0600 (т.е. запись и чтение только для root).

Вы должны запустить гасооп на обеих шлюзах. Вам также потребуется добавить правила для включения IKE трафика, передающегося по UDP через порт ISAKMP (Internet Security Association Key Management Protocol). Опять же, они должны быть расположены насколько возможно ближе к началу набора правил.

```
ipfw add 1 allow udp from A.B.C.D to W.X.Y.Z isakmp
ipfw add 1 allow udp from W.X.Y.Z to A.B.C.D isakmp
```

Как только гасооп будет запущен, вы можете попробовать выполнить `ping` с одного шлюза на другой. Соединение все еще не зашифровано, но гасооп установит параметры безопасности между двумя хостами — это может занять время и вы можете заметить небольшую задержку перед началом ответа команды `ping`.

Как только параметры безопасности установлены, вы можете просмотреть их используя `setkey(8)`. Запустите

```
setkey -D
```

на любом из хостов для просмотра информации о параметрах безопасности.

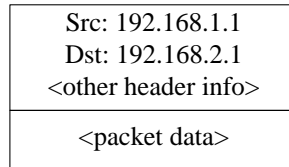
Это одна сторона проблемы. Другая сторона это настройка политики безопасности.



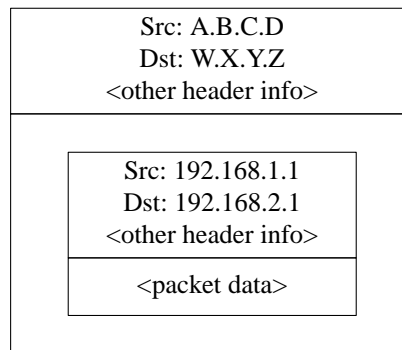
Для создания разумной политики безопасности давайте вспомним, что уже было настроено. Это рассмотрение относится к обоим концам соединения.

Каждый отправляемый IP пакет имеет заголовок, содержащий информацию о пакете. Заголовок включает IP адреса источника и назначения. Как мы уже знаем, приватные IP адреса, такие как 192.168.x.y, не могут появиться в интернет. Они должны быть сначала включены внутрь другого пакета. В этом пакете приватные IP адреса источника и назначения заменяются публичными IP адресами.

То есть исходящий пакет, который выглядит примерно так:



будет инкапсулирован в другой пакет, выглядящий примерно так:



Этой инкапсуляцией занимается устройство `gif`. Как вы можете видеть, теперь у пакета есть реальный IP адрес, исходный пакет был включен в этот пакет в виде данных, которые передаются через интернет.

Конечно, мы хотим зашифровать весь трафик между VPN. Вы можете сформулировать это на словах так:

“Если пакет отправляется с A.B.C.D, и предназначен для W.X.Y.Z, расшифровать его, используя необходимые параметры безопасности.”

“Если пакет отправляется с W.X.Y.Z, и предназначен для A.B.C.D, расшифровать его, используя необходимые параметры безопасности.”

Это похоже на желаемое, но не совсем то. Если вы сделаете это, весь трафик от и к W.X.Y.Z, даже если он не является частью VPN, будет зашифрован. Правильная политика такова:

“Если пакет отправляется с A.B.C.D, в нем инкапсулирован другой пакет и адрес назначения W.X.Y.Z, зашифровать его, используя необходимые параметры безопасности.”

“Если пакет отправляется с W.X.Y.Z, в нем инкапсулирован другой пакет и адрес назначения A.B.C.D, зашифровать его, используя необходимые параметры безопасности.”

Тонкое, но необходимое различие.

Политика безопасности также устанавливается с использованием `setkey(8)`. В `setkey(8)` предусмотрен язык определения политики `setkey(8)`. Вы можете или ввести инструкции по настройке со стандартного ввода, или использовать параметр `-f` для задания файла, содержащего эти инструкции.

Настройка на шлюзе #1 (где есть публичный IP адрес A.B.C.D) для включения шифрования всего предназначенного W.X.Y.Z трафика:

```
spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P out ipsec esp/tunnel/A.B.C.D-W.X.Y.Z/require;
```

Поместите эти команды в файл (например, `/etc/ipsec.conf`) и запустите

```
# setkey -f /etc/ipsec.conf
```

`spdadd` указывает `setkey(8)` добавить правило к базе данных политики безопасности. Остальная часть строки указывает какие пакеты будут соответствовать политике. A.B.C.D/32 и W.X.Y.Z/32 это IP адреса и сетевые маски, определяющие сети или хосты, к которым будет применяться данная политика. В данном случае мы хотим применить их к трафику между этими двумя хостами. Параметр `ipencap` сообщает ядру, что эта политика должна применяться только к пакетам, инкапсулирующим другие пакеты. Параметр `-P out` сообщает, что эта политика применяется к исходящим пакетам, и `ipsec` — то, что пакеты будут зашифрованы.

Оставшаяся часть строки определяет, как эти пакеты будут зашифрованы. Будет использоваться протокол `esp`, а параметр `tunnel` показывает, что пакет в дальнейшем будет инкапсулирован в IPsec пакет. Повторное использование A.B.C.D и W.X.Y.Z предназначено для выбора используемых параметров безопасности, и наконец параметр `require` разрешает шифрование пакетов, попадающих под это правило.

Это правило соответствует только исходящим пакетам. Вам потребуется похожее правило, соответствующее входящим пакетам.

```
spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P in ipsec esp/tunnel/W.X.Y.Z-A.B.C.D/require;
```

Обратите внимание, что вместо `in` используется `out` и IP адреса переставлены.

Другому шлюзу (с публичным IP адресом W.X.Y.Z) потребуются похожие правила.

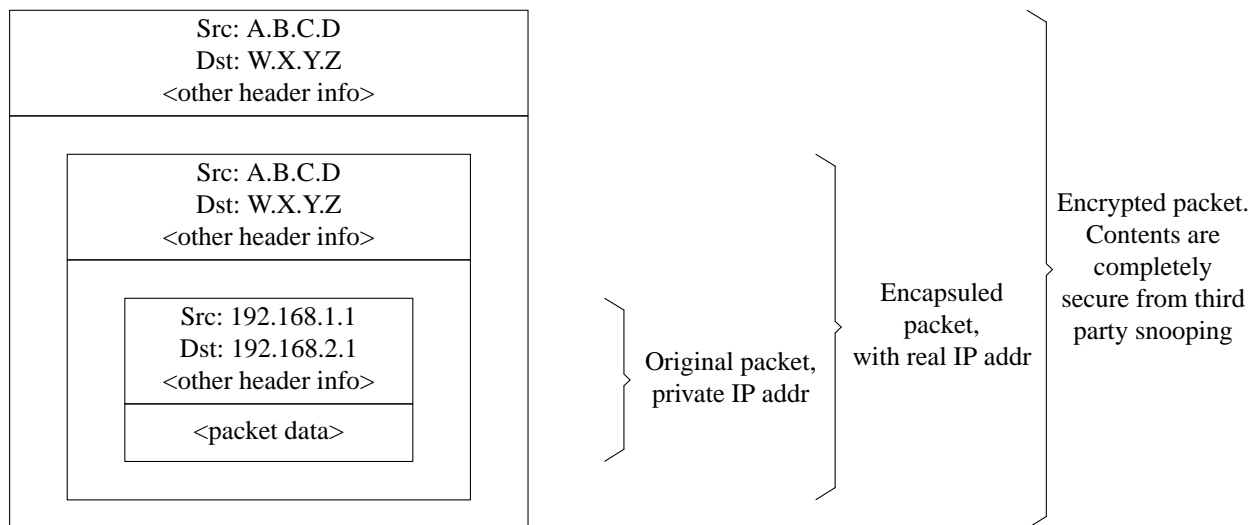
```
spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P out ipsec esp/tunnel/W.X.Y.Z-A.B.C.D/require;
spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P in ipsec esp/tunnel/A.B.C.D-W.X.Y.Z/require;
```

Наконец, вам потребуется добавить правила к межсетевому экрану для включения прохождения пакетов ESP и IPENCAP в обе стороны. На обоих хостах потребуется добавить следующие правила:

```
ipfw add 1 allow esp from A.B.C.D to W.X.Y.Z
ipfw add 1 allow esp from W.X.Y.Z to A.B.C.D
ipfw add 1 allow ipencap from A.B.C.D to W.X.Y.Z
ipfw add 1 allow ipencap from W.X.Y.Z to A.B.C.D
```

Поскольку правила симметричны, можно использовать их без изменения на обоих хостах

Исходящие пакеты теперь будут выглядеть примерно так:



Когда эти пакеты будут получены на удаленном конце VPN соединения, они будут расшифрованы (используя параметры безопасности, о которых договорился racoon). Затем они будут переданы интерфейсу gif, который "развернет" второй слой, оставив пакет с внутренними адресами, который сможет попасть во внутреннюю сеть.

Вы можете проверить безопасность тем же ping(8), который использовался ранее. Сначала войдите на шлюз A.B.C.D и запустите:

```
tcpdump dst host 192.168.2.1
```

В другой сессии на том же хосте запустите

```
ping 192.168.2.1
```

В этот момент вы должны увидеть примерно это:

```
XXX tcpdump output
```

Теперь, как видите, tcpdump(1) показывает ESP пакеты. Если вы попытаетесь просмотреть их с параметром -s, то вероятно увидите нечто непонятное, поскольку применяется шифрование.

Поздравляем. Вы только что настроили VPN между двумя удаленными сетями.

## Резюме

- Настройте оба ядра с:

```
options IPSEC
options IPSEC_ESP
```

- Установите security/racoon. Отредактируйте `${PREFIX}/etc/racoon/psk.txt` на обоих шлюзах, добавив запись для каждого IP адреса удаленного хоста и секретный ключ, который будет известен им обоим. Убедитесь, что режим доступа к файлу 0600.
- Добавьте к `/etc/rc.conf` на каждом хосте следующие строки:

```
ipsec_enable="YES"
ipsec_file="/etc/ipsec.conf"
```

- Создайте `/etc/ipsec.conf` на каждом хосте с необходимыми строками `spdadd`. На шлюзе #1 он будет таким:

```
spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P out ipsec
      esp/tunnel/A.B.C.D-W.X.Y.Z/require;
spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P in ipsec
      esp/tunnel/W.X.Y.Z-A.B.C.D/require;
```

А на шлюзе #2 таким:

```
spdadd W.X.Y.Z/32 A.B.C.D/32 ipencap -P out ipsec
      esp/tunnel/W.X.Y.Z-A.B.C.D/require;
spdadd A.B.C.D/32 W.X.Y.Z/32 ipencap -P in ipsec
      esp/tunnel/A.B.C.D-W.X.Y.Z/require;
```

- Добавьте правила к межсетевым экранам обоих хостов для включения IKE, ESP и IPENCAP трафика:

```
ipfw add 1 allow udp from A.B.C.D to W.X.Y.Z isakmp
ipfw add 1 allow udp from W.X.Y.Z to A.B.C.D isakmp
ipfw add 1 allow esp from A.B.C.D to W.X.Y.Z
ipfw add 1 allow esp from W.X.Y.Z to A.B.C.D
ipfw add 1 allow ipencap from A.B.C.D to W.X.Y.Z
ipfw add 1 allow ipencap from W.X.Y.Z to A.B.C.D
```

Двух приведенных шагов должно быть достаточно для настройки и включения VPN. Машины в каждой сети смогут обращаться друг к другу по IP адресам, и весь трафик через соединение будет автоматически надежно зашифрован.

## 14.12. OpenSSH

*Предоставил Chern Lee.*

**OpenSSH** это набор сетевых инструментов, используемых для защищенного доступа к удаленным компьютерам. Он может быть использован в качестве непосредственной замены `rlogin`, `rsh`, `rcp` и `telnet`. Кроме того, любые другие TCP/IP соединения могут быть безопасно туннелированы/перенаправлены через **SSH**. **OpenSSH** шифрует весь трафик, эффективно предотвращая кражу данных, перехват соединения и другие сетевые атаки.

**OpenSSH** поддерживается проектом OpenBSD, он основан на SSH v1.2.12 со всеми последними исправлениями и обновлениями, совместим с протоколами SSH версий 1 и 2. **OpenSSH** включен в базовую систему начиная с FreeBSD 4.0.

### 14.12.1. Преимущества использования OpenSSH

Обычно при использовании `telnet(1)` или `rlogin(1)` данные пересылаются по сети в незашифрованной форме. Перехватчик пакетов в любой точке сети между клиентом и сервером может похитить информацию о пользователе/пароле или данные, передаваемые через соединение. Для предотвращения этого **OpenSSH** предлагает различные методы шифрования.

### 14.12.2. Включение sshd

Убедитесь, что добавили в файл `rc.conf` следующую строку:

```
sshd_enable="YES"
```

При следующей загрузке системы запущен `sshd(8)`, даемон для **OpenSSH**. Вы можете также запустить **sshd** непосредственно, набрав в командной строке `sshd`.

### 14.12.3. SSH клиент

Утилита `ssh(1)` работает подобно `rlogin(1)`.

```
# ssh user@example.com
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)? yes
Host 'example.com' added to the list of known hosts.
user@example.com's password: *****
```

Вход продолжится так же, как если бы сессия была инициирована с использованием `rlogin` или `telnet`. SSH использует систему опознавательных ключей для проверки подлинности сервера при подключении клиента. Пользователю предлагается `yes` только при первом подключении. Дальнейшие попытки входа предваряются проверкой сохраненного ключа сервера. SSH клиент сообщит вам, если сохраненный ключ будет отличаться от только что полученного. Ключи серверов сохраняются в `~/.ssh/known_hosts`, или в `~/.ssh/known_hosts2` для SSH v2.

По умолчанию, сервер **OpenSSH** настроен для приема соединений SSH v1 и SSH v2. Клиент может выбирать между этими двумя протоколами. Версия 2 безопаснее своего предшественника.

Команде `ssh(1)` можно указать использование определенной версии протокола, запустив ее с параметром `-1` или `-2` для версии 1 или 2 соответственно.

### 14.12.4. Безопасное копирование

Команда `scp(1)` работает подобно `rcp(1)`; она копирует файл с удаленного компьютера, но делает это безопасным способом.

```
# scp user@example.com:/COPYRIGHT COPYRIGHT
user@example.com's password: *****
COPYRIGHT          100% |*****| 4735
00:00
#
```

Поскольку в предыдущем примере ключ сервера уже был сохранен, в этом примере он проверяется при использовании `scp(1)`.

Параметры, передаваемые `scp(1)`, похожи на параметры `cp(1)`, с файлом или файлами в качестве первого аргумента и приемником копирования во втором. Поскольку файлы передаются по сети через SSH, один или более аргументов принимают форму `user@host:<path_to_remote_file>`.

### 14.12.5. Настройка

Системные файлы настройки для даemons и клиента **OpenSSH** расположены в каталоге `/etc/ssh`.

Файл `ssh_config` используется для настройки клиента, а `sshd_config` для дамона.

Кроме того, параметры `sshd_program` (по умолчанию `/usr/sbin/sshd`), и `sshd_flags` `rc.conf` дают дополнительные возможности настройки.

### 14.12.6. ssh-keygen

Вместо использования паролей, с помощью `ssh-keygen(1)` пользователи могут аутентифицироваться ключами RSA:

```
% ssh-keygen -t rsa1
Initializing random number generator...
Generating p:  .++ (distance 66)
Generating q:  .....++ (distance 498)
Computing the keys...
Key generation complete.
Enter file in which to save the key (/home/user/.ssh/identity):
Enter passphrase:
Enter the same passphrase again:
Your identification has been saved in /home/user/.ssh/identity.
...
```

`ssh-keygen(1)` создаст пару публичного и приватного ключей, используемых для аутентификации. Приватный ключ сохраняется в `~/.ssh/identity`, а публичный в `~/.ssh/identity.pub`. Для включения аутентификации по ключам публичный ключ должен быть помещен в `~/.ssh/authorized_keys` на удаленном компьютере.

Это позволяет соединиться с удаленным компьютером с помощью RSA аутентификации вместо паролей.

**Замечание:** Параметр `-t rsa1` приведет к созданию RSA ключей, используемых SSH протоколом версии 1. Если вы хотите использовать RSA ключи с SSH протоколом версии 2, используйте команду `ssh-keygen -t rsa`.

Если при генерации ключей был использован пароль, каждый раз для при использовании приватного ключа он будет запрашиваться у пользователя.

DSA ключ для SSH протокола версии 2 может быть создан в тех же целях командой `ssh-keygen -t dsa`. Эта команда создаст публичный/приватный ключи DSA для использования только с SSH

протоколом версии 2. Публичный ключ сохраняется в `~/.ssh/id_dsa.pub`, а приватный ключ в `~/.ssh/id_dsa`.

Публичный ключ DSA также должен быть помещен в каталог `~/.ssh/authorized_keys` на удаленном компьютере.

Утилиты `ssh-agent(1)` и `ssh-add(1)` используются для управления множеством защищенных паролем приватных ключей.

**Внимание:** Параметры и имена файлов могут различаться для разных версий **OpenSSH**, установленных в системе, для решения проблем обратитесь к странице справочника `ssh-keygen(1)`.

### 14.12.7. SSH тунелирование

**OpenSSH** поддерживает возможность создания туннеля для пропуска соединения по другому протоколу через защищенную сессию.

Следующая команда указывает `ssh(1)` создать туннель для **telnet**:

```
% ssh -2 -N -f -L 5023:localhost:23 user@foo.example.com
%
```

Команда `ssh` используется со следующими параметрами:

`-2`

Указывает `ssh` использовать версию 2 протокола (не используйте этот параметр, если работаете со старыми SSH серверами).

`-N`

Означает использование в не-командном режиме, только для тунелирования. Если этот параметр опущен, `ssh` запустит обычную сессию.

`-f`

Указывает `ssh` запускаться в фоновом режиме.

`-L`

Означает локальный туннель в стиле `localport:remotehost:remoteport`.

`user@foo.example.com`

Удаленный сервер SSH.

Туннель SSH создается путем создания прослушивающего сокета на определенном порту `localhost`. Затем все принятые на локальном хосту/порту соединения переправляются на через SSH на определенный удаленный хост и порт.

В этом примере, порт `5023` на `localhost` перенаправляется на порт `23` на `localhost` удаленного компьютера. Поскольку `23` это порт **telnet**, будет создано защищенное соединение **telnet** через туннель SSH.

Этот метод можно использовать для любого числа небезопасных протоколов, таких как SMTP, POP3, FTP, и так далее.

#### Пример 14-1. Использование SSH для создания защищенного туннеля на SMTP

```
% ssh -2 -N -f -L 5025:localhost:25 user@mailserver.example.com
user@mailserver.example.com's password: *****
% telnet localhost 5025
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 mailserver.example.com ESMTP
```

Этот метод можно использовать вместе с ssh-keygen(1) и дополнительными пользовательскими учетными записями для создания более удобного автоматического SSH туннелирования. Ключи могут быть использованы вместо паролей, и туннели могут запускаться от отдельных пользователей.

### 14.12.7.1. Практические примеры SSH туннелирования

#### 14.12.7.1.1. Защищенный доступ к серверу POP3

На работе находится SSH сервер, принимающий соединения снаружи. В этой же офисной сети находится почтовый сервер, поддерживающий протокол POP3. Сеть или сетевое соединение между вашим домом и офисом могут быть или не быть полностью доверяемыми. По этой причине вам потребуется проверять почту через защищенное соединение. Решение состоит в создании SSH соединения к офисному серверу SSH и туннелирование через него к почтовому серверу.

```
% ssh -2 -N -f -L 2110:mail.example.com:110 user@ssh-server.example.com
user@ssh-server.example.com's password: *****
```

Когда туннель включен и работает, вы можете настроить почтовый клиент для отправки запросов POP3 на localhost, порт 2110. Соединение будет безопасно переправлено через туннель на mail.example.com.

#### 14.12.7.1.2. Прохождение через Драконовский Брандмауэр

Некоторые сетевые администраторы устанавливают на межсетевых экранах (брандмауэрах) драконовские правила, фильтруя не только входящие соединения, но и исходящие. Вам может быть разрешен доступ к удаленным компьютерам только по портам 22 и 80, для SSH и просмотра сайтов.

Вам может потребоваться доступ к другому (возможно, не относящемуся к работе) сервису, такому как Ogg Vorbis для прослушивания музыки. Если этот сервер Ogg Vorbis выдает поток не с портов 22 или 80, вы не сможете получить к нему доступ.

Решение состоит в создании SSH соединения с компьютером вне межсетевого экрана и использование его для туннелирования сервера Ogg Vorbis.

```
% ssh -2 -N -f -L 8888:music.example.com:8000 user@unfirewalled-system.example.org
user@unfirewalled-system.example.org's password: *****
```



Клиентскую программу теперь можно настроить на localhost порт 8888, который будет перенаправлен на music.example.com порт 8000, успешно обходя межсетевой экран.

### 14.12.8. Для дальнейшего чтения

OpenSSH (<http://www.openssh.com/>)

ssh(1) scp(1) ssh-keygen(1) ssh-agent(1) ssh-add(1)

sshd(8) sftp-server(8)

## 14.13. Списки контроля доступа файловой системы (ACL)

*Предоставил Tom Rhodes.*

В дополнение к другим расширениям файловой системы, таким как снимки (snapshots), FreeBSD 5.0 и более поздние версии системы предлагают защиту с помощью списков контроля доступа файловой системы (File System Access Control Lists, ACLs).

Списки контроля доступа расширяют стандартную модель прав UNIX высоко совместимым (POSIX.1e) способом. Эта возможность позволяет администратору получить преимущество от использования более интеллектуальной модели безопасности.

Для включения поддержки ACL в файловой системе UFS, следующая строка:

```
options UFS_ACL
```

должна быть добавлена в файл настройки ядра. Если параметр не добавлен, при попытке монтирования систем, поддерживающих ACL, появится предупреждающее сообщение. Этот параметр включен в ядро GENERIC. ACL основывается на дополнительных атрибутах, встроенных в файловую систему. Дополнительные атрибуты поддерживаются по умолчанию следующим поколением файловых систем UNIX, UFS2.

**Замечание:** Для включения дополнительных атрибутов в UFS1 требуется больше усилий по сравнению с UFS2. Производительность дополнительных атрибутов в UFS2 также существенно выше. По этим причинам для работы с списками контроля доступа предпочтительно использование UFS2

ACL включаются во время монтирования флагом `acls`, который добавляется к `/etc/fstab`. Этот флаг также можно сделать постоянным с помощью `tunefs(8)`, изменив флаг `ACL` в заголовке файловой системы. Вообще говоря, использование флага в суперблоке предпочтительно по нескольким причинам:

- Постоянный ACL флаг не может быть изменен путем перемонтирования системы (`mount(8) -u`), а только через `umount(8)` и `mount(8)`. Это означает, что ACL нельзя включить на корневой файловой системе после загрузки. Это также означает, что вы не можете изменить флаг на используемой файловой системе.

- Установка флага в суперблоке приводит к постоянному монтированию файловой системы с включенным ACL, даже если нет записи в `fstab` или при смене порядка устройств. Это предотвращает случайное монтирование файловой системы без ACL, которое может повлечь за собой проблемы с безопасностью.

**Замечание:** Мы можем изменить поведение ACL для включения флага без полного перемонтирования, но считаем, что желательно исключить случайное монтирование без ACL, поскольку вы можете попасть в неприятную ситуацию, если включите ACL, затем выключите их, затем опять включите без сброса расширенных атрибутов. Обычно, как только вы включили ACL в файловой системе, они не должны быть выключены, поскольку получающаяся защита файлов может быть не совместима с той, что применяется пользователями системы, и повторное включение ACL может подключить предыдущие списки контроля доступа к файлам, права на которые изменены, что приведет к непредсказуемому поведению.

Файловые системы с включенными ACLs показывают знак + при просмотре прав на файлы. Например:

```
drwx----- 2 robert  robert  512 Dec 27 11:54 private
drwxrwx---+ 2 robert  robert  512 Dec 23 10:57 directory1
drwxrwx---+ 2 robert  robert  512 Dec 22 10:20 directory2
drwxrwx---+ 2 robert  robert  512 Dec 27 11:57 directory3
drwxr-xr-x  2 robert  robert  512 Nov 10 11:54 public_html
```

Здесь мы видим, что каталоги `directory1`, `directory2`, и `directory3` используют преимущества ACL. Каталог `public_html` их не использует.

### 14.13.1. Использование ACL

ACL файловой системы можно просмотреть с помощью утилиты `getfacl(1)`. Например, для просмотра настроек ACL файла `test`, может использоваться команда:

```
% getfacl test
#file:test
#owner:1001
#group:1001
user::rw-
group::r--
other::r--
```

Для изменения ACL этого файла, вызовите утилиту `setfacl(1)`. Выполните:

```
% setfacl -k test
```

Параметр `-k` удалит все установленные на данный момент ACL из файла или файловой системы. Более предпочтительный метод это использование параметра `-b`, который оставит необходимые для работы ACL поля.

```
% setfacl -m u:trhodes:rw,group:web:r--,o:--- test
```

В вышеприведенной команде параметр `-m` использован для изменения записей ACL по умолчанию. Поскольку предустановленных записей не было (они были удалены предыдущей командой), эта команда восстановит параметры по умолчанию и задаст приведенные параметры. Имейте ввиду, при

добавлении пользователя или группы, которых нет в системе, на stdout будет выведена ошибка Invalid argument.

## 14.14. Сообщения безопасности FreeBSD

*Предоставил Tom Rhodes.*

Как многие и высококачественные операционные системы, FreeBSD публикует "Сообщения безопасности" ("Security Advisories"). Эти сообщения обычно отправляются по почте в списки рассылки, посвященные безопасности и публикуются в списке проблем только после выхода исправлений к соответствующим релизам. В этом разделе разъясняется, что такое сообщения безопасности, как их читать и какие меры принимать для исправления системы.

### 14.14.1. Как выглядит сообщение?

Сообщение безопасности FreeBSD выглядит подобно сообщению ниже, взятому из списка рассылки freebsd-security-notifications (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-security-notifications>).

```
=====
FreeBSD-SA-XX:XX.UTIL                                Security Advisory
                                                    The FreeBSD Project
```

```
Topic:                denial of service due to some problem❶

Category:             core❷
Module:               sys❸
Announced:           2003-09-23❹
Credits:              Person@EMAIL-ADDRESS❺
Affects:              All releases of FreeBSD❻
                      FreeBSD 4-STABLE prior to the correction date
Corrected:            2003-09-23 16:42:59 UTC (RELENG_4, 4.9-PRERELEASE)
                      2003-09-23 20:08:42 UTC (RELENG_5_1, 5.1-RELEASE-p6)
                      2003-09-23 20:07:06 UTC (RELENG_5_0, 5.0-RELEASE-p15)
                      2003-09-23 16:44:58 UTC (RELENG_4_8, 4.8-RELEASE-p8)
                      2003-09-23 16:47:34 UTC (RELENG_4_7, 4.7-RELEASE-p18)
                      2003-09-23 16:49:46 UTC (RELENG_4_6, 4.6-RELEASE-p21)
                      2003-09-23 16:51:24 UTC (RELENG_4_5, 4.5-RELEASE-p33)
                      2003-09-23 16:52:45 UTC (RELENG_4_4, 4.4-RELEASE-p43)
                      2003-09-23 16:54:39 UTC (RELENG_4_3, 4.3-RELEASE-p39)❷
FreeBSD only:         NO❸
```

For general information regarding FreeBSD Security Advisories, including descriptions of the fields above, security branches, and the following sections, please visit <http://www.FreeBSD.org/security/>.

I. Background❹

II. Problem Description(10)

## III. Impact(11)

## IV. Workaround(12)

## V. Solution(13)

## VI. Correction details(14)

## VII. References(15)

- ❶ Поле `Topic` показывает в чем именно заключается проблема. Это обычно введение в сообщение безопасности, упоминающее утилиту, в которой возникла ошибка.
- ❷ Поле `Category` относится к затронутой части системы и может быть выбрана из `core`, `contrib`, или `ports`. Категория `core` означает, что уязвимость затрагивает основной компонент операционной системы FreeBSD. Категория `contrib` означает, что уязвимость затрагивает программы, предоставленные проекту FreeBSD, например **sendmail**. Наконец, категория `ports` означает, что уязвимость затрагивает программное обеспечение, доступное из коллекции портов.
- ❸ Поле `Module` указывает на местоположение компонента, например `sys`. В этом примере мы видим, что затронут модуль `sys`, следовательно, эта уязвимость относится к компоненту, используемому в ядре.
- ❹ Поле `Announced` отражает дату публикации сообщения безопасности, или его анонсирования. Это означает, что команда обеспечения безопасности убедилась, что проблема существует и что патч помещен в репозиторий исходных текстов FreeBSD.
- ❺ Поле `Credits` упоминает частное лицо или организацию, обнаружившую уязвимость и сообщившую о ней.
- ❻ Поле `Affects` дает информацию о релизах FreeBSD, к которым относится данная уязвимость. Для базовой системы, просмотр вывода команды `ident` для файлов, затронутых уязвимостью, поможет определить ревизию. Номер версии портов приведен после имени порта в каталоге `/var/db/pkg`. Если система не синхронизируется с CVS репозиторием FreeBSD и не пересобирается ежедневно, высок шанс, что она затронута уязвимостью.
- ❼ Поле `Corrected` показывает дату, время, смещение во времени и релиз, в котором исправлена ошибка.
- ❽ Поле `FreeBSD only` показывает, существует ли эта уязвимость только в FreeBSD, или затрагивает и другие системы.
- ❾ Поле `Background` дает информацию именно о той утилите, для которой выпущено сообщение. Как правило информация о том, зачем утилита присутствует в FreeBSD, для чего она используется, и немного информации о том, как появилась эта утилита.

- (10) Поле `Problem Description` дает более глубокие разъяснения возникшей проблемы. Оно может включать информацию об ошибочном коде, или даже о том, как утилита может быть использована для создания брешы в системе безопасности.
- (11) Поле `Impact` описывает тип воздействия, который проблема может оказать на систему. Это может быть все, что угодно, от атаки на отказ в обслуживании до получения пользователями дополнительных привилегий, или даже получения атакующим прав суперпользователя.
- (12) Поле `Workaround` предлагает тем, системным администраторам, которые не могут обновить систему, обходной путь решения проблемы. Он может пригодиться при недостатке времени, отсутствии подключения к сети или по массе других причин. В любом случае, к безопасности нельзя относиться несерьезно, и необходимо либо применить указанный обходной путь, либо исправить систему.
- (13) Поле `Solution` предлагает инструкции по исправлению затронутой системы. Это пошаговое руководство, протестированный метод восстановления безопасности системы.
- (14) Поле `Correction Details` показывает ветвь CVS (имя релиза с точками, замененными на символы подчеркивания). Здесь также показан номер ревизии каждого файла из каждой ветви.
- (15) Поле `References` обычно упоминает другие источники информации. Это могут быть веб-страницы, книги, списки рассылки и группы новостей.

## Примечания

1. В FreeBSD стандартный пароль может быть до 128 символов длиной.

# Глава 15. Принудительный контроль доступа (MAC)

*Написал Tom Rhodes. Перевод на русский язык: Денис Пеплин.*

## 15.1. Краткий обзор

FreeBSD 5.X представляет новые расширения системы безопасности от проекта TrustedBSD, основанные на документах POSIX.1e. Два из наиболее важных нововведений в механизмах безопасности это списки контроля доступа файловой системы (Access Control Lists, ACLs) и принудительный контроль доступа Mandatory Access Control, MAC). Инфраструктура позволяет загружать новые модули контроля доступа, реализуя новые политики безопасности. Некоторые из них предоставляют защиту ключевых подсистем, защищая определенный сервис, в то время как другие предоставляют исчерпывающую систему безопасности с метками на всех субъектах и объектах. Контроль называется принудительным, поскольку применение контроля производится администраторами и системой, и не зависит от решения пользователей, как это происходит при обычном контроле доступа (Discretionary Access Control, DAC, стандартные файловые и System V IPC права в FreeBSD).

Вся эта глава фокусируется на инфраструктуре принудительного контроля доступа и настройке подключаемых модулей, реализующих различные политики безопасности.

После прочтения этой главы вы узнаете:

- Какие модули MAC включены в настоящее время в FreeBSD, какие политики с ними связаны.
- Что способны реализовать политики MAC, различие между политиками с метками (label) и без меток.
- Как эффективно настроить систему для использования инфраструктуры MAC.
- Как настроить различные политики, используемые модулями MAC.
- Как реализовать более защищенную среду, используя инфраструктуру MAC и приведенные примеры.
- Как протестировать настройку MAC, чтобы убедиться, что инфраструктура была реализована правильно.

Перед прочтением этой главы вам потребуется:

- Понимание основ UNIX и FreeBSD (Гл. 3).
- Ознакомиться с основами настройки/компилирования ядра (Гл. 8).
- Иметь некоторые понятия о безопасности и как она относится к FreeBSD (Гл. 14).

**Внимание:** Неправильное использование информации этой главы может вызвать потерю доступа к системе, проблемы у пользователей, или невозможность запуска XFree86. Что более важно, MAC не должен восприниматься как полная защита системы. Инфраструктура MAC лишь усиливает имеющуюся систему безопасности: без применения методов защиты и регулярных проверок, система никогда не станет полностью защищенной.

Необходимо также учесть, что примеры, приведенные в этой главе, это всего лишь примеры. Не рекомендуется дублирование этих настроек для реальных задач. Реализация политик безопасности требует вдумчивого планирования. Тот, кто не понял полностью как все это на самом деле работает, может столкнуться с необходимостью повторной полной перенастройки системы безопасности для многих файлов и каталогов.

### 15.1.1. Что не будет затронуто

В этой главе охвачен широкий спектр вопросов безопасности, относящихся к инфраструктуре MAC. Однако разработка политик MAC не будет затронута. Несколько модулей, включенных в инфраструктуру MAC, имеют особые характеристики, которые предназначены для тестирования и разработки новых модулей. Это относится к модулям/политикам `mac_test(4)`, `mac_stub(4)` и `mac_none(4)`. За дальнейшей информацией по этим модулям и различным предоставляемым ими механизмам, обратитесь к соответствующим страницам справочника.

## 15.2. Ключевые термины этой главы

Перед чтением этой главы необходимо ознакомиться с некоторыми ключевыми терминами. Это возможно разрешит возникающие вопросы и предотвратит перегрузку материала новыми терминами и информацией.

- *отдел* (compartment): Отдел это набор программ и данных, которые необходимо отделить, и где пользователи получают явный доступ к отдельным компонентам системы. Отдел представляет группирование, например рабочую группу, департамент, проект или тему. С использованием отделов возможна реализация политики с явно определенным доступом.
- *целостность* (integrity): Целостность, как ключевая концепция, это уровень доверия, который может быть присвоен данным. Поскольку целостность данных повышается, это дает возможность доверять данным.
- *метка* (label): Метка является инструментом безопасности, она может быть применена к файлам, каталогам и другим сущностям системы. Ее можно представить как штамп конфиденциальности; метка, помещенная на файл, описывает уровень секретности данного файла и разрешит доступ только файлам, пользователям, ресурсам и т.д. с теми же или меньшими установками безопасности. Некоторые из политик могут обрабатывать метки различными способами; это будет обсуждаться в разделе политик ниже.
- *multilabel* (множественные метки): свойство `multilabel` это параметр файловой системы, который может быть установлен в однопользовательском режиме с помощью утилиты `tunefs(8)`, во время загрузки через файл `fstab(5)`, или при создании новой файловой системы. Этот параметр позволяет администратору помещать различные метки MAC на различные объекты. разрешает помещение множественных MAC меток на файлы и каталоги файловой системы. Этот параметр применим только к политикам с метками.
- *объект* (object): Объект или системный объект это сущность, через которую информация проходит к субъекту. Это могут быть каталоги, файлы, поля, экраны, клавиатуры, память, магнитные

накопители, принтеры или любые другие устройства хранения/перемещения данных. В сущности это контейнер данных или ресурс системы; доступ к *объекту* фактически означает доступ к данным.

- *политика* (policy): Набор правил, определяющих как достичь объекта. *Политика* обычно документирует обращение с определенными элементами. В этой главе *политика* будет означать *политику безопасности*; т.е. коллекцию правил, которые будут контролировать поток данных и определять кто будет иметь доступ к этим данным.
- *чувствительность* (sensitivity): Обычно используется при обсуждении MLS. Уровень чувствительности это термин, используемый для описания того, насколько важны или секретны данные. Увеличение уровня чувствительности означает важность данных.
- *одиночная метка* (single label): означает, что вся файловая система использует одну метку для определения доступа всего потока данных. Когда файловая система использует эту установку, что происходит всегда, если не установлен параметр multilabel, ко всем файлам будет применяться одна и та же установка метки.
- *субъект* (subject): субъект это любая активная сущность, вызывающая перемещение информации между *объектами*; т.е. пользователь, пользовательский обработчик, системный процесс и т.д. В FreeBSD это почти всегда поток, работающий в процессе или представляющий пользователя.

### 15.3. Описание MAC

Усвоив все эти термины, рассмотрим как MAC повышает безопасность системы в целом. Различные политики, предоставляемые инфраструктурой MAC, могут быть использованы для защиты сети и файловых систем, блокирования доступа пользователей к определенным портам и сокетам, и так далее. Возможно, наилучшее использование политик это сочетание их вместе путем загрузки нескольких модулей одновременно, для создания многослойной защищенной среды. В многослойной среде безопасности несколько политик обеспечивают контролируемость защиты. Это отличается от усиления защиты, когда обычно усиливаются элементы системы, используемой в определенных целях. Единственным недостатком является дополнительная административная нагрузка в случае множественных меток файловой системы, установки сетевого доступа по пользователям, и т.д.

Эти недостатки минимальны по сравнению с длительным эффектом функционирования инфраструктуры; например, возможность выбора политик, необходимых для определенных конфигураций, уменьшает потерю производительности. Возможность удаления поддержки не используемых политик может увеличить общую производительность системы, а также дает гибкость выбора. Хорошая реализация удовлетворит общие требования безопасности и будет эффективно использовать различные политики, предоставляемые инфраструктурой.

Система, использующая возможности MAC, должна как минимум гарантировать, что пользователю не разрешается самостоятельно изменять атрибуты безопасности; все утилиты пользователя, программы и скрипты должны работать с ограничениями доступа, налагаемыми выбранной политикой; весь контроль правил доступа MAC находится в руках системного администратора.

Право выбора правильных политик безопасности принадлежит только системному администратору. В некоторых случаях может потребоваться ограничение доступа через сеть; для этого могут пригодиться `mac_portacl(4)`, `mac_ifoff(4)` и даже `mac_biba(4)`. В других случаях может быть необходима строгая конфиденциальность объектов в файловой системе. Для этого существуют политики `mac_bsdextended(4)` и `mac_mls(4)`.



Выбор политики может быть сделан на основе конфигурации сети. Возможно только определенным пользователям можно разрешить доступ через ssh(1) к сети или интернет. В таких ситуациях подойдет политика mac\_portacl(4). Но что необходимо сделать для файловых систем? Должен ли доступ к определенным каталогам быть запрещен для других групп или определенных пользователей? Или мы должны ограничить доступ пользователей или утилит к определенным файлам путем классификации определенных объектов?

В случае файловой системы, доступ может считаться конфиденциальным для отдельных пользователей, но не для всех. Например, большая команда разработчиков может быть разбита на небольшие группы. Разработчикам проекта А может быть не разрешен доступ к объектам, написанным разработчиками из проекта В. Хотя им может понадобиться доступ к объектам, созданным разработчиками проекта С; это реально встречающаяся ситуация. С помощью различных политик, предоставляемых инфраструктурой MAC, пользователи могут быть разделены на эти три группы и затем получить доступ к соответствующим областям без опасности утечки информации.

Таким образом, каждая политика имеет уникальный способ взаимодействия с общей безопасностью системы. Выбор политики должен быть основан на хорошо продуманной политике безопасности. Во многих случаях политика должна быть полностью пересмотрена и реализована заново для всей системы. Понимание различных политик, предоставляемых инфраструктурой MAC, поможет администраторам выбрать лучшую политику в своей ситуации.

Стандартное ядро FreeBSD не включает параметр MAC; необходимо добавить следующий параметр ядра перед тем, как пробовать какие-либо из примеров или применять информацию этой главы:

```
options MAC
```

Затем необходимо пересобрать и переустановить ядро.

**Предостережение:** Хотя различные страницы справочника для модулей MAC сообщают, что они могут быть встроены в ядро, возможна блокировка доступа системы к сети и другие побочные эффекты. Включение MAC очень похоже на включение брандмауэра, но необходимо быть внимательным, чтобы полностью не заблокировать систему. Необходимо предусмотреть возможность возврата к предыдущей конфигурации, а реализация MAC удаленно должна производиться с особой осторожностью.

## 15.4. Метки MAC

Метка MAC это атрибут безопасности, который может быть применен к субъектам и объектам всей системы.

При установке метки пользователь должен в точности понимать, что именно она делает. Атрибуты, доступные для объекта, зависят от загруженной политики, а политики интерпретируют свои атрибуты совершенно различным образом. Результатом недостаточного понимания настроек может стать их неправильная реализация, что может привести к неожиданному, и возможно нежелательному поведению системы.

Метка безопасности на объекте используется политикой для определения правил доступа. Для некоторых политик метка сама по себе содержит всю необходимую для этого информацию, в других моделях метки могут обрабатываться как часть большого набора правил, и т.д.

Например, установка метки в `biba/low` на файле присвоит этому файлу метку, обрабатываемую политикой Biba со значением `low`.

Несколько политик, поддерживающих метки в FreeBSD, предоставляют три определенные предустановленные метки. Это низкая, высокая и равная метки. Хотя они устанавливают контроль различными способами для каждой политики, вы можете быть уверены, что низкая метка задаст минимальные установки, равная метка означает отмену или недействительность, а высокая метка установит максимально возможные настройки в политиках Biba и MLS.

При применении в файловой системе одиночной метки, только одна метка может быть использована для объектов. Это вызовет установку одних и тех же прав доступа для всей системы, и во многих случаях это все, что необходимо. Тем не менее, существует несколько ситуаций, в которых на объекты и субъекты файловой системы могут быть установлены множественные метки. В этих ситуациях необходимо с помощью `tunefs(8)` установить параметр `multilabel`.

В случае Biba и MLS может быть установлена числовая метка для указания точного уровня иерархического контроля. Этот числовой уровень используется для разделения или сортировки информации по различным группам классификации, разрешающей доступ только к этой группе или группе с более высоким уровнем.

В большинстве случаев системный администратор использует только одну метку на всю файловую систему.

*Постойте, но это же похоже на DAC! Я думал, что MAC дает контроль только администратору.* Это утверждение все еще верно, только `root` контролирует и настраивает политики, так что пользователи помещаются в соответствующие категории/уровни доступа. Многие политики могут ограничить также и пользователя `root`. Базовый контроль над объектами затем передается группе, но пользователь `root` может отменить или изменить эти настройки в любое время. Данная иерархическая модель соответствует таким политикам как Biba и MLS.

### 15.4.1. Настройка меток

Практически все действия по настройке политики с метками могут быть выполнены с использованием утилит базовой системы. Эти команды обеспечивают простой интерфейс для настройки объектов или субъектов, или для изменения и проверки настроек.

Все настройки могут быть выполнены с использованием утилит `setfmac(8)` и `setpmac(8)`. Команда `setfmac` используется для установки меток MAC на системные объекты, а команда `setpmac` используется для установки меток на системные субъекты. Выполните:

```
# setfmac biba/high test
```

Если не произойдет ошибок, будет возвращено приглашение командной строки, как и после команд `chmod(1)` и `chown(8)`. В некоторых случаях может появиться ошибка `Permission denied`, и она обычно появляется при установке или изменении метки на объект с ограничениями.<sup>1</sup> Системный администратор для обхода этой проблемы может использовать следующие команды:

```
# setfmac biba/high test
Permission denied
# setpmac biba/low setfmac biba/high test
# getfmac test
test: biba/high
```

Как видно из примера выше, команда `setpmac` может быть использована для изменения установок политики путем присвоения иной метки вызывающему процессу. Утилита `getpmac` обычно используется с существующим на данный момент процессом, таким как **sendmail**, хотя она принимает PID вместо команды, ее действие аналогично. Если пользователи попытаются манипулировать файлами, к которым у них нет доступа в соответствии с правилами загруженных политик, функцией `mac_set_link` будет выдано сообщение об ошибке `Operation not permitted`.

#### 15.4.1.1. Пользователи и установки меток

Пользователям необходимо иметь метки, чтобы их файлы и процессы могли правильно взаимодействовать с определенной в системе политикой безопасности. Это настраивается через файл `login.conf` путем использования классов. Каждая политика, использующая метки, реализует установку класса пользователя.

Пример записи, содержащей все политики, приведенные ниже:

```
default:\
:copyright=/etc/COPYRIGHT:\
:welcome=/etc/motd:\
:setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
:path=~:/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:\
:manpath=/usr/share/man /usr/local/man:\
:nologin=/usr/sbin/nologin:\
:cputime=1h30m:\
:datasize=8M:\
:vmemoryuse=100M:\
:stacksize=2M:\
:memorylocked=4M:\
:memoryuse=8M:\
:filesize=8M:\
:coredumpsize=8M:\
:openfiles=24:\
:maxproc=32:\
:priority=0:\
:requirehome:\
:passwordtime=91d:\
:umask=022:\
:ignoretime@:\
:label=partition/13,mls/5,biba/10(5-15),lomac10[2]:
```

Параметр `label` используется для установки метки MAC по умолчанию для класса пользователя. Пользователи не смогут изменять это значение, поэтому его можно признать не опциональным. В реальной ситуации администратору никогда не потребуется включать каждую политику. Рекомендуется прочесть главу полностью перед реализацией любой из этих настроек.

**Замечание:** Пользователи могут изменить свою метку после входа; однако политика накладывает ограничение на это изменение. В примере выше политике Biba указано, что минимальная целостность процесса 5, максимальная 15, а эффективная целостность по умолчанию 10. Процесс будет работать на уровне 10, пока метка не будет изменена, например если пользователь использует команду `setpmac`, которую Biba ограничит диапазоном, установленным при входе.

Во всех случаях после изменения `login.conf`, база данных `login class capability` должна быть пересобрана с использованием команды `cap_mkdb` и это будет отражено в каждом последующем примере главы.

Полезно отметить, что количество пользователей, которым требуются различные классы, во многих сетях может быть велико. Необходимо тщательное планирование, поскольку управление такой сетью может серьезно усложниться.

В будущих версиях FreeBSD появится новый способ связывания пользователей с метками; однако, он будет доступен только через некоторое время после выхода FreeBSD 5.3.

#### 15.4.1.2. Сетевые интерфейсы и установка меток

Метки также могут быть установлены на сетевые интерфейсы, для контроля потока данных в сети. Во всех случаях они функционируют аналогично тому, как политики по отношению к объектам. Пользователи с высокими установками, например, `biba`, не смогут получить доступ к сетевым интерфейсам с низкими установками.

Для установки MAC меток на сетевых интерфейсах параметр `maclabel` может быть передан `ifconfig`. Например:

```
# ifconfig bge0 maclabel biba/equal
```

установит MAC метку `biba/equal` на интерфейс `bge(4)`. При использовании метки, подобной `biba/high(low-high)` вся метка должна быть взята в кавычки, иначе будет выдано сообщение об ошибке.

Каждая политика, использующая метки, снабжена переменной `sysctl`, которая может быть использована для отключения MAC меток на сетевых интерфейсах. Установка метки в `equal` будет иметь подобный эффект. Просмотрите вывод команды `sysctl`, страницы справочника для политик, или дальнейшую информацию из этой главы по этим переменным.

#### 15.4.2. Одиночные или множественные метки?

По умолчанию система будет использовать параметр `singlelabel`. Но что это означает для администратора? Существуют несколько различий между политиками, каждая из которых правильна сама по себе, но имеет свои доводы за и против относительно гибкости модели безопасности системы.

`singlelabel` (одиночная метка) разрешает использование только одной метки, например `biba/high`, для каждого объекта или субъекта. Ее преимущество в меньшей нагрузке на системного администратора, а недостаток в малой гибкости политик, поддерживающих метки. Многие администраторы в своих политиках безопасности могут предпочесть использование параметра `multilabel`.

С параметром `multilabel` каждый субъект или объект может иметь собственную метку MAC, в то время как со стандартным параметром `singlelabel` возможна только одна метка на весь раздел. Параметры `multilabel` и `singlelabel` требуются только для политик, реализующих метки, включая `Biba`, `Lomac`, `MLS` и `SEBSD`.

Во многих случаях `multilabel` может вообще не потребоваться. Предположим следующую ситуацию и модель безопасности:

- FreeBSD веб-сервер, использующий инфраструктуру MAC и набор различных политик.
- Этому компьютеру потребуется лишь одна метка, `biba/high`, для всей системы. Файловой системе не нужен параметр `multilabel`, поскольку по умолчанию работает одиночная метка.
- Но поскольку этот компьютер будет веб сервером, процесс веб сервера должен быть запущен с `biba/low` для предотвращения записи. Политика `Biba` и то, как она работает, будет обсуждаться позже, поэтому предыдущий комментарий сложно интерпретировать; просто продолжайте чтение. Сервер может использовать дисковый раздел с установленной меткой `biba/low` для большинства, если не для всех своих операций. В этом примере отсутствуют многие детали, такие как ограничения на данные, конфигурация системы и установки пользователей; однако, это лишь предварительный пример.

Если используется любая из политик, не поддерживающих метки, параметр `multilabel` не требуется. Сюда включаются политики `seeotheruids`, `portacl` и `partition`.

Необходимо также отметить, что использование параметра `multilabel` на разделе и установление модели безопасности, основанной на функциональности `multilabel`, может повлечь за собой множество дополнительной административной работы, поскольку всему в файловой системе должны быть присвоены метки. Это включает каталоги, файлы, и даже файлы устройств.

Следующая команда установит параметр `multilabel` на файловых системах. Это может быть сделано только в однопользовательском режиме:

```
# tuneefs -l enable /
```

Это не требуется для файловой системы подкачки.

**Замечание:** Некоторые пользователи сталкиваются с проблемами при установке флага `multilabel` на корневой раздел. В данном случае обратитесь к Разд. 15.16.

### 15.4.3. Настройка MAC переменными `sysctl`

Независимо от загрузки модулей, существует несколько частей MAC, которые могут быть настроены с использованием интерфейса `sysctl`. Эти переменные описаны ниже и во всех случаях значение 1 означает включение, а 0 — отключение:

- `security.mac.enforce_fs` по умолчанию установлена в 1 и включает политики MAC на файловых системах.
- `security.mac.enforce_kld` по умолчанию 1 и включает линкование политик MAC в ядре (см. `kld(4)`).
- `security.mac.enforce_network` по умолчанию 1 и включает сетевые политики MAC.
- `security.mac.enforce_pipe` по умолчанию 1 и включает политики MAC для каналов (`pipe`).
- `security.mac.enforce_process` по умолчанию 1 и включает политики MAC для процессов, использующих средства межпроцессного взаимодействия.
- `security.mac.enforce_socket` по умолчанию 1 и включает политики MAC на сокетах (см. страницу справочника `socket(2)`).

- `security.mac.enforce_system` по умолчанию 1 и включает политики MAC для действий системы, таких как учет (accounting) и перезагрузка.
- `security.mac.enforce_vm` по умолчанию 1 и включает политики MAC для системы виртуальной памяти.

**Замечание:** Каждая политика MAC поддерживает переменные `sysctl`. Они обычно попадают в дерево `security.mac.<polycname>`. Для просмотра всех переменных MAC, используйте следующую команду:

```
# sysctl -da | grep mac
```

Это должно быть интерпретировано так, что все основные политики MAC включены по умолчанию. Если модули встроены в ядро, система будет заблокирована, и скорее всего не сможет связаться с локальной сетью или с интернет, и т.д. Поэтому встраивание модулей в ядро не рекомендуется. Не потому, что это ограничит возможность отключения командой `sysctl`, а потому, что включение политик в виде модулей позволит администратору переключать политики системы без необходимости пересборки и переустановки новой системы.

## 15.5. Настройка модулей

Каждый модуль, включенный в инфраструктуру MAC, может быть или встроен в ядро, как упоминалось выше, или загружен в виде модуля ядра. Рекомендуется добавление имени модуля в файл `/boot/loader.conf`, этот модуль будет активирован в самом начале загрузки.

В последующих разделах будут обсуждаться различные модули MAC и их возможности. Реализация этих возможностей в определенных ситуациях также будет обсуждаться в этой главе. Некоторые модули поддерживают использование меток, которые контролируют доступ путем применения правил вида "это разрешено, а это нет". Настройка меток может контролировать доступ к файлам, сетевым коммуникациям и т.д. В предыдущем разделе было показано как флаг `multilabel` может быть установлен на файловые системы для включения контроля доступа по файлам или по разделам.

Конфигурация с одной меткой не допускает применение нескольких меток в системе, поэтому параметр `tunefs` называется `multilabel`.

### 15.5.1. Модуль MAC `seeotheruids`

Имя модуля: `mac_seeotheruids.ko`

Строка в конфигурации ядра: `options MAC_SEEOTHERUIDS`

Параметр загрузки: `mac_seeotheruids_load="YES"`

Модуль `mac_seeotheruids(4)` копирует и расширяет переменные `sysctl security.bsd.see_other_uids` и `security.bsd.see_other_gids`. Он не требует установки меток и может прозрачно работать с другими модулями.

После загрузки модуля, для управления им могут быть использованы следующие переменные `sysctl`:

- `security.mac.seeotheruids.enabled` включит модуль с настройками по умолчанию. Эти настройки запрещают пользователям просмотр процессов и сокетов, принадлежащих другим пользователям.
- `security.mac.seeotheruids.specificgid_enabled` позволит исключить определенные группы из этой политики. Для исключения определенной группы, используйте переменную `sysctl security.mac.seeotheruids.specificgid=XXX`. В примере выше необходимо заменить XXX на числовой ID группы.
- `security.mac.seeotheruids.primarygroup_enabled` используется для исключения определенной основной группы из этой политики. При использовании этой переменной `security.mac.seeotheruids.specificgid_enabled` может быть не установлена.

Необходимо отметить, что пользователь `root` не является исключением из этой политики. Это одно из самых существенных различий между MAC версией и обычными переменными, существующими по умолчанию: `security.bsd.seeotheruids`.

## 15.6. Модуль MAC `bsdextended`

Имя модуля: `mac_bsdextended.ko`

Строка конфигурации ядра: `options MAC_BSDEXTENDED`

Параметр загрузки: `mac_bsdextended_load="YES"`

Модуль `mac_bsdextended(4)` включает брандмауэр файловой системы. Политика этого модуля является расширением стандартной модели разрешений файловой системы, позволяя администратору создавать набор правил для защиты файлов, утилит и каталогов иерархии файловой системы в стиле брандмауэра.

Политика может быть создана с помощью утилиты, `ugidfw(8)`, синтаксис которой похож на синтаксис `ipfw(8)`. Другие инструменты могут быть написаны с использованием функций библиотеки `libugidfw(3)`.

При работе с этим модулем необходимо соблюдать особую осторожность; некорректное его использование может заблокировать доступ к отдельным частям файловой системы.

### 15.6.1. Примеры

После загрузки модуля `mac_bsdextended(4)` для просмотра текущей настройки правил может быть использована следующая команда:

```
# ugidfw list
0 slots, 0 rules
```

Как и можно было ожидать, правила не определены. Это означает, что доступ полностью открыт. Для создания правила, которое блокирует доступ всех пользователей, но не повлияет на `root`, просто запустите следующую команду:

```
# ugidfw add subject not uid root new object not uid root mode n
```

**Замечание:** В релизах FreeBSD до 5.3, параметр `add` не существует. Вместо него необходимо использовать `set`. Пример дан ниже.

Это очень плохая идея, поскольку такое правило запретит пользователям использовать даже самые простые команды, такие как `ls`. Более патристический список правил может быть таким:

```
# ugidfw set 2 subject uid user1 object uid user2 mode n
# ugidfw set 3 subject uid user1 object gid user2 mode n
```

Эти команды запретят весь и любой доступ пользователя `user1`, включая просмотр подкаталогов, к домашнему каталогу пользователя `user2`.

Вместо `user1` может быть задано `not uid user2`. Это включит те ограничения, о которых говорилось выше, для всех пользователей кроме одного.

**Замечание:** На пользователя `root` эти изменения не повлияют.

Материал выше должен дать общую идею как модуль `mac_bsdextended(4)` может быть использован в качестве средства защиты файловой системы. За дальнейшей информацией обращайтесь к страницам справочника `mac_bsdextended(4)` и `ugidfw(8)`.

## 15.7. Модуль MAC ifoff

Имя модуля: `mac_ifoff.ko`

Строка конфигурации ядра: `options MAC_IFOFF`

Параметр загрузки: `mac_ifoff_load="YES"`

Модуль `mac_ifoff(4)` существует только для отключения сетевых интерфейсов в работающей системе и удержания их от отправки пакетов во время начальной загрузки. Это не требует установления в системе каких-либо меток, нет и зависимости от других модулей MAC.

Большая часть управления может быть выполнена через переменные `sysctl`.

- `security.mac.ifoff.lo_enabled` включает/выключает весь трафик на `loopback (lo(4))` интерфейсе.
- `security.mac.ifoff.bpfrecv_enabled` включает/выключает весь трафик на интерфейсе Berkeley Packet Filter (`bpf(4)`).
- `security.mac.ifoff.other_enabled` включает/выключает весь трафик на всех других интерфейсах.

Одно из наиболее частых использований `mac_ifoff(4)` это сетевой мониторинг в среде, где сетевой трафик не должен быть разрешен во время загрузки. Другое предлагаемое применение это написание скрипта, использующего `security/aide` для автоматического блокирования сетевого трафика, если будут обнаружены новые или измененные файлы в защищаемых каталогах.



## 15.8. Модуль MAC portacl

Имя модуля: `mac_portacl.ko`

Строка конфигурации ядра: `MAC_PORTACL`

Параметр загрузки: `mac_portacl_load="YES"`

Модуль `mac_portacl(4)` используется для ограничения привязки (binding) к локальным портам TCP и UDP, используя различные переменные `sysctl`. По сути `mac_portacl(4)` делает возможной привязку к привилегированным портам, т.е. к портам с номерами меньше 1024 для не-root пользователей.

После загрузки этот модуль включит политику MAC на всех сокетах. Доступны следующие переменные `sysctl`:

- `security.mac.portacl.enabled` включает/отключает политику целиком. <sup>2</sup>
- `security.mac.portacl.port_high` установит наибольший номер порта, для которого `mac_portacl(4)` включает защиту.
- `security.mac.portacl.suser_exempt`, если установлена в ненулевое значение, исключает пользователя `root` из этой политики.
- `security.mac.portacl.rules` задает действующую политику `mac_portacl`: см. ниже.

Действующая политика `mac_portacl`, указанная в `security.mac.portacl.rules`, это текстовая строка в форме `rule[, rule, ...]` с таким количеством правил, которое требуется. Каждое правило задается в формате: `idtype:id:protocol:port`. Параметр `idtype` может принимать значения `uid` или `gid` и используется для интерпретации параметра `id`, в качестве `id` пользователя или группы соответственно. Параметр `protocol` используется для определения применимости этого правила к протоколу TCP или UDP, он может принимать значения `tcp` или `udp`. Последний параметр, `port`, задает номер порта, к которому разрешается привязка указанного пользователя или группы.

**Замечание:** Поскольку набор правил интерпретируется непосредственно ядром, для ID пользователя, группы и номера порта могут быть использованы только числовые значения. Т.е. имена пользователей, групп и сервисов портов не могут быть использованы.

По умолчанию в UNIX-подобных системах порты с номерами менее чем 1024 могут быть использованы только привилегированными процессами, т.е. теми, что запущены от `root`. С `mac_portacl(4)` для разрешения привязки непривилегированных процессов к портам с номерами ниже 1024 эти стандартные ограничения UNIX должны быть отменены. Это может быть выполнено путем установки переменных `sysctl(8)` `net.inet.ip.portrange.reservedlow` и `net.inet.ip.portrange.reservedhigh` в ноль.

Обратитесь к примерам ниже или к странице справочника `mac_portacl(4)` за дальнейшей информацией.

### 15.8.1. Примеры

Следующие примеры должны осветить обсуждение выше чуть лучше:

```
# sysctl security.mac.portacl.port_high=1023
# sysctl net.inet.ip.portrange.reservedlow=0 net.inet.ip.portrange.reservedhigh=0
```

Сначала мы настраиваем `mac_portacl(4)` для работы со стандартными привилегированными портами и отмены обычных ограничений UNIX на привязку.

```
# sysctl security.mac.portacl.suser_exempt=1
```

Пользователь `root` должен быть исключен из этой политики, для этого переменная `security.mac.portacl.suser_exempt` установлена в ненулевое значение. Модуль `mac_portacl(4)` теперь настроен на то поведение UNIX-подобных систем по умолчанию.

```
# sysctl security.mac.portacl.rules=uid:80:tcp:80
```

Разрешает пользователю с UID 80 (обычно это пользователь `www`) привязку к порту 80. Теперь пользователь `www` сможет запустить веб сервер даже без привилегии `root`.

```
# sysctl security.mac.portacl.rules=uid:1001:tcp:110,uid:1001:tcp:995
```

Разрешит пользователю с UID 1001 привязку к TCP портам 110 ("pop3") и 995 ("pop3s"). Это позволит данному пользователю запустить сервер, принимающий соединения на портах 110 и 995.

## 15.9. Политики MAC, использующие метки

В следующих нескольких разделах будут обсуждаться политики MAC, использующие метки.

С этого момента обсуждение будет сфокусировано на возможностях `mac_biba(4)`, `mac_lomac(4)`, `mac_partition(4)`, и `mac_mls(4)`.

**Замечание:** Это лишь примерные настройки, они не должны использоваться непосредственно в реальных задачах. Цель изложения в том, чтобы документировать и показать синтаксис, а также примеры реализации и тестирования.

Для правильной работы этих политик необходимо выполнить некоторые приготовления.

### 15.9.1. Приготовление к использованию политик с метками

В файл `login.conf` необходимо внести следующие изменения:

- Должен быть добавлен класс `insecure`, или другой подобный класс. Наличие класса `insecure` не обязательно, он приводится здесь в качестве примера; другие конфигурации могут использовать другое имя класса.
- Класс `insecure` должен использовать приведенные ниже настройки и определения. Некоторые из них могут быть изменены, но строка, определяющая метку по умолчанию, необходима и должна быть оставлена.

```
insecure:\
:copyright=/etc/COPYRIGHT:\
:welcome=/etc/motd:\
:setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
:path=~:/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:\
```

```

:manpath=/usr/share/man /usr/local/man:\
:nologin=/usr/sbin/nologin:\
:cputime=1h30m:\
:datasize=8M:\
:vmemoryuse=100M:\
:stacksize=2M:\
:memorylocked=4M:\
:memoryuse=8M:\
:filesize=8M:\
:coredumpsize=8M:\
:openfiles=24:\
:maxproc=32:\
:priority=0:\
:requirehome:\
:passwordtime=91d:\
:umask=022:\
:ignoretime@:\
:label=partition/13,mls/5,biba/low:

```

Перед тем, как переключать пользователей на новый класс, необходимо запустить команду `cap_mkdb(1)` на `login.conf(5)`.

Пользователю `root` также необходимо присвоить класс; иначе, почти любой команде, выполняемой от `root`, потребуется использование `setpmac`.

- Убедитесь, что все разделы, на которых будут установлены метки MAC, поддерживают параметр `multilabel`. Нам необходимо сделать это, поскольку многие из примеров здесь содержат различные метки в целях тестирования. Просмотрите вывод команды `mount` в качестве необходимой предосторожности.
- Переключите всех пользователей, которые будут использовать новые механизмы безопасности, на этот класс. Информация по этой процедуре находится в `rw(8)` или `vipw(8)`.

## 15.10. Модуль MAC partition

Имя модуля: `mac_partition.ko`

Строка настройки ядра: `options MAC_PARTITION`

Параметр загрузки: `mac_partition_load="YES"`

Политика `mac_partition(4)` распределяет процессы по "разделам" на основе их MAC меток. Это может быть представлено как особый тип `jail(8)`, хотя такое сравнение едва ли подходит.

Этот модуль должен быть добавлен в `loader.conf(5)`, чтобы политика была загружена и включена при загрузке системы.

Большая часть настройки этой политики выполняется с помощью утилиты `setpmac(8)`, которая будет описана ниже. Для данной политики имеется также следующая переменная `sysctl`:

- `security.mac.partition.enabled` включает MAC разделение процессов.

Когда эта политика включена, пользователям разрешено просматривать только собственные процессы, но не разрешено пользоваться определенными утилитами. Например, пользователю из класса `insecure` выше не будет разрешено использование команды `top`, а также многих других команд, которые должны порождать процесс.

Для присвоения утилитам меток `partition` используйте утилиту `setpmac`:

```
# setpmac partition/13 top
```

Команда `top` будет добавлена к метке, установленной для пользователей класса `insecure`. Обратите внимание, что все процессы, порожденные пользователями класса `insecure`, останутся с меткой `partition/13`.

### 15.10.1. Примеры

Следующая команда покажет вашу метку раздела и список процессов:

```
# ps -zx
```

Следующей командой можно просмотреть метку раздела процессов других пользователей и их запущенные процессы:

```
# ps -zu trhodes
```

**Замечание:** Пользователи могут увидеть процессы `root`, если не загружена политика `mac_seeotheruids(4)`.

Действительно "продвинутой" реализация должна отключать все сервисы через `/etc/rc.conf` и запускать их через скрипт, который установит правильный набор меток.

**Замечание:** Следующие политики поддерживают целочисленные установки вместо трех меток по умолчанию. Эти опции, включая их ограничения, описываются более подробно в страницах справочника модулей.

## 15.11. Модуль многоуровневой безопасности MAC (MLS)

Имя модуля: `mac_mls.ko`

Строка конфигурации ядра: `options MAC_MLS`

Параметр загрузки: `mac_mls_load="YES"`

Политика `mac_mls(4)` контролирует взаимодействие субъектов и объектов системы путем применения строгой политики к потоку информации.

В среде MLS, для каждого субъекта или объекта внутри отдела (`compartment`) устанавливается "уровень допуска". Поскольку количество уровней допуска может превышать шесть тысяч, для любого системного администратора задача настройки каждого субъекта или объекта может быть

слишком сложной. К счастью, существуют "постоянные" метки, которые уже включены в эту политику.

Эти метки `mls/low`, `mls/equal` и `mls/high`. Поскольку эти метки подробно описываются в справочнике, здесь мы дадим только краткое описание:

- Метка `mls/low` содержит минимальную настройку, что позволяет доминирование над ней всех других объектов. Все, что помечено с `mls/low`, находится на низком уровне доступа и доступ к более высоким уровням будет запрещен. Кроме того, эта метка предотвратит запись или передачу информации объектам с более высоким уровнем доступа.
- Метка `mls/equal` должна быть помещена на объекты, являющиеся исключением из политики.
- Метка `mls/high` это наибольший возможный уровень доступа. Объекты с этой меткой будут доминировать над всеми другими объектами системы; однако, утечка информации от них к объектам более низкого класса невозможна.

MLS представляет собой:

- Иерархические уровни безопасности с набором не иерархических категорий;
- Фиксированные правила: нет чтения сверху, нет записи вниз (субъект может иметь доступ на чтение объектов собственного уровня или ниже, но не выше. Аналогично, субъект может иметь доступ на запись в объекты своего уровня или выше, но не наоборот.);
- Секретность (предотвращение неавторизованного раскрытия данных);
- Основа для разработки систем, одновременно работающих с данными на нескольких уровнях секретности (без утечки информации).

Для настройки специальных сервисов и интерфейсов доступны следующие переменные `sysctl`:

- `security.mac.mls.enabled` используется для включения/отключения политики MLS.
- `security.mac.mls.ptys_equal` пометит все устройства `pty(4)` как `mls/equal` во время создания.
- `security.mac.mls.revocation_enabled` используется для запрета доступа к объектам после того, как их метка изменится в меньшую сторону.
- `security.mac.mls.max_compartments` используется для установки максимального количества уровней отделов на объекты; обычно это максимальное количество отделов, разрешенных в системе.

Для управления метками MLS существует команда `setfmac(8)`. Для присвоения метки объекту, выполните следующую команду:

```
# setfmac mls/5 test
```

Для получения метки MLS файла `test`, выполните следующую команду:

```
# getfmac test
```

Выше представлен краткий обзор возможностей политики MLS. Существует метод, связанный с созданием основного файла политики в каталоге `/etc`, где будет определена необходимая для политики MLS информация, которая будет передана команде `setfmac`. Этот метод будет описан после рассмотрения всех политик.

Итоги: объект с низким уровнем доступа не может прочесть данные объекта с высоким уровнем доступа. Базовая политика должна устанавливать `mls/high` на всем, что не должно быть прочитано, даже если туда необходимо записывать. На всем, куда нельзя писать, должна быть установлена метка `mls/low`, даже если это необходимо читать. Наконец, на всем остальном установите `mls/equal`. Все пользователи, помеченные как `insecure`, должны иметь метку `mls/low`.

## 15.12. Модуль MAC Biba

Имя модуля: `mac_biba.ko`

Строка конфигурации ядра: `options MAC_BIBA`

Параметр загрузки: `mac_biba_load="YES"`

Модуль `mac_biba(4)` загружает MAC политику Biba. Эта политика работает в основном так же, как и MLS, за исключением того, что правила потока информации изменены на противоположные. Они предназначены для предотвращения передачи потока секретной информации вверх, в то время как политика MLS предотвращает передачу потока секретной информации вниз; таким образом, большая часть этого раздела применима к обоим политикам.

В среде Biba, каждому субъекту или объекту присваивается метка "целостности". Эти метки состоят из иерархических уровней и не-иерархических компонентов. При возрастании уровня объекта или субъекта это повышает его целостность.

Поддерживаемые метки `biba/low`, `biba/equal`, и `biba/high`; описаны ниже:

- Метка `biba/low` обеспечивает наименьшую целостность объекта или субъекта. Установка ее на объект или субъект заблокирует их доступ к объектам или субъектам, имеющим более высокую метку. Тем не менее, у них остается доступ на чтение.
- Метка `biba/equal` должна помещаться только на объекты, исключаящиеся из политики.
- Метка `biba/high` разрешит запись в объекты с более низкой меткой, но не разрешит чтение из этих объектов. Рекомендуется помещать такую метку на объекты, влияющие на целостность всей системы.

Biba представляет собой:

- Иерархические уровни целостности с набором не иерархических категорий;
- Фиксированные правила: нет записи наверх, нет чтения снизу (обратно MLS). Субъект может иметь доступ на запись к объектам своего уровня или ниже, но не выше. Аналогично, субъект может иметь доступ на чтение к объектам своего уровня или выше, но не ниже;
- Целостность (предотвращение неавторизованного изменения данных);
- Уровни целостности (вместо уровней секретности MLS).

Для управления политикой Biba могут быть использованы следующие переменные `sysctl`:

- `security.mac.biba.enabled` может использоваться для включения/выключения политики Biba.
- `security.mac.biba.ptys_equal` может использоваться для отключения политики Biba на устройствах `pty(4)`.

- `security.mac.biba.revocation_enabled` включит отмену доступа к объектам, если метка изменена на более высокую, чем у субъекта.

Для выполнения настроек политики Biba на системных объектах, применяются команды `setfmac` и `getfmac`:

```
# setfmac biba/low test
# getfmac test
test: biba/low
```

Итоги: субъект с низким уровнем целостности не может писать в субъект с высоким уровнем целостности; субъект с высоким уровнем целостности не может читать из субъекта с низким уровнем целостности.

## 15.13. Модуль MAC LOMAC

Имя модуля: `mac_lomac.ko`

Строка конфигурации ядра: `options MAC_LOMAC`

Параметр загрузки: `mac_lomac_load="YES"`

В отличие от политики MAC Biba, политика `mac_lomac(4)` разрешает доступ к объектам с более низким уровнем целостности только после уменьшения уровня целостности, чтобы не нарушать каких-либо правил целостности.

MAC версия политики целостности Low-watermark, чтобы не пересекаться со старой реализацией `lomac(4)`, работает почти так же, как и Biba, за исключением использования плавающих меток для поддержки понижения метки субъекта через отдел для вспомогательной градации (`auxiliary grade compartment`). Этот вспомогательный отдел принимает вид `[auxgrade]`. При включении политики `lomac` с вспомогательной градацией метка должна выглядеть приблизительно так: `lomac/10[2]`, где номер 2 это вспомогательная градация.

Политика MAC LOMAC основана на тотальной пометке всех системных объектов метками целостности, разрешая субъектам читать из объектов с более низкой степенью целостности и с уменьшением метки субъекта для предотвращения последующей записи в объекты с более высокой степенью целостности. Параметр `[auxgrade]` обсуждался выше, таким образом политика может быть более совместимой и требовать меньшей первоначальной настройки, чем Biba.

### 15.13.1. Примеры

Как и для политик Biba и MLS, для установки меток на системные объекты и субъекты могут быть использованы утилиты `setfmac` и `setpmac`:

```
# setfmac /usr/home/trhodes lomac/high[low]
# getfmac /usr/home/trhodes lomac/high[low]
```

Обратите внимание, что вспомогательная градация здесь `low`, эта возможность предоставляется только политикой MAC LOMAC policy.

## 15.14. Реализация защищенной среды с MAC

Нижеследующая демонстрация реализует защищенную среду с использованием различных MAC модулей с соответственно настроенными политиками. Используйте этот пример только для тестирования, он не предназначен для удовлетворения всех требований к защите. Реализация этих политик без понимания принципа их работы неприменима в реальных задачах.

Перед началом процесса настройки, на каждую файловую систему необходимо установить параметр `multilabel`, который упоминался в начале этой главы. Невыполнение этого требования приведет к ошибкам.

### 15.14.1. Создание `insecure` класса пользователя

Начните процедуру добавлением следующего класса пользователя к файлу `/etc/login.conf`:

```
insecure:\
:copyright=/etc/COPYRIGHT:\
:welcome=/etc/motd:\
:setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
:path=~:/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin
:manpath=/usr/share/man /usr/local/man:\
:nologin=/usr/sbin/nologin:\
:cputime=1h30m:\
:datasize=8M:\
:vmemoryuse=100M:\
:stacksize=2M:\
:memorylocked=4M:\
:memoryuse=8M:\
:filesize=8M:\
:coredumpsize=8M:\
:openfiles=24:\
:maxproc=32:\
:priority=0:\
:requirehome:\
:passwordtime=91d:\
:umask=022:\
:ignoretime@:\
:label=partition/13,mls/5:
```

и добавлением следующей строки к `default` классу пользователя:

```
:label=mls/equal,biba/equal,partition/equal:
```

После завершения этих действий, для пересборки базы данных должна быть выполнена следующая команда:

```
# cap_mkdb /etc/login.conf
```



### 15.14.2. Загрузка с необходимыми модулями

Добавьте к `/boot/loader.conf` следующие строки, чтобы необходимые модули были загружены при старте системы:

```
mac_biba_load="YES"
mac_mls_load="YES"
mac_seeotheruids_load="YES"
mac_partition_load="YES"
```

### 15.14.3. Установка всех пользователей в `insecure`

Всем учетным записям, кроме `root` или системных пользователей теперь потребуется присвоить класс (`login class`). При отсутствии класса пользователи не смогут получить доступа к обычным командам, таким как `vi(1)`. Следующий скрипт `sh` сделает все необходимое:

```
# for x in `awk -F: '($3 >= 1001) && ($3 != 65534) { print $1 }' \
/etc/passwd`; do pw usermod $x -L insecure; done;
```

После этого изменения необходимо запустить команду `cap_mkdb` на файле `/etc/master.passwd`.

### 15.14.4. Завершение настройки

Должен быть создан файл контекста; следующий пример взят из примера политики от Robert Watson, он может быть помещен в `/etc/policy.contexts`:

```
# This is the default BIBA/MLS policy for this system.

.*                                biba/high,mls/high
/sbin/dhclient                   biba/high(low),mls/high(low)
/dev(/.*)?                       biba/equal,mls/equal
# This is not an exhaustive list of all "privileged" devices.
/dev/mdctl                       biba/high,mls/high
/dev/pci                         biba/high,mls/high
/dev/k?mem                       biba/high,mls/high
/dev/io                          biba/high,mls/high
/dev/agp.*                       biba/high,mls/high
(/var)?/tmp(/.*)?               biba/equal,mls/equal
/tmp/\.X11-unix                  biba/high(equal),mls/high(equal)
/tmp/\.X11-unix/. *              biba/equal,mls/equal
/proc(/.*)?                     biba/equal,mls/equal
/mnt.*                           biba/low,mls/low
(/usr)?/home                    biba/high(low),mls/high(low)
(/usr)?/home/. *                 biba/low,mls/low
/var/mail(/.*)?                  biba/low,mls/low
/var/spool/mqueue(/.*)?         biba/low,mls/low
(/mnt)?/cdrom(/.*)?             biba/high,mls/high
(/usr)?/home/(ftp|samba)(/.*)?  biba/high,mls/high
/var/log/sendmail\.st            biba/low,mls/low
/var/run/utmp                    biba/equal,mls/equal
/var/log/(lastlog|wtmp)          biba/equal,mls/equal
```

Эта политика обеспечит безопасность путем применения ограничений на нисходящий и восходящий потоки информации в применении к каталогам и утилитам, приведенным в левой части файла.

Он может быть внесен в систему следующими командами:

```
# setfsmac -ef /etc/policy.contexts /
# setfsmac -ef /etc/policy.contexts /usr
```

**Замечание:** Раскладка вышеприведенной файловой системы может быть различной для разных систем.

Файл `/etc/mac.conf` требует следующих изменений в основном разделе:

```
default_labels file ?biba,?mls
default_labels ifnet ?biba,?mls
default_labels process ?biba,?mls,?partition
default_labels socket ?biba,?mls
```

### 15.14.5. Тестирование настройки

Добавьте пользователя с помощью команды `adduser` и поместите его в класс `insecure` для этих тестов.

В примерах ниже тестирование `root` и обычных пользователей будет смешиваться; форма приглашения поможет различить этих пользователей.

#### 15.14.5.1. Основное тестирование меток

```
% getpmac
biba/15(15-15),mls/15(15-15),partition/15
# setpmac partition/15,mls/equal top
```

**Замечание:** Процесс `top` будет уничтожен перед тем, как мы запустим другой процесс `top`.

#### 15.14.5.2. Тестирование MAC `seeotheruids`

```
% ps Zax
biba/15(15-15),mls/15(15-15),partition/15 1096 #C: S 0:00.03 -su (bash)
biba/15(15-15),mls/15(15-15),partition/15 1101 #C: R+ 0:00.01 ps Zax
```

Просмотр процессов всех других пользователей должен быть запрещен.

#### 15.14.5.3. Тестирование MAC `partition`

Отключите политику MAC `seeotheruids` для остальных тестов:

```
# sysctl security.mac.seeotheruids.enabled=0
```

```
% ps Zax
LABEL                                PID  TT  STAT      TIME COMMAND
biba/equal(low-high),mls/equal(low-high),partition/15  1122 #C:  S+    0:00.02 top
biba/15(15-15),mls/15(15-15),partition/15             1096 #C:  S     0:00.05 -su (bash)
biba/15(15-15),mls/15(15-15),partition/15             1123 #C:  R+    0:00.01 ps Zax
```

Все пользователи должны видеть каждый процесс в своем разделе (partition).

#### 15.14.5.4. Тестирование меток Biba и MLS

```
# setpmac partition/15,mls/equal,biba/high\ (high-high\ ) top
% ps Zax
LABEL                                PID  TT  STAT      TIME  COMMAND
biba/high(high-high),mls/equal(low-high),partition/15  1251 #C:  S+    0:00.02 top
biba/15(15-15),mls/15(15-15),partition/15             1096 #C:  S     0:00.06 -su (bash)
biba/15(15-15),mls/15(15-15),partition/15             1157 #C:  R+    0:00.00 ps Zax
```

Политика Biba позволяет чтение объектов с более высокими метками.

```
# setpmac partition/15,mls/equal,biba/low top
% ps Zax
LABEL                                PID  TT  STAT      TIME COMMAND
biba/15(15-15),mls/15(15-15),partition/15             1096 #C:  S     0:00.07 -su (bash)
biba/15(15-15),mls/15(15-15),partition/15             1226 #C:  R+    0:00.01 ps Zax
```

Политика Biba не позволяет чтение объектов с более низкими метками; тем не менее, MLS разрешает это.

```
% ifconfig bge0 | grep maclabel
maclabel biba/low(low-low),mls/low(low-low)
% ping -c 1 192.0.34.166
PING 192.0.34.166 (192.0.34.166): 56 data bytes
ping: sendto: Permission denied
```

Пользователи не могут выполнить ping на example.com, или на любой домен по этой причине.

Для устранения этой ошибки, запустите следующую команду:

```
# sysctl security.mac.biba.trust_all_interfaces=1
```

Она устанавливает метку интерфейса по умолчанию в незащищенный режим, так что политика Biba по умолчанию не будет применена.

```
# ifconfig bge0 maclabel biba/equal\ (low-high\ ),mls/equal\ (low-high\ )
% ping -c 1 192.0.34.166
PING 192.0.34.166 (192.0.34.166): 56 data bytes
64 bytes from 192.0.34.166: icmp_seq=0 ttl=50 time=204.455 ms
--- 192.0.34.166 ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max/stddev = 204.455/204.455/204.455/0.000 ms
```

Установив более корректную метку, мы можем использовать ping.

Теперь создадим файлы для процедуры тестирования чтения и записи:

```
# touch test1 test2 test3 test4 test5
# getfmac test1
test1: biba/equal,mls/equal
# setfmac biba/low test1 test2; setfmac biba/high test4 test5; \
  setfmac mls/low test1 test3; setfmac mls/high test2 test4
# setfmac mls/equal,biba/equal test3 && getfmac test?
test1: biba/low,mls/low
test2: biba/low,mls/high
test3: biba/equal,mls/equal
test4: biba/high,mls/high
test5: biba/high,mls/equal
# chown testuser:testuser test?
```

Все эти файлы должны принадлежать пользователю testuser. Тесты на чтение:

```
% ls
test1 test2 test3 test4 test5
% ls test?
ls: test1: Permission denied
ls: test2: Permission denied
ls: test4: Permission denied
test3 test5
```

Доступ на чтение не должен быть разрешен для пар: (biba/low,mls/low), (biba/low,mls/high) и (biba/high,mls/high). Теперь несколько тестов на запись:

```
% for i in `echo test*`; do echo 1 > $i; done
-su: test1: Permission denied
-su: test4: Permission denied
-su: test5: Permission denied
```

Подобно тестам на чтение, доступ на запись должен быть запрещен для пар: (biba/low,mls/high) и (biba/equal,mls/equal).

```
% cat test?
cat: test1: Permission denied
cat: test2: Permission denied
1
cat: test4: Permission denied
```

А теперь от root:

```
# cat test2
1
```

## 15.15. Другой пример: Использование MAC для защиты веб сервера

Будет создано отдельное хранилище для веб данных, к которому пользователи должны иметь доступ. Это позволит biba/high управлять доступом к веб данным.

Начните с создания каталога для хранения веб данных:

```
# mkdir /usr/home/cvs
```

Теперь инициализируйте его командой cvs:

```
# cvs -d /usr/home/cvs init
```

Для начала необходимо включить политику biba, добавив mac\_biba\_enable="YES" в /boot/loader.conf. Предполагается, что ядро скомпилировано с поддержкой MAC.

Далее установите метку biba/high для всей системы по умолчанию.

В файл login.conf, класс default, необходимо внести следующие изменения:

```
:ignoretime@:\
:umask=022:\
:label=biba/high:
```

Каждого пользователя необходимо поместить в класс по умолчанию; такая команда:

```
# for x in `awk -F: '($3 >= 1001) && ($3 != 65534) { print $1 }' \
/etc/passwd`; do pw usermod $x -L default; done;
```

быстро решит эту задачу.

Теперь создадим другой класс, web, копию класса default с меткой, установленной в biba/low.

Создайте пользователя для работы с основными веб данными, хранящимися в репозитории **cvs**. Этого пользователя необходимо поместить в новый класс, web.

Поскольку метка по умолчанию biba/high, на репозитории она будет той же. Веб данные должны иметь ту же метку, чтобы у пользователей был доступ к ним на чтение/запись. Веб сервер должен иметь доступ к тем же данным, к которым есть доступ у пользователей с меткой biba/high, для этого необходимо понизить метку данных.

Все, что потребуется, это следующий sh(1) скрипт, который может быть запущен из cron(8):

```
PATN=/bin:/usr/bin:/usr/local/bin; export PATN;
CVSROOT=/home/repo; export CVSROOT;
cd /home/web;
cvs -qR checkout -P htdocs;
exit;
```

**Замечание:** Во многих случаях в веб файлы cvs необходимо поместить теги ld.

Этот скрипт теперь может быть помещен в домашний каталог пользователя web, необходимо также добавить следующую запись crontab(1):

```
# Выполнять checkout web данных под меткой biba/low каждые 12 часов:
0      */12      *      *      *      web      /home/web/checkout.sh
```

Эта запись будет извлекать HTML страницы каждые двенадцать часов.

Метод запуска веб сервера по умолчанию также должен быть изменен для запуска процесса с меткой biba/low. Это может быть сделано путем следующего изменения в скрипте /usr/local/etc/rc.d/apache.sh:

```
command="setpmac biba/low /usr/local/sbin/httpd"
```

Настройки **Apache** должны быть изменены для работы с политикой biba/low. В этом случае необходимо указать для хранения лог файлов каталог с меткой biba/low, иначе будут возвращены ошибки access denied.

**Замечание:** В этом примере необходимо указать в директиве docroot каталог /home/web/htdocs; или, **Apache** не сможет найти каталог с документами.

Необходимо также изменить другие параметры конфигурации, включая PID файл, Scoreboardfile, DocumentRoot, или любые другие настройки для каталогов, где необходим доступ на запись. При использовании biba будет запрещен доступ на запись во все каталоги сервера, на которых нет метки biba/low.

## 15.16. Решение проблем с инфраструктурой MAC

На стадии разработки несколько пользователей сообщали о проблемах при обычных настройках. Некоторые из этих проблем приведены ниже:

### 15.16.1. Параметр multilabel не может быть включен на /

Параметр multilabel не включается на моем корневом (/) разделе!

Похоже, что каждый пятидесятый пользователь сталкивается с этой проблемой; на самом деле, и у нас была эта проблема в первых настройках. Дальнейшие наблюдения за этой так называемой "ошибкой" привели меня к мнению, что это результат или некорректной документации, или неправильной интерпретации этой документации. Независимо от того, почему это случилось, для решения этой проблемы могут быть предприняты следующие шаги:

1. Отредактируйте /etc/fstab и установите для корневого раздела параметр только для чтения (ro).
2. Перегрузитесь в однопользовательский режим.
3. Запустите команду tuneefs -l enable на /.
4. Перегрузите систему в нормальный режим.
5. Запустите mount -urw / и измените параметр ro обратно на rw в /etc/fstab; перезагрузите систему опять.
6. Дважды проверьте вывод mount, чтобы убедиться, что параметр multilabel был установлен на корневой файловой системе.

### 15.16.2. Не могу запустить XFree86 после MAC

После настройки системы безопасности MAC, я больше не могу запускать XFree86!

Это может быть вызвано политикой MAC `partition` или путем неправильной установки меток одной из политик MAC. Для отладки попробуйте следующее:

1. Просмотрите сообщение об ошибке; если пользователь находится в классе `insecure`, проблема может быть в политике `partition`. Попробуйте установить класс пользователя обратно в `default` и пересобрать базу данных командой `cap_mkdb`. Если это не решит проблемы, попробуйте шаг два.
2. Дважды проверьте политики с метками. Убедитесь, что политики настроены правильно для рассматриваемого пользователя, приложения XFree86, и устройств в `/dev`.
3. Если проблема не решена, отправьте сообщение об ошибке и описание вашей системы в список рассылки TrustedBSD, находящийся на веб сайте TrustedBSD (<http://www.TrustedBSD.org>) или в Список рассылки, посвящённый вопросам и ответам пользователей FreeBSD (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions>).

### 15.16.3. Error: `_secure_path(3)` cannot stat `.login_conf`

При попытке переключения от `root` на другого пользователя системы, появляется сообщение об ошибке `_secure_path: unable to state .login_conf`.

Это сообщение обычно показывается, когда у пользователя более высокая метка, чем у пользователя, которым он пытается стать. Например, у пользователя системы `joe` метка по умолчанию `biba/low`. Пользователь `root`, метка которого `biba/high`, не может просматривать домашний каталог пользователя `joe`. Это не зависит от того, использует ли пользователь `root` команду `su joe` или нет. В этом сценарии модель целостности Biba не позволит `root` просматривать объекты с низким уровнем целостности.

### 15.16.4. Пользователя `root` нет!

В нормальном или даже однопользовательском режиме `root` не обнаруживается. Команда `whoami` возвращает 0 (нуль) и `su` возвращает `who are you?`. Что можно сделать?

Это может произойти, если политика с метками была отключена, или через `sysctl(8)`, или путем загрузки модуля политики. Если политика была постоянно или временно отключена, базу данных `login` необходимо перенастроить. Дважды проверьте `login.conf`, чтобы убедиться, что все параметры `label` были удалены и пересоберите базу данных командой `cap_mkdb`.

## Примечания

1. Есть и другие ситуации, в которых могут возникать различные ошибки. Например, файл может не находиться во владении пользователя, пытающегося изменить метку на объекте, объект может не существовать или существовать только для чтения. Политика может не позволить процессу изменить метку на файл, возможно в связи со свойствами файла, процесса, или нового значения метки. Например: пользователь с низким уровнем доверия пытается изменить метку на файле с

высоким уровнем доверия. Или пользователь с низким уровнем доверия пытается изменить метку файла с низким уровнем доверия на метку высокого уровня.

2. Вследствие ошибки переменная `sysctl security.mac.portacl.enabled` не будет работать в FreeBSD 5.2.1 или более ранних релизах.



# Глава 16. Устройства хранения

*Перевод на русский язык: Андрей Захватов.*

## 16.1. Краткий обзор

В этой главе описывается использование дисков во FreeBSD. К ним относятся диски в памяти, диски, подключенные по сети, обычные устройства хранения SCSI/IDE и устройства, использующие интерфейс USB.

После чтения этой главы вы будете знать:

- Терминологию, используемую во FreeBSD для описания организации данных на физическом диске (разделы и слайсы).
- Как добавить дополнительные винчестеры к вашей системе.
- Как настроить FreeBSD для использования дисковых устройств USB.
- Как настроить виртуальные файловые системы, такие, как диски в оперативной памяти.
- Как использовать квоты для ограничения использования дискового пространства.
- Как зашифровать диски, чтобы защитить их от взлома.
- Как создавать и записывать CD и DVD во FreeBSD.
- Различные варианты использования устройств хранения для резервных копий.
- Как использовать программы резервного копирования, имеющиеся для FreeBSD.
- Как выполнять резервное копирование на дискеты.
- Что такое мгновенные копии файловых систем и как их эффективно использовать

Перед прочтением этой главы вам потребуется:

- Узнать как настраивать и устанавливать новое ядро FreeBSD (Гл. 8).

## 16.2. Имена устройств

Далее приводится список физических устройств хранения информации, которые поддерживаются во FreeBSD, и имена устройств, которые им соответствуют.

**Таблица 16-1. Соглашения по именованию физических дисков**

Тип диска	Имя дискового устройства
Винчестеры IDE	ad
Приводы IDE CDROM	acd
Винчестеры SCSI и дисковые устройства USB	da
Приводы SCSI CDROM	cd

Тип диска	Имя дискового устройства
Различные нестандартные приводы CDROM	mcd для Mitsumi CD-ROM, scd для Sony CD-ROM, matcd для Matsushita/Panasonic CD-ROM а
Дискеты	fd
Ленточные приводы SCSI	sa
Ленточные приводы IDE	ast
Флэш-диски	fla для флэш-устройств DiskOnChip®
Диски RAID	aacd для Adaptec AdvancedRAID, mlxd и mlyd для Mylex, amrd для AMI MegaRAID, idad для Compaq Smart RAID, twed для 3ware® RAID.

Примечания: а. 5 октября 2002 года драйвер matcd(4) был удалён из ветки FreeBSD 4.X и отсутствует во FreeBSD 5.0 и 5.X.

## 16.3. Добавление дисков

*Изначальный текст предоставил David O'Brien.*

Предположим, что мы хотим установить новый диск SCSI на машину, имеющую в данный момент только один диск. Сначала выключим компьютер и установим диск в компьютер согласно инструкциям к компьютеру, контроллеру и от производителя диска. Из-за большого разнообразия этих процедур их рассмотрение выходит за рамки этого документа..

Войдите в систему как пользователь root. После того, как вы установили диск, просмотрите файл /var/run/dmesg.boot, чтобы убедиться, что новый диск был найден. Продолжая наш пример, только что добавленный диск будет называться da1 и мы хотим смонтировать его в каталог /1 (если вы добавляете диск IDE, то устройство будет называться wd1 в системах, предшествовавших 4.0, и ad1 в системах 4.X и 5.X).

Так как FreeBSD работает на IBM-PC совместимых компьютерах, она должна принимать во внимание разделы PC BIOS. В этом заключается отличие от традиционных разделов BSD. Диск PC может иметь до четырех записей разделов BIOS. Если диск на самом деле будет использоваться исключительно под FreeBSD, вы можете использовать режим *dedicated*. В противном случае FreeBSD будет располагаться в одном из разделов PC BIOS. Во FreeBSD разделы PC BIOS называются *слайсами*, чтобы не путать их с традиционными разделами BSD. Вы также можете использовать слайсы и с диском, предназначенным исключительно для FreeBSD, однако используемым в компьютере, на котором имеется дополнительная операционная система. Это нужно для того, чтобы не было путаницы с утилитой fdisk другой операционной системы.

В случае слайсов диск будет добавлен как /dev/dalsle. Это интерпретируется следующим образом: диск SCSI, устройство номер 1 (второй диск SCSI), слайс 1 (раздел PC BIOS 1), и раздел BSD e. В случае использования в выделенном режиме диск будет добавлен просто как /dev/da1e.

Вследствие использования 32-битных целых для хранения числа секторов, bsdlable(8) (называемый disklabel(8) в FreeBSD 4.X) ограничен  $2^{32}-1$  секторами на диск, или 2ТБ в в большинстве случаев. Формат fdisk(8) позволяет наличие первого сектора со смещением не более  $2^{32}-1$  и длину не более  $2^{32}-1$ , что ограничивает размер раздела до 2ТБ, а размер диска до 4ТБ в большинстве случаев. Формат sunlabel(8) ограничен  $2^{32}-1$  секторами на раздел и 8 разделами, что составляет 16ТБ. Для дисков большего раздела могут быть использованы разделы gpt(8).

### 16.3.1. Использование утилиты **sysinstall(8)**

#### 1. Использование **Sysinstall**

Вы можете использовать простое меню утилиты `/stand/sysinstall` для разбиения на разделы и разметки нового диска. Войдите как пользователь `root` или воспользуйтесь командой `su`. Запустите команду `/stand/sysinstall` и войдите в меню `Configure`. Внутри `FreeBSD Configuration Menu`, пролистайте и выберите пункт `Fdisk`.

#### 2. Редактор разделов **fdisk**

Работая с утилитой **fdisk**, мы можем выбрать **A** для того, чтобы использовать под `FreeBSD` полностью весь диск. Когда будет задан вопрос о том, хотите ли вы "сохранить совместимость с другими возможными операционными системами в будущем", ответьте **YES**. Запишите изменения на диск при помощи команды **W**. А теперь выйдите из редактора `FDISK`, нажав **q**. В этот момент вам будет задан вопрос о главной загрузочной записи. Так как вы добавляете диск к уже работающей системе, выберите `None`.

#### 3. Редактор метки диска

Теперь вам нужно выйти из **sysinstall** и запустить эту утилиту снова. Следуйте указаниям выше, но на этот раз выберите пункт `Label`. Вы перейдете к меню `Disk Label Editor`. Здесь вы создадите традиционные разделы `BSD`. На диске может быть до восьми разделов, имеющих метки `a-h`. Некоторые из меток разделов имеют особый смысл. Раздел `a` используется для размещения корневого раздела (`/`). По этой причине только ваш системный диск (например, тот, с которого происходит загрузка), должен иметь раздел `a`. Раздел `b` используется под раздел подкачки, и вы можете иметь много дисков с разделами подкачки. Раздел `c` используется для доступа ко всему диску в режиме эксклюзивного использования или ко всему слайсу `FreeBSD` при работе в режиме с использованием слайсов. Остальные разделы имеют обычное предназначение.

Редактор метки диска программы **sysinstall** использует раздел `e` для некорневого раздела и не для раздела подкачки. Внутри редактора метки диска создайте отдельную файловую систему, нажав **C**. Когда будет задан вопрос о том, будет ли это раздел с файловой системой (`FS`) или это будет раздел подкачки, выберите `FS` и наберите точку монтирования (например, `/mnt`). При добавлении диска после установки системы, программа **sysinstall** не будет автоматически создавать записи в файле `/etc/fstab`, поэтому точка монтирования не так уж и важна.

Теперь вы готовы записать новую метку на диск и создать на нем файловую систему. Сделайте это, набрав **W**. Пройгнорируйте сообщения об ошибках от **sysinstall** о невозможности смонтировать новый раздел. Полностью выйдите из редактора метки диска и из программы **sysinstall**.

#### 4. Завершение

Последний шаг заключается в редактировании файла `/etc/fstab` и добавлении записи для вашего нового диска.

### 16.3.2. Использовании утилит командной строки

#### 16.3.2.1. Работа со слайсами

Следующая настройка позволит вашему диску корректно работать с другими операционными

системами, которые могут быть установлены на вашем компьютере, и не вызовет конфликта с утилитами `fdisk` других операционных систем. Этот способ рекомендуется использовать для установок новых дисков. Используйте **эксклюзивный режим**, только если у вас есть реальные причины делать это!

```
# dd if=/dev/zero of=/dev/dal bs=1k count=1
# fdisk -BI dal # Инициализируем новый диск.
# disklabel -B -w -r dals1 auto # Размечаем его.
# disklabel -e dals1 # Редактируем только что созданную метку диска и добавляем разделы.
# mkdir -p /1
# newfs /dev/dals1e # Повторяем этот шаг для всех созданных разделов.
# mount /dev/dals1e /1 # Монтируем раздел(ы)
# vi /etc/fstab # Добавляем соответствующую запись/записи в файл /etc/fstab.
```

Если у вас установлен диск IDE, подставьте `ad` вместо `da`. На системах версий ранее 4.X используйте `wd`.

### 16.3.2.2. Эксклюзивный режим

Если вы не будете использовать новый диск совместно с другой операционной системой, то вы можете использовать режим **эксклюзивного использования**. Отметьте, что этот режим может ввести в заблуждение операционные системы от Microsoft; однако информацию они не разрушат. А вот OS/2 компании IBM будет "забирать себе" любой раздел, который она найдет и не сможет распознать.

```
# dd if=/dev/zero of=/dev/dal bs=1k count=1
# disklabel -Brw dal auto
# disklabel -e dal # create the 'e' partition
# newfs -d0 /dev/dale
# mkdir -p /1
# vi /etc/fstab # add an entry for /dev/dale
# mount /1
```

Альтернативный метод заключается в следующем:

```
# dd if=/dev/zero of=/dev/dal count=2
# disklabel /dev/dal | disklabel -BrR dal /dev/stdin
# newfs /dev/dale
# mkdir -p /1
# vi /etc/fstab # add an entry for /dev/dale
# mount /1
```

**Замечание:** Начиная с FreeBSD 5.1-RELEASE, на смену старой программе `disklabel(8)` пришла утилита `bsdlabel(8)`. У `bsdlabel(8)` отсутствуют некоторые устаревшие опции и параметры; в примере выше параметр `-r` не может использоваться с `bsdlabel(8)`. Для получения дополнительной информации обратитесь к справочной странице по `bsdlabel(8)`.

## 16.4. RAID

### 16.4.1. Программный RAID

#### 16.4.1.1. Конфигурация драйвера объединённого диска (CCD)

*Оригинальный текст предоставил Christopher Shumway. Изменения внёс Jim Brown.*

При выборе решения для организации хранилища самыми важными характеристиками являются скорость, надёжность и стоимость. Редко все эти характеристики наличествуют одновременно; обычно быстрое и надёжное устройство хранения стоит дорого, а при уменьшении стоимости в жертву приносятся скорость работы или надёжность.

При проектировании описываемой далее системы в качестве самого важного фактора была выбрана её стоимость, затем быстродействие и надёжность. Скорость передачи данных для этой системы ограничивалась только пропускной способностью сети. И, хотя надёжность очень важна, CCD-диск, описываемый ниже, обслуживал работу с данными, полные копии которых уже хранились на дисках CD-R, так они могли быть с лёгкостью обновлены.

При выборе решения для массового хранения данных первым шагом является определение ваших требований к нему. Если в ваших требованиях главными являются скорость или надёжность, а не стоимость, то ваш выбор будет отличаться от описываемой в этом разделе системы.

##### 16.4.1.1.1. Установка оборудования

Кроме системного IDE-диска, основу описываемого далее CCD-диска общим объёмом примерно в 90 Гбайт составили три IDE-диска Western Digital 30GB, 5400 RPM. В идеальном случае каждый диск IDE имеет собственный контроллер и кабель, но для минимизации стоимости дополнительные контроллеры IDE не использовались. Вместо этого диски были настроены при помощи переключателей так, что на каждом IDE-контроллере находилось по одному ведущему и одному ведомому диску.

До перезагрузки BIOS системы была настроена на автоматическое распознавание подключенных дисков. Более важно то, что при перезагрузке их распознала FreeBSD:

```
ad0: 19574MB <WDC WD205BA> [39770/16/63] at ata0-master UDMA33
ad1: 29333MB <WDC WD307AA> [59598/16/63] at ata0-slave UDMA33
ad2: 29333MB <WDC WD307AA> [59598/16/63] at ata1-master UDMA33
ad3: 29333MB <WDC WD307AA> [59598/16/63] at ata1-slave UDMA33
```

**Замечание:** Если FreeBSD не распознала все диски, проверьте корректность положения переключателей на них. На большинстве IDE-дисков имеется также переключатель "Cable Select". Он *не имеет* отношения к выбору ведущего и ведомого устройств. Для получения помощи по правильному положению переключателей обратитесь к документации по устройствам.

Затем определите, как сделать их частью файловой системы. Изучите справку по `vinum(8)` (Гл. 17) и `ccd(4)`. В нашем конкретном случае была выбрана технология `ccd(4)`.

#### 16.4.1.1.2. Настройка CCD

Драйвер `ccd(4)` позволяет вам взять несколько идентичных дисков и объединить их в одну логическую файловую систему. Для использования `ccd(4)` нужно ядро со встроенной поддержкой `ccd(4)`. Добавьте такую строку в файл конфигурации ядра, перестройте и установите новое ядро:

```
pseudo-device    ccd      4
```

В системах 5.X вместо этого вам нужно использовать такую строку:

```
device    ccd
```

**Замечание:** Во FreeBSD 5.X нет нужды указывать количество устройств `ccd(4)` так как драйвер устройства `ccd(4)` теперь клонируется сам — новые экземпляры устройств будут создаваться автоматически по необходимости.

Во FreeBSD 3.0 и последующих версиях поддержка `ccd(4)` также может быть обеспечена загрузкой подгружаемого модуля ядра.

Для настройки `ccd(4)` сначала вам нужно воспользоваться утилитой `disklabel(8)` для разметки дисков:

```
disklabel -r -w ad1 auto
disklabel -r -w ad2 auto
disklabel -r -w ad3 auto
```

При этом создаются метки для `ad1c`, `ad2c` и `ad3c`, которые занимают диск полностью.

**Замечание:** Начиная с FreeBSD 5.1-RELEASE, на смену старой программе `disklabel(8)` пришла утилита `bsdlabel(8)`. У `bsdlabel(8)` отсутствуют некоторые устаревшие опции и параметры; в примере выше параметр `-r` не может использоваться с `bsdlabel(8)`. Для получения дополнительной информации обратитесь к справочной странице по `bsdlabel(8)`.

Следующим шагом является изменение типа метки диска. Для редактирования дисков можно использовать утилиту `disklabel(8)`:

```
disklabel -e ad1
disklabel -e ad2
disklabel -e ad3
```

При этом в редакторе, задаваемом переменной окружения `EDITOR` (обычно это `vi(1)`), открывается текущая метка каждого диска.

Немодифицированная метка диска будет выглядеть примерно следующим образом:

```
8 partitions:
#          size      offset      fstype    [fsize bsize bps/cpg]
  c: 60074784         0      unused         0     0     0   # (Cyl.    0 - 59597)
```

Добавьте новый раздел `e` для использования драйвером `ccd(4)`. Как правило, он может быть скопирован с раздела `s`, но поле `fstype` *должно* иметь значение **4.2BSD**. Теперь метка диска должна выглядеть примерно так:

```
8 partitions:
#          size      offset      fstype    [fsize bsize bps/cpg]
  c: 60074784         0      unused         0     0     0   # (Cyl.  0 - 59597)
  e: 60074784         0      4.2BSD         0     0     0   # (Cyl.  0 - 59597)
```

#### 16.4.1.1.3. Построение файловой системы

Файл устройства для `ccd0c` может ещё не существовать, так что для его создания предварительно выполните такие команды:

```
cd /dev
sh MAKEDEV ccd0
```

**Замечание:** Во FreeBSD 5.0 `devfs(5)` будет управлять файлами устройств в каталоге `/dev` автоматически, так что в использовании `MAKEDEV` необходимости нет.

Теперь, когда все диски размечены, вы должны построить `ccd(4)`. Для этого используйте утилиту `ccdconfi g(8)` с параметрами, подобными следующим:

```
ccdconfig ccd0❶ 32❷ 0❸ /dev/ad1e❹ /dev/ad2e /dev/ad3e
```

Использование и значение каждого параметра описывается ниже:

- ❶ Первым аргументом является конфигурируемое устройство, в нашем случае `/dev/ccd0c`. Часть `/dev/` является необязательной.
- ❷ Чередование для файловой системы. Оно определяет размер единицы блока данных в количестве дисковых блоков, каждый из которых обычно имеет объём в 512 байт. Таким образом, при чередовании в 32 это будет составлять 16384 байт.
- ❸ Опции для `ccdconfi g(8)`. Если вы хотите включить зеркалирование диска, то можете задать это здесь. В нашей конфигурации зеркалирование для `ccd(4)` не предусмотрено, поэтому здесь задан 0 (ноль).
- ❹ Последним параметром для `ccdconfi g(8)` является список устройств для объединения в массив. Для каждого устройства нужно задавать полное имя.

После запуска `ccdconfi g(8)` устройство `ccd(4)` будет отконфигурировано. Может быть построена файловая система. Обратитесь к справке по команде `newfs(8)` для выяснения требуемых параметров, или просто запустите:

```
newfs /dev/ccd0c
```

#### 16.4.1.1.4. Автоматическое выполнение

Вообще говоря, вам потребуется монтировать `ccd(4)` при каждой перезагрузке. Для этого сначала вы должны отконфигурировать это устройство. Запишите вашу текущую конфигурацию в файл `/etc/ccd.conf` при помощи такой команды:

```
ccdconfig -g > /etc/ccd.conf
```

При перезагрузке скрипт `/etc/rc` запускает команду `ccdconfig -C`, если существует файл `/etc/ccd.conf`. При этом `ccd(4)` автоматически конфигурируется так, чтобы он мог быть смонтирован.

**Замечание:** Если при загрузке вы входите в однопользовательский режим, то перед тем, как выполнять монтирование `ccd(4)` по команде `mount(8)`, вам нужно для конфигурации массива запустить следующую команду:

```
ccdconfig -C
```

Для автоматического монтирования `ccd(4)` поместите запись о `ccd(4)` в файл `/etc/fstab`, чтобы он мог быть смонтирован во время загрузки системы:

```
/dev/ccd0c          /media          ufs      rw      2        2
```

#### 16.4.1.2. Менеджер томов Vinum

Менеджер томов `Vinum` является драйвером блочного устройства, который реализует виртуальные диски. Он отделяет дисковое оборудование от интерфейса блочного устройства и работает с данными таким образом, что в результате повышается гибкость, производительность и надёжность по сравнению с традиционным взглядом на дисковое хранилище как на кусок дискового пространства. `vinum(8)` реализует модели RAID-0, RAID-1 и RAID-5, как по отдельности, так и в комбинациях.

Обратитесь к Гл. 17 для получения более полной информации о `vinum(8)`.

### 16.4.2. Аппаратный RAID

FreeBSD поддерживает также целый ряд аппаратных контроллеров RAID. Эти устройства самостоятельно управляют RAID-подсистемой, без необходимости иметь специфичное для FreeBSD программное обеспечение управления массивом.

При помощи встроенной в адаптер BIOS, он сам управляет большинством дисковых операций. Далее следует краткое описание установки при помощи контроллера Promise IDE RAID. После установки адаптера и запуска системы, выдаётся запрос на ввод. Следуйте указаниям для входа в настройку адаптера. Отсюда вы можете объединить все подключенные диски. После этого во FreeBSD диск(и) будут выглядеть как один диск. Аналогично могут быть настроены и другие уровни RAID.



### 16.4.3. Перестроение массивов ATA RAID1

FreeBSD позволяет вам выполнять горячую замену вышедшего из строя диска. При этом требуется, чтобы вы заметили это до перезагрузки.

Вероятно, в файле `/var/log/messages` или в выдаче команды `dmesg(8)` вы увидите примерно следующее:

```
ad6 on monster1 suffered a hard error.
ad6: READ command timeout tag=0 serv=0 - resetting
ad6: trying fallback to PIO mode
ata3: resetting devices .. done
ad6: hard error reading fsbn 1116119 of 0-7 (ad6 bn 1116119; cn 1107 tn 4 sn 11)\\
status=59 error=40
ar0: WARNING - mirror lost
```

При помощи `atacontrol(8)` получите дополнительную информацию:

```
# atactrol list
ATA channel 0:
    Master:      no device present
    Slave:      acd0 <HL-DT-ST CD-ROM GCR-8520B/1.00> ATA/ATAPI rev 0

ATA channel 1:
    Master:      no device present
    Slave:      no device present

ATA channel 2:
    Master:      ad4 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
    Slave:      no device present

ATA channel 3:
    Master:      ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
    Slave:      no device present

# atactrol status ar0
ar0: ATA RAID1 subdisks: ad4 ad6 status: DEGRADED
```

1. Сначала вам нужно отключить диск от массива, чтобы его можно было без последствий извлечь:

```
# atactrol detach 3
```

2. Замените диск.

3. Повторно подключите диск в качестве резервного:

```
# atactrol attach 3
Master:  ad6 <MAXTOR 6L080J4/A93.0500> ATA/ATAPI rev 5
Slave:   no device present
```

4. Перестройте массив:

```
# atactrol rebuild ar0
```

5. Команда перестроения будет работать, пока не закончит. Однако имеется возможность открыть другой терминал (комбинацией клавиш **Alt+Fn**) и проверить состояние дел при помощи следующей команды:

```
# dmesg | tail -10
[выдача удалена]
ad6: removed from configuration
ad6: deleted from ar0 disk1
ad6: inserted into ar0 disk1 as spare

# atacontrol status ar0
ar0: ATA RAID1 subdisks: ad4 ad6 status: REBUILDING 0% completed
```

6. Дождитесь завершения этой операции.

## 16.5. USB устройства хранения

*Предоставил Marc Fonvieille.*

Множество современных устройств хранения используют Universal Serial Bus (USB): жесткие диски, брелоки USB, CD-R приводы, и т.д. FreeBSD предоставляет поддержку этих устройств.

### 16.5.1. Настройка

Драйвер `umass(4)` предоставляет поддержку устройств хранения USB. Если вы используете `GENERIC` ядро, изменять что-либо в настройках не потребуется. Если вы используете настроенное ядро, убедитесь, что в файле настройки присутствуют следующие строки:

```
device scbus
device da
device pass
device uhci
device ohci
device usb
device umass
```

Для доступа к устройствам хранения USB драйвер `umass(4)` использует подсистему SCSI, ваши устройства USB будут видны системе как SCSI устройства. В зависимости от чипсета USB на материнской плате, вам потребуется только один из параметров `device uhci` или `device ohci`. Однако, наличие обоих этих параметров не помешает. Не забудьте скомпилировать и установить новое ядро после добавления каких-либо строк.

**Замечание:** Если ваше USB устройство это пишущий привод CD-R или DVD, необходимо добавить в ядро SCSI CD-ROM драйвер, `cd(4)`, следующей строкой:

```
device cd
```

Поскольку устройство записи видно как SCSI диск, драйвер `atapi(4)` не должен использоваться в файле настройки.

Поддержка USB 2.0 контроллеров предоставляется в FreeBSD 5.X, и в ветви 4.X с FreeBSD 4.10-RELEASE. Добавьте:

```
device ehci
```

в файл настройки ядра для поддержки USB 2.0. Обратите внимание, что драйверы uhci(4) и ohci(4) все еще нужны, если необходима поддержка USB 1.X.

**Замечание:** В FreeBSD 4.X, необходимо запустить USB демона (usbd(8)), чтобы увидеть некоторые USB устройства. Для этого добавьте `usbd_enable="YES"` в файл `/etc/rc.conf` и перезагрузите компьютер.

## 16.5.2. Тестирование конфигурации

Конфигурация готова к тестированию, подключите устройство USB, и в буфере системных сообщений (dmesg(8)), диск должен отобразиться примерно так:

```
umass0: USB Solid state disk, rev 1.10/1.00, addr 2
GEOM: create disk da0 dp=0xc2d74850
da0 at umass-sim0 bus 0 target 0 lun 0
da0: <Generic Traveling Disk 1.11> Removable Direct Access SCSI-2 device
da0: 1.000MB/s transfers
da0: 126MB (258048 512 byte sectors: 64H 32S/T 126C)
```

Конечно, производитель, имя устройства (da0) и другие детали могут отличаться в зависимости от конфигурации.

Поскольку устройство USB видится как SCSI, команда `camcontrol` может быть использована для вывода списка устройств хранения USB, подключенных к системе:

```
# camcontrol devlist
<Generic Traveling Disk 1.11>          at scbus0 target 0 lun 0 (da0,pass0)
```

Если на диске есть файловая система, у вас должна быть возможность смонтировать ее. Разд. 16.3 поможет вам создать и отформатировать разделы на диске USB если потребуется.

Если вы отключите устройство (диск должен быть сначала размонтирован), вы должны увидеть в буфере системных сообщений что-то подобное:

```
umass0: at uhub0 port 1 (addr 2) disconnected
(da0:umass-sim0:0:0:0): lost device
(da0:umass-sim0:0:0:0): removing device entry
GEOM: destroy disk da0 dp=0xc2d74850
umass0: detached
```

## 16.5.3. Дополнительная информация

Помимо разделов Добавление дисков и Монтирование и размонтирование файловых систем, также может быть полезно чтение различных страниц справочника: `umass(4)`, `camcontrol(8)`, и `usbdevs(8)`.

## 16.6. Запись и использование оптических носителей (CD)

*Текст предоставил Mike Meyer.*

### 16.6.1. Введение

Компакт-диски (CD) имеют несколько особенностей, отличающих их от обычных дисков. Во-первых, на них невозможно производить запись. Они спроектированы с расчетом на то, что их можно читать последовательно без задержек на перемещение головки между дорожками. К тому же их гораздо проще переносить от системы к системе, чем носители близкого объема.

У CD имеются дорожки, но они представляют собой последовательность данных, читаемую последовательно, и не являются физической характеристикой диска. Для записи CD во FreeBSD вы готовите файлы данных, которые будут формировать дорожки на компакт-диске, а затем записываете дорожки на CD.

Файловая система ISO 9660 была разработана с учетом этих отличий. К сожалению, она унаследовала ограничения файловых систем, которые были тогда. К счастью, она дает механизм расширений, которые позволяют правильно записанным дискам обходить эти ограничения и при этом продолжать работать с системами, которые не поддерживают эти расширения.

Для создания файла данных, содержащего файловую систему ISO 9660, используется программа `mkisofs(8)`, которая включена в порт `sysutils/cdrtools`. Она имеет опции, поддерживающие различные расширения, и описана ниже.

Какой инструмент использовать для записи CD, зависит от того, является ли ваше устройство для записи CD устройством ATAPI или каким-либо другим. С устройствами для записи стандарта ATAPI используется программа `burncd`, которая является частью комплекта поставки системы. С устройствами SCSI и USB нужно использовать `cdrecord` из порта `sysutils/cdrtools`.

`burncd` поддерживает не все устройства. Для определения того, поддерживается ли устройство, посмотрите список поддерживаемых приводов CD-R/RW (<http://www.freebsd.dk/ata/>).

**Замечание:** Если вы используете FreeBSD 5.X, FreeBSD 4.8-RELEASE или более новые версии, то при работе с ATAPI-оборудованием через модуль ATAPI/CAM можно использовать утилиту `cdrecord` и другие инструменты для SCSI-приводов.

Если для записи CD вам нужна программа с графическим интерфейсом пользователя, взгляните на **X-CD-Roast** или **K3b**. Они доступны в виде пакетов или из портов `sysutils/xcdroast` и `sysutils/k3b`. Программам **X-CD-Roast** и **K3b** для работы с оборудованием ATAPI требуется модуль ATAPI/CAM.

### 16.6.2. mkisofs

Программа `mkisofs(8)`, поставляемая с портом `sysutils/cdrtools` создаёт файловую систему ISO 9660, которая является образом дерева каталогов в пространстве имён файловой системы UNIX. В самом простом случае она используется так:

```
# mkisofs -o imagefile.iso /path/to/tree
```

Эта команда создаст файл *imagefile.iso*, содержащий файловую систему ISO 9660, которая является копией дерева каталогов */path/to/tree*. Во время работы она будет преобразовывать имена файлов в имена, которые удовлетворяют ограничениям файловой системы ISO 9660, и исключит файлы, которые носят имена, неподходящие для файловой системы ISO.

Для того, чтобы обойти эти ограничения, имеется несколько опций. В частности, `-R` включает использование расширений Rock Ridge, распространенных в UNIX-системах, с `-J` будут применены расширения Joliet, используемые в системах от Microsoft, а `-hfs` может использоваться для создания файловых систем HFS, используемых в Mac OS.

Для CD, которые будут использоваться только с системами FreeBSD, может использоваться опция `-U`, отменяющая все ограничения на имена файлов. При использовании с опцией `-R` генерируется образ файловой системы, идентичный начальному дереву FreeBSD, хотя при этом стандарт ISO 9660 может нарушаться в нескольких местах.

Последней часто используемой опцией является `-b`. Она используется для указания загрузочного образа для использования при создании загрузочного CD в стандарте 'El Torito'. Этой опции указывается аргумент, который является маршрутом к загрузочному образу из корня дерева, записываемого на CD. По умолчанию, `mkisofs(8)` создает образ ISO в так называемом режиме "эмуляции флоппи-диска", и потому ожидает загрузочный образ размера строго 1200, 1440 или 2880 КБ. Некоторые загрузчики, в том числе и тот, что используется на дистрибутивных дисках FreeBSD, не используют режим эмуляции; в этом случае должна использоваться опция `-no-emul-boot`. Так что, если */tmp/myboot* содержит загрузочную систему FreeBSD с загрузочным образом в */tmp/myboot/boot/cdboot*, вы можете создать образ файловой системы ISO 9660 в */tmp/bootable.iso* следующим образом:

```
# mkisofs -R -no-emul-boot -b boot/cdboot -o /tmp/bootable.iso /tmp/myboot
```

Сделав это, и имея в ядре отконфигурированное устройство `vn` (для FreeBSD 4.X) или `md` (при использовании FreeBSD 5.X), вы можете смонтировать файловую систему, выполнив:

```
# vnconfig -e vn0c /tmp/bootable.iso
# mount -t cd9660 /dev/vn0c /mnt
```

в случае использования FreeBSD 4.X, а для FreeBSD 5.X:

```
# mdconfig -a -t vnode -f /tmp/bootable.iso -u 0
# mount -t cd9660 /dev/md0 /mnt
```

В этот момент вы можете проверить, что */mnt* и */tmp/myboot* идентичны.

Имеется много других опций, которые можно использовать с программой `mkisofs(8)` для тонкой настройки её поведения. В частности: модификации в размещении ISO 9660 и создание дисков в форматах Joliet и HFS. Обратитесь к справочным страницам по `mkisofs(8)` для получения более подробной информации.

### 16.6.3. burncd

Если ваше устройство для записи CD соответствует стандарту ATAPI, то для записи ISO-образа на компакт-диск вы можете воспользоваться командой `burncd`. `burncd` входит в базовый комплект

операционной системы и установлена как `/usr/sbin/burncd`. Использовать её очень просто, так как параметров у неё немного:

```
# burncd -f cddevice data imagefile.iso fixate
```

По этой команде файл `imagefile.iso` будет скопирован на `cddevice`. По умолчанию используется устройство `/dev/acd0` (или `/dev/acd0c` в FreeBSD 4.X). Для получения информации о параметрах, задающих скорость записи, выброс диска после записи и запись звуковых данных, обратитесь к `burncd(8)`.

#### 16.6.4. cdrecord

Если ваше устройство для записи CD не соответствует стандарту ATAPI, то для записи компакт-дисков вам нужно пользоваться программой `cdrecord`. `cdrecord` не входит в комплект поставки системы; вы должны установить её из порта `sysutils/cdrtools` или из соответствующего пакета. Изменения в системе могут приводить к тому, что откомпилированные версии этой программы работать не будут, или приводить к порче дисков. Поэтому вы должны при обновлении системы либо обновить порт, либо, если вы следуете -STABLE, обновить порт при появлении его новой версии.

Хотя `cdrecord` имеет много опций, в основном использовать её ещё проще, чем `burncd`. Запись образа ISO 9660 делается такой командой:

```
# cdrecord dev=device imagefile.iso
```

Тонким моментом при использовании `cdrecord` является определение правильного устройства `dev`. Чтобы задать параметр правильно, воспользуйтесь флагом `-scanbus` команды `cdrecord`, в результате чего может получиться примерно такой результат:

```
# cdrecord -scanbus
Cdrecord 1.9 (i386-unknown-freebsd4.2) Copyright (C) 1995-2000 Jörg Schilling
Using libscg version 'schily-0.1'
scsibus0:
0,0,0      0) 'SEAGATE ' 'ST39236LW      ' '0004' Disk
0,1,0      1) 'SEAGATE ' 'ST39173W      ' '5958' Disk
0,2,0      2) *
0,3,0      3) 'iomega ' 'jaz 1GB        ' 'J.86' Removable Disk
0,4,0      4) 'NEC      ' 'CD-ROM DRIVE:466' '1.26' Removable CD-ROM
0,5,0      5) *
0,6,0      6) *
0,7,0      7) *
scsibus1:
1,0,0     100) *
1,1,0     101) *
1,2,0     102) *
1,3,0     103) *
1,4,0     104) *
1,5,0     105) 'YAMAHA ' 'CRW4260      ' '1.0q' Removable CD-ROM
1,6,0     106) 'ARTEC   ' 'AM12S        ' '1.06' Scanner
1,7,0     107) *
```

Здесь приведены соответствующие значения параметров `dev` для имеющихся устройств. Найдите здесь ваше устройство для записи CD, а в качестве параметров для `dev` задавайте три числа через запятые. В нашем случае CRW-устройству соответствуют числа 1,5,0, так что правильным параметром будет `dev=1, 5, 0`. Имеется более простой способ задать эти значения; обратитесь к справочной информации о `cdrecord(1)` для выяснения подробностей. Там же находится информация о записи звуковых дорожек, управлении скоростью и другим вещам.

### 16.6.5. Копирование аудио CD

Вы можете копировать музыкальные CD, извлекая данные аудио с CD в набор файлов, а затем записывая эти файлы на чистый CD. Процесс несколько различен в случаях использования устройств ATAPI и SCSI.

#### Устройства SCSI

1. Используйте `cdda2wav` для извлечения данных аудио.

```
% cdda2wav -v255 -D2,0 -B -Owav
```

2. Воспользуйтесь `cdrecord` для записи файлов `.wav`.

```
% cdrecord -v dev=2,0 -dao -useinfo *.wav
```

Значение, соответствующее `2, 0`, должно быть установлено правильно, как это описано в Разд. 16.6.4.

#### Устройства ATAPI

1. Драйвер устройств ATAPI CD делает каждую дорожку доступной как `/dev/acd0t $n$` , где  $d$  является номером привода, а  $nn$  соответствует номеру дорожки, который записывается двумя десятичными цифрами с нулём в начале, если это нужно. Таким образом, первая дорожка на первом диске будет носить имя `/dev/acd0t01`, вторая будет именоваться `/dev/acd0t02`, третья будет носить имя `/dev/acd0t03` и так далее.

Удостоверьтесь, что соответствующий файл имеется в каталоге `/dev`.

```
# cd /dev
# sh MAKEDEV acd0t99
```

**Замечание:** Во FreeBSD 5.0 `devfs(5)` будет автоматически создавать объекты в каталоге `/dev` и управлять ими, так что использовать `MAKEDEV` не обязательно.

2. Извлеките каждую дорожку при помощи команды `dd(1)`. При извлечении файлов вы должны также использовать специфическое значение для размера блока.

```
# dd if=/dev/acd0t01 of=track1.cdr bs=2352
# dd if=/dev/acd0t02 of=track2.cdr bs=2352
...
```

3. Запишите извлечённые файлы на диск при помощи утилиты `burncd`. Вы должны указать, что это файлы с аудио, и что `burncd` должна зафиксировать диск по окончании работы.

```
# burncd -f /dev/acd0 audio track1.cdr track2.cdr ... fixate
```

### 16.6.6. Копирование компакт-дисков с данными

Вы можете скопировать CD с данными в файл образа, который функционально эквивалентен файлу образа, созданному командой `mkisofs(8)`, и вы можете использовать его для копирования любого CD с данными. В приводимом здесь примере предполагается, что ваш привод CDRом называется `acd0`. Подставьте название вашего привода CDRом. В FreeBSD 4.X к имени устройства должен быть добавлен символ `s` для указания на то, что берётся весь раздел, а в случае CDRом, весь диск.

```
# dd if=/dev/acd0 of=file.iso bs=2048
```

Теперь, когда вы имеете образ, вы можете записать его на CD так, как это описано выше.

### 16.6.7. Использование компакт-диски с данными

Теперь, после того, как вы создали стандартный CDRом с данными, вы, наверное, захотите смонтировать его и считать с него данные. По умолчанию `mount(8)` предполагает, что файловая система имеет тип `ufs`. Если вы попытаетесь выполнить что-то вроде:

```
# mount /dev/cd0 /mnt
```

вы получите сообщение `Incorrect super block`, и диск не смонтируется. CDRом не является файловой системой UFS, поэтому попытки смонтировать его таким образом будут терпеть неудачу. Вам просто нужно указать команде `mount(8)`, что файловая система имеет тип `ISO9660`, и всё должно заработать. Сделайте это, задав параметр `-t cd9660` при вызове `mount(8)`. К примеру, если вы хотите смонтировать устройство CDRом, `/dev/cd0`, в каталог `/mnt`, вы должны выполнить:

```
# mount -t cd9660 /dev/cd0 /mnt
```

Заметьте, что имя вашего устройства (`/dev/cd0` в этом примере) может быть другим, в зависимости от интерфейса, используемого в CDRом. Кроме того, параметр `-t cd9660` всего лишь задаёт выполнение утилиты `mount_cd9660(8)`. Пример выше может быть упрощён до:

```
# mount_cd9660 /dev/cd0c /mnt
```

Таким способом, вообще говоря, вы можете использовать компакт-диски любого производителя. Диски с некоторыми расширениями ISO 9660 могут, однако, работать со странностями. К примеру диски Joliet хранят все имена файлов в виде последовательностей двухбайтовых символов Unicode. Ядро FreeBSD (пока ещё) не может работать с Unicode, поэтому не английские символы выводятся в виде знаков вопроса. (Если вы работаете с FreeBSD 4.3 или более поздней версией, то в драйвере CD9660 имеется механизм для динамической загрузки соответствующей таблицы преобразования Unicode. Модули для некоторых распространённых кодировок могут быть получены из порта `sysutils/cd9660_unicode`.)

Время от времени вы можете получать сообщения `Device not configured` при попытке смонтировать CDRом. Это обычно означает, что привод CDRом полагает, что в нём нет диска, или что привод не виден на шине. Приводу CDRом может понадобиться несколько секунд, чтобы понять, что он был закрыт, так что будьте терпеливы.



Иногда SCSI CDROM может потеряться из-за того, что у него не было достаточно времени, чтобы ответить на сброс шины. Если у вас имеется SCSI CDROM, то, пожалуйста, добавьте следующий параметр в конфигурацию вашего ядра и перестройте его.

```
options SCSI_DELAY=15000
```

Это укажет вашей шине SCSI выдерживать 15-секундную паузу во время загрузки, чтобы дать вашему приводу CDROM шанс ответить на сброс шины.

### 16.6.8. Запись необработанных данных на компакт-диски

Вы можете предпочесть запись файла непосредственно на CD без создания файловой системы ISO 9660. Некоторые поступают так при создании резервных копий. Это выполняется гораздо быстрее, чем запись стандартного компакт-диска:

```
# burncd -f /dev/acd1 -s 12 data archive.tar.gz fixate
```

Для извлечения данных, записанных так на компакт-диск, вы должны считывать данные из файла непосредственного доступа к устройству:

```
# tar xzvf /dev/acd1
```

Вы не можете монтировать этот диск как обычный CDROM. Такой компакт-диск не может быть прочитан ни в какой другой операционной системе, кроме FreeBSD. Если вы хотите монтировать CD или обменяться данными с другой операционной системой, то вы должны использовать mkisofs(8) так, как это было описано выше.

### 16.6.9. Использование драйвера ATAPI/CAM

*Предоставил Marc Fonvieille.*

Этот драйвер позволяет работать с ATAPI-устройствами (приводы CD-ROM, CD-RW, DVD и так далее) через подсистему SCSI, таким образом расширяя использование таких приложений, как sysutils/cdrdao или cdrecord(1).

Для использования этого драйвера вам необходимо добавить в файл конфигурации ядра следующие строки:

```
device atapicam
device scbus
device cd
device pass
```

Кроме того, в файле конфигурации ядра должна быть следующая строка:

```
device ata
```

которая уже должна там присутствовать.

После этого перестройте и установите ваше новое ядро, выполните перезагрузку машины. В процессе загрузки ваш пишущий привод должен появиться примерно следующим образом:

```
acd0: CD-RW <MATSHITA CD-RW/DVD-ROM UJDA740> at atal-master PIO4
cd0 at atal bus 0 target 0 lun 0
cd0: <MATSHITA CDRW/DVD UJDA740 1.00> Removable CD-ROM SCSI-0 device
cd0: 16.000MB/s transfers
cd0: Attempt to query device size failed: NOT READY, Medium not present - tray closed
```

Теперь с ним можно работать через устройство `/dev/cd0`, например, чтобы смонтировать CD-ROM в каталог `/mnt`, просто наберите следующую команду:

```
# mount -t cd9660 /dev/cd0 /mnt
```

Для получения SCSI-адреса пишущего привода, вы можете, работая как пользователь `root`, запустить такую команду:

```
# camcontrol devlist
<MATSHITA CDRW/DVD UJDA740 1.00> at scbus1 target 0 lun 0 (pass0,cd0)
```

Таким образом, `1,0,0` будет SCSI-адресом для использования с `cdrecord(1)` и другими приложениями для работы со SCSI.

Для получения дополнительной информации об ATAPI/CAM и системе SCSI, обратитесь к страницам справочной системы по `atapicam(4)` и `cam(4)`.

## 16.7. Создание и использование оптических носителей (DVD)

*Предоставил Marc Fonvieille. Дополнения предоставил Andy Polyakov.*

### 16.7.1. Введение

DVD это следующее после CD поколение оптических носителей. DVD может вмещать больше данных чем любой CD и является современным стандартом распространения видео.

Для записываемых DVD существует пять физических форматов записи:

- DVD-R: Был первым форматом записываемых DVD. Стандарт DVD-R был создан DVD Forum (<http://www.dvdforum.com/forum.shtml>). Это формат для однократной записи.
- DVD-RW: Это перезаписываемая версия стандарта DVD-R. Носители DVD-RW могут быть перезаписаны около 1000 раз.
- DVD-RAM: Это также перезаписываемый формат, поддерживаемый DVD Forum. DVD-RAM может быть виден как съемный жесткий диск. Однако, этот носитель не совместим с большинством приводов DVD-ROM и проигрывателями DVD-Video; лишь несколько пишущих DVD поддерживают формат DVD-RAM.
- DVD+RW: Это перезаписываемый формат, созданный DVD+RW Alliance (<http://www.dvdrw.com/>). Носитель DVD+RW может быть перезаписан около 1000 раз.
- DVD+R: Этот формат — однократно записываемая версия формата DVD+RW.

Однослойный записываемый DVD может хранить до 4,700,000,000 байт, что равно 4.38 Гб, или 4485 Мб (1 килобайт это 1024 байт).

**Замечание:** Необходимо различать физический носитель и приложение. Например, DVD-Video это определенная файловая раскладка, которая может быть помещена на записываемый DVD любого физического формата: DVD-R, DVD+R, DVD-RW и т.д. Перед выбором типа носителя вы должны убедиться, что и устройство записи и DVD-Video проигрыватель (отдельный или DVD-ROM привод компьютера) совместимы с данным носителем.

## 16.7.2. Настройка

Для записи DVD будет использоваться программа **growisofs(1)**. Эта команда входит в набор утилит **dvd+rw-tools** (`sysutils/dvd+rw-tools`), который поддерживает все типы носителей DVD.

Эти утилиты используют подсистему **SCSI** для доступа к устройствам, следовательно необходимо добавить в ядро поддержку **ATAPI/CAM**. Если пишущий привод использует **USB** интерфейс, это добавление бесполезно и необходимо прочесть более подробную информацию по настройке устройств **USB** в Разд. 16.5

Вам также потребуется включить **DMA** доступ для устройств **ATAPI**, это можно сделать добавив в `/boot/loader.conf` следующую строку:

```
hw.ata.atapi_dma="1"
```

Перед использованием **dvd+rw-tools** вы должны свериться со списком совместимого оборудования **dvd+rw-tools** (<http://fy.chalmers.se/~appro/linux/DVD+RW/hcn.html>) с информацией по устройствам для записи DVD.

**Замечание:** Если вам нужен графический интерфейс пользователя, взгляните на **K3b** (`sysutils/k3b`), который предоставляет дружелюбный пользователю интерфейс к **growisofs(1)** и многим другим программам записи.

## 16.7.3. Запись DVD с данными

Команда **growisofs(1)** является оболочкой для **mkisofs**, она вызовет **mkisofs(8)** для создания файловой системы и запишет DVD. Это означает, что вам не потребуется создавать образ с данными перед началом процесса записи.

Для записи данных из каталога `/path/to/data` на DVD+R или DVD-R, используйте следующую команду:

```
# growisofs -dvd-compat -Z /dev/cd0 -J -R /path/to/data
```

Параметры `-J -R` передаются **mkisofs(8)** для создания файловой системы (в данном случае: файловая система **ISO 9660** с расширениями **Joliet** и **Rock Ridge**), обратитесь к странице справочника **mkisofs(8)** за более подробной информацией.

Параметр `-Z` используется для первой сессии записи в любом случае: для одной или нескольких сессий. Устройство DVD, `/dev/cd0`, должно быть изменено в соответствии с имеющимися настройками.

Параметр `-dvd-compat` закрывает диск и дозапись станет невозможна. Это должно улучшить совместимость с приводами DVD-ROM.

Возможна также запись предварительного (pre-mastered) образа, например, для записи `imagefile.iso` запустим:

```
# growisofs -dvd-compat -Z /dev/cd0=imagefile.iso
```

Скорость записи должна быть определена и автоматически установлена в соответствии с носителем и приводом. Если вы хотите явно указать скорость записи, используйте параметр `-speed=`. За дальнейшей информацией обратитесь к странице справочника `growisofs(1)`.

#### 16.7.4. Запись DVD-Video

DVD-Video это особая файловая система, базирующаяся на ISO 9660 и спецификациях micro-UDF (M-UDF). DVD-Video также представляет определенную иерархию структуры данных, поэтому для создания DVD потребуется особая программа, такая как `multimedia/dvdauthor`.

Если у вас уже есть образ файловой системы DVD-Video, просто запишите его как любой другой образ, примеры находятся в предыдущем разделе. Если вы создали DVD и результат находится в каталоге `/path/to/video`, для записи DVD-Video должна быть использована следующая команда:

```
# growisofs -Z /dev/cd0 -dvd-video /path/to/video
```

Параметр `-dvd-video` будет передан `mkisofs(8)` и укажет создать файловую систему DVD-Video. Помимо этого, параметр `-dvd-video` подразумевает параметр `growisofs(1) -dvd-compat`.

#### 16.7.5. Использование DVD+RW

В отличие от CD-RW, новый DVD+RW необходимо отформатировать перед первым использованием. Программа `growisofs(1)` позаботится об этом сама при необходимости, и это *рекомендованный* способ. Тем не менее, для форматирования DVD+RW вы можете использовать команду `dvd+rw-format`:

```
# dvd+rw-format /dev/cd0
```

Эту операцию необходимо выполнить лишь однажды, помните, что только новые носители DVD+RW необходимо форматировать. Затем запишите DVD+RW тем способом, который описан в предыдущем разделе.

Если вы хотите записать новые данные (полностью новую файловую систему, а не дописать данные) на DVD+RW, его не нужно очищать, просто запишите поверх предыдущей записи (создав новую начальную сессию) примерно так :

```
# growisofs -Z /dev/cd0 -J -R /path/to/newdata
```

Формат DVD+RW делает возможным легко дописать данные к предыдущей записи. Операция состоит в присоединении предыдущей сессии к существующей, это не мультисессионная запись, `growisofs(1)` *расширит* (grow) файловую систему ISO 9660, существующую на носителе.

Например, для дозаписи данных к предыдущей сессии на DVD+RW, используется следующая команда:

```
# growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

При последующих записях mkisofs(8) необходимо передавать те же параметры, что и при первой записи.

**Замечание:** Вы можете использовать параметр `-dvd-compat` для улучшения совместимости с приводами DVD-ROM. В случае DVD+RW это не мешает добавлению данных.

Если по какой-либо причине вам потребуется очистить носитель, используйте следующую команду:

```
# growisofs -Z /dev/cd0=/dev/zero
```

### 16.7.6. Использование DVD-RW

Существует два формата дисков DVD-RW: последовательно дополняемый и с ограниченной перезаписью. По умолчанию формат дисков DVD-RW последовательный.

Новый DVD-RW может быть записан непосредственно без необходимости форматирования, однако DVD-RW с данными в последовательном формате необходимо очистить перед созданием новой начальной сессии.

Для очистки DVD-RW в последовательном формате, запустите:

```
# dvd+rw-format -blank=full /dev/cd0
```

**Замечание:** Полная очистка (`-blank=full`) займет около одного часа на скорости 1x. Быструю очистку можно выполнить с параметром `-blank`, если DVD-RW будет записан в режиме Disk-At-Once (DAO). Для записи DVD-RW в режиме DAO, используйте команду:

```
# growisofs -use-the-force-luke=dao -Z /dev/cd0=imagefile.iso
```

Параметр `-use-the-force-luke=dao` не должен потребоваться, поскольку `growisofs(1)` попытается определить был ли носитель быстро очищен и включить DAO запись.

Фактически, лучше использовать режим с ограниченной перезаписью с любым DVD-RW, этот формат более гибкий, чем формат по умолчанию с последовательной записью.

Для записи данных на последовательный DVD-RW, используйте ту же команду, что и для других форматов DVD:

```
# growisofs -Z /dev/cd0 -J -R /path/to/data
```

Если вы хотите добавить данные к предыдущей записи, используйте параметр `growisofs(1)` `-m`. Однако при добавлении данных на DVD-RW в последовательном режиме, на диске будет создана новая сессия и в результате получится мультисессионный диск.

В формате DVD-RW с ограниченной перезаписью не требуется очищать носитель перед созданием новой начальной сессии, вам всего лишь нужно переписать диск с параметром `-Z`, подобно DVD+RW. Возможно также увеличение существующей файловой системы ISO 9660, записанной на диск тем же способом, как для DVD+RW с параметром `-m`. В результате получится односессионный DVD.

Для перевода DVD-RW в формат с ограниченной перезаписью, необходимо использовать следующую команду:

```
# dvd+rw-format /dev/cd0
```

Для перевода обратно в последовательный формат, выполните:

```
# dvd+rw-format -blank=full /dev/cd0
```

### 16.7.7. Мультисессия

Лишь несколько DVD-ROM и проигрывателей поддерживают мультисессионные DVD, в основном они в лучшем случае прочтут только первую сессию. DVD+R, DVD-R и DVD-RW в последовательном формате могут работать с несколькими сессиями, и это не относится к форматам DVD+RW и DVD-RW в формате ограниченной перезаписи.

Использование следующей команды после первой (не закрытой) сессии для DVD+R, DVD-R, или DVD-RW в последовательном формате, добавит на диск новую сессию:

```
# growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

Использование этой командной строки с DVD+RW или DVD-RW в режиме ограниченной перезаписи добавит данные, объединив новую сессию с предыдущей. В результате получится односессионный диск. Такой способ используется для добавления данных после первой записи на эти носители.

**Замечание:** Некоторый объем носителя используется между сессиями для завершения и начала сессии. Следовательно, для оптимизации объема хранения сессии должны быть большими. Количество сессий ограничено 154 для DVD+R, около 2000 для DVD-R и 127 для DVD+R Double Layer.

### 16.7.8. Дополнительная информация

Для получения дополнительной информации о DVD, можно запустить команду `dvd+rw-medainfo /dev/cd0`, диск должен находиться в приводе.

Дополнительная информация о **dvd+rw-tools** может быть найдена на странице справочника `growisofs(1)`, на веб сайте `dvd+rw-tools` (<http://fy.chalmers.se/~appro/linux/DVD+RW/>) и в архивах списка рассылки `cdwrite` (<http://lists.debian.org/cdwrite/>).

**Замечание:** Вывод `dvd+rw-medainfo` при записи или проблемный носитель необходимы для любого сообщения о проблеме. Без этого вывода будет совершенно невозможно помочь вам.

## 16.8. Дискеты

*Первоначальный текст предоставил Julio Merino. Перепечатал Martin Karlsson.*

Хранение данных на дискетах иногда бывает полезным, например, когда нет других съёмных носителей или когда необходимо перенести небольшой объём данных на другой компьютер.

В этом разделе будет описано, как использовать дискеты во FreeBSD. В основном речь пойдёт о форматировании и использовании дискет DOS размером 3.5 дюйма, однако общие принципы применимы и для других форматов гибких дисков.

### 16.8.1. Форматирование дискет

#### 16.8.1.1. Устройство

Доступ к гибким дискам, как, впрочем, и к остальным устройствам, осуществляется через соответствующие файлы в каталоге `/dev`. Чтобы обратиться к дискете при использовании релизов 4.X и ранее, необходимо работать с `/dev/fdN`, где *N* обозначает номер привода, обычно 0, или `/dev/fdNX`, где *X* обозначает букву.

В 5.0 и более новых релизах просто используйте `/dev/fdN`.

##### 16.8.1.1.1. Размер диска в 4.X и более ранних релизах

Имеются также устройства `/dev/fdN.size`, где *size* обозначает размер дискеты в килобайтах. Эти файлы устройств используются во время низкоуровневого форматирования для задания размера устройства. В последующих примерах будет использоваться размер в 1440kB.

Иногда записи в каталоге `/dev` необходимо создавать повторно. Для этого выполните следующее:

```
# cd /dev && ./MAKEDEV "fd*"
```

##### 16.8.1.1.2. Размер диска в 5.0 и последующих релизах

В 5.0 `devfs(5)` управляет файлами устройств в каталоге `/dev` в автоматическом режиме, так что использование `MAKEDEV` необязательно.

Требуемый размер диска передаётся утилите `fdformat(1)` при помощи параметра `-f`. Поддерживаемые размеры перечислены в `fdcontrol(8)`, но, по нашему мнению, лучше всего работает 1440kB.

#### 16.8.1.2. Форматирование

Перед тем, как дискетой можно будет воспользоваться, её необходимо отформатировать на низком уровне. Обычно это выполняется производителем, однако форматирование является хорошим способом проверить целостность носителя. Большинство гибких дисков предназначены для использования с размером 1440kB, однако возможно задать меньший или больший размер.

Для низкоуровневого форматирования дискет вам нужно использовать `fdformat(1)`. В качестве параметра этой утилите передаётся имя устройства.

Обратите внимание на появление сообщений об ошибках, так как они могут помочь определить, хорошая это дискета или плохая.

#### 16.8.1.2.1. Форматирование в 4.X и более ранних релизах

Для форматирования дискет используйте устройства `/dev/fdN.size`. Вставьте новую 3.5-дюймовую дискету в дисковод и введите команду:

```
# /usr/sbin/fdformat /dev/fd0.1440
```

#### 16.8.1.2.2. Форматирование в 5.0 и более новых релизах

Для форматирования гибких дисков используйте устройства `/dev/fdN`. Вставьте новую 3.5-дюймовую дискету в дисковод и введите команду:

```
# /usr/sbin/fdformat -f 1440 /dev/fd0
```

### 16.8.2. Метка диска

После низкоуровневого форматирования диска вам нужно поместить на него метку диска. Эта метка будет потом разрушена, но она будет нужна системе для определения размера диска и его характеристик.

Новая метка диска будет касаться диска в целом, и будет содержать полную информацию о параметрах дискеты. Значения геометрии для метки диска перечислены в файле `/etc/disktab`.

Теперь вы можете запустить `disklabel(8)` примерно так:

```
# /sbin/disklabel -B -r -w /dev/fd0 fd1440
```

**Замечание:** Начиная с FreeBSD 5.1-RELEASE, на смену старой программе `disklabel(8)` пришла утилита `bsdlabel(8)`. У `bsdlabel(8)` отсутствуют некоторые устаревшие опции и параметры; в примере выше параметр `-r` не может использоваться с `bsdlabel(8)`. Для получения дополнительной информации обратитесь к справочной странице по `bsdlabel(8)`.

### 16.8.3. Файловая система

Теперь ваша дискета готова к высокоуровневому форматированию. При этом на неё будет помещаться новая файловая система, которая позволит FreeBSD читать и записывать информацию на диск. После создания новой файловой системы метка диска уничтожается, так что если вы захотите переформатировать диск, вам придётся создавать метку диска повторно.

Файловой системой для дискеты может служить UFS или FAT. Вообще говоря, FAT для дискет подходит лучше.

Для размещения на дискете новой файловой системы, выполните:



```
# /sbin/newfs_msdos /dev/fd0
```

Теперь диск готов к работе.

#### 16.8.4. Использование дискет

Для работы с гибким диском смонтируйте его при помощи утилит `mount_msdos(8)` (для 4.X и более ранних релизов) или `mount_msdosfs(8)` (в 5.0 и последующих релизах). Можно также использовать пакет `emulators/mtools` из коллекции портов.

### 16.9. Создание и использование архивных копий на магнитной ленте

К наиболее часто используемым носителям на магнитной ленте следует отнести ленты шириной 4мм и 8мм, а также типа QIC, мини-картриджи и DLT.

#### 16.9.1. 4мм (DDS: Digital Data Storage)

Ленты шириной 4мм заменяют QIC в качестве наиболее предпочтительного носителя для создания резервных копий. Эта тенденция значительно усилилась после покупки компанией Соннег фирмы Archive, ведущего производителя накопителей QIC и последующего прекращения их выпуска. Накопители 4мм малы по размеру и мало шумят, но у них нет репутации носителя, обладающего надежностью приводов 8мм. Картриджи более дешевы и меньше по размеру (3 x 2 x 0.5 дюймов, 76 x 51 x 12 мм), чем 8мм-картриджи. Накопители для лент шириной 4мм, как и 8мм, имеют сравнительно малый срок службы головок, по причине использования в обоих случаях технологии спирального сканирования (helical scan).

Пропускная способность у таких накопителей начинается с цифры ~150 kB/s, пиковая достигает ~500 kB/s. Ёмкость накопителей начинается с 1.3 GB и может достигать 2.0 GB. Аппаратное сжатие, имеющееся на большинстве таких накопителей, даёт увеличение ёмкости примерно вдвое. Блоки многоприводных ленточных библиотек могут иметь до 6 накопителей в одном модуле с автоматической сменой ленты. Ёмкость библиотек может достигать 240 GB.

Стандарт DDS-3 в настоящее время поддерживает ёмкости лент вплоть до 12 ГБ (или 24 ГБ сжатой информации).

В накопителях 4мм, как и в приводах 8мм, используется технология спирального сканирования. Все плюсы и минусы этой технологии относятся как к 4мм, так и 8мм приводам.

Не следует использовать ленты после того, как они были подвергнуты 2000 проходов, или были использованы для создания 100 полных копий.

#### 16.9.2. 8мм (Exabyte)

Ленты шириной 8мм являются самым распространённым типом для ленточных SCSI-накопителей; они же являются наиболее удачным выбором при выборе типа носителей для обмена лентами. Наверное, каждый сервер имеет привод Exabyte шириной 8мм и объёмом 2 ГБ. Эти приводы удобны, они

работают надёжно и тихо. Картриджи дешевы и малы по размеру (4.8 x 3.3 x 0.6 дюймов; 122 x 84 x 15 мм). Одним минусом лент шириной 8мм является сравнительно малое время службы головок и лент из-за высокой скорости движения ленты вдоль головок.

Скорость передачи данных варьируется от ~250 kB/s до ~500 kB/s. Объём хранимых данных начинается с 300 МБ и может достигать 7 ГБ. Аппаратное сжатие, имеющееся практически на всех таких приводах, увеличивает ёмкость примерно вдвое. Эти приводы существуют как в виде отдельных модулей, так и в виде многоприводных ленточных библиотек с 6 приводами и 120 лентами в одном отсеке. Ленты сменяются автоматически модулем. Ёмкости библиотек достигают величин, превышающих 840 ГБ.

Модель Exabyte "Mammoth" поддерживает ёмкость ленты в 12 ГБ (24 ГБ со сжатием) и стоит примерно вдвое больше, чем обычный ленточный накопитель.

Данные на ленту записываются по технологии спирального сканирования, головки позиционируются под углом к носителю (примерно в 6 градусов). Лента оборачивается на 270 градусов вокруг шпульки, которая держит головки. Во время скольжения ленты вокруг шпульки последняя вращается. В результате достигается высокая плотность записи данных с очень близко лежащими дорожками, расположенными под наклоном по всей ленте.

### 16.9.3. QIC

Ленты и накопители формата QIC-150, наверное, являются наиболее распространенным типом носителей. Приводы лент формата QIC являются самыми дешёвыми "серьёзными" накопителями для резервного копирования. Минусом является стоимость носителей. Ленты формата QIC по сравнению с лентами шириной 8мм или 4мм являются дорогими, превосходя их по стоимости хранения одного гигабайта в пять раз. Однако если вам будут достаточно половины ленты, QIC может оказаться правильным выбором. QIC является *самым* распространенным типом привода. Каждый сайт имеет привод QIC какой-либо ёмкости. QIC имеет большое количество плотностей на физически похожих (иногда даже идентичных) лентах. Приводы QIC работают вовсе не тихо. Эти накопители громко осуществляют поиск перед тем, как начать запись данных и достаточно шумны в процессе чтения, записи или поиска. Ленты QIC имеют размеры (6 x 4 x 0.7 дюймов; 15.2 x 10.2 x 1.7 мм).

Скорость обмена данными лежит в границах от ~150 kB/s до ~500 kB/s. Ёмкость накопителей варьируется от 40 МБ до 15 ГБ. Аппаратное сжатие присутствует во многих современных накопителях QIC. Приводы QIC устанавливаются менее часто; они вытесняются накопителями DAT.

На ленту данные записываются в виде дорожек. Дорожки располагаются в длину вдоль всей ленты. Количество дорожек, и, в свою очередь, их ширина, меняется вместе с ёмкостью ленты. Большинство, если не все современные накопители обеспечивают обратную совместимость по крайней мере для чтения (однако зачастую и для режима записи). Формат QIC имеет хорошую репутацию в области надёжности хранения данных (механика устроена проще и более надёжна, чем в случае накопителей, построенных по технологии спирального сканирования).

Ленты не следует больше использовать после создания 5,000 резервных копий.

### 16.9.4. DLT

Формат DLT обладает самой высокой скоростью передачи данных среди всех перечисленных здесь накопителей. Лента шириной 1/2" (12.5мм) помещена в один картридж с катушкой (4 x 4 x 1 дюймов;

100 x 100 x 25 мм). Вдоль одной из сторон картриджа расположена сдвигающаяся крышечка. Механизм накопителя открывает эту крышку, чтобы вытащить конец ленты. На этом конце имеется овальное отверстие, которое используется для "захвата" ленты. Принимающая катушка размещена внутри накопителя. Все другие типы картриджей, перечисленные здесь (за исключением 9-дорожечных лент), имеют как подающий, так и принимающий барабаны внутри самого картриджа.

Скорость передачи данных равна примерно 1.5 MB/s, что в три раза больше скорости передачи данных для накопителей 4мм, 8мм или QIC. Ёмкость картриджей варьируется от 10 ГБ до 20 ГБ для одного накопителя. Приводы могут компоноваться как многоленточные роботизированные, так и многоленточные, многоприводные библиотеки лент, вмещающие от 5 до 900 лент и от 1 до 20 приводов, что даёт ёмкость хранилища от 50 ГБ до 9 ТБ.

Формат DLT Type IV поддерживает ёмкость до 70 ГБ со сжатием.

Данные на ленту записываются в виде дорожек, параллельных направлению движения (точно также, как и для лент QIC). Одновременно записываются две дорожки. Срок жизни головок чтения/записи сравнительно велик; как только лента перестаёт двигаться, одновременно прекращается трение между головками и лентой.

### 16.9.5. AIT

AIT - это новый формат фирмы Sony, который позволяет хранить до 50 ГБ (со сжатием) информации на одной ленте. Ленты содержат микросхемы памяти, на которых размещается каталог содержимого ленты. Этот каталог может быть быстро считан накопителем для определения расположения файлов на ленте, вместо того, чтобы тратить несколько минут на поиск, как это происходит с другими форматами. Такое программное обеспечение, как **SAMS:Alexandria**, может управлять сорока или большим количеством ленточных библиотек AIT, связываясь непосредственно с памятью лент для вывода их содержимого, определения того, какие файлы были скопированы на какую ленту, выбора нужной ленты, её загрузки и восстановления данных с ленты.

Библиотеки с такими функциями стоят в районе \$20,000, вывода их из ниши любительского рынка.

### 16.9.6. Использование новой ленты первый раз

Если вы попытаетесь прочитать или записать новую, абсолютно чистую ленту, в первый раз, то вам это не удастся. Выводимые на консоль сообщения будут выглядеть примерно так:

```
sa0(ncr1:4:0): NOT READY asc:4,1
sa0(ncr1:4:0): Logical unit is in process of becoming ready
```

На ленте отсутствует идентификационный блок (блок номер 0). Со времен принятия стандарта QIC-525 все накопители формата QIC записывают на ленту идентификационный блок (Identifier Block). Здесь имеется два решения:

- По команде `mt fsf 1` ленточный накопитель записывает идентификационный блок на ленту.
- Воспользуйтесь кнопкой на передней панели для выброса ленты.

Вставьте ленту повторно и по команде `dump` сбросьте данные на ленту.

Программа `dump` выдаст `DUMP: End of tape detected`, а на консоли будет выведено: `HARDWARE FAILURE info:280 asc:80,96`.

перематывайте ленту такой командой: `mt rewind`.

Последующие операции с лентой будут успешными.

## 16.10. Создание резервных копий на дискетах

### 16.10.1. Можно ли использовать дискеты для создания резервных копий моих данных?

На самом деле дискеты не подходят для создания резервных копий, потому что:

- Носитель ненадёжен, особенно если речь идет о больших сроках хранения.
- Создание резервных копий и восстановление данных происходит очень медленно.
- Дискеты имеют весьма ограниченную емкость (дни, когда весь винчестер копировался на десяток или около того дискет, давно прошли).

Несмотря на все это, если у вас нет другого способа сделать резервную копию ваших данных, то дискеты все же лучше, чем ничего.

Если вы используете дискеты, то проверьте, что они должны быть хорошего качества. Дискеты, которые валялись по всему офису в течении нескольких лет, не подойдут. Идеально использовать новые от известного производителя.

### 16.10.2. Итак, как же сделать резервную копию данных на дискетах?

Самым лучшим методом создания резервной копии на дискете является использование утилиты `tar(1)` с опцией `-m` (многотомные архивы), которая позволяет размещать архивы на нескольких дискетах.

Для копирования всех файлов в текущем каталоге и подкаталогах выполните следующее (работая как пользователь `root`):

```
# tar Mcvf /dev/fd0 *
```

Когда первая дискета окажется полностью заполненной, программа `tar(1)` выдаст запрос на следующий том (так как работа утилиты `tar(1)` не зависит от носителя, она имеет дело с томами; здесь это означает дискету).

Prepare volume #2 for /dev/fd0 and hit return:

Это сообщение будет повторяться (со все увеличивающимся номером тома) до тех пор, пока все указанные файлы не будут заархивированы.

### 16.10.3. Можно ли резервные копии подвергнуть компрессии?

К сожалению, tar(1) при создании многотомных архивов не позволяет использовать опцию -z. Вы конечно же, можете скомпрессировать все файлы утилитой gzip(1), программой gzip(1) скопировать их на дискеты, а затем распаковать файлы снова утилитой gunzip(1)!

### 16.10.4. Как восстановить данные из моих резервных копий?

Для полного восстановления архива воспользуйтесь такой командой:

```
# tar Mxvf /dev/fd0
```

Есть два подхода к восстановлению только нужных вам файлов. В первом вы можете начать с первой дискеты и выдать такую команду:

```
# tar Mxvf /dev/fd0 filename
```

Программа tar(1) будет выдавать запрос на подачу последующих дискет до тех пор, пока не найдет требуемый файл.

Как альтернатива, если вы знаете, на какой дискете расположен файл, то вы можете просто подать ее и дать ту же самую команду, что и выше. Заметьте, что если первый файл на дискете является продолжением предыдущего, то tar(1) выдаст предупреждение о том, что не может его восстановить, хотя вы этого и не просили делать!

## 16.11. Основы технологии резервного копирования

Тремя основными программами резервного копирования являются dump(8), tar(1) и cpio(1).

### 16.11.1. Dump и Restore

Для UNIX традиционными программами резервного копирования являются dump и restore. Они работают с приводом как с набором дисковых блоков, которые расположены ниже понятий файлов, связей и каталогов, создаваемых файловыми системами. Программа dump выполняет резервное копирование всей файловой системы, располагающейся на устройстве. Невозможно выполнить резервное копирование части файловой системы или дерева каталогов, которые располагаются более чем в одной файловой системе. Утилита dump не записывает на ленту файлы и каталоги, она записывает блоки данных, из которых строятся файлы и каталоги.

**Замечание:** Если вы используете программу dump для работы с корневым каталогом, при этом не будет выполняться резервное копирование /home, /usr и многих других каталогов, так как они обычно являются точками монтирования других файловых систем или символическими ссылками на эти файловые системы.

В программе dump имеются некоторые неудобства, оставшиеся от её ранних дней в составе Version 6 операционной системы AT&T UNIX (примерно 1975). Параметры, используемые по умолчанию, подходят для 9-дорожечных лент (6250 bpi), но не для современных носителей с высокой плотностью

записи информации (до 62,182 fpi). Для использования ёмкостей нынешних накопителей на магнитной ленте эти параметры могут быть заданы в командной строке.

При помощи `rdump` и `rrestore` возможно резервное копирование данных по сети на накопитель, подключенный к другому компьютеру. Обе программы используют в работе `rcmd(3)` и `ruserok(3)` для доступа к накопителю на магнитной ленте на удалённом компьютере. Поэтому пользователь, выполняющий резервное копирование, должен быть указан в файле `.rhosts` на удалённом компьютере. Аргументы для `rdump` и `rrestore` должны подходить для использования на другом компьютере. При выполнении копирования по команде `rdump` на компьютере с FreeBSD на накопитель Exabyte, подключенный к машине Sun по имени `komodo`, используйте такую команду:

```
# /sbin/rdump 0dsbfu 54000 13000 126 komodo:/dev/nsa8 /dev/da0a 2>&1
```

Будьте осторожны: есть проблемы с обеспечением безопасности при аутентификации посредством `.rhosts`. Внимательно рассмотрите вашу ситуацию.

Программы `dump` и `restore` можно использовать в более защищённом режиме посредством `ssh`.

### Пример 16-1. Использование `dump` через `ssh`

```
# /sbin/dump -0uan -f - /usr | gzip -2 | ssh -c blowfish \
targetuser@targetmachine.example.com dd of=/mybigfiles/dump-usr-10.gz
```

Либо воспользуйтесь встроенной в `dump` возможностью, задав переменную окружения `RSN`:

### Пример 16-2. Использование `dump` при работе через `ssh` с заданием `RSN`

```
# RSN=/usr/bin/ssh /sbin/dump -0uan -f targetuser@targetmachine.example.com:/dev/sa0
```

## 16.11.2. tar

Утилита `tar(1)` также восходит корнями к Version 6 системы AT&T UNIX (около 1975). `tar` работает с файловой системой; `tar` записывает на ленту файлы и каталоги. `tar` поддерживает не полный набор опций, имеющих в `cpio(1)`, однако он не требует необычного перенаправления в командной строке, которое используется в утилите `cpio`.

В большинстве версий `tar` создание резервных копий по сети не поддерживается. Версия GNU утилиты `tar`, которая используется во FreeBSD, поддерживает удалённые устройства в том же самом синтаксисе, что и `rdump`. Чтобы скопировать данные на накопитель Exabyte, подключенный к машине Sun по имени `komodo`, используйте такую команду:

```
# /usr/bin/tar cf komodo:/dev/nsa8 . 2>&1
```

В случае использования версий без поддержки удалённых устройств, вы можете воспользоваться перенаправлением вывода и командой `rsh` для отправки данных на удалённый ленточный накопитель.

```
# tar cf - . | rsh hostname dd of=tape-device obs=20b
```

Если вы беспокоитесь о безопасности создания резервных копий по сети, то вместо `rsh` вам нужно использовать `ssh`.

### 16.11.3. cpio

`cpio(1)` является оригинальной программой UNIX для обмена файлами на магнитных носителях. В утилите `cpio` имеются опции (кроме всего прочего), позволяющие выполнять изменение порядка следования байтов, поддерживающие различные форматы архивов и выполняющие перенаправление данных другим программам. Последняя возможность делает `cpio` прекрасным выбором для целей установки. `cpio` не знает о том, как работать с каталогами, список файлов должен даваться через `stdin`.

`cpio` не поддерживает создание резервных копий по сети. Вы можете воспользоваться перенаправлением вывода и программой `rsh` для отправки данных на удалённый накопитель.

```
# for f in directory_list; do
find $f >> backup.list
done
# cpio -v -o --format=newc < backup.list | ssh user@host "cat > backup_device"
```

Где `directory_list` это список директорий, с которых Вы хотите создать резервные копии, `user@host` это комбинация пользователь/хост которая описывает того кто занимается резервированием, и `backup_device` это устройство куда копии должны быть записаны (например, `/dev/nsa0`).

### 16.11.4. рах

`рах(1)` является ответом IEEE/POSIX на утилиты `tar` и `cpio`. В течение многих лет различные версии программ `tar` и `cpio` получались не совсем совместимыми. Так что вместо того, чтобы попытаться полностью их стандартизировать, POSIX создал новую утилиту для работы с архивами. `рах` пытается читать и писать различные форматы `cpio` и `tar`, и, кроме того, свои собственные новые форматы. Набор команд этой утилиты больше напоминает `cpio`, чем `tar`.

### 16.11.5. Amanda

**Amanda** (Advanced Maryland Network Disk Archiver) является целой клиент/серверной системой резервного копирования, а не отдельной программой. Сервер **Amanda** сможет осуществлять резервное копирование на единственный накопитель любого количества компьютеров, на которых имеется клиент **Amanda** и которые могут связываться по сети с сервером **Amanda**. Общей проблемой систем с большим количеством больших дисков является то, что время, требуемое для непосредственной записи данных на ленту, превышает лимит времени, выделенный на эту задачу. **Amanda** решает эту проблему. **Amanda** может использовать "промежуточный диск" для резервного копирования нескольких файловых систем одновременно. **Amanda** создаёт "наборы архивов": группа лент, используемых в некоторый период времени для создания полных копий всех файловых систем, перечисленных в конфигурационном файле системы **Amanda**. "Архивный набор" содержит также создаваемый каждую ночь инкрементальные (или дифференциальные) резервные копии всех файловых систем. Восстановление повреждённой файловой системы требует наличия самой последней полной копии и инкрементальных резервных копий.

Конфигурационный файл даёт прекрасный механизм для управления процессом резервного копирования и объёмом трафика, генерируемого системой **Amanda**. **Amanda** сможет использовать любую из перечисленных выше программ для записи данных на ленту. **Amanda** имеется в виде как порта, так и пакета, и по умолчанию она не установлена.

### 16.11.6. Не делать ничего

“Не делать ничего” - это не программа для компьютера, и в то же время это наиболее широко используемая стратегия резервного копирования. Здесь нет никаких первоначальных затрат. Здесь нет расписания, которому нужно следовать. Просто скажите нет. Если что-то случится с вашими данными, улыбнитесь и забудьте о них!

Если ваше время и данные практически ничего не стоят, то “не делать ничего” является самой подходящей программой для вашего компьютера. Но будьте осторожны, POSIX является весьма полезным инструментом, и через полгода вы можете обнаружить, что у вас есть набор файлов, представляющих для вас определенную ценность.

“Ничего не делать” является правильным методом резервного копирования для `/usr/obj` и других деревьев каталогов, которые могут быть в точности регенерированы вашим компьютером. Примером являются файлы, представляющие страницы этого Руководства в форматах HTML или PostScript. Они генерируются из входных файлов в формате SGML. Создавать резервные копии файлов в форматах HTML и PostScript не нужно. Исходные файлы в формате SGML копируются регулярно.

### 16.11.7. Какая программа резервного копирования самая лучшая?

dump(8) *Точка*. Elizabeth D. Zwicky протестировала все программы резервного копирования, обсуждаемые здесь. Беспроигрышным вариантом для сохранения всех ваших данных и особенностей файловых систем UNIX является dump. Элизабет создала файловые системы, содержащие большое количество необычных элементов (и некоторых не так уж необычных) и тестировала каждую из программ, выполняя резервное копирование и последующее восстановление этих файловых систем. В число необычных элементов входили: файлы с дырами, файлы с дырами и блоком пустого места, файлы с необычными символами в их именах, нечитаемые и незаписываемые файлы, устройства, меняющие свой размер во время резервного копирования, файлы, создаваемые и удаляемые во время копирования и тому подобное. Она представила результаты на конференции LISA V в октябре 1991 года. Посмотрите ссылку на сайте *torture-testing Backup and Archive Programs* (<http://berdmann.dyndns.org/zwicky/testdump.doc.html>).

### 16.11.8. Процедура восстановления при сбое

#### 16.11.8.1. До того, как случится катастрофа

Вам нужно выполнить всего лишь четыре шага для того, чтобы быть готовым к любому сбою.

Во-первых, распечатайте разметку диска для всех ваших дисков (к примеру, `disklabel da0 | lpr`), таблицу файловых систем (`/etc/fstab`) и все сообщения, выводимые при загрузке, каждого по два экземпляра.

Во-вторых, определите, все ли устройства присутствуют на загрузочной и аварийной дискетах (`boot.flp` и `fixit.flp`). Самым простым способом проверки является перезагрузка вашей машины с загрузочной дискетой, вставленной в дисковод и последующая проверка сообщений при загрузке. Если все имеющиеся у вас устройства здесь будут перечислены и будут работоспособны, перейдите к третьему шагу.

В противном случае вам необходимо будет создать две особым образом сформированные загрузочные дискеты, на которых помещено ядро, могущее смонтировать все ваши диски и получить доступ к



вашему стримеру. На этих дискетах должны быть: `fdisk`, `disklabel`, `newfs`, `mount` и какая-либо используемая вами программа резервного копирования. Эти программы должны быть скомпонованы статически. Если вы используете `dump`, то на дискете должна присутствовать и программа `restore`.

В-третьих, регулярно создавайте резервные копии на ленте. Любые изменения, которые вы делали после последнего резервного копирования, могут быть безвозвратно потеряны. На лентах включайте защиту от записи.

В-четвертых, проверяйте работу дискет (либо `boot.flp` и `fixit.flp`, либо двух дискет, которые вы сделали при выполнении второго шага) и лент с резервными копиями. Ведите журнал выполняемых действий. Храните эти записи вместе с загрузочной дискетой, распечатками и лентами. Вы просто обезумеете при восстановлении данных, если окажется, что записи могли бы избежать разрушения ваших резервных копий (Каким образом? Вместо команды `tar xvf /dev/sa0` вы могли случайно набрать `tar cvf /dev/sa0` и тем самым перезаписать вашу резервную копию).

Для дополнительной страховки, каждый раз создавайте загрузочные дискеты и две резервные копии на ленте. Храните одну из копий в каком-то удаленном месте и НЕ в том же здании, где находится ваш офис. Достаточно большое количество компаний во Всемирном Торговом Центре изучило это на своей шкуре. Это удаленное хранилище должно быть физически отделено на большое расстояние от ваших компьютеров и дисковых устройств.

### Пример 16-3. Скрипт для создания загрузочной дискеты

```
#!/bin/sh
#
# create a restore floppy
#
# format the floppy
#
PATH=/bin:/sbin:/usr/sbin:/usr/bin

fdformat -q fd0
if [ $? -ne 0 ]
then
    echo "Bad floppy, please use a new one"
    exit 1
fi

# place boot blocks on the floppy
#
disklabel -w -B /dev/fd0c fd1440

#
# newfs the one and only partition
#
newfs -t 2 -u 18 -l 1 -c 40 -i 5120 -m 5 -o space /dev/fd0a

#
# mount the new floppy
#
mount /dev/fd0a /mnt

#
```

```

# create required directories
#
mkdir /mnt/dev
mkdir /mnt/bin
mkdir /mnt/sbin
mkdir /mnt/etc
mkdir /mnt/root
mkdir /mnt/mnt # for the root partition
mkdir /mnt/tmp
mkdir /mnt/var

#
# populate the directories
#
if [ ! -x /sys/compile/MINI/kernel ]
then
    cat << EOM
The MINI kernel does not exist, please create one.
Here is an example config file:
#
# MINI -- A kernel to get FreeBSD onto a disk.
#
machine "i386"
cpu "I486_CPU"
ident MINI
maxusers 5

options  INET # needed for _tcp _icmpstat _ipstat
#        _udpstat _tcpstat _udb
options  FFS #Berkeley Fast File System
options  FAT_CURSOR #block cursor in syscons or pccons
options  SCSI_DELAY=15 #Be pessimistic about Joe SCSI device
options  NCONS=2    #1 virtual consoles
options  USERCONFIG #Allow user configuration with -c XXX

config kernel root on da0 swap on da0 and da1 dumps on da0

device isa0
device pci0

device fdc0 at isa? port "IO_FD1" bio irq 6 drq 2 vector fdintr
device fd0 at fdc0 drive 0

device ncr0

device scbus0

device sc0 at isa? port "IO_KBD" tty irq 1 vector scintr
device npx0 at isa? port "IO_NPX" irq 13 vector npxintr

device da0
device da1
device da2

```

```

device sa0

pseudo-device loop # required by INET
pseudo-device gzip # Exec gzipped a.out's
EOM
    exit 1
fi

cp -f /sys/compile/MINI/kernel /mnt

gzip -c -best /sbin/init > /mnt/sbin/init
gzip -c -best /sbin/fsck > /mnt/sbin/fsck
gzip -c -best /sbin/mount > /mnt/sbin/mount
gzip -c -best /sbin/halt > /mnt/sbin/halt
gzip -c -best /sbin/restore > /mnt/sbin/restore

gzip -c -best /bin/sh > /mnt/bin/sh
gzip -c -best /bin/sync > /mnt/bin/sync

cp /root/.profile /mnt/root

cp -f /dev/MAKEDEV /mnt/dev
chmod 755 /mnt/dev/MAKEDEV

chmod 500 /mnt/sbin/init
chmod 555 /mnt/sbin/fsck /mnt/sbin/mount /mnt/sbin/halt
chmod 555 /mnt/bin/sh /mnt/bin/sync
chmod 6555 /mnt/sbin/restore

#
# create the devices nodes
#
cd /mnt/dev
./MAKEDEV std
./MAKEDEV da0
./MAKEDEV da1
./MAKEDEV da2
./MAKEDEV sa0
./MAKEDEV pty0
cd /

#
# create minimum filesystem table
#
cat > /mnt/etc/fstab <<EOM
/dev/fd0a    /      ufs      rw  1  1
EOM

#
# create minimum passwd file
#
cat > /mnt/etc/passwd <<EOM

```

```

root:*:0:0:Charlie &:/root:/bin/sh
EOM

cat > /mnt/etc/master.passwd <<EOM
root::0:0::0:0:Charlie &:/root:/bin/sh
EOM

chmod 600 /mnt/etc/master.passwd
chmod 644 /mnt/etc/passwd
/usr/sbin/pwd_mkdb -d/mnt/etc /mnt/etc/master.passwd

#
# umount the floppy and inform the user
#
/sbin/umount /mnt
echo "The floppy has been unmounted and is now ready."

```

### 16.11.8.2. После сбоя

Главный вопрос: выжило ли ваше оборудование? Вы регулярно делали резервные копии, так что нет нужды беспокоиться о программном обеспечении.

Если оборудование было повреждено, должны быть заменены неисправные компоненты.

Если с оборудованием все в порядке, проверьте ваши дискеты. При использовании самостоятельно созданной загрузочной дискеты, загрузитесь в однопользовательском режиме (набрав `-s` в приглашении `boot:`). Пропустите следующий абзац.

Если вы используете дискеты `boot.flp` и `fixit.flp`, читайте дальше. Вставьте дискету `boot.flp` в первый дисковод и загрузите компьютер. На экран будет выведено оригинальное меню установки. Выберите пункт `Fixit--Repair mode with CDROM or floppy`. После вывода приглашения вставьте `fixit.flp`. `restore` и другие нужные вам программы находятся в `/mnt2/stand`.

Восстановите по отдельности каждую файловую систему.

Попробуйте выполнить команду `mount` (например, `mount /dev/da0a /mnt`) по отношению к корневому разделу вашего первого диска. Если метка диска была испорчена, то воспользуйтесь командой `disklabel` для переразбиения на разделы и разметки диска так, чтобы получившаяся метка совпала с той, которая вами была распечатана и сохранена. Для повторного создания файловых систем используйте утилиту `newfs`. Повторно смонтируйте корневой раздел дискеты в режиме чтения-записи (`mount -u -o rw /mnt`). Воспользуйтесь вашей программой резервного копирования и резервными копиями на лентах для восстановления данных для этой файловой системы (например, `restore vrf /dev/sa0`). Размонтируйте файловую систему (например, `umount /mnt`). Повторите эту процедуру для каждой файловой системы, которая была повреждена.

Как только ваша система заработает, сделайте резервную копию на новые ленты. Что бы ни вызвало сбой или потерю данных, это может случиться снова. Ещё один час, потраченный в этот момент, может спасти вас от неприятностей в будущем.

## 16.12. Сетевые файловые системы, файловые системы в памяти и с отображением в файл

*Реорганизацию и улучшения выполнил Marc Fonvieille.*

Кроме дисков, которые вы физически устанавливаете в ваш компьютер; дискеты, компакт-диски, винчестеры и так далее, FreeBSD воспринимает и другие типы дисков - *виртуальные диски*.

Сюда могут быть отнесены сетевые файловые системы, такие, как Network File System и Coda, а также файловые системы с организацией в памяти и создаваемые в файлах.

В зависимости от версии FreeBSD, которую вы используете, для создания и работы с файловыми системами, отображаемыми в оперативную память или файлы, вам нужно будет пользоваться разными инструментами.

**Замечание:** Пользователи FreeBSD 4.X для создания требуемых устройств должны использовать MAKEDEV(8). Во FreeBSD 5.0 и более поздних версиях для создания файлов устройств используется devfs(5), которая выполняет это прозрачно для пользователей.

### 16.12.1. Файловая система в файле во FreeBSD 4.X

Утилита `vnconfi g(8)` конфигурирует и позволяет использовать дисковые устройства на основе псевдо-устройств `vnnode`. `vnnode` представляет собой файл и отвечает за работу с файлом. Это означает, что `vnconfi g(8)` использует файлы для создания и работы с файловой системой. Одним из возможных способов использования является монтирование образов дисков или образов компакт-дисков, сброшенных в файлы.

Для использования `vnconfi g(8)` в конфигурационном файле ядра вам нужно включить поддержку `vn(4)`:

```
pseudo-device vn
```

Чтобы смонтировать имеющийся образ файловой системы:

**Пример 16-4. Использование `vnconfig` для монтирования имеющегося образа файловой системы во FreeBSD 4.X**

```
# vnconfig vn0 diskimage
# mount /dev/vn0c /mnt
```

Для создания нового образа файловой системы с помощью `vnconfi g(8)`:

**Пример 16-5. Создание нового диска в файле с помощью `vnconfig`**

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# vnconfig -s labels -c vn0 newimage
# disklabel -r -w vn0 auto
# newfs vn0c
Warning: 2048 sector(s) in last cylinder unallocated
/dev/vn0c: 10240 sectors in 3 cylinders of 1 tracks, 4096 sectors
```

```

5.0MB in 1 cyl groups (16 c/g, 32.00MB/g, 1280 i/g)
super-block backups (for fsck -b #) at:
 32
# mount /dev/vn0c /mnt
# df /mnt
Filesystem 1K-blocks   Used   Avail Capacity  Mounted on
/dev/vn0c  4927         1    4532      0%    /mnt

```

### 16.12.2. Файловые системы, отображаемые в файлы, во FreeBSD 5.X

Во FreeBSD 5.X для конфигурации и подключения дисков md(4), отображаемых в оперативную память, используется утилита mdconfi g(8). Для работы с mdconfi g(8) вам нужно подгрузить модуль md(4) или добавить поддержку этих устройств в файл конфигурации ядра:

```
device md
```

Утилита mdconfi g(8) поддерживает три типа виртуальных дисков, отображаемых в память: диски в памяти, которая выделяется запросами malloc(9) и диски в памяти, использующие в качестве устройств хранения файлы или раздел подкачки. Одним из возможных использований таких дисков является монтирование файлов с образами дискет или CD.

Для монтирования образа существующей файловой системы:

**Пример 16-6. Использование mdconfig для монтирования файла с образом существующей файловой системы во FreeBSD 5.X**

```

# mdconfig -a -t vnode -f diskimage -u 0
# mount /dev/md0 /mnt

```

Для создания образа новой файловой системы при помощи mdconfi g(8):

**Пример 16-7. Создание нового диска, отображаемого в файл, при помощи mdconfig**

```

# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# mdconfig -a -t vnode -f newimage -u 0
# disklabel -r -w md0 auto
# newfs md0c
/dev/md0c: 5.0MB (10240 sectors) block size 16384, fragment size 2048
using 4 cylinder groups of 1.27MB, 81 blks, 256 inodes.
super-block backups (for fsck -b #) at:
 32, 2624, 5216, 7808
# mount /dev/md0c /mnt
# df /mnt
Filesystem 1K-blocks   Used   Avail Capacity  Mounted on
/dev/md0c  4846         2    4458      0%    /mnt

```

Если в параметре -u вы не задали номер устройства, то mdconfi g(8) для выбора неиспользуемого устройства будет использовать функцию автоматического выделения в md(4). Имя выделенного устройства будет выдано на стандартное устройство вывода в виде, например, md4. Для получения

более полной информации о `mdconfi g(8)`, пожалуйста, обратитесь к соответствующей странице справочной системы.

**Замечание:** Начиная с FreeBSD 5.1-RELEASE, на смену старой программе `disklabel(8)` пришла утилита `bsdlabel(8)`. У `bsdlabel(8)` отсутствуют некоторые устаревшие опции и параметры; в примере выше параметр `-r` не может использоваться с `bsdlabel(8)`. Для получения дополнительной информации обратитесь к справочной странице по `bsdlabel(8)`.

Утилита `mdconfi g(8)` весьма полезна, однако для создания файла с файловой системой требуется произвести много действий. Вместе с FreeBSD 5.0 поставляется утилита под названием `mdmfs(8)`, которая создаёт диск `md(4)` при помощи `mdconfi g(8)`, размещает на нём файловую систему UFS при помощи `newfs(8)` и монтирует её командой `mount(8)`. Например, если вы хотите создать и смонтировать такой же образ файловой системе, как выше, просто наберите такую команду:

#### Пример 16-8. Настройка и монтирование диска, отображаемого в файл, при помощи команды `mdmfs`

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
# mdmfs -F newimage -s 5m md0 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity Mounted on
/dev/md0      4846      2  4458      0%  /mnt
```

Если вы используете параметр `md` без номера устройства, то `mdmfs(8)` будет использовать автоматическую нумерацию `md(4)` для автоматического выбора неиспользуемого устройства. Более полную информацию о `mdmfs(8)` можно найти на страницах справочной системы.

### 16.12.3. Файловая система в памяти во FreeBSD 4.X

Драйвер `md(4)` является простым и эффективным способом создания файловых систем в памяти во FreeBSD 4.X. Для выделения памяти используется `malloc(9)`.

Просто возьмите файловую систему, которую вы приготовили при помощи, скажем, `vnconfi g(8)` и:

#### Пример 16-9. Диск `md` в памяти во FreeBSD 4.X

```
# dd if=newimage of=/dev/md0
5120+0 records in
5120+0 records out
# mount /dev/md0c /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity Mounted on
/dev/md0c  4927        1   4532      0%  /mnt
```

Для получения более полной информации, пожалуйста, обратитесь к страницам справочной системы по `md(4)`.

### 16.12.4. Файловые системы с отображением в память во FreeBSD 5.X

При работе с файловыми системами, отображаемыми в файл или память, используются одни и те же утилиты: `mdconfi g(8)` или `mdmfs(8)`. Место для хранения файловых систем в памяти выделяется через `malloc(9)`.

#### Пример 16-10. Создание нового диска с отображением в память при помощи `mdconfig`

```
# mdconfig -a -t malloc -s 5m -u 1
# newfs -U md1
/dev/md1: 5.0MB (10240 sectors) block size 16384, fragment size 2048
using 4 cylinder groups of 1.27MB, 81 blks, 256 inodes.
with soft updates
super-block backups (for fsck -b #) at:
 32, 2624, 5216, 7808
# mount /dev/md1 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity Mounted on
/dev/md1      4846    2  4458    0%    /mnt
```

#### Пример 16-11. Создание нового диска с отображением в память при помощи `mdmfs`

```
# mdmfs -M -s 5m md2 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity Mounted on
/dev/md2      4846    2  4458    0%    /mnt
```

Вместо того, чтобы использовать файловую систему, опирающуюся на `malloc(9)`, возможно использовать память раздела подкачки, для чего нужно просто заменить `malloc` на `swap` в командной строке при вызове `mdconfi g(8)`. Утилита `mdmfs(8)` по умолчанию (без опции `-M`) создаёт диск в разделе подкачки. Для выяснения всех подробностей, пожалуйста, обратитесь к страницам справочной системы по `mdconfi g(8)` и `mdmfs(8)`.

### 16.12.5. Отключение диска, отображаемого в память, от системы

Если файловые системы, отображаемые в память или файл, больше не используются, вам нужно высвободить все ресурсы для системы. Первым делом нужно размонтировать файловую систему, затем воспользоваться `mdconfi g(8)` для отключения диска от системы и освободить ресурсы.

К примеру, чтобы отключить и освободить все ресурсы, используемые `/dev/md4`:

```
# mdconfig -d -u 4
```

Для выдачи информации об отконфигурированных устройствах `md(4)` используется команда `mdconfig -l`.

Во FreeBSD 4.X для отключения устройства используется команда `vnconfi g(8)`. Например, для отключения и освобождения всех ресурсов, используемых `/dev/vn4`:

```
# vnconfig -u vn4
```



## 16.13. Мгновенные копии файловых систем

*Текст предоставил Tom Rhodes.*

Во FreeBSD 5.0 вместе с технологией Отложенных обновлений представлена новая возможность: генерация мгновенных копий файловых систем.

Мгновенные копии позволяют пользователю создавать образы заданных файловых систем и работать с ними как с файлами. Файлы мгновенных копий должны создаваться в той файловой системе, над которой производится действие, и пользователь может создавать не более 20 мгновенных копий для каждой файловой системы. Активные копии записываются в суперблок, так что они остаются в силе между операциями монтирования и размонтирования в процессе системных перезагрузок. Если мгновенная копия больше не нужна, она может быть удалена стандартной командой `rm(1)`.

Мгновенные копии могут удаляться в любом порядке, однако всё использованное пространство не может быть использовано, так как другая мгновенная копия может претендовать на некоторые блоки из освобождённых.

В момент первоначального создания устанавливается флаг `schg` (обратитесь к страницам справочной системы по команде `chflags(1)`) для обеспечения того, что даже пользователь `root` не сможет произвести запись в мгновенную копию. Однако команда `unlink(1)` делает исключение для файлов мгновенных копий, позволяя их удалять при наличии установленного флага `schg`, так что нет необходимости снимать флаг `schg` перед удалением файла мгновенной копии.

Мгновенные копии создаются при помощи утилиты `mount(8)`. Чтобы создать мгновенную копию `/var` в файле `/var/snapshot/snap`, воспользуйтесь такой командой:

```
# mount -u -o snapshot /var/snapshot/snap /var
```

В качестве альтернативного средства создания мгновенных копий вы можете использовать утилиту `mksnap_ffs(8)`:

```
# mksnap_ffs /var /var/snapshot/snap
```

После создания мгновенной копии есть несколько способов её использования:

- Некоторые администраторы будут использовать файл мгновенной копии для целей создания резервной копии, так как мгновенная копия может быть перенесена на CD или магнитную ленту.
- Утилита проверки целостности файловой системы, `fsck(8)`, может быть запущена над мгновенной копией. Полагая, что файловая система была в порядке, когда она была смонтирована, вы всегда должны получать нормальный (и неизменный) результат. Это именно то, что выполняет фоновый процесс `fsck(8)`.
- Запустить утилиту `dump(8)` с мгновенной копией. Будет создаваться дамп, соответствующий файловой системе на момент создания мгновенной копии. Утилита `dump(8)` при использовании опции `-L` тоже может работать с мгновенными копиями, создавать их дампы, а затем удалять за один проход.
- Смонтировать командой `mount(8)` мгновенную копию как замороженный образ файловой системы. Чтобы смонтировать командой `mount(8)` мгновенную копию `/var/snapshot/snap`, запустите:

```
# mdconfig -a -t vnode -f /var/snapshot/snap -u 4
# mount -r /dev/md4 /mnt
```

Теперь вы можете пройти по иерархии вашей зафиксированной файловой системы `/var`, смонтированной в каталог `/mnt`. Всё будет в том же самом состоянии, в каком это было во время создания мгновенной копии. Единственным исключением будет то, что любые ранее сделанные мгновенные копии будут видны как файлы нулевой длины. Когда использование мгновенной копии закончено, она может быть удалена командой:

```
# umount /mnt
# mdconfig -d -u 4
```

Для получения более полной информации о `softupdates` и мгновенных копиях файловых систем, включая техническое описание, вы можете посетить сайт Маршалла Кёрка МакКузика (Marshall Kirk McKusick) по адресу <http://www.mckusick.com/>.

## 16.14. Квотирование файловых систем

Квоты - это опциональная возможность операционной системы, которая позволяет ограничивать объем дискового пространства и/или количество файлов для конкретного пользователя или членов определенной группы в рамках одной файловой системы. Чаще всего эта возможность используется в системах разделения времени, когда желательно ограничить количество ресурсов, которые может использовать один пользователь или группа пользователей. Это позволит не допустить ситуации, когда один пользователь или группа пользователей заполняют всё доступное дисковое пространство.

### 16.14.1. Настройка вашей системы на использование дисковых квот

Перед тем, как попытаться использовать дисковые квоты, необходимо убедиться, что квоты включены в ваше ядро. Это делается добавлением следующей строки в конфигурационный файл вашего ядра:

```
options QUOTA
```

В стандартном ядре `GENERIC` это по умолчанию не включено, так что для использования дисковых квот вам нужно будет настроить, откомпилировать и установить собственное ядро. Пожалуйста, обратитесь к Гл. 8 за дополнительной информацией о настройке ядра.

Затем вам потребуется включить квотирование дисков в файле `/etc/rc.conf`. Это делается добавлением такой строки:

```
enable_quotas="YES"
```

Для более полного контроля над запуском квотирования имеется дополнительная переменная для настройки. Как правило, при загрузке целостность квот каждой файловой системы проверяется программой `quotacheck(8)`. При работе программы `quotacheck(8)` проверяется точное соответствие данных в базе данных квот данным в файловой системе. Это весьма долгий процесс, что отражается на времени загрузки системы. Если вам захочется пропустить этот шаг, то для этого предназначена специальная переменная в файле `/etc/rc.conf`:

```
check_quotas="NO"
```

Если вы работаете с FreeBSD версий до 3.2-RELEASE, то настройка делается проще, и она состоит только из одной переменной. Задайте следующее в вашем файле `/etc/rc.conf`:

```
check_quotas="YES"
```

Наконец, вам потребуется отредактировать файл `/etc/fstab` для включения дисковых квот на уровне файловых систем. Это то место, где вы можете включить квоты для пользователей, для групп или для обеих этих категорий для всех ваших файловых систем.

Для включения пользовательских квот для файловой системы, добавьте параметр `userquota` в поле параметров файловой системы, на которой вы хотите включить квотирование, в файле `/etc/fstab`. Например:

```
/dev/dals2g    /home    ufs rw,userquota 1 2
```

Подобным же образом для включения квотирования на уровне групп, воспользуйтесь параметром `groupquota` вместо `userquota`. Чтобы включить квотирование как для пользователей, так и для групп, измените строчку следующим образом:

```
/dev/dals2g    /home    ufs rw,userquota,groupquota 1 2
```

По умолчанию файлы квот хранятся в корневом каталоге файловой системы в файлах с именами `quota.user` и `quota.group` соответственно для пользовательских и групповых квот. Для получения подробной информации обратитесь к команде `fstab(5)`. Хотя справочная страница по `fstab(5)` утверждает, что вы можете указать другое местоположение файлов с квотами, этого делать не рекомендуется, потому что различные утилиты для работы с квотами не могут нормально работать в такой ситуации.

На этом этапе вы должны перезагрузить вашу систему с новым ядром. Скрипт `/etc/rc` автоматически запустит соответствующие команды для создания начальных файлов для всех квот, которые вы создали в файле `/etc/fstab`, так что нет нужды вручную создавать никаких файлов квот нулевой длины.

При нормальной работе вам не потребуется вручную запускать программы `quotacheck(8)`, `quotaon(8)` или `quotaoff(8)`. Однако вам нужно хотя бы прочесть страницы справочника по этим командам, просто чтобы ознакомиться с их функциями.

## 16.14.2. Установка квот

Как только вы настроили вашу систему на использование квот, проверьте, что они действительно были задействованы. Простым способом сделать это является запуск такой команды:

```
# quota -v
```

Вы должны увидеть однострочную информацию, отражающую использование диска и текущие ограничения для каждой файловой системы, на которой включено квотирование.

Теперь вы действительно готовы задавать ограничения при помощи команды `edquota(8)`.

У вас есть несколько вариантов того, как приводить в действие ограничения по объему дискового пространства, который могут занимать пользователь или группа, а также по количеству файлов, которые они могут создать. Вы можете ограничивать размещение ресурсов на основе объема дискового пространства (квотирование блоков), количества файлов (квотирование inode) или их комбинации.

Каждое из этих ограничений, в свою очередь, делится на две категории: мягкие и жёсткие ограничения.

Жёсткое ограничение не может быть превышено. Как только пользователь достиг своих ограничений, ресурсы соответствующей файловой системы ему больше выделяться не будут. Например, если пользователь имеет жесткое ограничение в 500 блоков на файловой системе и в текущий момент он использует 490 блоков, то пользователь может получить дополнительно еще 10 блоков. Попытка получить еще 11 блоков окончится неудачно.

С другой стороны, мягкие ограничения могут быть превышены в течении некоторого периода времени. Этот период времени также называют периодом отсрочки, который по умолчанию равен одной неделе. Если пользователь превышает своё мягкое ограничение в течение периода времени, превышающего отсрочку, то это мягкое ограничение становится жестким и последующее выделение ресурсов будет запрещено. Когда пользователь вернётся обратно к отметке, меньшей, чем мягкое ограничение, то период отсрочки будет сброшен.

Далее приводится пример того, что вы можете наблюдать при запуске команды `edquota(8)`. Когда вызывается команда `edquota(8)`, вы оказываетесь в редакторе, заданном переменной окружения `EDITOR`, или в редакторе `vi`, если переменная `EDITOR` не задана, и можете редактировать квоты.

```
# edquota -u test
```

```
Quotas for user test:
/usr: blocks in use: 65, limits (soft = 50, hard = 75)
inodes in use: 7, limits (soft = 50, hard = 60)
/usr/var: blocks in use: 0, limits (soft = 50, hard = 75)
inodes in use: 0, limits (soft = 50, hard = 60)
```

Для каждой файловой системы, на которой включено квотирование, вы должны увидеть две строки. В одной строке приведены ограничения на блоки, а в другой на количество inode. Например, чтобы увеличить ограничения на количество блоков для пользователя с мягкого ограничения в 50 и жёсткого ограничения в 75, на мягкое ограничение в 500 и жёсткое ограничение в 600, измените:

```
/usr: blocks in use: 65, limits (soft = 50, hard = 75)
```

на:

```
/usr: blocks in use: 65, limits (soft = 500, hard = 600)
```

Новые ограничения вступят в силу после выхода из редактора.

Иногда желательно установить ограничения квот на некоторый диапазон UID (идентификаторов пользователей). Это можно сделать при помощи параметра `-p` в команде `edquota(8)`. Во-первых, установите желаемое ограничение для пользователя, а затем запустите команду `edquota -p protouser startuid-enduid`. Например, если пользователь `test` имеет желаемые ограничения, то для дублирования этих ограничений на пользователей с UID от 10000 до 19999 может быть использована такая команда:

```
# edquota -p test 10000-19999
```

Дополнительную информацию можно получить из справочной страницы по команде `edquota(8)`.

### 16.14.3. Проверка ограничений и использования диска

Для проверки квот и использования дисков вы можете использовать команды `quota(1)` или `repquota(8)`. Команда `quota(1)` может быть использована для проверки квот отдельных пользователей, групп, а также использования дисков. Пользователь может только проверить собственную квоту и квоту той группы, к которой он принадлежит. Только администратор системы может проверить квоты всех пользователей и групп. Команду `repquota(8)` можно использовать для получения суммарной статистики всех квот и использования дисков для файловых систем с включенными квотами.

Далее приведен пример вывода команды `quota -v` для пользователя, который имеет ограничения на двух файловых системах.

```
Disk quotas for user test (uid 1002):
  Filesystem  blocks   quota  limit  grace  files  quota  limit  grace
    /usr       65*     50     75   5days     7     50     60
    /usr/var    0      50     75      0     50     60
```

В этом примере для файловой системы `/usr` пользователь превысил свое мягкое ограничение в 50 блоков на 15 блоков и имеет 5 дней до истечения отсрочки. Отметьте знак звездочки `*`, который указывает на превышение пользователем своего ограничения.

Как правило, файловые системы, на которых пользователь не занимает дискового пространства, не показываются в выводе команды `quota(1)`, даже если ему выделена квота на этой файловой системе. При использовании параметра `-v` эти файловые системы выводятся, как, например, файловая система `/usr/var` в примере выше.

### 16.14.4. Квоты в NFS

Квоты определяются подсистемой квот на сервере NFS. Демон `rpc.rquotad(8)` предоставляет информацию о квотах для программы `quota(1)` на клиентах NFS, позволяя пользователям на этих машинах смотреть свою статистику о квотах.

Включите `rpc.rquotad` в файле `/etc/inetd.conf` следующим образом:

```
rquotad/1      dgram rpc/udp wait root /usr/libexec/rpc.rquotad rpc.rquotad
```

Теперь перезапустите `inetd`:

```
# kill -HUP `cat /var/run/inetd.pid`
```

## 16.15. Шифрование дисковых разделов

*Текст предоставил Lucky Green.*

FreeBSD предоставляет прекрасную возможность по защите от несанкционированного доступа к данным. Права на доступ к файлам и технология принудительного контроля доступа MAC (Mandatory

Access Control) (смотрите see Гл. 15) помогают предотвратить несанкционированный доступ посторонних лиц к данным, при условии работы операционной системы и компьютера. Однако права доступа, контролируемые операционной системой, не имеют значения, если нападающий получает физический доступ к компьютеру и может просто перенести жёсткий диск на другую машину для копирования и дальнейшего анализа важных данных.

Вне зависимости от того, как атакующий завладел жёстким диском или выключенным компьютером, технология **gbde** (GEOM Based Disk Encryption - шифрование диска на уровне GEOM) может защитить данные файловой системы компьютера даже против очень заинтересованной атакующей стороны с достаточными ресурсами. В отличие от громоздких систем шифрования, которые шифруют отдельные файлы, **gbde** шифрует в прозрачном режиме файловую систему в целом, при этом данные в открытом виде на диск никогда не записываются.

### 16.15.1. Включение **gbde** в ядре

1. Получите права пользователя `root`

Настройка **gbde** требует права доступа администратора системы.

```
% su -
Password:
```

2. Проверьте номер версии операционной системы

Для работы **gbde(4)** требуется FreeBSD 5.0 и выше.

```
# uname -r
5.0-RELEASE
```

3. Включите поддержку **gbde(4)** в конфигурационный файл ядра

При помощи вашего любимого текстового редактора добавьте следующую строку в файл конфигурации вашего ядра:

```
options GEOM_BDE
```

Выполните конфигурацию, компиляцию и установку нового ядра FreeBSD. Этот процесс описан в Гл. 8.

Перезагрузитесь, запустив новое ядро.

### 16.15.2. Подготовка зашифрованного жёсткого диска

В следующем примере предполагается, что в вашу систему вы добавляете новый винчестер, на котором будет располагаться единственный раздел с зашифрованными данными. Этот раздел будет монтироваться в каталог `/private`. **gbde** может также использоваться для шифрования `/home` и `/var/mail`, но это требует более сложной последовательности действий, что выходит за рамки этого вводного материала.

1. Подключите новый жёсткий диск

Установите новый диск в систему, как это описано в Разд. 16.3. В рамках этого примера раздел, соответствующий новому жёсткому диску, будет называться `/dev/ad4s1c`. Устройства

/dev/ad0s1\* представляют существующие стандартные разделы FreeBSD нашей тестовой системы.

```
# ls /dev/ad*
/dev/ad0      /dev/ad0s1b   /dev/ad0s1e   /dev/ad4s1
/dev/ad0s1    /dev/ad0s1c   /dev/ad0s1f   /dev/ad4s1c
/dev/ad0s1a   /dev/ad0s1d   /dev/ad4
```

## 2. Создайте каталог для размещения файлов блокировок GBDE

```
# mkdir /etc/gbde
```

Файл блокировки **gbde** содержит информацию, которая нужна **gbde** для доступа к зашифрованному разделу. Не имея доступа к файлу блокировки, **gbde** не сможет расшифровать данные, хранимые в зашифрованном разделе, без значительного ручного вмешательства, что программно не поддерживается. Каждый зашифрованный раздел использует отдельный файл блокировки.

## 3. Инициализируйте раздел gbde

Перед началом работы с разделом **gbde** его необходимо проинициализировать. Эта инициализация производится только один раз:

```
# gbde init /dev/ad4s1c -i -L /etc/gbde/ad4s1c
```

gbde(8) запустит редактор, что позволит вам задать в шаблоне различные конфигурационные параметры. При работе с файловыми системами UFS1 и UFS2 задайте значение sector\_size равным 2048:

```
$FreeBSD: src/sbin/gbde/template.txt,v 1.1 2002/10/20 11:16:13 phk Exp $
#
# Sector size is the smallest unit of data which can be read or written.
# Making it too small decreases performance and decreases available space.
# Making it too large may prevent filesystems from working. 512 is the
# minimum and always safe. For UFS, use the fragment size
#
sector_size      =          2048
[...]
```

gbde(8) дважды запросит ввод пароля, который будет использоваться для защиты данных. Пароль в обоих случаях должен вводиться одинаковый. Возможности **gbde** по защите ваших данных полностью зависят от качества выбранной вами ключевой фразы.<sup>1</sup>

По команде **gbde init** создаётся файл блокировок для вашего раздела **gbde**, который в нашем случае будет иметь имя **/etc/gbde/ad4s1c**.

**Предостережение:** Резервные копии файлов блокировок **gbde** должны храниться вместе с содержимым шифруемых разделов. Хотя удаление только блокировочного файла не сможет противостоять дешифрации атакующим раздела **gbde**, без этого файла даже легитимный пользователь не сможет получить доступ к данным без определённых и значительных усилий, что не поддерживается **gbde(8)** и его разработчиком.

## 4. Подключите зашифрованный раздел к системе

```
# gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c
```

Будет выдан запрос на ввод ключевой фразы, которую вы выбирали во время инициализации зашифрованного раздела. Новое защищённое устройство будет видно в каталоге `/dev` под названием `/dev/device_name.bde`:

```
# ls /dev/ad*
/dev/ad0          /dev/ad0s1b      /dev/ad0s1e      /dev/ad4s1
/dev/ad0s1        /dev/ad0s1c      /dev/ad0s1f      /dev/ad4s1c
/dev/ad0s1a       /dev/ad0s1d      /dev/ad4          /dev/ad4s1c.bde
```

## 5. Создайте файловую систему на зашифрованном устройстве

Как только защищённое устройство будет подключено к системе, вы сможете создать на нём файловую систему. Для этого используется утилита `newfs(8)`. Так как инициализация новой файловой системы UFS2 происходит быстрее, чем инициализация файловой системы старого формата UFS1, то рекомендуется использовать `newfs(8)` с параметром `-O2`.

**Замечание:** В FreeBSD 5.1-RELEASE и последующих релизах параметр `-O2` используется по умолчанию.

```
# newfs -U -O2 /dev/ad4s1c.bde
```

**Замечание:** Запуск команды `newfs(8)` должен выполняться над подключенном разделе **gbde**, который идентифицируется по расширению `*.bde` в имени устройства.

## 6. Смонтируйте зашифрованный раздел

Создайте точку монтирования для зашифрованной файловой системы.

```
# mkdir /private
```

Смонтируйте защищённую файловую систему.

```
# mount /dev/ad4s1c.bde /private
```

## 7. Проверьте доступность зашифрованной файловой системы

Защищённая файловая система теперь должна быть доступна утилите `df(1)` и доступной для использования.

```
% df -h
Filesystem      Size  Used Avail Capacity  Mounted on
/dev/ad0s1a     1037M   72M   883M     8%
/devfs           1.0K   1.0K    0B   100%    /dev
/dev/ad0s1f      8.1G   55K   7.5G     0%    /home
/dev/ad0s1e     1037M   1.1M   953M     0%    /tmp
/dev/ad0s1d      6.1G   1.9G   3.7G    35%    /usr
/dev/ad4s1c.bde  150G   4.1K  138G     0%    /private
```



### 16.15.3. Монтирование имеющихся зашифрованных файловых систем

После каждой загрузки для каждой защищённой файловой системы перед их использованием должны выполняться повторное подключение к системе, проверка на наличие ошибок и монтирование.

Требуемые для этого команды должны выполняться пользователем `root`.

1. Подключение `gbde`-раздела к системе

```
# gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c
```

Будет выдан запрос на ввод ключевой фразы, выбранной на этапе инициализации зашифрованного раздела `gbde`.

2. Проверка файловой системы на наличие ошибок

Так как защищаемая файловая система не может пока быть указана в файле `/etc/fstab` для автоматического монтирования, то она должны проверяться на наличие ошибок посредством ручного запуска `fsck(8)` до её монтирования.

```
# fsck -p -t ffs /dev/ad4s1c.bde
```

3. Монтирование зашифрованной файловой системы

```
# mount /dev/ad4s1c.bde /private
```

Теперь защищённая файловая система доступна для работы.

#### 16.15.3.1. Автоматическое монтирование зашифрованных разделов

Для автоматического подключения, проверки и монтирования зашифрованного раздела можно создать скрипт, но во соображениях безопасности в этом скрипте пароля для `gbde(8)` быть не должно. Поэтому рекомендуется запускать такие скрипты вручную, а пароль задавать с консоли или сеанса `ssh(1)`.

### 16.15.4. Криптографическая защита, применяемая в `gbde`

`gbde(8)` шифрует содержимое секторов при помощи 128-битного AES в режиме CBC. Каждый сектор диска шифруется различным ключом AES. Более полная информация о системе шифрования `gbde`, включая алгоритм генерации ключей для секторов из ключевой фразы, вводимой пользователем, можно найти на страницах справочной системы о `gbde(4)`.

### 16.15.5. Вопросы совместимости

`sysinstall(8)` несовместим с устройствами, зашифрованными `gbde`. Все устройства `*.bde` перед запуском `sysinstall(8)` должны быть отключены от системы, или эта утилита аварийно завершит работу на этапе обнаружения устройств. Для отключения защищённого устройства, используемого в нашем примере, воспользуйтесь такой командой:

```
# gbde detach /dev/ad4s1c
```

Также заметьте, что, так как `vinum(4)` работает не через подсистему `geom(4)`, то вы не можете использовать тома `vinum` с `gbde`.

## Примечания

1. Советы по выбору легкозапоминающихся ключевых фраз можно найти на сайте Diceware Passphrase (<http://world.std.com/~reinhold/diceware.html>).

# Глава 17. Менеджер дискового пространства **Vinum**

*Изначально написано Greg Lehey. Перевод на русский язык: Дмитрий Морозовский.*

## 17.1. Краткая аннотация

Какие бы диски у вас ни были, они всегда будут подвержены ограничениям:

- Слишком маленькие
- Слишком медленные
- Недостаточно надежные

Для того, чтобы обойти эти ограничения, можно использовать несколько (и, возможно, избыточное число) дисков.

В дополнение к поддержке разнообразных контроллеров RAID, базовая система FreeBSD включает Менеджер дискового пространства **Vinum** — драйвер, реализующий поддержку виртуальных дисков.

**Vinum** позволяет более гибко оперировать дисковым пространством, повысить производительность и надежность дисковой подсистемы за счет реализации моделей RAID-0, RAID-1 и RAID-5, а также их комбинаций.

В данной главе кратко рассматриваются потенциальные проблемы традиционной системы хранения данных и методы их решения при помощи **Vinum**.

## 17.2. Диски слишком малы

*Vinum* (произносится *Винум*, с ударением на первом слоге) — так называемый *Менеджер дисковых томов* — представляет собой виртуальный дисковый драйвер, призванный решить три вышеописанные проблемы. Взглянем на них более подробно. Предлагаются (и реализованы) следующие пути:

Объемы дисков растут, тем не менее, растут и требования к объемам систем хранения данных. Вы запросто можете оказаться в ситуации, когда требуемый объем файловой системы превышает размеры доступных дисков. Надо признать, что в настоящее время данная проблема стоит не так остро, как 10 лет назад, но тем не менее она существует. Некоторые системы выходят из этого тупика посредством создания мета-устройств, распределяющих хранящиеся данные по нескольким дискам.

## 17.3. Ограниченная пропускная способность

Современным системам часто необходим одновременный доступ ко многим данным. В частности, крупный FTP или HTTP-сервер может обслуживать тысячи одновременных соединений, поступающих по нескольким 100 Mbit/s каналам во внешний мир, что ощутимо превышает скорость передачи данных большинства дисков.

Современные диски могут передавать данные со скоростями до 70 MB/s; однако, эти цифры труднодостижимы в случае, когда к диску обращается большое число независимых процессов, каждый из которых может получить лишь часть этого значения. Интересным будет взглянуть на проблему с точки зрения дисковой подсистемы: важным параметром в нашем случае будет загрузка подсистемы фактом передачи фрагмента данных, а именно время, в течение которого диски, участвующие в передаче, будут заняты.

При любом запросе диск сначала должен спозиционировать головки, дожидаться подхода к головкам первого сектора из необходимых, и лишь затем выполнить обращение. Данная операция может рассматриваться как атомарная: нет никакого смысла ее прерывать.

Рассмотрим типичный запрос на передачу 10 kB информации. Современные высокопроизводительные диски подводят головки в нужную позицию в среднем за 3.5 миллисекунды. Самые быстрые диски вращаются со скоростью 15000 об/мин, так что среднее время на подход первого сектора к головке (rotational latency, половина времени одного оборота) составит еще 2 миллисекунды. При линейной скорости передачи данных в 70 MB/s собственно чтение/запись займет около 150 микросекунд — исчезающе мало по сравнению с временем позиционирования. В нашем случае, эффективная скорость передачи данных падает почти до 1 MB/s и, очевидно, сильно зависит от размера передаваемого блока.

Традиционным и очевидным решением этой проблемы является принцип "больше шпинделей": вместо использования одного большого диска можно применить несколько дисков меньшего размера. Диски позиционируют головки и передают данные независимо, так что эффективная пропускная способность возрастает примерно во столько раз, сколько дисков мы применяем.

Точная цифра, разумеется, будет несколько ниже: диски могут передавать данные параллельно, но у нас нет средства обеспечить строго равномерное распределение нагрузки по всем дискам. Нагрузка на один диск неизбежно будет больше чем на другой.

Равномерность распределения нагрузки на диски серьезно зависит от способа распределения по ним данных. В терминах дальнейшего обсуждения, будет удобно представить пространство хранения набором большого количества секторов с данными, которые адресуются по номеру, подобно страницам в книге. Наиболее очевидным методом будет поделить виртуальный диск на группы расположенных последовательно секторов размером с физический диск (которые будут подобны разделам книги). Этот метод называется *конкатенацией* или *сцеплением* (*concatenation*); его преимуществом является то, что он не налагает никаких ограничений на размеры применяемых дисков. Конкатенация эффективна, если нагрузка на дисковое пространство распределена равномерно. В случае концентрации нагрузки в малой области диска увеличение производительности не будет заметно. Организация секторов на сцепленных единицах хранения показана на Рис. 17-1.

**Рисунок 17-1. Организация сцепленных дисков**

Disk 1	Disk 2	Disk 3	Disk 4
0	6	10	12
1	7	11	13
2	8		14
3	9		15
4			16
5			17

Альтернативным подходом будет разделение адресного пространства на компоненты одного, сравнительно небольшого размера, и расположение их последовательно на разных устройствах. Например, первая группа из 256 секторов будет расположена на первом физическом диске, вторая — на следующем и т.д.  $n+1$ -я группа попадает на первый диск вслед за первой. Такое расположение называется *перемежающимся* (*striping*) или RAID-0.<sup>1</sup> Перемежение требует дополнительных усилий для нахождения нужного блока данных и может приводить к дополнительным нагрузкам на подсистемы ввода-вывода, если передаваемый блок пересекает границу stripe (тем самым попадая на разные диски), зато обеспечивает более равномерное распределение нагрузки по физическим дискам. Распределение блоков по физическим дискам в случае striping иллюстрируется Рис. 17-2.

**Рисунок 17-2. Организация с перемежением**

Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15
16	17	18	19
20	21	22	23

## 17.4. Целостность данных

Наконец, слабым местом современных дисков является их ограниченная надежность. Несмотря на то, что за последние несколько лет она ощутимо выросла, из всех компонентов сервера отказ дисков наиболее вероятен. Отказ может привести к катастрофическим результатам: замена отказавшего диска и восстановление данных может занять несколько дней.

Традиционным путем решения проблемы надежности является *зеркалирование (mirroring)*, обеспечивающее хранение всей информации в двух копиях на различных физических носителях. С момента изобретения аббревиатуры RAID эту технику также называют RAID уровня 1 или просто RAID-1. Любой запрос на запись в таком томе приводит к записи в оба подтома, чтение может производиться из любой половины, так что данные остаются доступны в случае отказа одного из дисков.

Зеркалирование имеет два слабых места:

- Цена: требуется вдвое больше дисков.
- Проблемы с производительностью: запись производится на оба диска, требуя вдвое большей пропускной способности шины. Производительность чтения не уменьшается, более того, часто увеличивается.

Альтернативным решением является хранение *контрольных сумм (четности)*, реализованное в RAID уровней 2, 3, 4 и 5. Наиболее интересен RAID-5. В реализации Vinum, это вариант организации тома с перемежением, при котором один из блоков в страйпе выделяется для хранения четности остальных  $n-1$  блоков. Как требует спецификация RAID-5, положение блока четности меняется от страйпа к страйпу.

**Рисунок 17-3. Организация RAID-5**

Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	Parity
3	4	Parity	5
6	Parity	7	8
Parity	9	10	11
12	13	14	Parity
15	16	Parity	17

По сравнению с зеркалированием преимуществом RAID-5 является гораздо меньшее требование к объему дисков. Скорость чтения сравнима с чтением в случае томов с перемежением, а вот запись происходит ощутимо медленнее (примерно вчетверо медленнее чтения). При отказе одного из дисков

массив продолжает работать в "деградировавшем" режиме: запросы на чтение с оставшихся дисков производятся обычным образом, а блоки с отказавшего диска перевычисляются из данных остальных блоков страйпа.

## 17.5. Объекты Vinum

Для обеспечения необходимой функциональности Vinum использует четырехуровневую иерархию объектов:

- "Видимая снаружи" сущность — виртуальный диск, называемый *томом* (*volume*). Тома в основном аналогичны дискам UNIX, хотя имеются и мелкие различия. На тома нет ограничений по размеру.
- Тома образуются из *наборов* (*plex*), каждый из которых представляет полное адресное пространство тома. Данный уровень иерархии, таким образом, реализует избыточность. Наборы являются аналогами отдельных дисков в зеркалированном массиве; содержимое наборов идентично.
- Поскольку Vinum работает в среде подсистемы хранения данных UNIX, многодисковые наборы можно было бы реализовать на базе дисковых разделов UNIX. На практике, подобная реализация недостаточно гибка (диски UNIX могут иметь весьма ограниченное число разделов). Вместо этого Vinum вводит еще один уровень абстракции: единый дисковый раздел UNIX (*drive* в терминах Vinum) делится на непрерывные области, называемые *поддисками* (*subdisk*), которые и будут "строительным материалом" для наборов.
- Поддиски, как уже упоминалось, располагаются внутри *приводов* (*drive*) Vinum, существующих дисковых разделов UNIX. Привод может содержать неограниченное количество поддисков. Небольшая область в начале привода зарезервирована под хранение информации о конфигурации и состоянии Vinum; все остальное пространство пригодно для хранения данных.

Сейчас мы опишем, как эта иерархия обеспечивает необходимую функциональность для Vinum.

### 17.5.1. Размер тома

Наборы могут состоять из большого количества поддисков, распределенных по разным приводам Vinum. Стало быть, размеры отдельных дисков не ограничивают размер набора, а следовательно, и тома.

### 17.5.2. Избыточность

Vinum реализует избыточность посредством связывания с томом нескольких наборов. Содержимое каждого набора является полной копией содержимого тома. Количество наборов в томе может быть от одного до восьми.

Хотя набор представляет данные тома целиком, отдельные части содержимого тома могут быть представлены не всеми наборами. Во-первых, для некоторых частей набора поддиски могут быть не определены; во-вторых, часть набора может быть потеряна из-за отказа диска. До тех пор, пока хотя бы один набор может обеспечить данные для полного адресного пространства тома, том полностью функционален.

### 17.5.3. Производительность

Vinum поддерживает как конкатенацию, так и перемежение на уровне наборов:

- *Сцепленный набор* использует пространство поддисков последовательно, склеивая их "встык".
- *Набор с перемежением* разбивает данные по поддискам в соответствии с размером страйпа. Поддисков должно быть по меньшей мере два (чтобы отличить набор от сцепленного), и все они должны быть одинакового размера.

### 17.5.4. Организация наборов: что выбрать?

Vinum, распространяемый с FreeBSD версии 5.3 поддерживает два вида организации наборов:

- Сцепленные наборы наиболее гибки в использовании: они могут содержать любое количество поддисков произвольного размера. Такой набор может быть расширен "на лету" путем добавления дополнительных поддисков. Поддержка сцепленных наборов требует меньших затрат процессорного времени, чем поддержка наборов с перемежением (хотя различие вряд ли поддается измерению). С другой стороны, они наиболее чувствительны к концентрации нагрузки в одной области тома, при которой один из дисков принимает на себя всю нагрузку, а остальные бездействуют.
- Основным преимуществом наборов с перемежением (RAID-0) является распределение "горячих точек" нагрузки; вы можете даже полностью уравнивать ее, выбрав оптимальный размер страйпа (около 256 kB). Недостатки такой организации — более сложный код и ограничения на поддиски: все они должны быть строго одного размера. Кроме того, процесс добавления поддиска в набор с перемежением "на ходу" является настолько нетривиальной задачей, что в настоящее время Vinum не поддерживает эту операцию. Дополнительное (тривиальное) ограничение состоит в том, что набор с перемежением должен содержать как минимум два поддиска, иначе он будет неотличим от сцепленного.

Преимущества и недостатки различных методов организации наборов описаны в Табл. 17-1.

**Таблица 17-1. Методы организации наборов Vinum**

Тип набора	Поддисков, мин.	Расширяется "на лету"	Поддиски строго одного размера	Применение
сцепленный (concatenated)	1	да	нет	Крупные системы хранения, требующие максимальной гибкости и умеренной производительности
с перемежением (striped)	2	нет	да	Высокая производительность, в том числе в случае параллельного доступа к данным



## 17.6. Несколько примеров

Vinum ведет *базу данных конфигурации*, в которой описаны все объекты Vinum в отдельной системе. Начальная конфигурация создается пользователем при помощи системной утилиты `vinum(8)` из одного или нескольких конфигурационных файлов. Копия конфигурации хранится в начале каждого дискового раздела (привода) Vinum. Все копии обновляются при изменении состояния томов, поэтому после перезапуска состояние объектов Vinum восстанавливается.

### 17.6.1. Конфигурационный файл

Конфигурационный файл описывает объекты Vinum. Описание простого тома может быть таким:

```
drive a device /dev/da3h
volume myvol
plex org concat
sd length 512m drive a
```

Здесь описываются четыре объекта Vinum:

- Строка *drive* объявляет дисковый раздел (*привод*) и его местоположение на физическом диске. Приводу дано символьное имя *a*. Разделение символьных имен и имен устройств дает возможность перемещать физические диски (например, по разным контроллерам, или менять их местами) без изменения конфигурации.
- Строка *volume* описывает том. Единственным требуемым параметром является имя тома *myvol*.
- Строка *plex* определяет набор. Единственный обязательный параметр — метод организации набора, в нашем случае *concat* (сцепленный). Давать набору имя в явном виде не обязательно: Vinum автоматически сгенерирует имя набора из имени тома и суффикса *.px*, где *x* — номер набора в томе. В нашем случае набор будет называться *myvol.p0*.
- Наконец, строка *sd* описывает поддиск. Минимальными требованиями к его описанию являются имя *привода*, на котором он будет располагаться, и его размер. Как и в случае набора, имя указывать не обязательно: имя поддиска будет построено добавлением *.sx* к имени набора, где *x* будет номером поддиска в наборе. Наш поддиск получит имя *myvol.p0.s0*.

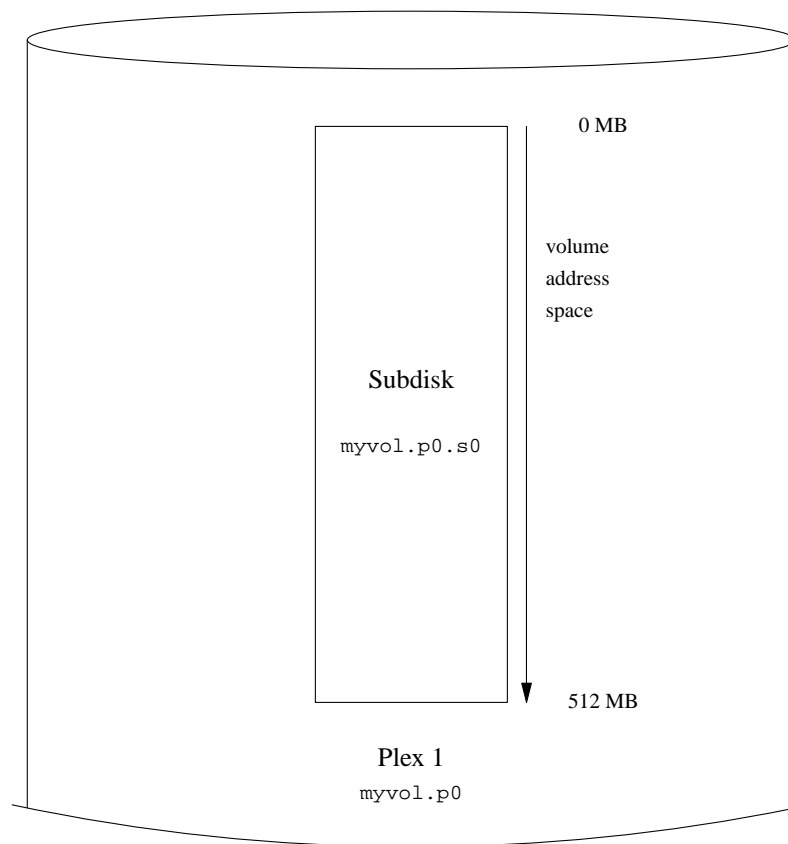
В результате обработки такого конфигурационного файла `vinum(8)` выдаст нам следующее:

```
# vinum -> create config1
Configuration summary
Drives:      1 (4 configured)
Volumes:     1 (4 configured)
Plexes:      1 (8 configured)
Subdisks:    1 (16 configured)
```

D a	State: up	Device /dev/da3h	Avail: 2061/2573 MB (80%)
V myvol	State: up	Plexes: 1	Size: 512 MB
P myvol.p0	C State: up	Subdisks: 1	Size: 512 MB
S myvol.p0.s0	State: up	PO: 0	B Size: 512 MB

На этом кратком листинге показан формат вывода `vinum(8)`. Графически созданный нами том представлен на Рис. 17-4.

Рисунок 17-4. Простой том Vinit



Этот и последующие рисунки изображают том, содержащий один или несколько наборов, каждый из которых, в свою очередь, состоит из одного или нескольких поддисков. В первом тривиальном примере том состоит из одного набора, представленного одним поддиском.

Построенный нами только что том не имеет никаких преимуществ перед обычным дисковым разделом. Он содержит единственный набор, так что не обеспечивает избыточность; набор состоит из одного поддиска, поэтому методика распределения дисковых блоков ничем не отличается от дискового раздела. В последующих параграфах мы рассмотрим более интересные конфигурации.

### 17.6.2. Повышаем надежность: зеркалирование

Надежность тома может быть повышена при помощи зеркалирования. Планируя зеркалированный том, важно не забыть о том, чтобы поддиски каждого набора располагались на разных физических дисках, чтобы отказ одного из них не привел к выходу из строя более чем одного набора. Вот конфигурация, определяющая зеркалированный том:

```

drive b device /dev/da4h
volume mirror
    plex org concat
        sd length 512m drive a
    plex org concat
        sd length 512m drive b

```

Как мы видим, нет необходимости вновь описывать привод *a*, поскольку *Vinum* сохраняет состояние уже сконфигурированных объектов. После обработки этих определений конфигурация будет выглядеть так:

```

Drives:      2 (4 configured)
Volumes:     2 (4 configured)
Plexes:      3 (8 configured)
Subdisks:    3 (16 configured)

D a          State: up      Device /dev/da3h      Avail: 1549/2573 MB (60%)
D b          State: up      Device /dev/da4h      Avail: 2061/2573 MB (80%)

V myvol      State: up      Plexes:      1 Size:      512 MB
V mirror     State: up      Plexes:      2 Size:      512 MB

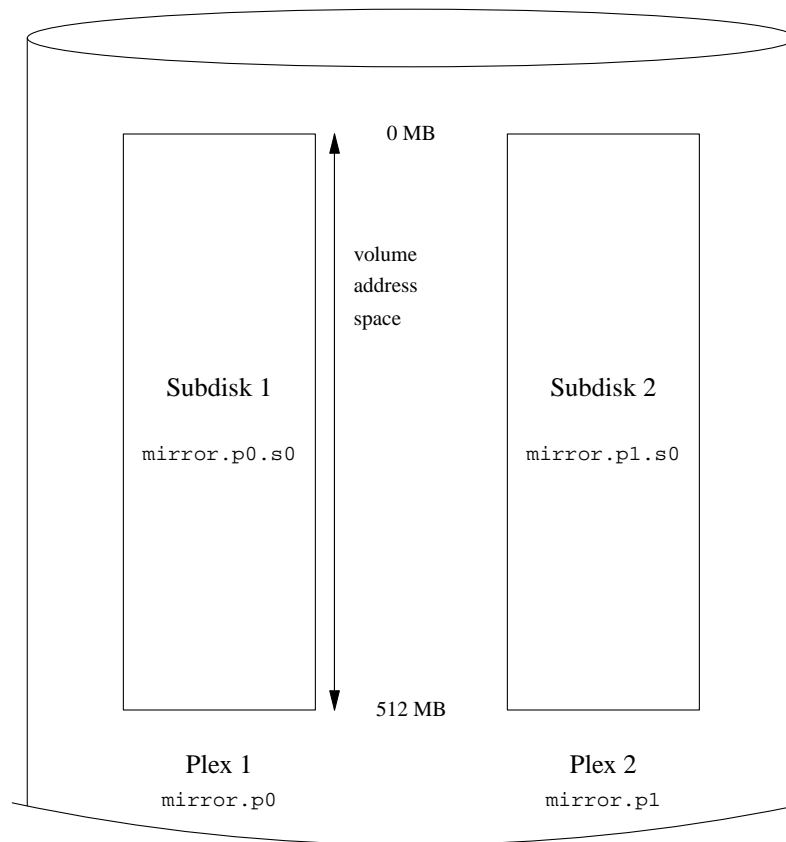
P myvol.p0   C State: up      Subdisks:    1 Size:      512 MB
P mirror.p0  C State: up      Subdisks:    1 Size:      512 MB
P mirror.pl  C State: initializing Subdisks:    1 Size:      512 MB

S myvol.p0.s0 State: up      PO:      0 B Size:      512 MB
S mirror.p0.s0 State: up      PO:      0 B Size:      512 MB
S mirror.pl.s0 State: empty   PO:      0 B Size:      512 MB

```

Рис. 17-5 иллюстрирует структуру полученного тома.

Рисунок 17-5. Зеркалированный том Vinum



В данном примере каждый набор содержит все 512 MB адресного пространства тома. Как и в предыдущем случае, каждый набор состоит из одного поддиска.

### 17.6.3. Оптимизируем производительность

Зеркалированный том из предыдущего примера гораздо более отказоустойчив, чем обычный том, но его производительность ниже: каждый запрос на запись выливается в две операции физической записи, что вдвое увеличивает необходимую пропускную способность шины дисковой подсистемы. Увеличение производительности требует иного подхода: вместо зеркалирования данные распределяются (перемежением) по максимальному количеству физических дисков. Следующий пример конфигурации создает том с перемежением на четырех дисках:

```
drive c device /dev/da5h
drive d device /dev/da6h
volume stripe
plex org striped 512k
```

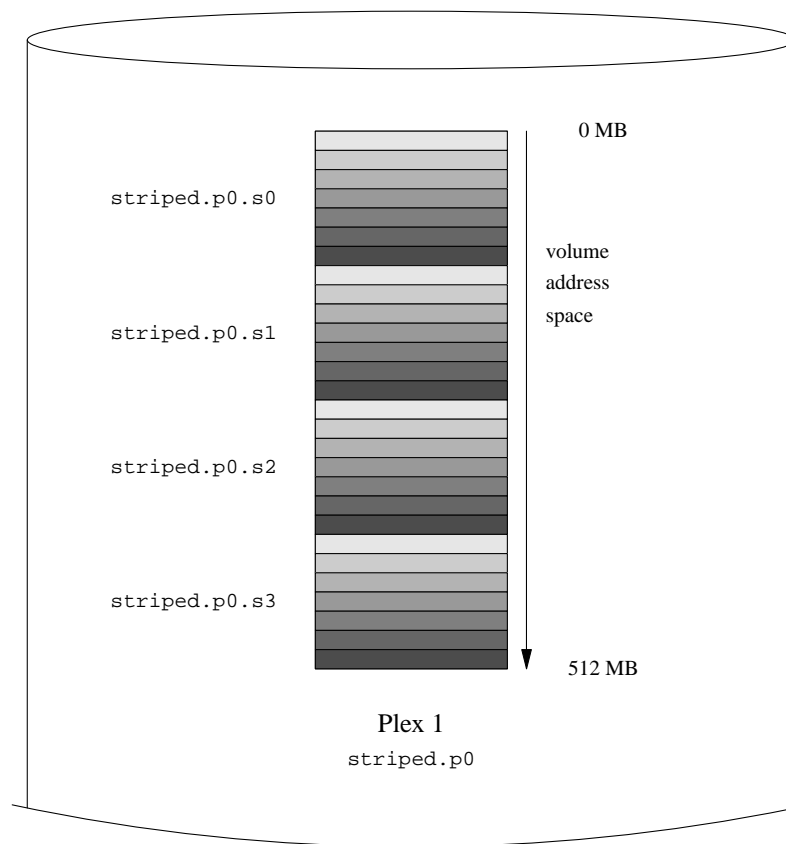
```
sd length 128m drive a
sd length 128m drive b
sd length 128m drive c
sd length 128m drive d
```

Как и ранее, нет необходимости переопределять уже сконфигурированные приводы. Общий вид базы конфигурации Vinum после создания нового тома будет таким:

```
Drives:      4 (4 configured)
Volumes:     3 (4 configured)
Plexes:      4 (8 configured)
Subdisks:    7 (16 configured)
```

D a	State: up	Device /dev/da3h	Avail: 1421/2573 MB (55%)
D b	State: up	Device /dev/da4h	Avail: 1933/2573 MB (75%)
D c	State: up	Device /dev/da5h	Avail: 2445/2573 MB (95%)
D d	State: up	Device /dev/da6h	Avail: 2445/2573 MB (95%)
V myvol	State: up	Plexes: 1 Size: 512 MB	
V mirror	State: up	Plexes: 2 Size: 512 MB	
V striped	State: up	Plexes: 1 Size: 512 MB	
P myvol.p0	C State: up	Subdisks: 1 Size: 512 MB	
P mirror.p0	C State: up	Subdisks: 1 Size: 512 MB	
P mirror.pl	C State: initializing	Subdisks: 1 Size: 512 MB	
P striped.pl	State: up	Subdisks: 1 Size: 512 MB	
S myvol.p0.s0	State: up	PO: 0 B Size: 512 MB	
S mirror.p0.s0	State: up	PO: 0 B Size: 512 MB	
S mirror.pl.s0	State: empty	PO: 0 B Size: 512 MB	
S striped.p0.s0	State: up	PO: 0 B Size: 128 MB	
S striped.p0.s1	State: up	PO: 512 kB Size: 128 MB	
S striped.p0.s2	State: up	PO: 1024 kB Size: 128 MB	
S striped.p0.s3	State: up	PO: 1536 kB Size: 128 MB	

Рисунок 17-6. Том с перемежением



Новосозданный том представлен на Рис. 17-6. Плотность заштрихованных участков показывает расположение страйпов в адресном пространстве набора (от светлых к темным).

#### 17.6.4. Отказоустойчивость и производительность одновременно

При наличии достаточного количества дисков можно создать том, сочетающий повышенную отказоустойчивость и высокую производительность по сравнению со стандартными дисковыми разделами UNIX. Типичная конфигурация может быть такой:

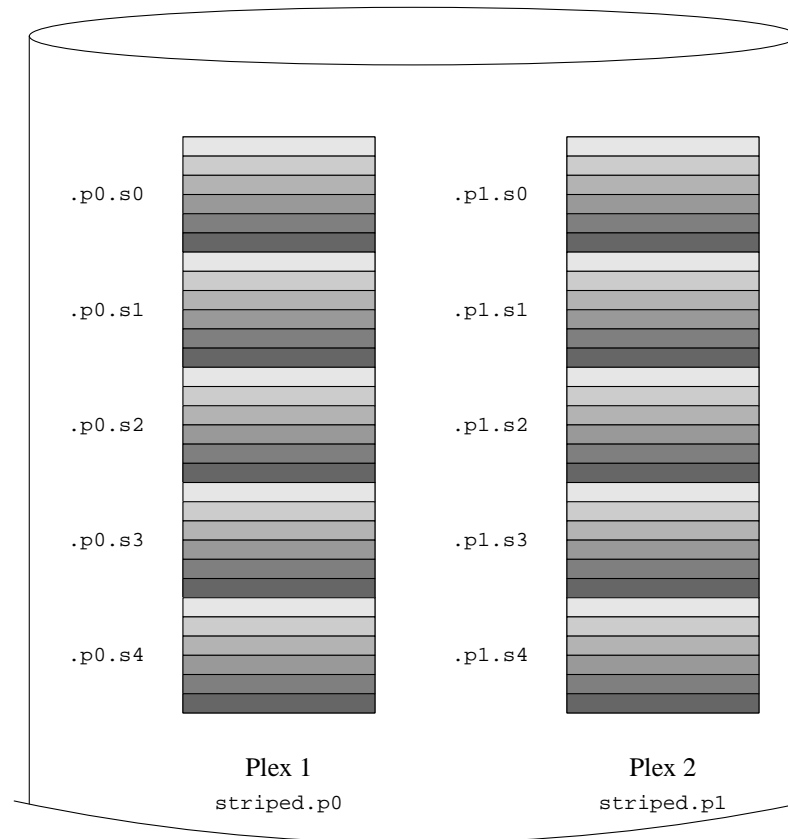
```
volume raid10
  plex org striped 512k
    sd length 102480k drive a
    sd length 102480k drive b
    sd length 102480k drive c
    sd length 102480k drive d
    sd length 102480k drive e
```

```
plex org striped 512k
sd length 102480k drive c
sd length 102480k drive d
sd length 102480k drive e
sd length 102480k drive a
sd length 102480k drive b
```

Как вы можете заметить, поддиски второго набора смещены на два привода относительно поддисков первого. В результате даже при запросе на запись, пересекающем границы страйпа, не возникнет двух обращений к одному физическому диску.

Рис. 17-7 отражает структуру нового тома.

**Рисунок 17-7. Зеркалированный том с перемежением**



## 17.7. Правила именования объектов

Как уже было описано, Vinum автоматически именует создаваемые наборы и поддиски, хотя эти имена и могут быть переопределены. На самом деле, мы не рекомендовали бы переопределять стандартные имена: опыт с дисковым менеджером VERITAS показал, что гибкость в именовании объектов не дает ощутимого преимущества, а запутать пользователя может.

Имена объектов могут состоять из любых непробельных символов. Впрочем, рекомендуем ограничиться буквами, цифрами и подчеркиваниями. Имена томов, наборов и поддисков могут быть до 64 символов длиной; максимальная длина имени привода — 32 символа.

Для объектов Vinum в иерархии `/dev/vinum` создаются файлы устройств. Приведенный выше пример конфигурации создаст следующий набор устройств:

- Управляющие устройства `/dev/vinum/control` и `/dev/vinum/controld`, используемые системной утилитой `vinum(8)` и демоном Vinum соответственно.
- Блочные и символьные устройства для каждого из томов. Основные устройства, используемые Vinum'ом. Блочные устройства именуются в соответствии с именами томов; символьные, в соответствии с традицией именования устройств BSD, имеют префикс *r*. Таким образом, вышеописанная конфигурация будет включать блочные устройства `/dev/vinum/myvol`, `/dev/vinum/mirror`, `/dev/vinum/striped`, `/dev/vinum/raid5` и `/dev/vinum/raid10`, и символьные устройства `/dev/vinum/rmyvol`, `/dev/vinum/rmirror`, `/dev/vinum/rstriped`, `/dev/vinum/rraid5` и `/dev/vinum/rraid10`.
- Каталог `/dev/vinum/drive` с записями для каждого привода. В реальности, каждая запись является символьной ссылкой на соответствующий файл дискового устройства.
- Каталог `/dev/vinum/volume` с записями для томов. В нем содержатся поддиректории для каждого набора, внутри которых, в свою очередь, имеются поддиректории для каждого из поддисков.
- Каталоги `/dev/vinum/plex`, `/dev/vinum/sd` и `/dev/vinum/rsd`, содержащие блочные устройства для наборов, блочные и символьные устройства для каждого из поддисков.

Например, для конфигурации, описываемой как

```
drive drive1 device /dev/sd1h
drive drive2 device /dev/sd2h
drive drive3 device /dev/sd3h
drive drive4 device /dev/sd4h
  volume s64 setupstate
    plex org striped 64k
      sd length 100m drive drive1
      sd length 100m drive drive2
      sd length 100m drive drive3
      sd length 100m drive drive4
```

после обработки `vinum(8)`, созданный набор устройств в каталоге `/dev/vinum` будет таким:

```
brwx----- 1 root wheel 25, 0x40000001 Apr 13 16:46 Control
brwx----- 1 root wheel 25, 0x40000002 Apr 13 16:46 control
brwx----- 1 root wheel 25, 0x40000000 Apr 13 16:46 controld
drwxr-xr-x 2 root wheel 512 Apr 13 16:46 drive
drwxr-xr-x 2 root wheel 512 Apr 13 16:46 plex
crwxr-xr-- 1 root wheel 91, 2 Apr 13 16:46 rs64
```



```

drwxr-xr-x  2 root  wheel           512 Apr 13 16:46 rsd
drwxr-xr-x  2 root  wheel           512 Apr 13 16:46 rvol
brwxr-xr--  1 root  wheel    25,    2 Apr 13 16:46 s64
drwxr-xr-x  2 root  wheel           512 Apr 13 16:46 sd
drwxr-xr-x  3 root  wheel           512 Apr 13 16:46 vol

/dev/vinum/drive:
total 0
lrwxr-xr-x  1 root  wheel    9 Apr 13 16:46 drive1 -> /dev/sd1h
lrwxr-xr-x  1 root  wheel    9 Apr 13 16:46 drive2 -> /dev/sd2h
lrwxr-xr-x  1 root  wheel    9 Apr 13 16:46 drive3 -> /dev/sd3h
lrwxr-xr-x  1 root  wheel    9 Apr 13 16:46 drive4 -> /dev/sd4h

/dev/vinum/plex:
total 0
brwxr-xr--  1 root  wheel    25, 0x10000002 Apr 13 16:46 s64.p0

/dev/vinum/rsd:
total 0
crwxr-xr--  1 root  wheel    91, 0x20000002 Apr 13 16:46 s64.p0.s0
crwxr-xr--  1 root  wheel    91, 0x20100002 Apr 13 16:46 s64.p0.s1
crwxr-xr--  1 root  wheel    91, 0x20200002 Apr 13 16:46 s64.p0.s2
crwxr-xr--  1 root  wheel    91, 0x20300002 Apr 13 16:46 s64.p0.s3

/dev/vinum/rvol:
total 0
crwxr-xr--  1 root  wheel    91,    2 Apr 13 16:46 s64

/dev/vinum/sd:
total 0
brwxr-xr--  1 root  wheel    25, 0x20000002 Apr 13 16:46 s64.p0.s0
brwxr-xr--  1 root  wheel    25, 0x20100002 Apr 13 16:46 s64.p0.s1
brwxr-xr--  1 root  wheel    25, 0x20200002 Apr 13 16:46 s64.p0.s2
brwxr-xr--  1 root  wheel    25, 0x20300002 Apr 13 16:46 s64.p0.s3

/dev/vinum/vol:
total 1
brwxr-xr--  1 root  wheel    25,    2 Apr 13 16:46 s64
drwxr-xr-x  3 root  wheel           512 Apr 13 16:46 s64.plex

/dev/vinum/vol/s64.plex:
total 1
brwxr-xr--  1 root  wheel    25, 0x10000002 Apr 13 16:46 s64.p0
drwxr-xr-x  2 root  wheel           512 Apr 13 16:46 s64.p0.sd

/dev/vinum/vol/s64.plex/s64.p0.sd:
total 0
brwxr-xr--  1 root  wheel    25, 0x20000002 Apr 13 16:46 s64.p0.s0
brwxr-xr--  1 root  wheel    25, 0x20100002 Apr 13 16:46 s64.p0.s1
brwxr-xr--  1 root  wheel    25, 0x20200002 Apr 13 16:46 s64.p0.s2
brwxr-xr--  1 root  wheel    25, 0x20300002 Apr 13 16:46 s64.p0.s3

```

Заметим, что, несмотря на то что наборы и поддиски не рекомендуется называть каким-либо специальным образом, приводы Vinum должны быть поименованы. Именование позволяет отвязать приводы от физических устройств, и при этом обеспечить их автоматическое распознавание. Имена приводов могут достигать длины в 32 символа.

### 17.7.1. Создание файловых систем

Тома с точки зрения системы аналогичны дискам, за одним малым исключением: в отличие от дисков UNIX, тома Vinum не содержат таблиц разделов. В результате потребовалось модифицировать некоторые утилиты работы с дисками, в первую очередь newfs(8), которая ранее использовала последний символ имени тома для определения идентификатора раздела. Например, дисковое устройство может именоваться /dev/ad0a — первый раздел (a) первого (0) IDE-диска (ad) — или /dev/da2h — восьмой раздел (h) третьего (2) диска SCSI (da). Том Vinum может называться, например, /dev/vinum/concat — как легко видеть, имя тома никак не связано с именем раздела.

Обычно newfs(8) пытается интерпретировать имя раздела и сообщает об ошибке при невозможности такой интерпретации:

```
# newfs /dev/vinum/concat
newfs: /dev/vinum/concat: can't figure out file system partition
```

**Замечание:** Дальнейшее относится к версиям FreeBSD до 5.0:

Для создания файловых систем на томе Vinum следует использовать опцию newfs(8) -v:

```
# newfs -v /dev/vinum/concat
```

## 17.8. Создание конфигурации Vinum

Стандартное (GENERIC) ядро FreeBSD не включает Vinum. Хотя и можно собрать специальное ядро с включенной поддержкой Vinum, этот вариант не рекомендуется. Обычный способ активизации Vinum — загрузка модуля для ядра (kld). При этом, явно использовать команду kldload(8) нет необходимости: при старте утилита vinum(8) проверит наличие поддержки Vinum в ядре и при необходимости загрузит модуль автоматически.

### 17.8.1. Активация

Vinum хранит конфигурационную информацию на дисковых разделах в той же форме, что используется в файлах конфигурации при создании объектов. Впрочем, в них применяются некоторые ключевые слова, не разрешенные в файлах конфигурации. Например, хранимая на диске база может выглядеть так:

```
volume myvol state up
volume bigraid state down
plex name myvol.p0 state up org concat vol myvol
plex name myvol.p1 state up org concat vol myvol
plex name myvol.p2 state init org striped 512b vol myvol
plex name bigraid.p0 state initializing org raid5 512b vol bigraid
```

```
sd name myvol.p0.s0 drive a plex myvol.p0 state up len 1048576b driveoffset 265b plexoffset 0b
sd name myvol.p0.s1 drive b plex myvol.p0 state up len 1048576b driveoffset 265b plexoffset 1048576b
sd name myvol.p1.s0 drive c plex myvol.p1 state up len 1048576b driveoffset 265b plexoffset 0b
sd name myvol.p1.s1 drive d plex myvol.p1 state up len 1048576b driveoffset 265b plexoffset 1048576b
sd name myvol.p2.s0 drive a plex myvol.p2 state init len 524288b driveoffset 1048841b plexoffset 0b
sd name myvol.p2.s1 drive b plex myvol.p2 state init len 524288b driveoffset 1048841b plexoffset 524288b
sd name myvol.p2.s2 drive c plex myvol.p2 state init len 524288b driveoffset 1048841b plexoffset 1048576b
sd name myvol.p2.s3 drive d plex myvol.p2 state init len 524288b driveoffset 1048841b plexoffset 1572864b
sd name bigraid.p0.s0 drive a plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 0b
sd name bigraid.p0.s1 drive b plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 4194304b
sd name bigraid.p0.s2 drive c plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 8388608b
sd name bigraid.p0.s3 drive d plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 12582912b
sd name bigraid.p0.s4 drive e plex bigraid.p0 state initializing len 4194304b driveoff set 1573129b plexoffset 16777216b
```

Видно, что каждый объект имеет явно описанное имя, а поддиски еще и явное положение на приводе (и то и другое может, хотя это и не рекомендуется, устанавливаться пользователем). Помимо этого, для каждого объекта хранится его состояние (и установка состояния напрямую пользователю недоступна). *Vinum* не хранит в конфигурационных базах информацию о приводах: она создается при сканировании дисковых разделов, помеченных как *Vinum*. Это дает возможность *Vinum* правильно идентифицировать диски при смене имени устройства.

### 17.8.1.1. Автоматическая активация

Для автоматического старта *Vinum* при загрузке системы добавьте следующую строку в файл конфигурации системы `/etc/rc.conf`:

```
start_vinum="YES" # set to YES to start vinum
```

Если в вашей системе нет файла `/etc/rc.conf`, создайте его с таким содержимым. Данная строка вызовет активацию `kld` модуля *Vinum* при загрузке, а также старт всех объектов, упомянутых в конфигурации *Vinum*. Активация *Vinum* происходит до монтирования файловых систем, так что возможны автоматическая проверка (`fsck(8)`) и монтирование файловых систем на томах *Vinum*.

При старте с помощью команды `vinum start`, *Vinum* читает базы конфигурации с одного из приводов. В нормальной ситуации все приводы содержат идентичную информацию о конфигурации, так что не имеет значения, какой именно диск будет читаться. В случае краха *Vinum* определяет, какая копия является наиболее свежей, в дальнейшем использует ее, а также обновляет ее на оставшихся приводах.

## 17.9. *Vinum* для корневой файловой системы

Сервер, все информационные файловые системы которого дублированы, хотелось бы оснастить и зеркалированной корневой файловой системой. Создание такой конфигурации не вполне тривиально по сравнению с зеркалированием прочих файловых систем:

- Корневая файловая система должна быть доступна для чтения в самом начале процесса загрузки, так что инфраструктура *Vinum* должна к этому моменту уже работать.
- Том с корневой файловой системой содержит, помимо прочего, системный загрузчик и ядро, которые должны читаться "родными" (*native*) утилитами компьютера (*BIOS* для машин архитектуры *PC*); обеспечить поддержку ими тонкостей *Vinum* зачастую невозможно.

В данном разделе термин "корневой том" означает том Vinum, содержащий корневую файловую систему. Неплохой идеей является назвать такой том "root", хотя это, разумеется, и необязательно. Все наши примеры, впрочем, будут использовать именно это имя.

### 17.9.1. Активизация Vinum на ранней стадии процесса загрузки

Для обеспечения этого необходимо следующее:

- Vinum должен быть доступен ядру еще на этапе загрузки. Метод, описанный в Разд. 17.8.1.1, неприменим; на самом деле, параметр `start_vinum` *не* должен быть установлен. Одним из вариантов является сборка ядра с поддержкой Vinum, что возможно, но, как правило, нежелательно. Более удобный вариант — загрузка модуля ядра Vinum при помощи `/boot/loader` (Разд. 12.3.3), для чего в файл `/boot/loader.conf` следует добавить строку  
`vinum_load="YES"`
- Vinum должен быть активирован достаточно рано, поскольку требуется предоставить том для корневой файловой системы. По умолчанию Vinum в ядре не начинает поиск приводов, содержащих информацию о томах Vinum, до команды администратора (или одного из стартовых скриптов) `vinum start`.

**Замечание:** Данный раздел описывает необходимые действия для FreeBSD версии 5.x и старше. Шаги, необходимые в случае FreeBSD версии 4.x, описаны ниже: Разд. 17.9.5.

Строка

```
vinum.autostart="YES"
```

в файле `/boot/loader.conf`, указывает Vinum автоматически просканировать все диски для сбора информации о томах в процессе загрузки ядра.

Обращаем ваше внимание, что нет необходимости как-либо специально сообщать ядру, где находится корневая файловая система. Загрузчик (`/boot/loader`) найдет необходимое имя устройства в `/etc/fstab` и передаст его ядру. В момент монтирования корневой файловой системы ядро передаст имя устройства соответствующему драйверу для декодирования (трансляции в пару идентификаторов устройств — `major/minor device number`).

### 17.9.2. Загрузчик должен прочесть корневой том Vinum

В настоящее время начальный загрузчик FreeBSD ограничен размером всего в 7.5 KB, и этот размер фактически исчерпан (загрузчик должен уметь прочесть файл `/boot/loader` с файловой системы формата UFS и передать ему управление). Невозможно разместить в загрузчике внутренние структуры Vinum, чтобы он мог считать настройку Vinum и самостоятельно определить элементы загрузочного тома. Поэтому, для создания у загрузчика иллюзии, что загрузка происходит со стандартного раздела "a" требуются некоторые дополнительные ухищрения.

Для того, чтобы такая загрузка вообще была возможной, корневой том должен отвечать следующим требованиям:

- быть только зеркалированным (ни перемежение, ни RAID5 невозможны);
- содержать ровно один поддиск на каждом из наборов.

Заметим, что возможно (и является, вообще говоря, основной целью), чтобы корневой том содержал несколько наборов, каждый с копией корневой файловой системы. В процессе загрузки, впрочем, используется только одна из копий (на этапе поиска начального загрузчика и его конфигурационных файлов, ядра, модулей и т.п. до момента монтирования корневой файловой системы). Для обеспечения возможности загрузки поддиск каждого из наборов должен быть отображен в псевдо-раздел "a". Вообще говоря, эти псевдо-разделы не обязаны находиться на одних и тех же местах дисков; тем не менее, во избежание излишней путаницы, рекомендуется создавать тома с одинаково устроенными дисками для зеркалирования.

Для создания псевдо-разделов "a" необходимо для каждого из дисков, содержащих копию корневого тома, проделать следующее:

1. Определить положение (смещение от начала устройства) и размер поддиска, являющегося частью корневого тома:

```
# vinum l -rv root
```

Отметим, что все размеры и смещения в терминах Vinum указаны в байтах. Для получения номеров блоков, используемых в утилите `disklabel`, все числа надо поделить на 512.

2. Выполнить команду

```
# disklabel -e devname
```

для каждого из дисков, на котором будет расположен корневой том. *devname* будет или именем диска (например, `da0`) для дисков без таблицы слайсов, или именем слайса (`ad0s1`).

Если на устройстве уже есть раздел "a" (скорее всего, это предыдущая инкарнация корневой файловой системы), он должен быть переименован (чтобы быть доступным в будущем, на всякий случай; при этом стартовый загрузчик больше не должен выбирать его по умолчанию). Не забудьте, что активный (например, смонтированный) раздел не может быть переименован, так что переименование нужно производить или загрузившись с диска 'Fixit', или в два шага (для конфигурации с зеркалированием сначала переименовать раздел на втором диске, затем, после перезагрузки, на первом).

Затем, адрес начала нового раздела "a" вычисляется как сумма начального смещения раздела Vinum и подсчитанного выше адреса поддиска внутри привода. Совместно с вычисленным размером эти значения вносятся в поля "offset" и "size" строки "a" `disklabel(8)`; Поле "fstype" должно быть 4.2BSD. Значения полей "fsize", "bsize" и "cpq" желательно заполнить в соответствии с имеющейся файловой системой, хотя в обсуждаемом контексте это и не строго обязательно.

Как можно заметить, новосозданный раздел "a" располагается внутри раздела Vinum. Утилита `disklabel` разрешает разделам пересекаться только в случае, если один из них корректно описан как имеющий тип "vinum".

3. Готово! Сконструированный псевдо-раздел "a" создан на каждом из устройств, содержащих реплики корневого тома. Крайне важно проверить результат еще раз, выполнив команду

```
# fsck -n /dev/devnamea
```

Следует помнить, что все файлы, содержащие загрузочную конфигурацию, должны быть построены в соответствии с новой корневой файловой системой; скорее всего, эта информация не будет соответствовать текущему положению вещей. В особенности, следует обратить внимание на содержимое файлов `/etc/fstab` и `/boot/loader.conf`.

После перезагрузки начальный загрузчик должен определить данные новой корневой файловой системы на основе Vinum и действовать в соответствии с ними. В завершение процесса инициализации ядра, после упоминания всех определившихся устройств, должно появиться сообщение вида:

```
Mounting root from ufs:/dev/vinum/root
```

### 17.9.3. Пример конфигурации корневой файловой системы на базе Vinum

После создания корневого тома, вывод команды `vinum l -rv root` будет примерно таким:

```
...
Subdisk root.p0.s0:
Size:          125829120 bytes (120 MB)
State: up
Plex root.p0 at offset 0 (0 B)
Drive disk0 (/dev/da0h) at offset 135680 (132 kB)

Subdisk root.pl.s0:
Size:          125829120 bytes (120 MB)
State: up
Plex root.pl at offset 0 (0 B)
Drive disk1 (/dev/dalh) at offset 135680 (132 kB)
```

Из этой информации нас более всего интересует смещение в 135680 байт относительно раздела `/dev/da0h`. После деления на 512 получим 265 дисковых блоков для утилиты `disklabel`.

Аналогичным образом, размер тома составит 245760 512-байтных блоков. Так же устроена реплика тома на диске `/dev/dalh`.

Разметка разделов (`disklabel`) будет выглядеть примерно так:

```
...
8 partitions:
#      size  offset  fstype  [fsize bsize bps/cpg]
a:   245760    281   4.2BSD   2048 16384    0 # (Cyl.  0*- 15*)
c:  71771688     0  unused     0     0    # (Cyl.  0 - 4467*)
h:  71771672    16   vinum                # (Cyl.  0*- 4467*)
```

Как уже отмечалось, размер ("size") псевдо-раздела "a" соответствует значению, вычисленному ранее; смещение ("offset") равно сумме смещения поддиска внутри раздела Vinum ("h") и смещения самого этого раздела относительно начала диска (слайса). Так мы избегаем проблем, описанных ниже (Разд. 17.9.4.3). Заметим также, что раздел "a" целиком размещен внутри раздела "h", описывающего все данные Vinum на этом диске.

Заметим, что в описанном примере все дисковое пространство отдано Vinum. Корневого раздела, существовавшего до настройки Vinum, нет, поскольку это вновь установленный диск, предназначенный для использования исключительно в Vinum.

## 17.9.4. Проблемы и их устранение

Если что-то пошло не так, должен быть путь для восстановления доступа к информации. Далее описаны некоторые известные проблемные ситуации и способы их устранения.

### 17.9.4.1. Загрузчик работает, но система не грузится

Если по каким-то причинам система не может завершить загрузку, загрузчик может быть прерван нажатием пробела в течение первых 10 (по умолчанию) секунд. Вы можете посмотреть переменные загрузчика (такие как `vinum.autostart`) при помощи команды `show` и изменить их содержимое командами `set` и `unset`.

Если единственной проблемой было отсутствие загруженного модуля ядра Vinum, поможет просто команда `load vinum`.

Процесс загрузки должен быть продолжен командой `boot -as`. Параметры `-as` заставят ядро спросить о корневой файловой системе (параметр `-a`) и остановить процесс загрузки в однопользовательском (параметр `-s`) режиме. При этом корневая файловая система будет смонтирована в режиме "только для чтения" (`read-only`). В результате, даже если будет смонтирован лишь один набор из многонаборного тома, риска рассинхронизации наборов нет.

Ответом на приглашение ввести адрес корневой файловой системы может быть имя любого устройства, указывающего на файловую систему, пригодную для загрузки. При корректно построенной карте файловых систем (`/etc/fstab`) значением по умолчанию должно быть что-то вроде `ufs:/dev/vinum/root`. Распространенной альтернативой будет, например, `ufs:da0d` (раздел, содержащий корневую файловую систему в эпоху "до Vinum"). Будьте осторожны, монтируя в качестве корневой файловой системы раздел "a", ссылающийся внутрь привода Vinum. В зеркалированном томе смонтируется только часть файловой системы. Если вам потребуется изменить ее содержимое, необходимо будет также удалить и создать заново остальные наборы тома в конфигурации Vinum, иначе они будут содержать несинхронизированные данные.

### 17.9.4.2. Работает только основной загрузчик

Если `/boot/loader` не загружается, а основной загрузчик все еще пригоден к работе (в начале процесса загрузки появляется одиночный минус в первой колонке экрана), можно попытаться прервать основной загрузчик нажатием пробела в этот момент. При этом загрузка будет остановлена на второй стадии (см. Разд. 12.3.2). Можно попробовать загрузиться с другого раздела, например, содержащего предыдущую копию корневой файловой системы (бывший раздел "a", см. выше).

### 17.9.4.3. Ничего не грузится, загрузчик падает

Это происходит, когда загрузчик на диске затерт Vinum'ом. К сожалению, Vinum оставляет лишь 4 KB в начале своего раздела до записи своих управляющих блоков. Две стадии первоначального загрузчика в совокупности с меткой диска BSD (`disklabel`) требуют 8 KB. Так что попытка создать раздел Vinum по смещению 0 диска или слайса, который должен быть загруженным, затрет загрузчик.

Что хуже, попытка разрешить описанную ситуацию посредством загрузки с диска 'Fixit' и перезаписи начального загрузчика при помощи команды `disklabel -v` (как описано в Разд. 12.3.2) приведет к тому, что загрузчик затрет управляющий заголовок Vinum, и тот не сможет найти свой диск. Хотя собственно конфигурация Vinum при этом не потеряется, и все данные могут быть восстановлены посредством создания объектов на их предыдущих местах, очень сложно окончательно исправить ситуацию. Весь раздел Vinum должен быть смещен по крайней мере на 4 КВ, так чтобы загрузчики и заголовок Vinum более не пересекались.

### 17.9.5. Отличия для FreeBSD версий 4.X

В системах под управлением FreeBSD 4.X отсутствуют некоторые функции ядра, необходимые для автоматического сканирования дисков Vinum'ом; кроме того, код, определяющий номера устройств корневой файловой системы, недостаточно продвинут для того, чтобы понимать конструкции вида `/dev/vinum/root`. Требуется приложение дополнительных усилий.

Во-первых, в файле `/boot/loader.conf` должен быть явно указан список дисков, которые Vinum будет сканировать:

```
vinum.drives="/dev/da0 /dev/da1"
```

Важно, чтобы были описаны все приводы, на которых могут встретиться данные Vinum. Не произойдет ничего плохого, если будет описано *больше* дисков, чем необходимо. Также, нет нужды описывать все слайсы и/или разделы (Vinum сканирует их автоматически).

Поскольку подпрограммы разбора имени корневой файловой системы и определения номеров устройств воспринимают только "классические" имена, такие как `/dev/ad0s1a`, для них не подходят имена типа `/dev/vinum/root`. Имя корневого тома должно быть сообщено Vinum отдельно. Для этого служит переменная загрузчика `vinum.root`. Соответствующая строка в файле `/boot/loader.conf` будет выглядеть так:

```
vinum.root="root"
```

Процедура инициализации ядра выглядит так: перед определением корневого устройства для загрузки проверяется, не установил ли какой-либо модуль соответствующий параметр ядра. В случае положительного ответа *и* при совпадении основного (major) номера устройства драйвера и установленной файловой системы автоопределение прекращается, что дает возможность передать продолжение процесса загрузки и монтирование корневого тома Vinum.

Следует отметить, впрочем, что обработчик ответа на запрос имени корневой файловой системы (`boot -a`) не может разобрать имя тома Vinum. Можно ввести имя устройства, отличное от устройства Vinum (в этом случае произойдет стандартная процедура разбора, так что можно указать, например, `ufs:da0d`). Имена же, подобные `ufs:vinum/root` не могут быть распознаны. Единственным выходом из этой ситуации будет перезагрузка и введение имени устройства заново (префикс `/dev/` в ответе на запрос "askroot" всегда можно опустить).

## Примечания

1. RAID — сокращение от термина *Redundant Array of Inexpensive Disks* (массив недорогих дисков с



*резервированием*); различные виды RAID предоставляют разные формы защиты от сбоев. RAID-0, вообще говоря, не является *RAID*, поскольку не обеспечивает резервирования.

# Глава 18. Локализация - I18N/L10N

## использование и настройка

*Предоставил Andrey A. Chernov. Переписал Michael C. Wu. Перевод на русский язык: Александр Пересунко, Денис Пеплин.*

### 18.1. Краткий обзор

FreeBSD - это очень распределенный проект, пользователи и контрибьюторы которого находятся в самых разных частях света. В этой главе рассказывается о возможностях интернационализации и локализации FreeBSD, которые позволяют не-англоговорящим пользователям делать свою работу. Существует много подходов в i18n реализации - как на системном уровне, так и на уровне приложений, так что, где это возможно, мы будем давать читателю ссылки на более углубленные источники документации.

После прочтения этой главы, вы будете знать:

- Как разные языки и региональные настройки кодируются в современных операционных системах.
- Как установить региональные настройки для вашей оболочки.
- Как настроить вашу консоль для отличных от английского языков.
- Как эффективно использовать X Window System с разными языками.
- Где найти дополнительную документацию по созданию i18n-совместимых приложений.

Перед чтением этой главы вам следует::

- Знать как установить дополнительные приложения сторонних разработчиков (Гл. 4).

### 18.2. Основы

#### 18.2.1. Что такое I18N/L10N?

Разработчики сократили интернационализацию в термин I18N, по числу символов между первым и последним символом интернационализации. В названии L10N используется тот же принцип от "локализации". Совместно используемые I18N/L10N методы, протоколы и приложения позволяют пользователям использовать языки по своему выбору.

I18N приложения написаны с применением набора I18N посредством библиотек. Это позволяет разработчикам писать простые файлы и переводить отображаемые меню и надписи на разные языки. Мы настоятельно рекомендуем программистам следовать этому соглашению.

## 18.2.2. Почему я должен использовать I18N/L10N?

I18N/L10N используется всегда, когда вы хотите просмотреть, ввести или обработать данные на отличных от английского языках.

## 18.2.3. Какие языки поддерживаются в проекте I18N?

I18N и L10N не являются специфичными для FreeBSD. На текущий момент у вас есть выбор между самыми распространенными языками мира, включая, но не ограничиваясь: китайский, немецкий, японский, корейский, французский, русский, вьетнамский и др.

## 18.3. Использование локализации

При всем своем величии, I18N не специфична для FreeBSD и является, по сути, соглашением. Следование этому соглашению помогает FreeBSD, и мы это приветствуем.

Настройки локализации базируются на трех основных терминах: код языка, код страны и кодировка. Имена локализаций строятся из этих частей следующим образом:

`код_языка_код_страны.кодировка`

### 18.3.1. Язык и код страны

Чтобы настроить FreeBSD (или любую другую UNIX-систему, поддерживающую I18N) на использование специфического языка, пользователю нужно выяснить коды этого языка и страны (коды страны сообщают приложению какой именно диалект языка этой страны нужно использовать). К тому же, веб-браузеры, SMTP/POP серверы, веб-серверы и т.п. делают свой выбор в зависимости от этого. Далее показаны примеры кодов языка/страны:

Код языка/страны	Описание
en_US	Английский - Соединенные Штаты
ru_RU	Русский для России
zh_TW	Традиционный китайский для Тайваня

### 18.3.2. Кодировки

Некоторые языки используют не соответствующие ASCII кодировки, которые используют для кодирования 8 бит, расширенное или многобайтовое представление символов (подробности см. в `multibyte(3)`). Старые приложения не распознают такие кодировки и ошибочно принимают их за управляющие символы. Новые приложения обычно распознают 8-битные символы. В зависимости от реализации, пользователю может потребоваться компиляция приложения с поддержкой расширенного или многобайтного представления символов, или же просто правильная его настройка. Чтобы иметь возможность вводить и обрабатывать символы с расширенным или многобайтным представлением, Коллекция портов FreeBSD (<http://www.FreeBSD.org/ru/ports/index.html>) предоставляет поддержку разных языков для различных приложений. Смотрите документацию по I18N в

соответствующем порту FreeBSD.

В частности, пользователю необходимо читать прилагаемую к приложению документацию чтобы решить, как правильно его настроить или указать правильные параметры `configure/Makefile/компилятору`.

Следует помнить следующие вещи:

- Наборы символов, использующие один байт для кодирования символа (см. `multibyte(3)`), например, ISO-8859-1, ISO-8859-15, KOI8-R, CP437.
- Расширенные или многобайтные кодировки, например, EUC, Big5.

Вы можете посмотреть актуальный список наборов символов на сайте IANA Registry (<http://www.iana.org/assignments/character-sets>).

**Замечание:** Вместо этого, FreeBSD версий 4.5 и выше используют X11-совместимые кодировки для локализации.

### 18.3.3. Приложения, использующие I18N

В системе Портов и Пакетов FreeBSD в название приложений, поддерживающих I18N, включена аббревиатура I18N для легкой их идентификации. И, тем не менее, не все приложения поддерживают требуемый язык.

### 18.3.4. Настройка локализации

Обычно достаточно экспортировать значение имени локализации в переменной `LANG` Вашей оболочки. Это можно сделать в пользовательском `~/.login_conf` файле или в инициализационном файле пользовательской оболочки (`~/.profile`, `~/.bashrc`, `~/.cshrc`). Нет надобности устанавливать отдельные параметры локализации такие как `LC_TYPE`, `LC_TIME`. За дополнительной информацией, пожалуйста, обращайтесь к документации FreeBSD по нужному вам языку.

Вам следует установить следующие переменные окружения в ваших конфигурационных файлах:

- `LANG` для семейства функций POSIX `setlocale(3)`
- `MM_CHARSET` набор символов MIME для приложений

Эти настройки включают конфигурацию оболочки пользователя, конфигурацию конкретных приложений и конфигурацию X11.

#### 18.3.4.1. Методы настройки локализации

Существует два метода настройки локализации и оба описаны ниже. Первый (рекомендуемый) заключается в задании переменных в классе логина, другой метод заключается в задании переменных окружения в инициализационном файле системной оболочки.

### 18.3.4.1.1. Метод Login Classes

Этот метод позволяет установить переменные окружения, необходимые для установки локализации и набора символов MIME, однократно для любых оболочек, вместо того, чтобы добавлять установку этих переменных в инициализационный файл каждой оболочки. Настройка на уровне пользователя может быть сделана самим пользователем, а Настройка на уровне администратора требует привилегий суперпользователя.

#### 18.3.4.1.1.1. Настройка на уровне пользователя

Вот минимальный пример файла `.login_conf` в домашнем каталоге пользователя, в котором задаются обе переменные для кодировки Latin-1:

```
me:\
:charset=ISO-8859-1:\
:lang=de_DE.ISO8859-1:
```

Это пример `.login_conf`, который задает переменные для Традиционного Китайского языка в кодировке BIG-5. Обратите внимание, что задается намного больше переменных, потому что некоторые программы некорректно воспринимают переменные окружения локализации для Китая, Японии и Кореи.

```
#Пользователи, которые не хотят использовать
денежные единицы
#и форматы времени Тайваня, могут вручную изменить каждую переменную
me:\
:lang=zh_TW.Big5:\
:lc_all=zh_TW.Big5:\
:lc_collate=zh_TW.Big5:\
:lc_ctype=zh_TW.Big5:\
:lc_messages=zh_TW.Big5:\
:lc_monetary=zh_TW.Big5:\
:lc_numeric=zh_TW.Big5:\
:lc_time=zh_TW.Big5:\
:charset=big5:\
:xmodifiers="@im=xcin": #Setting the XIM Input Server
```

За подробностями обращайтесь к разделу Настройка на уровне администратора и `login.conf(5)`.

#### 18.3.4.1.1.2. Настройка на уровне администратора

Проверьте, что класс логина пользователя в `/etc/login.conf` задает нужный язык. Убедитесь, что эти настройки присутствуют в `/etc/login.conf`:

```
language_name:accounts_title:\
:charset=MIME_charset:\
:lang=locale_name:\
:tc=default:
```

Используя наш предыдущий пример с Latin-1, это должно выглядеть так:

```
german:German Users Accounts:\
```

```
:charset=ISO-8859-1:\
:lang=de_DE.ISO8859-1:\
:tc=default:
```

#### Изменение класса логина с помощью *vipw(8)*

Используйте *vipw* для добавления новых пользователей и отредактируйте запись о пользователе, чтобы она была похожа на эту:

```
user:password:1111:11:language:0:0:User Name:/home/user:/bin/sh
```

#### Изменение класса логина с помощью *adduser(8)*

Используйте *adduser* для добавления новых пользователей и сделайте следующее:

- Установите `defaultclass = язык` в `/etc/adduser.conf`. Помните, что в этом случае вы должны задать `default` класс для всех пользователей других языков.
- Другой способ заключается в задании класса логина каждый раз, когда *adduser(8)* попросит об этом:  
Enter login class: default []:
- Еще один способ - использовать следующее для каждого пользователя другого языка, которого вы хотите добавить:  
**# adduser -class язык**

#### Изменение класса логина с помощью *pw(8)*

Если для добавления новых пользователей вы используете утилиту *pw(8)*, запускайте ее с такими параметрами:

```
# pw useradd
    имя_пользователя -L язык
```

#### 18.3.4.1.2. Метод инициализационных файлов оболочки

**Замечание:** Этот метод не рекомендуется, потому что он требует разных настроек для каждой оболочки. Вместо этого используйте Метод класса логина.

Чтобы добавить локализацию и набор символов MIME, просто установите две переменные окружения, приведенные ниже, в инициализационные файлы оболочек `/etc/profile` и/или `/etc/csh.login`. Для примера мы используем немецкий язык:

В `/etc/profile`:

```
LANG=de_DE.ISO8859-1; export LANG
MM_CHARSET=ISO-8859-1; export MM_CHARSET
```

Или в `/etc/csh.login`:

```
setenv LANG de_DE.ISO8859-1
setenv MM_CHARSET ISO-8859-1
```

Как вариант, вы можете добавить установки, приведенные выше в `/usr/share/skel/dot.profile` (соответствует установкам в `/etc/profile` выше), или `/usr/share/skel/dot.login` (соответствует установкам в `/etc/csh.login` выше).

Для X11:

В `$HOME/.xinitrc`:

```
LANG=de_DE.ISO8859-1; export LANG
```

Или:

```
setenv LANG de_DE.ISO8859-1
```

В зависимости от вашей оболочки (см. выше).

### 18.3.5. Настройка консоли

Для всех однобайтовых кодовых таблиц, укажите подходящие консольные шрифты в `/etc/rc.conf` для требуемого языка:

```
font8x16=имя_шрифта
font8x14=имя_шрифта
font8x8=имя_шрифта
```

The *имя\_шрифта* берется из каталога `/usr/share/syscons/fonts`, без указания расширения `.fnt`.

Убедитесь также в установке правильных значений `keymap` и `screenmap` для вашей однобайтовой кодовой таблицы через `/stand/sysinstall`. В **sysinstall** выберите **Configure**, затем **Console**. Или добавьте следующие строки в `/etc/rc.conf`:

```
scrnmap=screenmap_name
keymap=keymap_name
keychange="fkey_number sequence"
```

*screenmap\_name* берется из каталога `/usr/share/syscons/scrnmaps`, без суффикса `.scm`. Screenmap с соответствующим шрифтом обычно используется в качестве обходного пути для расширения 8 бит до 9 бит в псевдографическую область VGA адаптера, например для удаления букв из этой области если используется 8-битный шрифт.

Если демон мыши **moused** включен добавлением в `/etc/rc.conf` следующей строки:

```
moused_enable="YES"
```

прочтите информацию о курсоре мыши в следующем параграфе.

По умолчанию драйвер `syscons(4)` выделяет для курсора мыши в таблице символов диапазон `0xd0-0xd3`. Если в вашем языке используется этот диапазон, переместите этот диапазон. В FreeBSD версий до 5.0, добавьте следующую строку в файл конфигурации ядра:

```
options SC_MOUSE_CHAR=0x03
```

Для FreeBSD версий 4.4 и выше включите следующую строку в `/etc/rc.conf`:

```
mousechar_start=3
```

`keymap_name` берется из каталога `/usr/share/syscons/keymaps`, без расширения `.kbd`. Если вы не знаете какую таблицу использовать, запустите `kbdmap(1)` для тестирования таблицы без перезагрузки.

`keychange` обычно требуется для сопоставления функциональных клавиш выбранному типу терминала, поскольку последовательности функциональных клавиш не могут быть определены в раскладке клавиатуры.

Убедитесь также, что для всех записей `ttyv*` в `/etc/ttys` установлен подходящий тип терминала. Имеющиеся типы соответствуют следующим кодировкам:

Кодовая таблица	Тип терминала
ISO-8859-1 or ISO-8859-15	cons25l1
ISO-8859-2	cons25l2
ISO-8859-7	cons25l7
KOI8-R	cons25r
KOI8-U	cons25u
CP437 (VGA default)	cons25
US-ASCII	cons25w

Для расширенных кодировок, а также для многобайтовых кодировок, используйте порт FreeBSD из каталога `/usr/ports/language`. Некоторые порты выводят информацию на консоль, хотя для системы выглядят как последовательные псевдотерминалы, следовательно вы должны зарезервировать достаточное количество псевдотерминалов и для X11 и для последовательных псевдоконсолей. Вот неполный список приложений для использования этих кодировок в консоли:

Язык	Приложение
Традиционный китайский (BIG-5)	chinese/big5con
Японский	japanese/kon2-16dot или japanese/mule-freewnn
Корейский	korean/han

### 18.3.6. Настройка X11

Хотя X11 не является частью проекта FreeBSD, мы включили сюда некоторую информацию для пользователей FreeBSD. За детальной информацией обращайтесь к Веб сайту XFree86 (<http://www.xfree86.org/>) или к сайту того сервера X11, который вы используете.

В `~/Xresources`, вы можете внести дополнительные I18N настройки приложений (настройки шрифтов, меню и т.п.).



### 18.3.6.1. Отображение шрифтов

Установите сервер **Xorg** (`x11-servers/xorg-server`) или **XFree86** (`x11-servers/XFree86-4-Server`), затем TrueType шрифты для используемого языка. После установки корректного значения locale меню, сообщения и т.д. должны появиться на соответствующем языке.

### 18.3.6.2. Ввод не-англоязычных символов

Протокол X11 Input Method (XIM) это новый стандарт для всех клиентов X11. Все приложения X11 должны быть написаны как XIM клиенты, получающие ввод с серверов XIM Input. Существует несколько XIM серверов, доступных для различных языков.

### 18.3.7. Настройка принтера

Некоторые однобайтовые кодовые таблицы обычно поддерживаются принтерами на аппаратном уровне. Расширенные, или мультибайтовые кодовые таблицы требуют дополнительной настройки. Мы рекомендуем использование **apsfilter**. Кроме того, вы можете преобразовать документ в формат PostScript или PDF с помощью поддерживающих соответствующий язык программ.

### 18.3.8. Ядро и файловые системы

Файловая система FreeBSD (fast filesystem, FFS) поддерживает 8-битные символы, ее можно использовать с любой односимвольной кодовой таблицей (см. `multibyte(3)`), но кодовые таблицы не хранятся в файловой системе; т.е. ей ничего не известно о порядке расположения символов в кодовой таблице. Официально FFS пока не поддерживает расширенные или много символьные кодовые таблицы. Однако, для некоторых расширенных или многобайтовых кодировок есть неофициальные патчи, включающие их поддержку в FFS. Это всего лишь временные не переносимые решения или исправления кода и мы решили не включать их в дерево исходных текстов. Обращайтесь к сайтам по соответствующим языкам за дальнейшей информацией и патчами.

Файловая система MS-DOS в FreeBSD может быть настроена для поддержки перекодировки между MS-DOS, кодовыми таблицами Unicode и выбранной кодовой таблицей FreeBSD. Обращайтесь к `mount_msdos(8)` за подробностями.

## 18.4. Компиляция I18N программ

Многие порты FreeBSD были созданы с поддержкой I18N. Некоторые из них помечены символами -I18N в имени порта. Эти и многие другие программы имеют встроенную поддержку I18N и не нуждаются в специальной настройке.

Однако, некоторые приложения, такие как **MySQL**, требуют указания кодировки в `Makefile`. Это обычно делается правкой `Makefile` или передачей параметров **configure** при сборке из исходных текстов.

## 18.5. Локализация FreeBSD для поддержки определенных языков

### 18.5.1. Русский язык (кодировка KOI8-R)

*Первоначально предоставил Andrey A. Chernov.*

За дополнительной информацией о кодировке KOI8-R, обращайтесь к странице KOI8-R References (Russian Net Character Set) (<http://koi8.pp.ru/>).

#### 18.5.1.1. Настройка локализации

Поместите в `~/.login_conf` следующие строки:

```
me:My Account:\
:charset=KOI8-R:\
:lang=ru_RU.KOI8-R:
```

Обратитесь к вышеизложенному материалу этой главы за соответствующей информацией.

#### 18.5.1.2. Настройка консоли

- Для FreeBSD версий до 5.0 добавьте к файлу настройки ядра следующую строку:

```
options SC_MOUSE_CHAR=0x03
```

Для FreeBSD версий 4.4 и выше вставьте следующую строку в `/etc/rc.conf`:

```
mousechar_start=3
```

- Используйте в `/etc/rc.conf` такие настройки:

```
keymap="ru.koi8-r"
scrnmap="koi8-r2cp866"
font8x16="cp866b-8x16"
font8x14="cp866-8x14"
font8x8="cp866-8x8"
```

- Для каждого устройства `ttyv*` в файле `/etc/ttys`, используйте `cons25r` в качестве типа терминала.

Обратитесь к вышеизложенному материалу этой главы за примерами настройки консоли.

#### 18.5.1.3. Настройка принтера

Большинство принтеров для России поставляются с аппаратной кодовой страницей CP866, требуется специальный выходной фильтр для перекодировки из KOI8-R в CP866. Такой фильтр устанавливается по умолчанию в `/usr/libexec/lpr/ru/koi2alt`. Запись для принтера в `/etc/printcap` должна выглядеть приблизительно так:

```
lp|Russian local line printer:\
```

```
:sh:of=/usr/libexec/lpr/ru/koi2alt:\
:lp=/dev/lpt0:sd=/var/spool/output/lpd:lf=/var/log/lpd-errs:
```

Обратитесь к `printcap(5)` за подробным описанием.

#### 18.5.1.4. MS-DOS FS и русские имена файлов

Следующий пример записи в `fstab(5)` включает поддержку русских имен файлов при монтировании файловых систем MS-DOS:

```
/dev/ad0s2      /dos/c  msdos   rw,-Wkoi2dos,-Lru_RU.KOI8-R 0 0
```

Параметр `-L` выбирает имя используемой локализации, а параметр `-W` задает таблицу перекодировки. При использовании параметра `-W` убедитесь, что раздел `/usr` смонтирован перед разделом MS-DOS, поскольку таблицы перекодировки расположены в каталоге `/usr/libdata/msdosfs`. За более подробной информацией обращайтесь к странице справочника `mount_msdos(8)`.

#### 18.5.1.5. Настройка X11

1. Выполните сначала не-X настройку локализации.

**Замечание:** Локализация KOI8-R может не работать со старыми релизами **XFree86** (ниже 3.3). На настоящий момент в FreeBSD в качестве системы X Window по умолчанию используется **XFree86 4.X**. Поэтому если только вы не используете старые версии FreeBSD, это не должно стать проблемой.

2. Зайдите в каталог `russian/X.language` и выполните следующую команду:

```
# make install
```

Вышеуказанный порт установит последнюю версию KOI8-R шрифтов. В **XFree86 3.3** уже есть некоторые KOI8-R шрифты, но эти масштабируются лучше.

Проверьте раздел "Files" в файле `/etc/XF86Config`. Следующие строки должны быть добавлены *перед* всеми записями `FontPath`:

```
FontPath  "/usr/X11R6/lib/X11/fonts/cyrillic/misc"
FontPath  "/usr/X11R6/lib/X11/fonts/cyrillic/75dpi"
FontPath  "/usr/X11R6/lib/X11/fonts/cyrillic/100dpi"
```

Если вы используете видеорежим с высоким разрешением, поменяйте местами строки для 75 dpi и 100 dpi.

3. Для настройки ввода на русском языке, добавьте нижеприведенные строки в раздел "Keyboard" файла `XF86Config`.

Для **XFree86 3.X**:

```
XkbLayout  "ru"
XkbOptions "grp:caps_toggle"
```

Для **XFree86 4.X**:

```
Option "XkbLayout"      "ru"
Option "XkbOptions"     "grp:caps_toggle"
```

Убедитесь, что `XkbDisable` отключен (закомментирован) в этом разделе.

Переключателем RUS/LAT служит **CapsLock**. Прежняя функция **CapsLock** все еще доступна в комбинации **Shift+CapsLock** (только в режиме LAT).

Если на вашей клавиатуре есть клавиши "Windows", и некоторые не-алфавитные клавиши работают неправильно в режиме RUS, добавьте следующую строку в файл `XF86Config`.

Для **XFree86 3.X**:

```
XkbVariant "winkeys"
```

Для **XFree86 4.X**:

```
Option "XkbVariant" "winkeys"
```

**Замечание:** Ввод на русском языке может не работать со старыми версиями **XFree86**, прочтите предупреждение выше с более подробной информацией. Ввод на русском языке может также не работать с нелокализованными приложениями. Минимально локализованные приложения должны вызывать функцию `XtSetLanguageProc (NULL, NULL, NULL)`; в начале программы. Обратитесь к KOI8-R for X Window (<http://koi8.pp.ru/xwin.html>) за дальнейшими инструкциями по локализации приложений X11.

## 18.5.2. Локализация на традиционном китайском для Тайваня

У проекта FreeBSD-Taiwan есть Chinese HOWTO для FreeBSD по адресу <http://netlab.cse.yzu.edu.tw/~statue/freebsd/zh-tut/>, в котором используются многие порты на китайском. На данный момент редактором FreeBSD Chinese HOWTO является Shen Chuan-Hsing <[statue@freebsd.sinica.edu.tw](mailto:statue@freebsd.sinica.edu.tw)>.

Chuan-Hsing Shen <[statue@freebsd.sinica.edu.tw](mailto:statue@freebsd.sinica.edu.tw)> создал Chinese FreeBSD Collection (CFC) (<http://netlab.cse.yzu.edu.tw/~statue/cfc/>), используя FreeBSD-Taiwan zh-L10N-tut. Пакеты и скрипты доступны по адресу <ftp://freebsd.csie.nctu.edu.tw/pub/taiwan/CFC/>.

## 18.5.3. Локализация на немецкий язык (для всех ISO 8859-1 языков)

Slaven Rezac <[eserte@cs.tu-berlin.de](mailto:eserte@cs.tu-berlin.de)> написал учебник по использованию umlauts в FreeBSD. Учебник написан на немецком и доступен по адресу <http://www.de.FreeBSD.org/de/umlaut/>.

## 18.5.4. Локализация на японский и корейский языки

За информацией по японской локализации, обратитесь к <http://www.jp.FreeBSD.org/>, а за информацией о корейской локализацией к <http://www.kr.FreeBSD.org/>.

### 18.5.5. Не-англоязычная документация FreeBSD

Отдельные контрибьюторы FreeBSD перевели части документации FreeBSD на другие языки. Эти переводы доступны на основном сайте (<http://www.FreeBSD.org/ru/index.html>) или из каталога `/usr/share/doc`.

# Глава 19. На переднем крае разработок

*Реструктурирование, реорганизацию и частичное обновление выполнил Jim Mock. Оригинальный текст написал Jordan Hubbard, Poul-Henning Kamp, John Polstra, Nik Clayton. Перевод на русский язык: Андрей Захватов.*

## 19.1. Краткий обзор

Между релизами над FreeBSD ведется постоянная работа. Для тех, кто хочет быть на переднем крае, есть несколько простых методов для поддержания своей системы в соответствии с последними разработками. Будьте осторожны — передний край не для всех! Эта глава поможет вам решить, хотите ли вы отслеживать систему в процессе работы над ней или останетесь верным одному из выпущенных релизов.

После чтения этой главы вы будете знать:

- Разницу между двумя ветвями разработки: FreeBSD-STABLE и FreeBSD-CURRENT.
- Как поддерживать вашу систему в актуальном состоянии при помощи **CVSup**, **CVS** или **CTM**.
- Как перестраивать и переустанавливать базовую систему полностью при помощи `make buildworld` (и других).

Перед чтением этой главы вы должны:

- Полностью настроить своё подключение к сети (Гл. 24).
- Знать, как устанавливать дополнительное программное обеспечение других разработчиков (Гл. 4).

## 19.2. FreeBSD-CURRENT против FreeBSD-STABLE

Во FreeBSD имеется две ветки разработки: FreeBSD-CURRENT и FreeBSD-STABLE. Этот раздел описывает каждую из них и объясняет, как синхронизировать вашу систему с любой из веток. Сначала будет обсуждаться ветка FreeBSD-CURRENT, затем FreeBSD-STABLE.

### 19.2.1. Как следовать текущим разработкам во FreeBSD

Пока вы читаете этот текст, помните, что FreeBSD-CURRENT является ”передовым краем” работ над FreeBSD. Предполагается, что пользователи FreeBSD-CURRENT технически более грамотны и могут решать проблемы с системой самостоятельно. Если вы являетесь во FreeBSD новичком, вам лучше сначала дважды подумать, прежде чем её устанавливать.

#### 19.2.1.1. Что такое FreeBSD-CURRENT?

FreeBSD-CURRENT является последними рабочими версиями исходных текстов FreeBSD. Сюда включаются неоконченные работы, экспериментальные изменения и промежуточные механизмы, которые могут присутствовать, а могут и отсутствовать в следующем официальном релизе программного обеспечения. Хотя многие из разработчиков FreeBSD выполняют компиляцию из

исходных текстов FreeBSD-CURRENT ежедневно, случаются периоды, когда исходные тексты заведомо не могут быть откомпилированы. Такие проблемы обычно решаются так быстро, как это возможно, но всё-таки момент, когда вы загрузили исходные тексты FreeBSD-CURRENT, может повлиять на то, содержат они мину замедленного действия или очень нужную функциональность!

### 19.2.1.2. Кому нужна FreeBSD-CURRENT?

FreeBSD-CURRENT предназначается трём основным заинтересованным группам:

1. Участники проекта FreeBSD, активно работающие над некоторой частью дерева исходных текстов и для кого работа в "current" является абсолютной необходимостью.
2. Участники проект FreeBSD, которые являются активными тестерами. Они тратят свое время на исправление проблем для того, чтобы FreeBSD-CURRENT оставалась, насколько это возможно, нормально работающей системой. Есть также люди, которые вносят важные предложения по изменениям и общему направлению развития FreeBSD и присылают свои патчи, реализующие эти изменения.
3. Те, кто просто хотят быть в курсе всех изменений или используют текущие исходные тексты для ознакомительных целей (к примеру, для *чтения*, но не для использования). Такие люди также иногда высказывают замечания или предоставляют код.

### 19.2.1.3. Чем FreeBSD-CURRENT не является?

1. Быстрым способом получить предварительную версию, в случае, если вы услышали, что здесь появилась некая крутая возможность, и вы хотите быть первым в вашем микрорайоне, у кого она есть. Здесь быть первым из тех, кто имеет это программное обеспечение означает также быть первым из тех, кто столкнулся с ошибками в нём.
2. Быстрым способом получения исправлений. Любая версия FreeBSD-CURRENT является в равной мере как источником исправлений существующих ошибок, так и источником появления новых.
3. "Официально поддерживаемой" каким бы то ни было способом. Мы прилагаем все усилия, чтобы помочь тем, кто изначально принадлежит одной из трех "признанных" групп пользователей FreeBSD-CURRENT, но у нас просто *нет времени* на техническую поддержку. Это не потому, что мы гадкие и злые люди, которые ни за что не будут помогать другим (если бы это было так, мы бы не создали FreeBSD). Мы просто не в силах отвечать на сотни сообщений в день *и* работать над FreeBSD! Если бы стоял выбор между тем, отвечать ли на множество вопросов об экспериментально коде или продолжать работу над совершенствованием FreeBSD, большинство разработчиков проголосовало бы за последнее.

### 19.2.1.4. Использование FreeBSD-CURRENT

1. Подпишитесь на списки рассылки `freebsd-current` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-current>) и `cvcs-all` (<http://lists.FreeBSD.org/mailman/listinfo/cvcs-all>). Это не просто хорошая идея, это *необходимость*. Если вы не являетесь участником списка рассылки `freebsd-current` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-current>), то вы не увидите замечаний, высказываемых

о текущем состоянии системы и в итоге можете столкнуться со множеством проблем, которые уже были найдены и решены другими. Ещё хуже, если вы пропустите важные сообщения, касающиеся жизнеспособности вашей системы.

Список рассылки `cvvs-all` (<http://lists.FreeBSD.org/mailman/listinfo/cvvs-all>) позволит вам для каждого изменения увидеть соответствующую запись в журнале коммитов, а они порой содержат относящуюся к делу информацию о возможных побочных эффектах.

Чтобы подключиться к этим и другим доступным спискам рассылки, перейдите по ссылке <http://lists.FreeBSD.org/mailman/listinfo> и щёлкните на списке, к которому вы хотите подключиться. Инструкции по дальнейшим действиям размещены там же.

2. Загрузите исходные тексты с зеркального сайта FreeBSD. Вы можете сделать это одним из следующих двух способов:
  - а. При помощи программы `cvsup` с `sup`-файлом `standard-supfile`, который можно найти в каталоге `/usr/share/examples/cvsup`. Это наиболее рекомендуемый метод, так как он позволяет вам загрузить набор исходных текстов один раз полностью, а затем загружать только произошедшие изменения. Многие запускают `cvsup` при помощи программы `cron` и получают самые свежие исходные тексты автоматически. Измените примерный файл `supfile` выше и отконфигурируйте `cvsup` для вашего окружения.
  - б. При помощи **СТМ**. Если у вас очень плохое подключение (дорогое или предоставляющее доступ только к электронной почте), то **СТМ** можно рассматривать как вариант. Однако в нем много "подводных камней", и его использование может привести к появлению неправильных файлов. Это привело к тому, что этот способ используется редко, что, в свою очередь, увеличивает шанс появления периодов его неработы. Мы рекомендуем использовать **CVSup** всем, чья скорость подключения равна 9600 bps и выше.
3. Если вам нужны исходные тексты для компиляции и запуска, а не просто для ознакомления, то загружайте исходные тексты ветки `FreeBSD-CURRENT` *полностью*, а не отдельные ее части. Причиной является то, что многие части исходных текстов зависят от других обновлений где-то еще, и попытка компиляции лишь некоторой части программ в этом случае гарантированно вызовет проблемы.
 

Перед тем, как компилировать `FreeBSD-CURRENT`, внимательно прочтите файл `Makefile` в каталоге `/usr/src`. В процессе обновления вы по крайней мере раз должны пройти через установку нового ядра и перестроение всех компонентов системы. Чтение списка рассылки `freebsd-current` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-current>) и `/usr/src/UPDATING` позволит вам быть в курсе всех процедур, которые иногда бывают необходимы в процессе работы над следующим релизом.
4. Будьте активным подписчиком! Если вы работаете с `FreeBSD-CURRENT`, мы хотим знать, что вы думаете о ней, особенно если у вас есть соображения по ее улучшению или исправлению ошибок. Пожелания, к которым прилагается код, всегда принимаются с большим энтузиазмом!



## 19.2.2. Работа с веткой **stable** во FreeBSD

### 19.2.2.1. Что такое FreeBSD-STABLE?

FreeBSD-STABLE является нашей веткой разработки, из которой делаются основные релизы. Изменения в этой ветке происходят с разной скоростью, и при этом предполагается, что сначала они были выполнены для FreeBSD-CURRENT в целях тестирования. Однако эта ветка *остаётся* веткой для разработки, а это значит, что в любой момент времени исходные тексты FreeBSD-STABLE могут оказаться неприменимы для некоторой задачи. Это просто ещё одна ветка при разработке, а не ресурс для конечных пользователей.

### 19.2.2.2. Кому нужна FreeBSD-STABLE?

Если вы заинтересованы в отслеживании процесса разработки FreeBSD или хотите принять в нём участие, особенно в той мере, насколько это касается выпуска следующего релиза FreeBSD с "точкой", то вам необходимо отслеживать FreeBSD-STABLE.

Хотя правда то, что исправления, касающиеся безопасности, также делаются и в ветке FreeBSD-STABLE, вам *не нужно* для этого отслеживать FreeBSD-STABLE. Каждый бюллетень по безопасности FreeBSD описывает, как решить проблему для тех релизов, которых он касается <sup>1</sup>, а отслеживание ветки разработки в полном объёме только ради исправлений пробелов в безопасности приводит к появлению большого количества дополнительных ненужных изменений.

Хотя мы прилагаем все усилия, чтобы ветка FreeBSD-STABLE всегда компилировалась и работала, этого нельзя гарантировать. Кроме того, несмотря на то, что перед включением в FreeBSD-STABLE, код разрабатывается в FreeBSD-CURRENT, гораздо большее количество людей работают с FreeBSD-STABLE, чем с FreeBSD-CURRENT. Поэтому неудивительно, что в FreeBSD-STABLE иногда обнаруживаются ошибки и всплывают непредвиденные ситуации, которые не проявляли себя в FreeBSD-CURRENT.

По этим причинам мы *не* рекомендуем слепо отслеживать FreeBSD-STABLE, и, что особенно важно, вы не должны обновлять какие-либо сервера, находящиеся в активной эксплуатации, до FreeBSD-STABLE без предварительного тщательного тестирования кода в вашей среде разработки.

Если у вас нет возможности сделать это, то мы рекомендуем работать с самой последним релизом FreeBSD и использовать механизм обновления бинарных файлов для перехода от релиза к релизу.

### 19.2.2.3. Использование FreeBSD-STABLE

1. Подпишитесь на список рассылки freebsd-stable (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable>). Это позволит вам узнавать о зависимостях процесса компиляции, которые могут появиться в ветке FreeBSD-STABLE или любых других проблемах, требующих особого внимания. В этом списке рассылки разработчики также делают объявления о спорных исправлениях или добавлениях, давая пользователям возможность высказать свое мнение о возможных тонких моментах.

Список рассылки cvs-all (<http://lists.FreeBSD.org/mailman/listinfo/cvs-all>) позволит вам для каждого изменения увидеть соответствующую запись в журнале коммитов, а они порой содержат относящуюся к делу информацию о возможных побочных эффектах.

Чтобы подключиться к этим и другим доступным спискам рассылки, перейдите по ссылке <http://lists.FreeBSD.org/mailman/listinfo> и щёлкните на списке, к которому вы хотите подключиться. Инструкции по дальнейшим действиям размещены там же.

2. Если вы устанавливаете новую систему и хотите, чтобы она работала максимально стабильно, то можете просто загрузить самый свежий снэпшот ветки по адресу

(<ftp://snapshots.jp.FreeBSD.org/pub/FreeBSD/snapshots/>) и установить его, как любой другой релиз. Либо вы можете установить самый последний релиз FreeBSD-STABLE, загрузив его с зеркалирующих сайтов, а затем следовать инструкциям ниже по обновлению исходных текстов вашей системы до самой последней версии FreeBSD-STABLE.

Если вы уже работаете с предыдущим релизом FreeBSD и хотите обновить его из исходных текстов, то вы можете легко это сделать с зеркального сайта FreeBSD. Это можно сделать одним из двух способов:

- a. При помощи программы `cvsup` с `sup`-файлом `stable-supfile` из каталога `/usr/share/examples/cvsup`. Это наиболее рекомендуемый метод, так как он позволяет вам загрузить набор исходных текстов один раз полностью, а затем загружать только произошедшие изменения. Многие запускают `cvsup` при помощи программы `cron` и получают самые свежие исходные тексты автоматически. Измените примерный файл `supfile` выше и отконфигурируйте `cvsup` для вашего окружения.
  - b. При помощи **CTM**. Если у вас нет быстрого и недорогого подключения к Интернет, то это как раз тот метод, которым вы должны воспользоваться.
3. Итак, если вам нужен быстрый доступ к исходным текстам и нагрузка на каналы связи для вас не проблема, то используйте `cvsup` или `ftp`. В противном случае воспользуйтесь **CTM**.
  4. Перед тем, как компилировать FreeBSD-STABLE, внимательно прочтите файл `Makefile` в каталоге `/usr/src`. В процессе обновления вы по крайней мере раз должны пройти через установку нового ядра и перестроение всех компонентов системы. Чтение списка рассылки `freebsd-stable` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable>) и `/usr/src/UPDATING` позволит вам быть в курсе всех процедур, которые иногда бывают необходимы при переходе к следующему релизу.

### 19.3. Синхронизация ваших исходных текстов

Имеются различные способы использования Интернет (или почтового) подключения для того, чтобы иметь самые последние версии исходных текстов любого проекта FreeBSD, в зависимости от того, чем вы интересуетесь. Основной сервис, который мы предлагаем, это Анонимный CVS, CVSup и CTM.

**Внимание:** Хотя имеется возможность обновлять только часть дерева исходных текстов, процедурой, которую мы настоятельно советуем, является обновление всего дерева и перекомпиляция пользовательских программ (то есть тех, которые работают в пространстве имен пользователя, например те, что находятся в каталогах `/bin` и `/sbin`) и ядра. Обновление только части дерева исходных текстов, только текстов ядра или

только текстов пользовательских программ часто приводит к возникновению проблем. Эти проблемы могут варьироваться от ошибок компиляции до аварийных остановов системы или порчи данных.

**Анонимный CVS** и **CVSup** используют модель *pull* обновления исходных текстов. В случае **CVSup** пользователь (или скрипт программы **cron**) вызывают **cvsup**, а она работает с каким-либо сервером **cvsupd**, чтобы выполнить обновление ваших файлов. Обновления, которые вы получаете, актуальны с точностью до минуты, и вы получаете их тогда и только тогда, когда сами захотите. Вы можете с легкостью ограничить обновления конкретными файлами или каталогами, которые представляют для вас интерес. Обновления создаются на лету сервером согласно тому, что у вас есть и что вы хотите иметь. **Анонимный CVS** гораздо проще, чем **CVSup** в том смысле, что он представляет собой всего лишь расширение **CVS**, позволяющее загрузить изменения непосредственно с удаленного хранилища **CVS**. **CVSup** может делать это гораздо более эффективно, однако **анонимным CVS** легче пользоваться.

**СТМ**, с другой стороны, не сравнивает последовательно исходные тексты, имеющиеся у вас, с теми, что находятся в главном архиве и вообще ни коим образом не касается наших серверов. Вместо этого несколько раз в день на главной машине **СТМ** запускается скрипт, находящий изменения в файлах с момента своего предыдущего запуска; все замеченные изменения сжимаются, помечаются последовательным номером и кодируются для передачи по электронной почте (в форме печатаемых символов ASCII). После получения эти "дельта-файлы **СТМ**" могут быть переданы утилите **stm\_gmail(1)**, которая осуществит автоматическое декодирование, проверку и применение изменений к пользовательской копии исходных текстов. Этот процесс гораздо более эффективен, чем **CVSup**, и требует меньше ресурсов нашего сервера, так как он сделан по модели *push*, а не *pull*.

Несомненно, есть и минусы. Если вы случайно уничтожили часть вашего архива, то **CVSup** обнаружит и загрузит поврежденную часть. **СТМ** этого делать не будет, и если вы уничтожили какую-то часть вашего дерева исходных текстов (и у вас нет архивной копии), то вам нужно будет начать с самого начала (с последнего "базового дельта-файла"), перестроив всё с помощью **СТМ**, или, используя **анонимный CVS**, просто удалить поврежденную часть и пересинхронизироваться.

## 19.4. Пересборка "world"

После того, как вы синхронизировали ваше локальное дерево исходных текстов с некоторой версией FreeBSD (FreeBSD-STABLE, FreeBSD-CURRENT и так далее), то можете использовать эти исходные тексты для перестроения системы.

**Создайте резервную копию:** Невозможно переоценить важность создания резервной копии вашей системы до того, как вы будете это делать. Хотя перестроение системы (пока вы следуете этим инструкциям) является простой задачей, вы всегда можете допустить ошибку, или ошибка может оказаться в исходных текстах, что может привести к тому, что система перестанет загружаться.

Обязательно сделайте резервную копию. И держите под рукой аварийную (**fixit**) дискету или загрузочный компакт диск. Может быть, вам никогда не приходилось ими пользоваться, но, постучав по дереву, всегда лучше подготовиться, чем потом сожалеть.

**Подпишитесь на соответствующий список рассылки:** Ветки FreeBSD-STABLE и FreeBSD-CURRENT кода по природе своей являются *изменяющимися*. В разработке FreeBSD участвуют люди, и время от времени случаются ошибки.

Иногда эти ошибки достаточно безобидны и приводят к выводу нового диагностического сообщения. Бывает, что изменение оказывается катастрофическим, и система не может загрузиться или разрушаются файловые системы (или что-нибудь ещё хуже).

Если возникают подобные проблемы, в соответствующем списке рассылки публикуется сообщение "heads up", в котором описывается природа проблемы и затрагиваемые системы. Когда проблема решается, публикуется сообщение "all clear".

Если вы пытаетесь отслеживать FreeBSD-STABLE или FreeBSD-CURRENT и не читаете Список рассылки, посвящённый обсуждению FreeBSD-STABLE (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable>) или Список рассылки, посвящённый обсуждению FreeBSD-CURRENT (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-current>) соответственно, то вы напрашиваетесь на неприятности.

**Не используйте `make world`:** Множество старой документации рекомендует использование `make world`. При этом пропускаются многие важные шаги, и использование этой команды возможно лишь в том случае, если вы точно знаете, что делаете. Почти во всех обстоятельствах `make world` это неправильный способ, вместо него необходимо использовать описанную здесь процедуру.

### 19.4.1. Канонический способ обновления вашей системы

Для обновления вашей системы вы должны воспользоваться следующей процедурой:

```
# make buildworld
# make buildkernel
# make installkernel
# reboot
```

Вы должны загрузиться в однопользовательский режим (например, по команде `boot -s` из приглашения загрузчика). Затем выполните:

```
# mergemaster -p
# make installworld
# mergemaster
# reboot
```

**Прочтите более полное описание:** Описанная выше последовательность является только краткой выжимкой для того, чтобы помочь вам начать. Вы должны всё же прочесть последующие разделы для полного понимания каждого шага, особенно если собираетесь использовать собственную конфигурацию ядра.

### 19.4.2. Прочтите `/usr/src/UPDATING`

Перед тем, как делать что-либо, прочтите `/usr/src/UPDATING` (или соответствующий файл в вашей копии исходных текстов). В этом файле содержится важная информация о проблемах, с которыми вы можете столкнуться, или указан порядок, в котором вы должны запускать определенные команды. Если в файле `UPDATING` написано нечто, противоречащее тому, что вы здесь читаете, то нужно следовать указаниям в `UPDATING`.

**Важно:** Чтение `UPDATING` не заменит подписки на соответствующий список рассылки, как это и описано выше. Эти два условия являются дополняющими, а не взаимоисключающими друг друга.

### 19.4.3. Проверьте содержимое `/etc/make.conf`

Просмотрите файлы `/usr/share/examples/etc/make.conf` (`/etc/defaults/make.conf` в FreeBSD 4.X) и `/etc/make.conf`. Первый содержит некоторые предопределенные по умолчанию значения – большинство из них закомментировано. Чтобы воспользоваться ими при перестроении системы из исходных текстов, добавьте их в файл `/etc/make.conf`. Имейте в виду, что все, добавляемое вами в `/etc/make.conf`, используется также каждый раз при запуске команды `make`, так что полезно задать здесь значения, подходящие вашей системе.

Вероятно стоит скопировать строки `CFLAGS` и `NOPROFILE`, расположенные в `/usr/share/examples/etc/make.conf` (или `/etc/defaults/make.conf` в FreeBSD 4.X), в файл `/etc/make.conf` и раскомментировать их.

Посмотрите на другие определения (`CORTFLAGS`, `NOPORTDOCS` и так далее) и решите, нужны ли они вам.

### 19.4.4. Обновите файлы в каталоге `/etc`

Каталог `/etc` содержит значительную часть информации о конфигурации вашей системы, а также скрипты, работающие в начале работы системы. Некоторые из этих скриптов меняются от версии к версии FreeBSD.

Некоторые конфигурационные файлы также используются в ежедневной работе системы. В частности, файл `/etc/group`.

Случалось, что установочная часть `make installworld` ожидала существования определённых имен пользователей или групп. При обновлении существует вероятность, что эти пользователи или группы не существуют. Это вызывает проблемы при обновлении. В некоторых случаях `make buildworld` проверяет наличие этих пользователей или групп.

Свежим примером этого является добавление пользователя `smmsp`. Пользователи столкнулись с прерыванием процесса установки, когда `mtree(8)` пыталась создать `/var/spool/clientmqueue`.

Выходом является просмотр файла `/usr/src/etc/group` и сравнение списка групп в нем с вашим собственным. Если в новом файле есть группы, отсутствующие в вашем, то скопируйте их. Таким же образом вы должны переименовывать все группы в `/etc/group`, которые имеют тот же самый `GID`, но другое название в `/usr/src/etc/group`.

Начиная с 4.6-RELEASE, вы можете запустить `mergemaster(8)` в режиме, предвещающем построение системы, задаваемым опцией `-p`. Она будет сравнивать только те файлы, которые необходимы для

успешного выполнения целей `buildworld` или `installworld`. Если ваша старая версия утилиты `mergemaster` не поддерживает опцию `-p`, воспользуйтесь новой версией из дерева исходных текстов при первом запуске:

```
# cd /usr/src/usr.sbin/mergemaster
# ./mergemaster.sh -p
```

**Подсказка:** Если вы параноик, можете поискать файлы, владельцем которых является та группа, которую вы переименовываете или удаляете:

```
# find / -group GID -print
```

выдаст список всех файлов, владельцем которых является группа `GID` (задаваемая именем или численным значением ID).

### 19.4.5. Перейдите в однопользовательский режим

Вам может понадобиться откомпилировать систему в однопользовательском режиме. Кроме обычного выигрыша в скорости процесса, переустановка системы затрагивает много важных системных файлов, все стандартные выполнимые файлы системы, библиотеки, `include`-файлы и так далее. Изменение их на работающей системе (в частности, в которой активно работают пользователи) может привести к неприятностям.

Другим способом является компиляция системы в многопользовательском режиме с последующим переходом в однопользовательский режим для выполнения установки. Если вы хотите поступить именно так, просто следуйте инструкциям до момента окончания построения. Вы можете отложить переход в однопользовательский режим до завершения целей `installkernel` или `installworld`.

Как администратор, вы можете выполнить:

```
# shutdown now
```

на работающей системе, что переведет ее в однопользовательский режим.

Либо вы можете выполнить перезагрузку и в приглашении загрузчика задать флаг `-s`. После этого система загрузится в однопользовательском режиме. В приглашении командного процессора вы должны запустить:

```
# fsck -p
# mount -u /
# mount -a -t ufs
# swapon -a
```

Эти команды выполняют проверку файловых систем, повторно монтируют `/` в режиме чтения/записи, монтируют все остальные файловые системы UFS, перечисленные в файле `/etc/fstab` и включают подкачку.

**Замечание:** Если часы в вашей CMOS настроены на местное время, а не на GMT (это имеет место, если команда `date(1)` выдаёт неправильное время и зону), то вам может понадобиться запустить следующую команду:

```
# adjkerntz -i
```

Это обеспечит корректную настройку местного часового пояса — без этого впоследствии вы можете столкнуться с некоторыми проблемами.

### 19.4.6. Удалите `/usr/obj`

При перестроении частей системы они помещаются в каталоги, которые (по умолчанию) находятся в `/usr/obj`. Структура повторяет структуру `/usr/src`.

Вы можете ускорить выполнение процесса `make buildworld` и, возможно, избавить себя от некоторой головной боли, связанной с зависимостями, удалив этот каталог.

На некоторых файлах из `/usr/obj` могут быть установлены специальные флаги (обратитесь к `chflags(1)` за дополнительной информацией), которые сначала должны быть сняты.

```
# cd /usr/obj
# chflags -R noschg *
# rm -rf *
```

### 19.4.7. Перекомпилируйте исходные тексты

#### 19.4.7.1. Сохраните вывод

Неплохо сохранить вывод, получаемый при работе программы `make(1)`, в файл. Если что-то вдруг пойдет не так, вы будете иметь копию сообщения об ошибке и полную картину того, где она произошла. Хотя это может и не помочь в определении причин происходящего, это может помочь другим, если вы опишите вашу проблему в одном из списков рассылки FreeBSD.

Проще всего это сделать при помощи команды `script(1)` с параметром, в котором указано имя файла, в который нужно сохранить вывод. Вы должны сделать это непосредственно перед тем, как перестроить систему, а по окончании процесса набрать **exit**.

```
# script /var/tmp/mw.out
Script started, output file is /var/tmp/mw.out
# make world
... compile, compile, compile ...
# exit
Script done, ...
```

Если вы делаете это, *не* сохраняйте вывод в `/tmp`. Этот каталог может быть очищен при следующей перезагрузке. Лучше сохранить его в `/var/tmp` (как в предыдущем примере) или в домашнем каталоге пользователя `root`.

### 19.4.7.2. Компиляция базовых компонентов системы

Вы должны находиться в каталоге `/usr/src`:

```
# cd /usr/src
```

(если, конечно, ваш исходный код не находится в другом месте, в случае чего вам нужно перейти в соответствующий каталог).

Для полного перестроения системы используется команда `make(1)`. Эта команда читает инструкции из файла `Makefile`, описывающего, как должны быть перестроены программы, которые составляют систему FreeBSD, в каком порядке они должны быть построены и так далее.

Общий формат командной строки, которую вы будете набирать, таков:

```
# make -x -DVARIABLE target
```

В этом примере `-x` является параметром, который вы передаете в `make(1)`. Обратитесь к справочной странице программы `make(1)`, которая содержит список возможных параметров.

`-DVARIABLE` передает переменную в `Makefile`. Поведение `Makefile` определяется этими переменными. Это те же самые переменные, которые задаются в `/etc/make.conf`, и это — еще один способ их задания.

```
# make -DNOPROFILE=true target
```

является другим способом указания того, что библиотеки для профилирования строить не нужно, и соответствует строке

```
NOPROFILE=      true # Обход построения библиотек для профилирования
```

в файле `/etc/make.conf`.

`target` указывает программе `make(1)` на то, что вы хотите сделать. Каждый файл `Makefile` определяет некоторое количество различных "целей", и ваш выбор цели определяет то, что будет делаться.

Некоторые цели, перечисленные в файле `Makefile`, не предназначены для вызова. Просто они используются в процессе построения для разбиения его на этапы.

В большинстве случаев вам не нужно передавать никаких параметров в `make(1)`, так что ваша команда будет выглядеть примерно так:

```
# make target
```

Начиная с версии FreeBSD 2.2.5 (на самом деле впервые это было сделано в ветке FreeBSD-CURRENT, а затем адаптировано в FreeBSD-STABLE где-то между 2.2.2 и 2.2.5) цель `world` была разделена на две: `buildworld` и `installworld`. Начиная с версии FreeBSD 5.3, `world` изменена так, что она более не работает, поскольку для большинства пользователей ее выполнение представляет опасность.

Как указывают на это названия, `buildworld` строит полностью новое дерево в каталоге `/usr/obj`, а `installworld` устанавливает это дерево на используемой машине.

Это весьма полезно по двум причинам. Во-первых, это позволяет вам безопасно строить систему, зная, что компоненты вашей рабочей системы затронуты не будут. Построение "самодостаточно". По этой



причине вы можете спокойно запустить `buildworld` на машине, работающей в многопользовательском режиме без опаски получить какие-либо проблемы. Но всё же рекомендуется запускать цель `installworld` в однопользовательском режиме.

Во-вторых, это позволяет вам использовать монтирование по NFS для обновления многих машин в сети. Если у вас есть три машины, А, В и С, которые вы хотите обновить, запустите `make buildworld` и `make installworld` на машине А. Хосты В и С должны будут затем смонтировать по NFS каталоги `/usr/src` и `/usr/obj` с машины А, и вы сможете запустить `make installworld` для установки результатов построения на машинах В и С.

Хотя цель `world` всё ещё имеется в наличии, вам настоятельно рекомендуется не пользоваться ею.

Выполните

```
# make buildworld
```

В настоящее время имеется возможность задавать команде `make` параметр `-j`, который приводит к запуску нескольких одновременно работающих процессов. Наиболее полезно использовать это на многопроцессорных машинах. Однако, так как процесс компиляции больше всего требователен к подсистеме ввода/вывода, а не к производительности процессора, это можно использовать и на машинах с одним процессором.

На типичной машине с одним CPU вы должны запускать:

```
# make -j4 buildworld
```

`make(1)` будет иметь до 4 одновременно работающих процессов. Эмпирические замеры, опубликованные как-то в списке рассылки, показывают, что в среднем это дает наибольшее увеличение производительности.

Если у вас многопроцессорная машина и вы используете ядро с настройками для SMP, попробуйте использовать значения между 6 и 10 и посмотрите, как это отразится на скорости работы.

Имейте в виду, что это все еще экспериментальная возможность, и изменения в дереве исходных текстов иногда могут быть не совместимы с параллельной сборкой. Если система не может быть построена с использованием этого параметра, попробуйте еще раз без него перед тем, как сообщать о проблемах.

### 19.4.7.3. Время на построение

На время компиляции влияет множество факторов, но на данный момент Pentium III частотой 500 МГц и 128 МБ ОЗУ справляется с построением дерева FreeBSD-STABLE примерно за 2 часа без дополнительных хитростей и убыстряющих процесс уловок. Дерево FreeBSD-CURRENT строится несколько дольше.

## 19.4.8. Откомпилируйте и установите новое ядро

Чтобы получить полную отдачу от вашей новой системы, вы должны перекомпилировать ядро. Это практически необходимость, так как отдельные структуры в памяти могут меняться, и программы типа `ps(1)` и `top(1)` не будут работать, пока версии ядра и исходных текстов системы не будут совпадать.

Самым простым и надежным способом сделать это является компиляция и установка ядра на основе GENERIC. Хотя в GENERIC могут оказаться не все необходимые для работы вашей системы устройства, в нем имеется все необходимое для перезагрузки вашей системы обратно в однопользовательский режим. Это является хорошей проверкой на правильность работы новой системы. После загрузки с ядром GENERIC и проверки работоспособности системы вы можете построить новое ядро на основе вашего обычного конфигурационного файла ядра.

В современных версиях FreeBSD важно выполнить buildworld перед сборкой нового ядра.

**Замечание:** Если вы хотите построить собственное ядро и уже подготовили файл конфигурации, просто используйте `KERNCONF=MYKERNEL` следующим образом:

```
# cd /usr/src
# make buildkernel KERNCONF=MYKERNEL
# make installkernel KERNCONF=MYKERNEL
```

Для FreeBSD 4.2 и более старых версий вы должны заменять `KERNCONF=` на `KERNEL=`. 4.2-STABLE, загруженная до 2 февраля 2001 года, не распознаёт `KERNCONF=`.

Заметьте, что, если вы установили `kern.securelevel` в значение, превышающее 1, и установили флаг `noschg` или подобный на бинарный файл ядра, то вы будете вынуждены перейти в однопользовательский режим для того, чтобы воспользоваться `installkernel`. В противном случае вы должны выполнять эти команды без проблем. Обратитесь к справочным страницам об `init(8)` для получения подробной информации о `kern.securelevel` и `chflags(1)` для получения информации о различных флагах файлов.

Если вы производите обновление до версии FreeBSD ниже 4.0, то нужно использовать старую процедуру построения ядра. Однако рекомендуется использовать новую версию программы `confi g(8)` со следующей командной строкой.

```
# /usr/obj/usr/src/usr.sbin/config/configKERNELNAME
```

### 19.4.9. Перезагрузитесь в однопользовательский режим

Для проверки работоспособности ядра вы должны перезагрузить систему и перейти в однопользовательский режим. Сделайте это, следуя указаниям в Разд. 19.4.5.

### 19.4.10. Установите новые версии системных программ

Если вы компилировали достаточно свежую версию FreeBSD, в которой имеется команда `make buildworld`, то для установки новых версий программ вы должны теперь выполнить команду `installworld`.

Запустите

```
# cd /usr/src
# make installworld
```

**Замечание:** Если при выполнении команды `make buildworld` вы задавали значения каких-либо переменных, то при выполнении `make installworld` вы должны задать те же самые переменные. Это не всегда так для остальных параметров; например, при выполнении `installworld` никогда не должен использоваться параметр `-j`.

Например, если вы выполняли команду:

```
# make -DNOPROFILE buildworld
```

то результат её выполнения должен устанавливаться командой

```
# make -DNOPROFILE installworld
```

В противном случае будет делаться попытка установить библиотеки для профилирования, которые не компилировались на этапе выполнения команды `make buildworld`.

### 19.4.11. Обновите файлы, не обновленные по команде `make installworld`

При перестроении системы не будут обновляться некоторые каталоги (в частности, `/etc`, `/var` и `/usr`) с конфигурационными файлами.

Самым простым способом обновить такие файлы является запуск утилиты `mergemaster(8)`, хотя можно сделать это и вручную, если вам так больше нравится. Вне зависимости от выбранного вами способа обязательно сделайте резервную копию каталога `/etc` на случай, если произойдёт что-то непредвиденное.

#### 19.4.11.1. `mergemaster`

*Текст предоставил Tom Rhodes.*

Утилита `mergemaster(8)` является скриптом для оболочки Боурна, которая поможет вам в определении разницы между вашими конфигурационными файлами в каталоге `/etc` и конфигурационными файлами из дерева исходных текстов `/usr/src/etc`. Это является рекомендуемым способом синхронизации системных конфигурационных файлов с теми, что размещены в дереве исходных текстов.

`mergemaster` была интегрирована в базовый комплект системы FreeBSD между выпусками 3.3-RELEASE и 3.4-RELEASE, что означает её наличие во всех системах `-STABLE` и `-CURRENT`, начиная с 3.3.

Для начала просто наберите `mergemaster` в приглашении командной строки и посмотрите, что происходит. `mergemaster` построит временное окружение для пользователя `root`, начиная от `/`, а затем заполнит его различными системными конфигурационными файлами. Эти файлы затем будут сравниваться с теми, что установлены в вашей системе. В этот момент файлы, которые имеют отличия, будут выданы в формате `diff(1)`, где знак `+` будет означать добавленные или изменённые строки, а знак `-` будет означать строки, которые были либо полностью удалены, либо заменены на новые. Обратитесь к страницам справочной системы по команде `diff(1)` для получения более полной информации о синтаксисе команды `diff(1)` и формате выдачи отличий в файлах.

Затем `mergemaster(8)` выдаст вам каждый файл, в котором есть изменения, и в этот момент у вас есть возможность либо удалить новый файл (который будем считать временным), установить временный

файл в его неизменённом виде, объединить временный файл с установленным на данный момент, либо просмотреть выдачу diff(1) ещё раз.

Выбор удаления временного файла укажет mergemaster(8) на то, что мы хотим оставить наш текущий файл без изменений и удалить его новую версию. Делать это не рекомендуется, если только у вас нет причин вносить изменения в текущий файл. Вы можете получить помощь в любое время, набрав ? в приглашении mergemaster(8). Если пользователь выбирает пропуск файла, запрос появится снова после того, как будут обработаны все остальные файлы.

Выбор установки немодифицированного временного файла приведёт к замене текущего файла новым. Для большинства немодифицированных файлов это является подходящим вариантом.

Выбор варианта с объединением файла приведёт к вызову текстового редактора, содержащего текст обоих файлов. Теперь вы можете объединить их, просматривая оба файла на экране, и выбирая те части из обоих, что подходят для окончательного варианта. Когда файлы сравниваются на экране, то нажатие **l** выбирает содержимое слева, а нажатие **r** выбирает содержимое справа. В окончательном варианте будет файл, состоящий из обеих частей, который и будет установлен. Этот вариант используется для файлов, настройки в которых изменялись пользователем.

Выбор повторного просмотра diff(1)-разниц выдаст вам разницы между файлами, как это делала утилита mergemaster(8) до того, как запросила вас о выборе.

После того, как утилита mergemaster(8) закончит работу с системными файлами, она выдаст запрос относительно других параметров. mergemaster(8) может запросить вас относительно перестроения файла паролей и/или запуска MAKEDEV(8) при использовании FreeBSD версий, меньших чем 5.0, и завершит запросом на удаление оставшихся временных файлов.

#### 19.4.11.2. Обновление в ручном режиме

Однако если вы хотите произвести обновление вручную, то вы не можете просто скопировать файлы из /usr/src/etc в /etc и получить работающую систему. Некоторые из этих файлов сначала нужно "установить". Это нужно потому, что каталог /usr/src/etc *не* является копией того, что должен содержать ваш каталог /etc. Кроме того, есть файлы, которые должны присутствовать в /etc, но которых нет в /usr/src/etc.

Если вы используете mergemaster(8) (как это рекомендуется), то вы можете перейти сразу к следующему разделу.

Вручную проще всего сделать это, установив файлы в новый каталог, а затем пройти по ним, отмечая разницу.

**Сделайте резервную копию вашего каталога /etc:** Хотя, в теории, никаких автоматических действий с этим каталогом не производится, всегда лучше чувствовать себя уверенным. Так что скопируйте имеющийся каталог /etc в какое-нибудь безопасное место. Запустите что-то вроде:

```
# cp -Rp /etc /etc.old
```

-R задает выполнение рекурсивного копирования, а -p сохраняет даты, владельца файлов и тому подобное.

Вам нужно создать шаблонную структуру каталогов для установки нового содержимого `/etc` и других файлов. Подходящим местом является `/var/tmp/root`, и в нём потребуется разместить некоторое количество подкаталогов.

```
# mkdir /var/tmp/root
# cd /usr/src/etc
# make DESTDIR=/var/tmp/root distrib-dirs distribution
```

Эти команды приведут к созданию нужной структуры каталогов и установке файлов. Множество каталогов, созданных в `/var/tmp/root`, будут пустыми и должны быть удалены. Проще всего сделать это так:

```
# cd /var/tmp/root
# find -d . -type d | xargs rmdir 2>/dev/null
```

Эти команды удалят все пустые каталоги. (Стандартный поток диагностических сообщений перенаправляется в `/dev/null` для исключения предупреждений о непустых каталогах.)

Теперь `/var/tmp/root` содержит все файлы, которые должны быть помещены в соответствующие места в `/`. Теперь пройдите по каждому из этих файлов и определите, чем они отличаются от имеющихся у вас файлов.

Заметьте, что некоторые из файлов, которые были установлены в каталог `/var/tmp/root`, имеют первым символом `'''`. На момент написания единственными такими файлами являлись файлы начальных скриптов командных процессоров в `/var/tmp/root/` и `/var/tmp/root/root/`, хотя могут быть и другие (зависит от того, когда вы это читаете). Обязательно пользуйтесь командой `ls -a`, чтобы выявить их.

Проще всего сделать это путём сравнения двух файлов при помощи команды `diff(1)`:

```
# diff /etc/shells /var/tmp/root/etc/shells
```

Эта команда покажет разницу между вашим файлом `/etc/shells` и новым файлом `/var/tmp/root/etc/shells`. Используйте это для определения того, переносить ли сделанные вами изменения или скопировать поверх вашего старого файла.

**Называйте новый корневой каталог (`/var/tmp/root`) по дате, чтобы вы смогли легко выявить разницу между версиями:** Частое перестроение системы означает также и частое обновление `/etc`, которое может быть несколько обременительным.

Вы можете ускорить этот процесс, сохраняя копию последнего набора измененных файлов, которые вы перенесли в `/etc`. Следующая процедура подаст вам одну идею о том, как это сделать.

1. Выполните перестроение системы обычным образом. Когда вы вам потребуется обновить `/etc` и другие каталоги, дайте целевому каталогу имя на основе текущей даты. Если вы делаете это 14 февраля 1998 года, то вы можете сделать следующее:

```
# mkdir /var/tmp/root-19980214
# cd /usr/src/etc
# make DESTDIR=/var/tmp/root-19980214 \
    distrib-dirs distribution
```

2. Перенесите изменение из этого каталога, как это описано выше.

Не удаляйте каталог `/var/tmp/root-19980214` после окончания этого процесса.

3. Когда вы загрузите самую последнюю версию исходного кода и перестроите систему, выполните шаг 1. Это даст вам новый каталог, который может называться `/var/tmp/root-19980221` (если вы ждете неделю между обновлениями).
4. Теперь вы можете видеть изменения, которые были сделаны за прошедшую неделю, выполнив при помощи команды `diff(1)` рекурсивное сравнение двух каталогов:

```
# cd /var/tmp
# diff -r root-19980214 root-19980221
```

Как правило, здесь содержится гораздо меньше отличий, чем между каталогами `/var/tmp/root-19980221/etc` и `/etc`. Так как отличий меньше, то и легче перенести эти изменения в ваш каталог `/etc`.

5. Теперь вы можете удалить более старый из двух каталогов `/var/tmp/root-*`:

```
# rm -rf /var/tmp/root-19980214
```

6. Повторяйте этот процесс всякий раз, когда вам нужно перенести изменения в каталог `/etc`.

Для автоматической генерации имён каталогов можно использовать команду `date(1)`:

```
# mkdir /var/tmp/root-`date +%Y%m%d`
```

### 19.4.12. Обновите `/dev`

**Замечание:** Если вы работаете с FreeBSD 5.0 или более поздними версиями, то можете спокойно пропустить этот раздел. В этих версиях `devfs(5)` используется для выделения файлов устройств в режиме, прозрачном для пользователя.

Если вы используете `DEVFS`, то этого можно не делать.

В большинстве случаев утилита `mergemaster(8)` обнаружит, что необходимо обновить файлы устройств, и предложит сделать это автоматически. Эти указания описывают, как обновить файлы устройств вручную.

Для безопасности этот процесс делается в несколько шагов.

1. Скопируйте `/var/tmp/root/dev/MAKEDEV` в `/dev`:

```
# cp /var/tmp/root/dev/MAKEDEV /dev
```

Если вы использовали `mergemaster(8)` для обновления `/etc`, то ваш скрипт `MAKEDEV` уже должен быть обновлен, так что его не нужно проверять (утилитой `diff(1)`) и копировать вручную в случае необходимости.

2. Теперь выведите текущее содержимое вашего каталога `/dev`. Этот список должен содержать права, владельцев, старшее и младшее числа каждого файла, но не должен содержать информацию о времени. Проще всего это сделать, отрезав при помощи `awk(1)` часть информации:

```
# cd /dev
# ls -l | awk '{print $1, $2, $3, $4, $5, $6, $NF}' > /var/tmp/dev.out
```

- Повторно создайте все файлы устройств:

```
# sh MAKEDEV all
```

- Создайте ещё один список содержимого каталога, на этот раз в `/var/tmp/dev2.out`. Теперь просмотрите оба эти файла и поищите файлы устройств, которые вы забыли создать. Таких быть не должно, но лишний раз удостовериться не помешает.

```
# diff /var/tmp/dev.out /var/tmp/dev2.out
```

Скорее всего, вы заметите разногласия в именовании дисковых слайсов, что решается такими командами, как:

```
# sh MAKEDEV sd0s1
```

для повторного создания устройств слайсов. Точное название зависит от вашей системы и может отличаться от приведённого.

### 19.4.13. Обновите `/stand`

**Замечание:** Этот шаг описан только для полноты. Он может быть безболезненно опущен. Если вы работаете с FreeBSD 5.2 или более поздней версией, то для пользователей каталог `/rescue` автоматически обновляется до текущего состояния, а статически компилируемые выполнимые файлы во время выполнения команды `make installworld`, поэтому каталог `/stand` обновлять не нужно.

В целях полноты обновления вам может потребоваться обновить также файлы в каталоге `/stand`. Эти файлы представляют собой жёсткие ссылки на выполнимый файл `/stand/sysinstall`. Этот файл должен быть статически скомпонован, чтобы его работа не зависела от других файловых систем (в частности, от наличия смонтированной файловой системы `/usr`).

```
# cd /usr/src/release/sysinstall
# make all install
```

### 19.4.14. Перезагрузка

Теперь вы сделали всё. После того, как вы проверили, что всё на месте, можете перегрузить систему. Простая команда `shutdown(8)` должна это сделать:

```
# shutdown -r now
```

### 19.4.15. Завершение

Теперь у вас имеется успешно обновлённая система FreeBSD. Поздравляем!

Если что-то работает неправильно, можно с лёгкостью перестроить конкретную часть системы. Например, если вы случайно удалили файл `/etc/magic` в процессе обновления или переноса `/etc`, то команда `file(1)` перестанет работать. В таком случае это можно исправить вот так:

```
# cd /usr/src/usr.bin/file
```

```
# make all install
```

## 19.4.16. Вопросы?

### 1. Нужно ли полностью перестраивать систему при каждом изменении?

Простого ответа на этот вопрос нет, так как это зависит от характера изменения. Например, если вы только что выполнили **CVSup**, и оказалось, что с момента последнего его запуска были изменены следующие файлы:

```
src/games/cribbage/instr.c
src/games/sail/pl_main.c
src/release/sysinstall/config.c
src/release/sysinstall/media.c
src/share/mk/bsd.port.mk
```

то перестраивать всю систему незачем. Вы можете просто перейти в соответствующий подкаталог и выдать команду `make all install`, этого будет достаточно. Однако, если меняется что-то важное, например, `src/lib/libc/stdlib`, то вы должны перестроить всю систему или по крайней мере те ее части, которые скомпонованы статически.

В конце концов, выбор за вами. Может быть вам нравится перестраивать систему, скажем, каждый вечер, а изменения скачивать ночью. Или вы можете захотеть перестраивать только те вещи, которые менялись, но быть уверенным, что отслежены все изменения.

И, конечно же, всё это зависит от того, как часто вы хотите делать обновление, и отслеживаете ли вы FreeBSD-STABLE или FreeBSD-CURRENT.

### 2. Компиляция прерывается с большим количеством ошибок по сигналу 11 (или с другим номером сигнала). Что случилось?

Как правило, это говорит о проблемах с оборудованием. (Пере)построение системы является эффективным стресс-тестом для вашего оборудования и частенько выявляет проблемы с памятью. Обычно это проявляется в виде неожиданных сбоев компилятора или получения странных программных сигналов.

Явным указателем на это является то, что при перезапуске процедуры построения она прекращается в различные моменты времени.

В этом случае вы мало что можете сделать, разве что попробовать заменить комплектующие вашей машины для определения сбойщей компоненты.

### 3. Могу ли я удалить каталог `/usr/obj` после окончания?

Если отвечать коротко, то да.

Каталог `/usr/obj` содержит все объектные файлы, которые создаются во время фазы компиляции. Обычно одним из первых шагов в процессе `make buildworld` является удаление этого каталога. В этом случае сохранение `/usr/obj` после окончания имеет мало смысла; вдобавок, он будет занимать большой объём дискового пространства (на данный момент около 340 МБ).



Однако если вы точно знаете, что делаете, то можете заставить процедуру `make buildworld` пропустить этот шаг. Это позволит последующие построения выполняться гораздо быстрее, так как большинство исходных текстов не нужно будет перекомпилировать. Обратной стороной медали этого подхода является вероятность появления некоторых проблем с зависимостями, что может привести к прерыванию построения по странным причинам. Это частенько вызывает шум в списках рассылки FreeBSD, когда кто-либо жалуется на прерывание процесса построения, не обращая внимания на то, что он пытается срезать углы на повороте.

#### 4. Могут ли быть продолжены прерванные процессы построения?

Это зависит от того, насколько далеко зашел процесс построения перед тем, как вы обнаружили проблему.

*В общем случае* (и это несложное и быстрое правило) процесс `make buildworld` строит новые копии необходимых инструментальных средств (таких, как `gcc(1)` и `make(1)`) и системные библиотеки. Затем эти средства и библиотеки устанавливаются. Новые инструментальные средства и библиотеки затем используются для перестроения самих себя, и повторно устанавливаются. Система в целом (теперь включая обычные пользовательские программы, такие, как `ls(1)` или `grep(1)`) теперь перестраивается с новыми системными файлами.

Если вы на последнем шаге, и вы знаете это (потому что просматривали вывод, который сохраняете), то вы можете (достаточно безболезненно) выполнить команду:

```
... исправление проблемы ...
# cd /usr/src
# make -DNOCLEAN all
```

При этом результат предыдущего запуска `make buildworld` откатываться не будет.

Если вы видите сообщение:

```
-----
Building everything..
-----
```

в выводе команды `make buildworld`, то делать так достаточно безопасно.

Если этого сообщения не было, или вы в этом не уверены, то всегда лучше обезопасить себя, и начать построение с самого начала.

#### 5. Как ускорить процесс построения системы?

- Работайте в однопользовательском режиме.
- Разместите каталоги `/usr/src` и `/usr/obj` в отдельных файловых системах, располагающихся на разных дисках. Если это возможно, то разместите эти диски на разных дисковых контроллерах.
- Ещё лучше разместить эти файловые системы на нескольких дисках при помощи устройства `ccd(4)` (драйвер объединённых дисков).
- Выключите генерацию профилирующего кода (установив `'NOPROFILE=true'` в файле `/etc/make.conf`). Вам это скорее всего никогда не понадобится.

- Также в `/etc/make.conf` установите значение `CFLAGS` во что-то типа `-O -pipe`. Оптимизация `-O2` выполняется гораздо медленнее, а разница между `-O` и `-O2` обычно несущественна. `-pipe` позволяет компилятору использовать для связи вместо временных файлов программные каналы, что уменьшает обращение к диску (за счет оперативной памяти).
- Передайте утилите `make(1)` параметр `-jл` для запуска параллельно нескольких процессов. Обычно это помогает вне зависимости от того, сколько процессоров установлено в вашей машине.
- Файловая система, на которой располагается каталог `/usr/src`, может быть смонтирована (или перемонтирована) с опцией `noatime`. При этом запись на диск информации о времени последнего доступа к файлам будет отключена. Скорее всего, вам эта информация и не нужна.

```
# mount -u -o noatime /usr/src
```

**Внимание:** В примере предполагается, что `/usr/src` располагается на собственной файловой системе. Если это не так (то есть он является частью, скажем, `/usr`), то вам нужно использовать точку монтирования той файловой системы, а не `/usr/src`.

- Файловая система, на которой располагается `/usr/obj`, может быть смонтирована (или перемонтирована) с параметром `async`. Это приведёт к тому, что операции записи на диск будут выполняться асинхронно. Другими словами, запись будет завершаться немедленно, но данные записываться на диск несколькими секундами позже. Это позволит объединять операции записи и приведёт к значительному приросту производительности.

**Внимание:** Имейте в виду, что эта опция делает вашу файловую систему менее устойчивой. С этой опцией имеется больше шансов, что при перезагрузке машины после неожиданного сбоя при пропадании напряжения файловая система окажется в невосстановимом состоянии.

Если каталог `/usr/obj` — это все, что есть в этой файловой системе, то это не проблема. Если на той же самой файловой системе имеются какие-то важные данные, то проверьте давность ваших резервных копий перед включением этой опции.

```
# mount -u -o async /usr/obj
```

**Внимание:** Как и раньше, если каталог `/usr/obj` располагается не на собственной файловой системе, то в примере замените его на имя соответствующей точки монтирования.

## 6. Что мне делать, если что-то пошло не так?

Скрупулезно проверьте, чтобы в вашем окружении не было мешающих остатков от предыдущих построений. Это достаточно просто.

```
# chflags -R noschg /usr/obj/usr
# rm -rf /usr/obj/usr
# cd /usr/src
# make cleandir
```

```
# make cleandir
```

Да, команду `make cleandir` действительно нужно выполнять дважды.

После этого повторите весь процесс снова, начиная с `make buildworld`.

Если у вас все еще есть проблемы, пришлите текст ошибки и выдачу команды `uname -a` на адрес списка рассылки `freebsd-questions` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions>). Будьте готовы ответить на другие вопросы о конфигурации вашей системы!

## 19.5. Отслеживание исходных текстов для нескольких машин

*Текст предоставил Mike Meyer.*

Если у вас множество машин, для которых вы хотите отслеживать одно и то же дерево исходных текстов, то загрузка кода и перестроение системы полностью выглядит как ненужная трата ресурсов: дискового пространства, пропускной способности сети и процессорного времени. Так оно и есть, и решением является выделение одной машины, которая выполняет основной объём работы, в то время как остальные используют результаты работы посредством NFS. В этом разделе описывается именно этот метод.

### 19.5.1. Подготовка

Первым делом определите набор машин, на которых выполняется один и тот же набор бинарных программ, и мы будем называть его *набором для построения*. Каждая машина может иметь собственное уникальное ядро, но они будут работать с одними и теми же программами пользователя. Из этого набора выберите машину, которая будет являться *машиной для построения*. Она станет машиной, на которой будут строиться ядро и всё окружение. В идеальном случае с достаточно незагруженным CPU для выполнения команд `make buildworld` и `make buildkernel`. Вам также потребуется выбрать машину, которая будет *тестовой* для проверки обновлений программного обеспечения прежде, чем оно будет запущено в промышленную эксплуатацию. Это *должна* быть машина, которая может быть в нерабочем состоянии достаточно долго. Это может быть машина для построения, но не обязательно.

Все машины в этом наборе для построения должны монтировать каталоги `/usr/obj` и `/usr/src` с одной и той же машины и в одну и ту же точку монтирования. В идеальном случае они располагаются на разных дисках машины построения, но они могут также монтироваться по NFS на этой машине. Если у вас имеется несколько наборов для построения, то каталог `/usr/src` должен быть на машине построения, а по NFS он должен быть смонтирован на остальных.

Наконец, удостоверьтесь в том, что файл `/etc/make.conf` на всех машинах набора для построения соответствует машине построения. Это означает, что машина построения должна строить все части основного системного набора, которые будут устанавливаться на каждой машине из набора для построения. Кроме того, у каждой машины построения должно быть задано имя ядра посредством переменной `KERNCONF` в файле `/etc/make.conf`, а машина построения должна перечислить их все в переменной `KERNCONF`, причём первым должно быть имя её собственного ядра. Машина построения

должна хранить конфигурационные файлы ядра каждой машины в каталоге `/usr/src/sys/arch/conf`, если на ней будут строиться соответствующие ядра..

## 19.5.2. Основные системные компоненты

Теперь, когда всё это сделано, вы готовы к построению. Постройте ядро и всё окружение так, как это описано в Разд. 19.4.7.2 на машине построения, но ничего не устанавливайте. После того, как процесс построения завершится, перейдите к тестовой машине и установите только что построенное ядро. Если эта машина монтирует каталоги `/usr/src` и `/usr/obj` посредством NFS, то при перезагрузке в однопользовательский режим вам потребуется задействовать сеть и смонтировать их. Самым простым способом сделать это является переход во многопользовательский режим и запуск команды `shutdown now` для перехода в однопользовательский режим. После этого вы можете установить новое ядро и всё окружение, а затем выполнить команду `mergeMASTER` обычным образом. После выполнения этих действий перезагрузитесь для возвращения к обычному режиму работы во многопользовательском режиме с этой машиной.

После того, как вы убедитесь в нормальной работе всего на тестовой машине, проведите ту же самую процедуру для установки нового программного обеспечения на каждой из оставшихся машин из набора для построения.

## 19.5.3. Порты

Те же самые идеи могут использоваться и для дерева портов. Первым критическим шагом является монтирование `/usr/ports` с одной и той же машины на всех компьютерах в наборе для построения. Затем вы можете корректно настроить `/etc/make.conf` для использования общего каталога с дистрибутивными файлами. Вы должны задать переменную `DISTDIR` так, чтобы она указывала на общедоступный каталог, доступный тому пользователю, который отображается в пользователя `root` для ваших точек монтирования NFS. Каждая машина должна задавать `WRKDIRPREFIX` так, чтобы она указывала на локальный каталог построения. Наконец, если вы собираетесь строить и распространять пакеты, то должны задать переменную `PACKAGES` так, чтобы она указывала на каталог, соответствующий `DISTDIR`.

## Примечания

1. Это не совсем так. Мы не можем поддерживать старые релизы FreeBSD бесконечно долго, хотя мы поддерживаем их многие годы. Полное описание текущей политики безопасности относительно старых релизов FreeBSD можно найти по адресу <http://www.FreeBSD.org/ru/security/> (<http://www.FreeBSD.org/ru/security/>).

# IV. Сетевые коммуникации

FreeBSD это одна из наиболее широко используемых в высокопроизводительных сетевых серверах операционных систем. Главы этой части книги охватывают:

- Последовательные соединения
- PPP и PPP через Ethernet
- Электронную почту
- Запуск сетевых серверов
- Другую сетевую тематику повышенной сложности

Эти главы предназначены для получения дополнительной информации. Нет необходимости читать их в определенной последовательности, или читать их все перед тем, как начать использовать FreeBSD в сети.

# Глава 20. Последовательные соединения

*Перевод на русский язык: Денис Пеплин.*

## 20.1. Краткое описание

В UNIX всегда была поддержка последовательных соединений. Фактически, самые первые UNIX машины использовали последовательные линии для пользовательского ввода/вывода. Многие изменилось с тех пор, когда среднестатистический "терминал" состоял из 10-символов-в-секунду последовательного принтера и клавиатуры. Эта глава рассказывает о некоторых способах, которыми FreeBSD использует последовательные соединения.

Прочитав эту главу, вы узнаете:

- Как подсоединить терминалы к системе FreeBSD.
- Как использовать модем для дозвона на удаленные хосты.
- Как разрешить удаленным пользователям входить в вашу систему с помощью модема.
- Как загрузить систему с последовательной консоли.

Перед прочтением этой главы вам потребуется:

- Узнать как настраивать и устанавливать новое ядро (Гл. 8).
- Понять, что такое права доступа и процессы UNIX (Гл. 3).
- Кроме этого вам потребуется техническое руководство на последовательное оборудование (модем или мультипортовую карту), которую вы хотите использовать с FreeBSD.

## 20.2. Введение

### 20.2.1. Терминология

bps

Бит в секунду (Bits per Second) — скорость передачи данных

DTE

Терминальное оборудование (Data Terminal Equipment) — например, ваш компьютер

DCE

Оборудование связи (Data Communications Equipment) — ваш модем

RS-232

Стандарт EIA для аппаратных последовательных соединений

При упоминании скорости передачи данных, в этой главе не используется термин "бод" ("baud"). Бод означает количество электрических импульсов, которые могут быть переданы за период времени, а "bps" это *корректный* термин для использования (он хотя бы не создает столько проблем как предыдущий).

## 20.2.2. Кабели и порты

Для подсоединения модема или терминала к системе FreeBSD потребуется последовательный порт и подходящий кабель для последовательного устройства. Если вы уже знаете о аппаратном обеспечении и требуемых кабелях, можете пропустить этот раздел.

### 20.2.2.1. Кабели

Есть несколько различных видов последовательных кабелей. Два наиболее часто используемых в нашей ситуации типа это нуль-модемный и стандартный ("прямой") RS-232 кабель. Документация на оборудование должна описывать тип требуемого кабеля.

#### 20.2.2.1.1. Нуль-модемные кабели

Нуль модемный кабель пропускает некоторые сигналы, такие как "signal ground", напрямую, а другие "заворачивает". Например, контакт "send data" на одном конце соединяется с контактом "receive data" на другом.

Если вы изготавливаете собственный кабели, то можете сделать собственный кабель для использования с терминалами. Эта таблица показывает названия сигналов RS-232C и номера контактов на разъеме DB-25.

Сигнал	Контакт		Контакт	Сигнал
SG	7	соединен с	7	SG
TD	2	соединен с	3	RD
RD	3	соединен с	2	TD
RTS	4	соединен с	5	CTS
CTS	5	соединен с	4	RTS
DTR	20	соединен с	6	DSR
DCD	8		6	DSR
DSR	6	соединен с	20	DTR

**Замечание:** Соедините "Data Set Ready" (DSR) и "Data Carrier Detect" (DCD) внутри корпуса коннектора, а затем подключите к "Data Terminal Ready" (DTR) на другой стороне.

#### 20.2.2.1.2. Стандартные кабели RS-232C

Стандартный последовательный кабель пропускает все RS-232C сигналы напрямую. Так, "send data" на одном конце кабеля соединяется с контактом "send data" на другом конце. Этот тип кабеля

предназначен для подсоединения модема, а также подходит для некоторых терминалов.

### 20.2.2.2. Порты

Последовательные порты это устройства, через которые данные передаются между компьютером с FreeBSD и терминалом. Этот раздел описывает типы существующих портов и их адресацию в FreeBSD.

#### 20.2.2.2.1. Типы портов

Существует несколько типов последовательных портов. Перед изготовлением кабеля, вам потребуется убедиться, что он подходит к портам терминала и системы FreeBSD.

Большинство терминалов используют порты DB25. Персональные компьютеры, включая PC под управлением FreeBSD, используют порты DB25 или DB9. Если у вас есть мультипортовая последовательная карта для PC, там могут быть RJ-12 или RJ-45 порты.

Обратитесь к сопровождающей документации на оборудование за информацией об используемых портах. Можно также определить тип используемых портов по их внешнему виду.

#### 20.2.2.2.2. Имена портов

В FreeBSD доступ к каждому последовательному порту может быть получен через файл в каталоге `/dev`. Есть два различных типа файлов:

- Порты входящих соединений (dial-in) называются `/dev/ttydN`, где *N* это номер порта начиная с нуля. Обычно, порты входящих соединений используются для терминалов. Для корректной работы этим портам требуется, чтобы последовательный кабель передавал сигнал data carrier detect (DCD).
- Порты исходящих соединений (call-out) называются `/dev/cuaaN`. Они обычно используются не для терминалов, а только для модемов. Вы можете использовать эти порты если последовательный кабель или терминал не поддерживает сигнал DCD.

Если вы соединили терминал с первым последовательным портом (COM1 в MS-DOS), используйте `/dev/ttyd0` для доступа к терминалу. Если терминал соединен со вторым последовательным портом (известным также как COM2), используйте `/dev/ttyd1`, и так далее.

### 20.2.3. Настройка ядра

FreeBSD с настройками по умолчанию поддерживает последовательные порты. В мире MS-DOS они известны как COM1, COM2, COM3, и COM4. На данный момент в FreeBSD есть поддержка как "простых" мультипортовых карт с последовательными интерфейсами, таких как BocaBoard 1008 и 2016, так и более "умных" мультипортовых карт, например карт Digiboard и Stallion Technologies. Тем не менее, ядро по умолчанию определяет только стандартные COM порты.

Чтобы увидеть, как ядро определяет последовательные порты, просмотрите сообщения, выводимые во время загрузки ядра, или используйте команду `/sbin/dmesg` для вывода сообщений ядра еще раз. В частности, обратите внимание на сообщения, начинающиеся с символов `sio`.



**Подсказка:** Для просмотра только тех сообщений, которые содержат слово `sio`, используйте команду:

```
# /sbin/dmesg | grep 'sio'
```

Например, в системе с четырьмя последовательными портами, появятся такие специфичные для последовательных портов сообщения:

```
sio0 at 0x3f8-0x3ff irq 4 on isa
sio0: type 16550A
sio1 at 0x2f8-0x2ff irq 3 on isa
sio1: type 16550A
sio2 at 0x3e8-0x3ef irq 5 on isa
sio2: type 16550A
sio3 at 0x2e8-0x2ef irq 9 on isa
sio3: type 16550A
```

Если ядро не распознает все последовательные порты, вам возможно потребуется настроить ядро FreeBSD. За детальной информацией по настройке ядра обращайтесь к главе Гл. 8.

Строки соответствующих устройств в файле конфигурации ядра FreeBSD 4.X будут выглядеть примерно так:

```
device sio0 at isa? port IO_COM1 irq 4
device sio1 at isa? port IO_COM2 irq 3
device sio2 at isa? port IO_COM3 irq 5
device sio3 at isa? port IO_COM4 irq 9
```

и примерно так для FreeBSD 5.X:

```
device sio
```

Вы можете закомментировать или полностью удалить эти строки для отсутствующих устройств в FreeBSD 4.X; в FreeBSD 5.X вам потребуется отредактировать файл `/boot/device.hints` для настройки последовательных портов. Обратитесь к странице справочника `sio(4)` за дополнительной информацией о настройке последовательных портов и мультипортовых карт. Будьте осторожны при использовании настроек, которые работали в предыдущих версиях FreeBSD, поскольку флаги устройств и синтаксис изменились в новых версиях.

**Замечание:** `port IO_COM1` это синоним для `port 0x3f8`, `IO_COM2` для `0x2f8`, `IO_COM3` для `0x3e8`, и `IO_COM4` для `0x2e8`. Это наиболее часто используемые для соответствующих последовательных портов адреса. Наиболее часто используемые прерывания 4, 3, 5, и 9. Имейте ввиду, что обычные последовательные порты не могут совместно использовать прерывания на ISA PC (на мультипортовых картах есть электроника, позволяющая всем чипам 16550A на плате совместно использовать одно или два IRQ).

## 20.2.4. Специальные файлы устройств

К большинству устройств ядра можно получить доступ через "специальные файлы устройств", расположенные в каталоге `/dev`. К устройствам `sio` можно получить доступ через `/dev/ttydN`

(устройства входящих вызовов, dial-in) и `/dev/cuaaN` (устройства исходящих вызовов, call-out). FreeBSD предоставляет также устройства инициализации (`/dev/ttyidN` и `/dev/cuaiaN`), устройства блокировки (`/dev/ttyldN` и `/dev/cualaN`). Первые используются для инициализации параметров порта при каждом его открытии (таких как `crtsets` для модемов, использующих сигналы RTS/CTS для управления потоком). Устройства блокировки используются для установки флага блокировки на порт и предотвращения изменения определенных параметров пользователями или программами; обратитесь к страницам справочника `termios(4)`, `sio(4)` и `stty(1)` соответственно за информацией о параметрах терминала, блокировании и инициализации устройств и настройке терминала.

#### 20.2.4.1. Создание специальных файлов устройств

**Замечание:** FreeBSD 5.0 включает файловую систему `devfs(5)`, которая автоматически создает файлы устройств по мере необходимости. Если вы работаете с версией FreeBSD, поддерживающей `devfs` просто пропустите этот раздел.

В каталоге `/dev` находится shell скрипт, называющийся `MAKEDEV`, который управляет специальными файлами устройств. Чтобы использовать `MAKEDEV` для создания специальных файлов устройств исходящих соединений для порта COM1 (порт 0), зайдите (`cd`) в каталог `/dev` и выполните команду `MAKEDEV ttyd0`. Точно так же, для создания специальных файлов устройств исходящих соединений для порта COM2 (порт 1), используйте `MAKEDEV ttyd1`.

`MAKEDEV` создаст не только специальный файл устройства `/dev/ttydN` но также `/dev/cuaaN`, `/dev/cuaiaN`, `/dev/cualaN`, `/dev/ttyldN`, и `/dev/ttyidN`.

После создания специальных файлов устройств, обязательно проверьте права на файлы (особенно на файлы `/dev/cua*`), чтобы убедиться, что только те пользователи, которым дан доступ на эти файлы, могут читать и писать в них — возможно, вы не хотите, чтобы обычные пользователи использовали модемы для звонков. Права по умолчанию на `/dev/cua*` должны подойти:

```
crw-rw----   1 uucp      dialer    28, 129 Feb 15 14:38 /dev/cuaa1
crw-rw----   1 uucp      dialer    28, 161 Feb 15 14:38 /dev/cuaia1
crw-rw----   1 uucp      dialer    28, 193 Feb 15 14:38 /dev/cuala1
```

Эти права позволяют пользователю `uucp` и пользователям из группы `dialer` использовать устройства исходящих вызовов.

#### 20.2.5. Настройка последовательных портов

Устройство `ttydN` (или `cuaaN`) это обычное устройство, которое потребуется открыть для приложений. Когда процесс открывает устройство применяются настройки ввода/вывода терминала по умолчанию. Вы можете посмотреть эти настройки с помощью команды

```
# stty -a -f /dev/ttyd1
```

Если вы измените настройки устройства, они будут действовать до его закрытия. После повторного открытия, оно вернется к настройкам по умолчанию. Для изменения настроек по умолчанию, вы можете открыть и изменить установки "начального состояния" устройства. Например, для включения

по умолчанию режима CLOCAL, 8-битного соединения и контроля передачи XON/XOFF для ttyd5, выполните:

```
# stty -f /dev/ttyd5 clocal cs8 ixon ixoff
```

Инициализация последовательных устройств контролируется файлом `/etc/rc.serial`. Этот файл определяет настройки последовательных устройств по умолчанию.

Для предотвращения изменения программами отдельных установок, настройте "состояние блокировки" устройства. Например, для установки значения скорости ttyd5 в 57600 bps, выполните:

```
# stty -f /dev/ttyd5 57600
```

Теперь приложение, открывающее ttyd5 и пытающееся изменить скорость порта, получит скорость 57600 bps.

И конечно, сделайте запись начальных значений и состояния блокировки устройств доступной только учетной записи root.

## 20.3. Терминалы

*Предоставил Sean Kelly.*

Терминалы предоставляют удобный и дешевый способ доступа к системе FreeBSD, когда вы не сидите за консолью компьютера и не подключены к сети. Этот раздел описывает использование терминалов в FreeBSD.

### 20.3.1. Пользователи и типы терминалов

В первых системах UNIX не было консолей. Вместо этого, пользователи входили и запускали программы через терминалы, которые были подключены к последовательным портам компьютеров. Это очень похоже на использование модема и программного обеспечения терминала для дозвона до удаленной системы и выполнения только-текстовой работы.

Консоли современных PC поддерживают высококачественную графику, но возможность входа по последовательному порту на сегодняшний день все еще доступна почти в каждой UNIX подобной операционной системе; FreeBSD не исключение. Используя терминал, подключенный к неиспользуемому последовательному порту, вы можете войти и запустить текстовую программу, которую обычно запускаете в текстовой консоли или в окне xterm системы X Window.

Для корпоративных пользователей, вы можете подсоединить множество терминалов к системе FreeBSD и поставить их на столы пользователей. Для домашнего пользователя, устаревший IBM PC или Macintosh может быть подключен в качестве терминала к более мощному компьютеру под управлением FreeBSD. Вы можете превратить однопользовательский компьютер в мощную многопользовательскую систему.

В FreeBSD три вида терминалов:

- Простые (dumb) терминалы
- PC, работающие в качестве терминалов

- X терминалы

В оставшейся части раздела описывается каждый вид.

### 20.3.1.1. Простые терминалы

Простые терминалы это специализированное оборудование, позволяющее соединять компьютеры через последовательные линии. Они называются "простыми", поскольку их вычислительных возможностей хватает только для отображения, отправки и получения текста. Вы не сможете запустить на них никаких программ. Компьютер, к которому подсоединяется терминал, предоставляет все возможности для запуска текстовых редакторов, компиляторов, почтовых программ, игр и так далее.

Есть сотни видов простых терминалов, изготовленных различными производителями, включая DEC VT-100 и Wyse WY-75. Почти любой терминал может работать с FreeBSD. Некоторые high-end терминалы даже могут отображать графику, но только отдельные программные пакеты могут получить преимущество от этих расширенных возможностей.

Простые терминалы популярны в рабочей среде, где не требуется доступ к графическим приложениям, например тем, которые предоставляет система X Window.

### 20.3.1.2. PC, работающие в качестве терминалов

Если простые терминалы могут только отображать, отправлять и получать текст, возможностей абсолютно любого персонального компьютера хватит для работы в роли простого терминала. Все, что вам потребуется, это подходящий кабель и какая-нибудь программа *эмулятора терминала*.

Это популярная домашняя конфигурация. Например, когда ваша вторая половина занята работой на системной консоли FreeBSD, вы можете одновременно выполнять только-текстовую работу с менее мощного персонального компьютера, подключенного к системе FreeBSD.

### 20.3.1.3. X терминалы

X терминалы это наиболее сложный тип существующих терминалов. Вместо подключения к последовательному порту, они обычно подключаются к сети, например Ethernet. Вместо работы только с текстовыми приложениями, они могут отображать любое X приложение.

Мы представляем X терминалы только ради полноты описания. Тем не менее, эта глава *не* охватывает установку, настройку или использование X терминалов.

## 20.3.2. Настройка

Этот раздел описывает, что нужно сделать для настройки системы FreeBSD и включения входа в систему через терминал. Предполагается, что вы уже подключили терминал и настроили ядро для включения поддержки последовательного порта, к которому он подключен.

Обратитесь к главе Гл. 12 за информацией о процессе `init`, отвечающем за контроль над всеми процессами и за инициализацию системы во время загрузки. Одна из задач, выполняемых `init` — чтение файла `/etc/ttys` и запуск процесса `getty` на доступных терминалах. Процесс `getty` отвечает за чтение имени пользователя и запуск программы `login`.

Таким образом, для настройки терминалов в системе FreeBSD необходимо выполнить следующие действия под `root`:

1. Добавить строку к `/etc/ttys` для файла из каталога `/dev`, представляющего последовательный порт, если этой строки еще нет.
2. Настроить запуск команды `/usr/libexec/getty` на этом порту и указать соответствующий тип `getty` в файле `/etc/gettytab`.
3. Указать тип терминала по умолчанию.
4. Переключить порт в состояние "on" ("включен")
5. Указать, должен ли порт быть "secure" ("безопасным")
6. Заставить `init` перечитать файл `/etc/ttys`.

Опционально, вы можете настроить свой тип `getty` для использования на шаге 2, добавив описание в файл `/etc/gettytab`. За описанием обратитесь к страницам справочника `gettytab(5)` и `getty(8)`.

### 20.3.2.1. Добавление строки в `/etc/ttys`

В файле `/etc/ttys` находится список всех портов системы FreeBSD, на которые возможен вход. Например, там находится первая виртуальная консоль `ttv0`. Вы можете войти на консоль с помощью этой записи. Файл содержит записи и для других виртуальных консолей, последовательных портов, и псевдо-терминалов. Название файла последовательного порта из каталога `/dev` приводится без префикса `/dev` (например, устройство `/dev/ttyv0` будет записано как `ttv0`).

Установка FreeBSD по умолчанию включает файл `/etc/ttys` с поддержкой первых четырех последовательных портов: от `ttvd0` до `ttvd3`. Если вы подключаете терминал к одному из этих портов, добавлять записи терминалов не потребуется.

#### Пример 20-1. Добавление записей терминалов в `/etc/ttys`

Предположим, вы хотите подключить два терминала к системе: Wyse-50 и старый 286 IBM PC с эмулятором терминала VT-100. Мы подключаем Wyse к второму последовательному порту и 286 к шестому последовательному порту (порт на мультипортовой карте). Соответствующие строки в `/etc/ttys` будут выглядеть так:

```
ttvd1❶  "/usr/libexec/getty std.38400"❷  wy50❸  on❹  insecure❺
ttvd5   "/usr/libexec/getty std.19200"   vt100  on   insecure
```

- ❶ Первое поле, как правило, указывает имя специального файла терминала, в соответствии с его именем в `/dev`.
- ❷ Второе поле — это команда, исполняемая для этого терминала, обычно `getty(8)`. `getty` инициализирует и открывает линию, устанавливает ее скорость, приглашает пользователя к вводу имени пользователя, а затем выполняет программу `login(1)`.

Программа `getty` принимает один (опциональный) параметр в командной строке, тип `getty`. Тип `getty` определяет характеристики терминальной линии, такие как значение `bps` и четность.

Программа `getty` считывает эти характеристики из файла `/etc/gettytab`.

Файл `/etc/gettytab` содержит множество записей для терминалов, как для старых так и для новых. Почти во всех случаях запись, начинающаяся с текста `std`, предназначена для работы с аппаратными терминалами. Эти записи игнорируют четность. Запись `std` есть для каждого значения `bps` от 110 до 115200. Конечно, вы можете добавить собственные записи в этот файл. Страница справочника `gettytab(5)` содержит дополнительную информацию.

При установке типа `getty` в файле `/etc/ttys` убедитесь в наличии соответствующей записи терминала.

Например, Wyse-50 не использует четность и соединяется на 38400 bps. 286 PC не использует четность и соединяется на 19200 bps.

- ③ Третье поле определяет тип терминала, обычно подключаемого к этой линии `tty`. Для портов входящих соединений обычно используется значение `unknown` или `dialup`, поскольку пользователь может подключить практически любой тип терминала или программу. Для аппаратных терминалов тип не меняется, поэтому вы можете поместить в это поле определенный тип терминала из базы данных `termcap(5)`.

Например, Wyse-50 использует реальный тип терминала, а 286 PC, работающий с **Procomm**, настроен на эмуляцию VT-100.

- ④ Четвертое поле определяет должен ли порт быть включен. Размещение здесь `on` укажет процессу `init` запустить программу, указанную во втором поле, `getty`. Если вы поместите `off` в это поле, команда `getty` не будет запущена и вход на этот порт станет невозможен.
- ⑤ Последнее поле используется, чтобы указать, является ли порт безопасным. Пометка порта безопасным означает, что вы доверяете ему достаточно для того, чтобы разрешить учетной записи `root` (или любой учетной записи с `UID 0`) входить с этого порта. Небезопасные порты не разрешат вход `root`. На небезопасном порту пользователи должны войти с через непривилегированную учетную запись, а затем использовать `su(1)` или подобный механизм для получения привилегий суперпользователя.

Настоятельно рекомендуется использовать "insecure" даже для терминалов, находящихся за закрытыми дверями. Довольно легко использовать `su` после входа, если вам потребуются привилегии суперпользователя.

### 20.3.2.2. Заставьте `init` перечитать `/etc/ttys`

После выполнения необходимых изменений в файле `/etc/ttys`, вам потребуется отправить сигнал `SIGHUP` (`hangup`) процессу `init`, чтобы заставить его перечитать его файл настройки. Например:

```
# kill -HUP 1
```

**Замечание:** `init` это всегда первый из запущенных в системе процессов, поэтому его `PID` всегда 1.

Если все установлено правильно, все кабели на месте и терминалы включены, процесс `getty` должен быть запущен на каждом терминале и вы увидите приглашение ко входу на каждом терминале.

### 20.3.3. Решение проблем с соединением

Даже при самом внимательном отношении к деталям, при настройке терминала все же могут возникнуть проблемы. В этом разделе приведен список симптомов и предлагается несколько решений.

#### 20.3.3.1. Не появляется приглашение ко входу

Убедитесь, что терминал подключен и его питание включено. Убедитесь, что эмулятор терминала запущен на соответствующем порту.

Убедитесь, что кабель хорошо подключен и к терминалу и к компьютеру с FreeBSD. Убедитесь, что правильно выбран тип кабеля.

Убедитесь, что терминал и FreeBSD имеют одинаковые установки значения bps и четности. Если у вас видео терминал, убедитесь, что контраст и яркость включены. Если это принт-терминал, убедитесь, что бумага и чернила в порядке.

Убедитесь, что процесс `getty` запущен и обслуживает терминал. Например, для получения списка запущенных процессов `getty` с помощью `ps`, выполните:

```
# ps -axww|grep getty
```

Вы должны увидеть строку для соответствующего терминала. Например, если `getty` запущена на втором последовательном порту `ttyd1` и использует запись `std.38400` из файла `/etc/gettytab`, отобразится следующее:

```
22189  d1  Is+      0:00.03 /usr/libexec/getty std.38400 ttyd1
```

Если процесс `getty` не запущен, убедитесь, что вы включили порт в `/etc/ttys`. Не забудьте также запустить `kill -HUP 1` после изменения файла `ttys`.

Если процесс `getty` запущен, но на терминале по-прежнему не отображается приглашение ко входу, или если приглашение отображается, но войти невозможно, терминал или кабель, возможно, не поддерживают квитирование связи. Попробуйте изменить поле в `/etc/ttys` с `std.38400` на `3wire.38400`. Запись `3wire` похожа на `std`, но игнорирует квитирование связи. Вам может потребоваться уменьшить скорость соединения или включить программный контроль передачи при использовании `3wire` для предотвращения переполнений буфера.

#### 20.3.3.2. Вместо приглашения ко входу на экране появляется "мусор"

Убедитесь, что терминал и FreeBSD имеют одинаковые установки значения bps и четности. Проверьте процесс `getty`, чтобы убедиться, что используется подходящий тип `getty`. Если это не так, отредактируйте `/etc/ttys` и запустите `kill -HUP 1`.

#### 20.3.3.3. Символы появляются дважды, пароль отображается при вводе

Переключите терминал (или программу эмулятора терминала) с `'half duplex'` или `'local echo'` на `'full duplex'`.

## 20.4. Входящие соединения по модему

*Предоставил Guy Helmer. Дополнил Sean Kelly.*

Настройка системы FreeBSD для поддержки входящих соединений очень похожа на подсоединение терминалов за исключением того, что вы работаете с модемами вместо терминалов.

### 20.4.1. Внешние и внутренние модемы

Внешние модемы более удобны для дозвона, поскольку легко могут быть настроены с помощью параметров, сохраняемых в энергонезависимой памяти. На них обычно есть индикаторы, отображающие состояние основных RS-232 сигналов. Мигающие индикаторы впечатляют, но кроме того они также очень полезны для индикации правильной работы модема.

Внутренние модемы обычно не снабжаются энергонезависимой памятью, поэтому их настройка может ограничиваться установкой DIP переключателей. Если на внутреннем модеме есть индикаторы, их обычно сложно увидеть при закрытой крышке корпуса.

#### 20.4.1.1. Модемы и кабели

Если вы используете внешний модем, несомненно потребуются подходящий кабель. Стандартный RS-232C кабель должен подойти, если подключены все обычные сигналы:

- Transmitted Data (TD)
- Received Data (RD)
- Request to Send (RTS)
- Clear to Send (CTS)
- Data Set Ready (DSR)
- Data Terminal Ready (DTR)
- Carrier Detect (CD)
- Signal Ground (SG)

FreeBSD требуются сигналы RTS и CTS для контроля передачи на скоростях выше 2400 bps, сигнал CD для определения, был ли ответ на сигнал или произошло отключение линии, и сигнал DTR для сброса модема после завершения сессии. Некоторые кабели не поддерживают все необходимые сигналы, поэтому, если вы столкнулись с проблемами, например, если сессия не завершается после отсоединения линии, причиной возможно являются проблемы с кабелем.

Как и другие UNIX подобные операционные системы, FreeBSD использует аппаратные сигналы для определения того, был ли ответ на звонок или линия была отключена и требуется завершить работу модема и сбросить его в начальное состояние. FreeBSD избегает отправлять команды модему или просматривать отчеты о статусе от модема. Если вы знакомы с настройкой BBS, это может показаться неудобным.



## 20.4.2. Рекомендации по последовательным интерфейсам

FreeBSD поддерживает интерфейсы, основанные на NS8250, NS16450, NS16550, и NS16550A EIA RS-232C (ССИТТ V.24). Устройства 8250 и 16450 снабжены односимвольным буфером. Устройство 16550 снабжено 16-ти символьным буфером, который повышает производительность системы. (Ошибки в 16550 делают невозможным использование 16-символьного буфера, поэтому используйте 16550A если возможно). Поскольку устройства с односимвольным буфером предъявляют большие требования к операционной системе, чем с 16-ти символьным буфером, предпочтительны устройства на 16550A. Если в системе много активных последовательных портов или нагрузка велика, устройства на 16550A лучше подходят для поддержки соединений с малым количеством ошибок.

## 20.4.3. Краткий обзор

Как и с терминалами, `init` запускает процесс `getty` на каждом настроенном для входящих звонков последовательном порту. Например, если модем подключен к `/dev/ttyd0`, команда `ps ax` может вывести следующее:

```
4850 ?? I      0:00.09 /usr/libexec/getty V19200 ttyd0
```

Когда пользователь дозванивается на подключенный модем, модем выдает сигнал CD (Carrier Detect). Ядро определяет, что несущая обнаружена и завершает открытие порта командой `getty`. `getty` отправляет приглашение `login:` на указанной скорости. `getty` ожидает в ответ набор символов, и, как правило, получает неправильный набор (обычно это происходит из-за того, что скорость соединения модема отличается от скорости `getty`). `getty` пробует подобрать скорость линии до тех пор, пока не получит правильный набор символов.

После того, как будет введено имя пользователя, `getty` выполняет `/usr/bin/login`, которая завершает вход, запрашивая пароль пользователя и запуская оболочку.

## 20.4.4. Файлы настройки

Есть три файла настройки системы в каталоге `/etc`, которые возможно потребуются отредактировать для включения удаленного доступа по модему в FreeBSD. Первый, `/etc/gettytab`, содержит информацию по настройке демона `/usr/libexec/getty`. Второй, `/etc/ttys`, содержит информацию, указывающую `/sbin/init` на каких устройствах `tty` должны быть запущены процессы `getty`. Наконец, вы можете поместить команды инициализации портов в скрипт `/etc/rc.serial`.

В UNIX есть две школы настройки модемов для входящих соединений. Одна предпочитает настраивать модемы и системы так, что не важно на какой скорости подсоединяется удаленный пользователь. Локальный интерфейс RS-232 компьютер-модем работает на жестко заданной скорости. Преимущество этой настройки в том, что удаленный пользователь всегда сразу видит приглашение ко входу. Обратная сторона в том, что система не знает, какова на самом деле скорость передачи данных, поэтому полноэкранные программы, такие как Emacs, не настраивают свои методы отображения на экране для работы с медленными соединениями.

Другая школа настраивает интерфейс RS-232 для работы с различной скоростью в зависимости от скорости подсоединения удаленного пользователя. Например, соединение модемов по протоколу V.32bis (14.4 Кбит/с) установит скорость порта RS-232 равной 19.2 Кбит/с, а соединение на скорости 2400 бит/с установит скорость RS-232 равной 2400 бит/с. Поскольку `getty` не понимает сообщений модема о скорости соединения, `getty` выдает приглашение `login:` на установленной по умолчанию

скорости и считывает символы, полученные в ответе. Если пользователь видит "мусор" вместо приглашения ко входу, это означает, что нужно нажимать Enter до тех пор, пока не появится приглашение ко входу. Если скорости не совпадают, `getty` получает все, что вводит пользователь, в виде "мусора", пробует переключиться на другую скорость и выдает приглашение `login:` опять. Эта процедура может продолжаться до отвращения, но обычно требуется одно или два нажатия клавиши перед появлением нормально выглядящего приглашения. Очевидно, эта последовательность входа не так хороша, как метод с фиксированной скоростью, но при низкой скорости соединения работать с полноэкранными программами станет проще.

В этом разделе делается попытка дать сбалансированную информацию для настройки, но предпочтение будет отдано установке скорости соединения с модемом в соответствии скорости подключения.

#### 20.4.4.1. `/etc/gettytab`

`/etc/gettytab` это файл в стиле `termcap(5)`, содержащей информацию по настройке `getty(8)`. Пожалуйста, обратитесь к странице справочника `gettytab(5)` за полной информацией о формате файла и за списком возможностей `getty`.

##### 20.4.4.1.1. Настройка фиксированной скорости

Если вы зафиксировали скорость соединения модема на определенной скорости, редактировать файл `/etc/gettytab` скорее всего не потребуется.

##### 20.4.4.1.2. Настройка изменяемой скорости

Вам потребуется сделать запись в `/etc/gettytab` для предоставления `getty` информации о скоростях, которые предполагается использовать для модема. Если у вас 2400 бит/с модем, возможно, подойдет существующая запись `D2400`.

```
#
# Fast dialup terminals, 2400/1200/300 rotary (can start either way)
#
D2400|d2400|Fast-Dial-2400:\
        :nx=D1200:tc=2400-baud:
3|D1200|Fast-Dial-1200:\
        :nx=D300:tc=1200-baud:
5|D300|Fast-Dial-300:\
        :nx=D2400:tc=300-baud:
```

Если у вас более скоростной модем, вам возможно потребуется добавить запись в `/etc/gettytab`; вот запись, которую вы можете использовать для 14.4 Кбит/с модема с максимальной скоростью интерфейса 19.2 Кбит/с:

```
#
# Additions for a V.32bis Modem
#
um|V300|High Speed Modem at 300,8-bit:\
        :nx=V19200:tc=std.300:
un|V1200|High Speed Modem at 1200,8-bit:\
        :nx=V300:tc=std.1200:
```

```
uo|V2400|High Speed Modem at 2400,8-bit:\
:nx=V1200:tc=std.2400:
up|V9600|High Speed Modem at 9600,8-bit:\
:nx=V2400:tc=std.9600:
uq|V19200|High Speed Modem at 19200,8-bit:\
:nx=V9600:tc=std.19200:
```

Эта настройка включает 8-битные соединения без программного контроля четности.

В примере выше скорость порта будет переключаться в цикле начиная с 19.2 Кбит/с (для соединения по V.32bis), затем 9600 бит/с (для V.32), 2400 бит/с, 1200 бит/с, 300 бит/с, и обратно на 19.2 Кбит/с.

Переключение скоростей в цикле реализовано с помощью `nx=` ("next table"). Каждая из линий использует `tc=` ("table continuation") для указания "стандартных" (std) настроек на каждой скорости.

Если у вас 28.8 Кбит/с модем и/или вы хотите получить преимущество от сжатия на скорости 14.4 Кбит/с, потребуются скорости выше, чем 19.2 Кбит/с. Вот пример записи из `gettytab` для начала соединения на скорости 57.6 Кбит/с:

```
#
# Additions for a V.32bis or V.34 Modem
# Starting at 57.6 Kbps
#
vm|VH300|Very High Speed Modem at 300,8-bit:\
:nx=VH57600:tc=std.300:
vn|VH1200|Very High Speed Modem at 1200,8-bit:\
:nx=VH300:tc=std.1200:
vo|VH2400|Very High Speed Modem at 2400,8-bit:\
:nx=VH1200:tc=std.2400:
vp|VH9600|Very High Speed Modem at 9600,8-bit:\
:nx=VH2400:tc=std.9600:
vq|VH57600|Very High Speed Modem at 57600,8-bit:\
:nx=VH9600:tc=std.57600:
```

Если у вас медленный CPU или сильно загруженная система без последовательных портов на базе 16550A, на скорости 57.6 Кбит/с могут возникнуть ошибки `sio "silo"`.

#### 20.4.4.2. /etc/ttys

Настройка файла `/etc/ttys` была описана в Прим. 20-1. Настройка модемов похожа, но потребуется передавать `getty` различные аргументы и указывать различные типы терминалов. Общий формат для фиксированной и переменной скорости такой:

```
ttyd0    "/usr/libexec/getty xxx"    dialup on
```

Первый пункт в строке выше это специальный файл устройства для этой записи — `ttyd0` означает, что `getty` будет запущена на `/dev/ttyd0`. Второй пункт, `"/usr/libexec/getty xxx"` (`xxx` будет замещено на запись из `gettytab` для начальной скорости), это процесс, который будет запущен на данном устройстве. Третий пункт, `dialup`, это тип терминала по умолчанию. Четвертый параметр, `on`, указывает `init`, что линия включена. Может быть пятый параметр, `secure`, но он должен использоваться только для терминалов, которые физически безопасны (таких как системная консоль).

Тип терминала по умолчанию (dialup в примере выше) может зависеть от личных предпочтений. dialup это традиционный тип терминала по умолчанию на линиях для дозвона, который позволяет пользователям, зная что тип терминала dialup, автоматически настраивать свой тип терминала. Однако, автор находит более легким указание vt102 в качестве типа терминала по умолчанию, поскольку пользователи работают на своих удаленных системах с эмулятором терминала VT102.

После внесения изменений в `/etc/ttys`, вы можете отправить процессу `init` сигнал HUP пересчитать файл. Используйте команду

```
# kill -HUP 1
```

для отправки сигнала. Если вы настраиваете систему в первый раз, то возможно захотите подождать, пока модем(ы) правильно настроится и соединится перед отправкой сигнала `init`.

#### 20.4.4.2.1. Настройка фиксированной скорости

Для настройки соединения с фиксированной скоростью, в файле `ttys` должна быть запись с фиксированной скоростью для `getty`. Для модема, скорость порта которого фиксирована на значении 19.2 Кбит/с, строка в `ttys` может выглядеть так:

```
ttyd0  "/usr/libexec/getty std.19200"  dialup on
```

Если скорость модема фиксирована на другом значении, подставьте соответствующее значение в `std.speed` вместо `std.19200`. Убедитесь, что вы используете тип, описанный в `/etc/gettytab`.

#### 20.4.4.2.2. Настройка переменной скорости

В настройке с переменной скоростью, запись в `ttys` должна обращаться к соответствующей "auto-baud" (sic) записи в `/etc/gettytab`. Например, если вы добавите предложенную выше запись для подключения модема с переменной скоростью, которая начинается с 19.2 Кбит/с (запись в `gettytab` начинается с `V19200`), запись в `ttys` может выглядеть так:

```
ttyd0  "/usr/libexec/getty V19200"  dialup on
```

#### 20.4.4.3. `/etc/rc.serial`

Для высокоскоростных модемов, таких как V.32, V.32bis и V.34, требуется использование аппаратного контроля передачи (RTS/CTS). Вы можете добавить команды `stty` к файлу `/etc/rc.serial` для установки флага аппаратного контроля передачи в ядре FreeBSD для модемных портов.

Например, для установки флага `termios crtscts` на последовательном порту номер 1 (COM2) при инициализации устройств для входящей и исходящей связи, в `/etc/rc.serial` должны быть добавлены следующие строки:

```
# Serial port initial configuration
stty -f /dev/ttyid1 crtscts
stty -f /dev/cuaial1 crtscts
```

## 20.4.5. Настройка модема

Если параметры вашего модема могут быть сохранены в энергонезависимой памяти, потребуется использовать терминальную программу (например, Telix под MS-DOS или `tip` под FreeBSD) для установки параметров. Подсоединитесь к модему, используя ту же скорость соединения, которую использует `getty` в качестве начальной скорости, и настройте модем для соответствия следующим требованиям:

- CD включен после соединения
- DTR включен во время работы; сброс DTR отключает линию и переводит модем в начальное состояние
- CTS контроль переданных данных
- Контроль потока XON/XOFF отключен
- RTS контроль принятых данных
- "Тихий" режим (без кодов возврата)
- Эхо команд отключено

Прочтите документацию на модем для определения какие команды и/или DIP переключатели требуются чтобы установить эти настройки.

Например, для установки вышеуказанных параметров на внешнем 14,400 модеме U.S. Robotics® Sportster®, требуется отправить модему следующие команды:

```
ATZ
AT&C1&D2&H1&I0&R2&W
```

Вы, возможно, захотите настроить и другие параметры модема, такие как использование сжатия V.42bis и/или MNP5.

Внешний U.S. Robotics Sportster 14,400 модем также снабжен некоторыми DIP переключателями, которые требуется установить; для других модемов эти настройки могут быть использованы в качестве примера:

- Переключатель 1: вверх — нормальный DTR
- Переключатель 2: N/A (визуальные коды возврата/числовые коды возврата)
- Переключатель 3: вверх — подавление кодов возврата
- Переключатель 4: вниз — без эхо, offline команды
- Переключатель 5: вверх — авто ответ
- Переключатель 6: вверх — нормальный контроль несущей
- Переключатель 7: вверх — загрузить установки по умолчанию из NVRAM
- Переключатель 8: N/A (Smart/Dumb режимы)

Коды возврата должны быть отключены/подавлены для устранения проблем, которые могут возникнуть, если `getty` ошибочно выдаст приглашение `login:` модему в командном режиме и модем вернет (echo) эту команду или код возврата. Эта последовательность может привести к дополнительному и бессмысленному обмену командами между `getty` и модемом.

### 20.4.5.1. Настройка фиксированной скорости

Для настройки фиксированной скорости вам потребуется настроить модем с поддержкой постоянной скорости обмена данными модем-компьютер независимо от скорости соединения. На внешнем модеме U.S. Robotics Sportster 14,400 эти команды зафиксируют скорость передачи модем-компьютер на скорости, которая установлена при выполнении команды:

```
ATZ
AT&B1&W
```

### 20.4.5.2. Настройка переменной скорости

Для настройки переменной скорости вам потребуется настроить модем с поддержкой изменения скорости передачи данных через последовательный порт в соответствии с скоростью соединения. Следующие команды зафиксируют скорость передачи данных с коррекцией ошибок внешнего модема U.S. Robotics Sportster 14,400 на значении, которое установлено при выполнении команды, но сделают возможным изменение скорости последовательного порта для соединений без коррекции ошибок:

```
ATZ
AT&B2&W
```

### 20.4.5.3. Проверка настроек модема

Большинство высокоскоростных модемов предоставляют команды для просмотра текущих параметров модема в виде, отчасти приспособленном для чтения. Для внешних модемов U.S. Robotics Sportster 14,400 команда AT15 отображает установки, сохраненные в энергонезависимой памяти. Для просмотра действующих параметров модема (с учетом положения DIP переключателей), используйте команду ATZ, а затем AT14.

Если ваш модем другого производителя, проверьте руководство к модему для аккуратной проверки параметров настройки модема.

## 20.4.6. Решение проблем

Вот несколько шагов, которые нужно выполнить для проверки настроек.

### 20.4.6.1. Проверьте систему FreeBSD

Подсоедините модем к системе FreeBSD, загрузите систему, и, если на модеме есть индикаторы, посмотрите, загорелся ли индикатор DTR при появлении приглашения login: на системной консоли — если он загорелся, это означает, что FreeBSD запустила процесс getty на соответствующем коммуникационном порту и модем ожидает входящего звонка.

Если индикатор DTR не загорелся, войдите на консоль системы FreeBSD и выполните команду ps ax, чтобы увидеть, пытается ли FreeBSD запустить процесс getty на соответствующем порту. Вы должны увидеть строки вроде этих среди показанных процессов:

```
114 ?? I      0:00.10 /usr/libexec/getty V19200 ttyd0
```

```
115 ?? I      0:00.10 /usr/libexec/getty V19200 ttyd1
```

Если вы видите что-то другое, вроде этого:

```
114 d0 I      0:00.10 /usr/libexec/getty V19200 ttyd0
```

и модем все еще не принимает звонок, это означает, что `getty` завершила открытие коммуникационного порта. Это может означать проблему с кабелем или неправильную настройку модема, поскольку `getty` не должна открывать коммуникационный порт, пока модем не установит CD (обнаружение несущей).

Если вы не видите процессов `getty`, ожидающих открытия соответствующего порта `ttydN`, внимательно проверьте записи в `/etc/ttys` и попробуйте найти ошибки, если они есть. Проверьте также лог файл `/var/log/messages`, нет ли там сообщений от `init` или `getty`, имеющих отношение к проблеме. Если сообщения есть, проверьте еще раз файлы настройки `/etc/ttys` и `/etc/gettytab`, как и соответствующие специальные файлы устройств `/dev/ttydN`, чтобы обнаружить ошибки, отсутствующие записи или отсутствующие специальные файлы устройств.

#### 20.4.6.2. Попробуйте позвонить на модем

Попробуйте дозвониться до системы; убедитесь, что используете 8 бит без четности и 1 стоп бит на удаленной системе. Если вы не получите приглашение сразу, или получите случайные данные, попробуйте нажимать `Enter` примерно раз в секунду. Если вы все еще не видите приглашения `login:` после нескольких попыток, попробуйте отправить команду `BREAK`. Если вы используете для дозвона высокоскоростной модем, попробуйте позвонить еще раз после фиксирования скорости интерфейса дозваниваемого модема (например, с помощью команды `AT&V1` для модема `U.S. Robotics Sportster`).

Если вы все еще не можете получить приглашение `login:`, проверьте `/etc/gettytab` еще раз и убедитесь, что

- Имя параметра `getty`, указанного в `/etc/ttys`, совпадает с именем параметра в `/etc/gettytab`
- Каждая запись `px=` соответствует имени другой записи в `gettytab`
- Каждая запись `tc=` соответствует имени другой записи в `gettytab`

Если система `FreeBSD` не отвечает на звонок, убедитесь, что модем настроен для ответа на звонок при включении `DTR`. Если модем настроен правильно, проверьте, что `DTR` включается, взглянув на индикаторы модема (если они есть).

Если вы проверили все несколько раз и все еще не добились результата, сделайте перерыв и вернитесь к настройкам позже. Если опять ничего не получилось, возможно вам потребуется отправить письмо в Список рассылки, посвященный вопросам и ответам пользователей `FreeBSD` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions>), описав модем и возникшую проблему, участники рассылки попробуют помочь вам.

## 20.5. Исходящие соединения по модему

Текст, приведенный ниже, это советы, позволяющие настроить ваш хост для доступа к другому компьютеру через модем. Они подходят для установления терминальной сессии с удаленным хостом.

Это подходит для входа на BBS.

Этот вид соединения может очень выручить, если требуется получить файл из интернет и есть проблемы с PPP. Если вам требуется зайти куда-то по FTP, а PPP не работает, используйте терминальную сессию для получения файла по FTP. Затем используйте `zmodem` для сброса его на свой компьютер.

### 20.5.1. Мой модем **Stock Hayes** не поддерживается, что я могу сделать?

На самом деле, страница руководства для `tip` устарела. Встроенная поддержка `generic Hayes` уже есть. Используйте `at=hayes` в файле `/etc/remote`.

Драйвер Hayes не умеет работать с некоторыми расширенными возможностями более новых модемов — сообщения вроде `BUSY`, `NO DIALTONE`, или `CONNECT 115200`. Вы должны отключить эти сообщения при использовании `tip` (с помощью `ATX0&W`).

Таймаут дозвона для `tip` составляет 60 секунд. Ваш модем должен использовать меньшее значение, или `tip` решит, что возникли проблемы со связью. Попробуйте `ATS7=45&W`.

**Замечание:** Оригинальная `tip` не полностью поддерживает модемы Hayes. Решить это проблему можно отредактировав файл `tipconf.h` в каталоге `/usr/src/usr.bin/tip/tip`. Конечно, для этого вам потребуются исходные тексты.

Замените строку `#define HAYES 0` на `#define HAYES 1`. Затем выполните `make` и `make install`. После этого все должно работать отлично.

### 20.5.2. Как нужно выполнять команды AT?

Сделайте то, что называется "прямой" записью в файле `/etc/remote`. Например, если модем подключен к первому последовательному порту, `/dev/cuaa0`, добавьте следующую строку:

```
cuaa0:dv=/dev/cuaa0:br#19200:pa=none
```

Используйте для `br` наибольшее значение `bps`, поддерживаемое модемом. Для подключения к модему выполните `tip cuaa0`.

Если в системе нет устройства `/dev/cuaa0`, выполните:

```
# cd /dev
# sh MAKEDEV cuaa0
```

Или используйте `cu` под `root` так:

```
# cu -lline -sspeed
```

`line` это последовательный порт (например `/dev/cuaa0`), а `speed` это скорость (например 57600). После ввода команд AT наберите `~`, для выхода.



### 20.5.3. Знак @ не работает для pn!

Знак @ в телефонном номере указывает tip взять телефонный номер из /etc/phones. Но знак @ это также специальный символ в таких файлах как /etc/remote. Экранируйте его с помощью обратной косой черты:

```
pn=\\@
```

### 20.5.4. Как я могу позвонить по телефонному номеру из командной строки?

Поместите так называемую "generic" запись в файл /etc/remote. Например:

```
tip115200|Dial any phone number at 115200 bps:\
:dv=/dev/cuaa0:br#115200:at=hayes:pa=none:du:
tip57600|Dial any phone number at 57600 bps:\
:dv=/dev/cuaa0:br#57600:at=hayes:pa=none:du:
```

Затем вы можете сделать следующее:

```
# tip -115200 5551234
```

Если вы предпочитаете cu команде tip, используйте generic запись для cu:

```
c115200|Use cu to dial any number at 115200bps:\
:dv=/dev/cuaa1:br#57600:at=hayes:pa=none:du:
```

и выполните:

```
# cu 5551234 -s 115200
```

### 20.5.5. Должен ли я вводить значение bps каждый раз?

Создайте запись tip1200 или cu1200, но используйте то значение bps, которое записано в поле br. tip считает, что хорошее значение по умолчанию это 1200 bps, поэтому обращается к записи tip1200. Тем не менее, значение bps будет другим.

### 20.5.6. Я получаю доступ ко множеству хостов через терминальный сервер

Вместо ожидания соединения и ввода каждый раз CONNECT <host>, используйте возможность tip cm. Вот пример записи в /etc/remote:

```
pain|pain.deep13.com|Forrester's machine:\
:cm=CONNECT pain\n:tc=deep13:
muffin|muffin.deep13.com|Frank's machine:\
:cm=CONNECT muffin\n:tc=deep13:
deep13:Gizmonics Institute terminal server:\
:dv=/dev/cuaa2:br#38400:at=hayes:du:pa=none:pn=5551234:
```

Она позволит вам вводить tip pain или tip muffin для соединения с хостами pain или muffin, и tip deep13 для доступа к терминальному серверу.

### 20.5.7. Может ли `tip` соединяться более через одну линию для каждого сайта?

Эта проблема часто возникает в университете, где несколько модемных линий и несколько тысяч студентов, пытающихся их использовать.

Создайте запись для университета в `/etc/remote` и используйте `@` для `pn`:

```
big-university:\
    :pn=\@:tc=dialout
dialout:\
    :dv=/dev/cuaa3:br#9600:at=courier:du:pa=none:
```

Затем, создайте список телефонов для университета в `/etc/phones`:

```
big-university 5551111
big-university 5551112
big-university 5551113
big-university 5551114
```

`tip` попыбует связаться с каждым в указанном порядке, затем прекратит попытки. Если вы хотите продолжать соединяться, запустите `tip` в цикле.

### 20.5.8. Почему я должен дважды нажать **Ctrl+P** для отправки **Ctrl+P** один раз?

**Ctrl+P** это "управляющий" символ по умолчанию, используемый для указания `tip` того, что далее идут символьные данные. Вы можете сделать любой другой символ управляющим с помощью экранирования `~s`, которое означает "установить переменную".

Введите `~sf force=single-char`, завершив ввод новой строкой. *single-char* это любой одиночный символ. Если вы не введете *single-char*, управляющим символом станет `nul`, который можно получить, введя **Ctrl+2** или **Ctrl+Space**. Хорошее значение для *single-char* это **Shift+Ctrl+6**, которое используется только на некоторых терминальных серверах.

Вы можете использовать в качестве управляющего символа все, что захотите, поместив его в файл `$HOME/.tiprc`:

```
force=<single-char>
```

### 20.5.9. Почему все, что я ввожу, вдруг стало отображаться в верхнем регистре??

Вы нажали **Ctrl+A**, "повышающий символ" `tip`, который был специально введен для тех, у кого не работает клавиша `caps-lock`. Используйте `~s` как в примере выше для установки переменной `raisechar` в подходящее значение. Фактически, вы можете установить ее в то же значение, что и управляющий символ, если не собираетесь использовать ни один из них.

Вот пример `.tiprc`, отлично подходящий для пользователей **Emacs**, которым часто требуется вводить **Ctrl+2** и **Ctrl+A**:

```
force=^^
raisechar=^^
```

Символ ^^ это **Shift+Ctrl+6**.

### 20.5.10. Могу ли я передавать файлы с помощью `tip`?

Если вы соединяетесь с другой системой UNIX, возможны передача и прием файлов с помощью команды `~р` (`put`) и `~т` (`take`). Эти команды запускают `cat` и `echo` в удаленной системе для приема и передачи файлов. Синтаксис следующий:

```
~р local-fi le [remote-fi le]
```

```
~т remote-fi le [local-fi le]
```

Коррекции ошибок нет, поэтому возможно лучше использовать другой протокол, например `zmodem`.

### 20.5.11. Как мне запустить `zmodem` с `tip`?

Для получения файла запустите отправляющую программу на удаленной стороне. Затем, наберите `~с rz` для начала локального приема файла.

Для отправки файлов запустите принимающую программу на удаленной стороне. Затем, наберите `~с sz` *файлы* для отправки их на удаленную систему.

## 20.6. Настройка последовательной консоли

*Предоставил Kazutaka YOKOTA. Оригинальный документ написал Bill Paul.*

### 20.6.1. Введение

FreeBSD может загружаться при использовании в качестве консоли текстового терминала на последовательном порту. Такая конфигурация может быть полезна в двух случаях: для системных администраторов, устанавливающих FreeBSD на компьютеры без подключенных клавиатуры или монитора, и для разработчиков, производящих отладку ядра или драйверов устройств.

Как описано в Гл. 12, процесс загрузки FreeBSD состоит из трех стадий. Первые две стадии реализованы в блоке загрузки, находящемся в начале слайса FreeBSD на загрузочном диске. На третьей стадии загрузочный блок запускает загрузчик (`/boot/loader`).

Для настройки последовательной консоли вам потребуется настроить блок загрузки, загрузчик и ядро.

### 20.6.2. Настройка последовательной консоли, краткая версия

В этом разделе предполагается, что вы используете настройки по умолчанию, знаете как подключиться к последовательным портам и просто хотите увидеть краткий обзор настройки последовательной консоли. Если эти шаги вызывают у вас затруднения, обратитесь к более подробному разъяснению всех этих параметров и расширенных настроек в Разд. 20.6.3.

1. Подключитесь к последовательному порту. Последовательная консоль будет на COM1.
2. Выполните `echo -h > /boot.config` для включения последовательной консоли для загрузчика и ядра.
3. Отредактируйте `/etc/ttys` и измените `off` на `on` для записи `ttyd0`. Это включит приглашение на вход на последовательной консоли так же, как обычно настраиваются видео консоли.
4. Команда `shutdown -r now` перезагрузит систему с включенной последовательной консолью.

### 20.6.3. Настройка последовательной консоли

1. Подготовьте кабель.

Вам потребуется нуль-модемный или стандартный последовательный кабель и нуль-модемный адаптер. Обратитесь к Разд. 20.2.2, где рассматриваются последовательные кабели.

2. Отключите клавиатуру.

Большинство систем PC тестируют клавиатуру во время включения (POST) и выдают ошибку если клавиатура не обнаружена. Некоторые системы при отсутствии клавиатуры выдают звуковой сигнал и не загружаются пока клавиатура не будет подключена.

Если компьютер сообщает об ошибке, но все же загружается, вам не потребуется делать что-то еще. (Некоторые компьютеры с Phoenix BIOS просто сообщают `Keyboard failed` и продолжают загрузку).

Если компьютер не загружается без клавиатуры, вам потребуется настроить BIOS так, чтобы отсутствие клавиатуры игнорировалось (если это возможно). Обратитесь к руководству по материнской плате за деталями о том, как это сделать.

**Подсказка:** Установка параметра клавиатуры "Not installed" в настройках BIOS не означает, что вы не сможете использовать клавиатуру. Все, что делает этот параметр — указывает BIOS не тестировать клавиатуру во время загрузки, поэтому ее отсутствие не вызывает ошибки. Вы можете оставить клавиатуру подключенной, даже если с флагом "Not installed" и она все еще будет работать.

**Замечание:** Если в к системе подключена PS/2® мышь, отключите ее, как и клавиатуру. Мышь PS/2 использует часть оборудования совместно с клавиатурой, поэтому если оставить ее подключенной, тестирование клавиатуры может ошибочно выдать наличие последней. Например, система Gateway 2000 Pentium 90 MHz ведет себя именно так. К тому же, это не проблема, поскольку мышь без клавиатуры как правило не нужна.

3. Подключите текстовый терминал к COM1 (sio0).

Если у вас нет текстового терминала, используйте старый PC/XT с модемной программой, или последовательный порт на другом компьютере UNIX. Если порта COM1 (sio0) нет, подключите его. На данный момент нет способа использовать другой порт вместо COM1 без перекомпиляции загрузочных блоков. Если вы уже используете COM1 для подключения другого устройства, временно удалите это устройство установите новый загрузочный блок и ядро как только FreeBSD

заработает. (Предполагается, что COM1 будет доступен на файловом/вычислительном/терминальном сервере в любом случае; если вам действительно требуется COM1 для чего-то другого (и вы не можете переключить это на COM2 (sio1)), возможно не стоит беспокоиться об этом сейчас.)

4. Убедитесь, что в файле настройки ядра установлены соответствующие флаги для COM1 (sio0).

Подходящие флаги такие:

0x10

Включает поддержку консоли для этого устройства. Если установлен этот флаг, другие игнорируются. На данный момент поддержка консоли может быть включена не более чем на одном устройстве; предпочтительно на первом (в соответствии с порядком в конфигурационном файле) с установкой этого флага. Эта опция сама по себе не сделает последовательный порт консолью. Установите следующий флаг или используйте опцию `-h`, описанную ниже, вместе с этим флагом.

0x20

Включает поддержку консоли на устройстве (если нет другой консоли с более высоким приоритетом), независимо от наличия описываемой ниже опции `-h`. Этот флаг заменил опцию COMCONSOLE в FreeBSD версий 2.X. Флаг 0x20 должен использоваться вместе с флагом 0x10.

0x40

Резервирует это устройство (совместно с флагом 0x10) и делает устройство недоступным для обычной работы. Вы не должны использовать этот флаг для устройства последовательного порта, которое будет использоваться в качестве последовательной консоли. Используйте этот флаг только если устройство предназначено для удаленной отладки ядра. Обратитесь к Руководству для разработчиков ([http://www.FreeBSD.org/doc/en\\_US.ISO8859-1/books/developers-handbook/index.html](http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/developers-handbook/index.html)) за дополнительной информацией по удаленной отладке.

**Замечание:** В FreeBSD 4.0 или выше семантика флага 0x40 немного другая и для удаленной отладки используется другой флаг.

Пример:

```
device sio0 at isa? port IO_COM1 flags 0x10 irq 4
```

Обратитесь к странице справочника sio(4) за подробностями.

Если флаги не были установлены, вам потребуется запустить UserConfig (на другой консоли) или пересобрать ядро.

5. Создайте `boot.config` в корневом каталоге раздела `a` на загрузочном диске.

Этот файл сообщит загрузочному блоку способ загрузки системы. Для активации последовательной консоли вам потребуется одна или несколько следующих опций — несколько опций могут быть указаны на одной строке:

-h

Переключает внутреннюю и последовательную консоль. Вы можете использовать ее для переключения устройств консоли. Например, при загрузке с внутренней (видео) консоли, вы можете использовать -h для запуска загрузчика и ядра с использованием последовательного порта в качестве устройства консоли. При загрузке с последовательной консоли, вы можете использовать опцию -h для указания загрузчику и ядру использовать в качестве консоли видео дисплей.

-D

Переключает одно- и двухконсольную конфигурации. В одноконсольной конфигурации консоль может быть либо внутренней (видео дисплей), либо последовательным портом, в зависимости от состояния опции -h. В двухконсольной конфигурации и видео дисплей и последовательный порт станут консолями одновременно, независимо от состояния опции -h. Имейте ввиду, что конфигурация с двумя консолями работает только во время работы загрузочного блока. Как только управление переходит к загрузчику, остается только одна консоль, указанная опцией -h.

-P

Указывает загрузочному блоку протестировать клавиатуру. Если клавиатура не найдена, автоматически устанавливаются параметры -D и -h.

**Замечание:** По причине ограничений на размер в существующей версии загрузочного блока, опция -P может протестировать только расширенные клавиатуры. Клавиатуры с менее чем 101 клавишами (и без клавиш F11 и F12) могут быть не обнаружены. Клавиатуры некоторых лэптопов могут быть не найдены из-за этого ограничения. Если это случилось, вы не сможете использовать опцию -P. К сожалению, не существует обходного пути решения этой проблемы.

Используйте или опцию -P для автоматического выбора консоли, или опцию -h для активации последовательной консоли.

Вы можете включить также другие опции, описанные в `boot(8)`.

Опции, за исключением -P, будут переданы загрузчику (`/boot/loader`). Загрузчик определит будет ли консолью внутреннее видео устройство или последовательный порт, проверив только состояние опции -h. Это означает, что если вы включите в `/boot.config` опцию -D, но не -h, то сможете использовать консоль только во время работы загрузочного блока; загрузчик будет использовать внутреннее видео устройство в качестве консоли.

## 6. Загрузите компьютер.

Когда вы включите компьютер FreeBSD, загрузочный блок выведет содержимое `/boot.config` на консоль. Например:

```
/boot.config: -P
keyboard: no
```

Вторая строка появится только если вы поместите -P в `/boot.config` и отражает наличие/отсутствие клавиатуры. Эти сообщения выводятся либо на последовательную, либо на внутреннюю консоль, или на обе, в зависимости от параметров в `/boot.config`.

Опции	Сообщения выводятся на
нет	внутренняя консоль
-h	последовательная консоль
-D	последовательная и внутренняя консоли
-Dh	последовательная и внутренняя консоли
-P, клавиатура присутствует	внутренняя консоль
-P, клавиатура отсутствует	последовательная консоль

После вывода вышеприведенных сообщений, происходит небольшая пауза перед тем, как запускается загрузчик и на консоли появляются следующие сообщения. В нормальной ситуации вам не потребуется прерывать загрузку в этот момент, но это можно сделать, чтобы убедиться, что все настроено правильно.

Нажмите на консоли любую клавишу кроме Enter для прерывания процесса загрузки. Загрузочный блок выдаст приглашение к дальнейшим действиям. Оно выглядит примерно так:

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/boot/loader
boot:
```

Убедитесь, что сообщение выше появилось на последовательной, внутренней консоли или на обеих, в зависимости от опций в `/boot.config`. Если сообщение появилось там, где должно было появиться, нажмите Enter для продолжения процесса загрузки.

Если вам нужна последовательная консоль, но на терминале не видно приглашения, это означает проблемы с настройками. Введите `-h` и нажмите Enter/Return (если это возможно) для указания загрузочному блоку (а также загрузчику и ядру) выбрать последовательный порт в качестве консоли. Когда система загрузится, проверьте настройки еще раз и определите, что было сделано неправильно.

После запуска загрузчика и перехода в третью стадию процесса загрузки вы все еще можете переключиться между внутренней консолью и последовательной консолью путем установки соответствующих переменных окружения в загрузчике. Обращайтесь к разделу Разд. 20.6.6.

## 20.6.4. Итоги

Здесь приведены краткие итоги по различным настройкам, рассмотренным в этом разделе и выбираемым в соответствии с ними консолям.

### 20.6.4.1. Вариант 1: вы устанавливаете для `sio0` флаги `0x10`

```
device sio0 at isa? port IO_COM1 flags 0x10 irq 4
```

Параметры в <code>/boot.config</code>	Консоль для загрузочного блока	Консоль для загрузчика	Консоль для ядра
нет	внутренняя	внутренняя	внутренняя
-h	последовательная	последовательная	последовательная

Параметры в /boot.config	Консоль для загрузочного блока	Консоль для загрузчика	Консоль для ядра
-D	последовательная и внутренняя	внутренняя	внутренняя
-Dh	последовательная и внутренняя	последовательная	последовательная
-P, клавиатура присутствует	внутренняя	внутренняя	внутренняя
-P, клавиатура отсутствует	последовательная и внутренняя	последовательная	последовательная

#### 20.6.4.2. Вариант 2: вы устанавливаете для `sio0` флаги `0x30`

```
device sio0 at isa? port IO_COM1 flags 0x30 irq 4
```

Параметры в /boot.config	Консоль для загрузочного блока	Консоль для загрузчика	Консоль для ядра
нет	внутренняя	внутренняя	последовательная
-h	последовательная	последовательная	последовательная
-D	последовательная и внутренняя	внутренняя	последовательная
-Dh	последовательная и внутренняя	последовательная	последовательная
-P, клавиатура присутствует	внутренняя	внутренняя	последовательная
-P, клавиатура отсутствует	последовательная и внутренняя	последовательная	последовательная

### 20.6.5. Приемы работы с последовательной консолью

#### 20.6.5.1. Установка более высокой скорости порта

По умолчанию, последовательный порт настроен так: 9600 бит/с, 8 бит, без четности, 1 стоп бит. Если вам необходимо изменить скорость, потребуется перекомпиляция как минимум загрузочных блоков. Добавьте следующую строку к `/etc/make.conf` и скомпилируйте новый загрузочный блок:

```
BOOT_COMCONSOLE_SPEED=19200
```

Обратитесь к Разд. 20.6.5.2 за подробными инструкциями по сборке и установке новых загрузочных блоков.

Если последовательная консоль настраивается не путем установки параметра `-h`, или последовательная консоль, используемая ядром, отличается от той, что используется загрузочным блоком, потребуется добавить следующие опции к файлу настройки ядра и собрать новое ядро:



```
options CONSPEED=19200
```

### 20.6.5.2. Использование для консоли другого последовательного порта вместо `sio0`

Использование другого последовательного порта вместо `sio0` для консоли потребует кое-какой перекомпиляции. Если вы по каким-либо причинам хотите использовать другой последовательный порт, перекомпилируйте загрузочный блок, загрузчик и ядро согласно приведенной ниже инструкции.

1. Получите исходные тексты ядра (глава Гл. 19)
2. Отредактируйте `/etc/make.conf` и установите `BOOT_COMCONSOLE_PORT` в соответствии с адресом порта, который вы хотите использовать (`0x3F8`, `0x2F8`, `0x3E8` или `0x2E8`). Могут быть использованы только устройства от `sio0` до `sio3` (от `COM1` до `COM4`); мультипортовые последовательные карты не будут работать. Установка прерываний не требуется.
3. Создайте файл настройки ядра и добавьте соответствующие флаги для порта, который планируется использовать. Например, если вы хотите использовать для консоли `sio1` (`COM2`):

```
device sio1 at isa? port IO_COM2 flags 0x10 irq 3
```

или

```
device sio1 at isa? port IO_COM2 flags 0x30 irq 3
```

Флаги для других последовательных устройств не устанавливайте.

4. Соберите и установите загрузочный блок и загрузчик:

```
# cd /sys/boot
# make clean
# make
# make install
```

5. Соберите и установите ядро.
6. Запишите загрузочный блок на загрузочный диск с помощью `disklabel(8)` и загрузитесь с новым ядром.

### 20.6.5.3. Вход в отладчик DDB с последовательной линии

Если вы хотите войти в отладчик ядра с последовательной консоли (полезно для удаленной диагностики, но опасно если вы введете неправильный **BREAK** на последовательном порту!), потребуется собрать ядро со следующими параметрами:

```
options BREAK_TO_DEBUGGER
options DDB
```

### 20.6.5.4. Получение приглашения на последовательной консоли

Хотя это не обязательно, вам может потребоваться приглашение *login* по последовательной линии, в дополнение к уже доступным загрузочным сообщениям и отладочной сессии ядра. Здесь описано как сделать это.

Откройте файл `/etc/ttys` с помощью редактора и найдите строки:

```
ttyd0 "/usr/libexec/getty std.9600" unknown off secure
ttyd1 "/usr/libexec/getty std.9600" unknown off secure
ttyd2 "/usr/libexec/getty std.9600" unknown off secure
ttyd3 "/usr/libexec/getty std.9600" unknown off secure
```

Строки от `ttyd0` до `ttyd3` соответствуют портам от `COM1` до `COM4`. Измените `off` на `on` для требуемого порта. Если вы изменили скорость последовательного порта, может потребоваться изменить `std.9600` для соответствия текущим настройкам, например `std.19200`.

Возможно, вы захотите заменить тип терминала `unknown` на тип реально используемого терминала.

После редактирования файла потребуется выполнить `kill -HUP 1` для включения новых настроек.

## 20.6.6. Изменение консоли из загрузчика

Предыдущий раздел описывает настройку последовательной консоли изменением параметров загрузочного блока. Этот раздел показывает, как указать консоль, вводя команды и переменные окружения для загрузчика. Поскольку загрузчик загружается после загрузочного блока, на третьей стадии загрузочного процесса, настройки загрузчика превагируют над настройками загрузочного блока.

### 20.6.6.1. Настройка последовательной консоли

Вы можете прямо указать загрузчику и ядру использовать последовательную консоль, записав одну строку в `/boot/loader.rc`:

```
set console=comconsole
```

Это сработает независимо от настроек загрузочного блока, рассмотренных в предыдущем разделе.

Поместите эту строку в самое начало `/boot/loader.rc`, чтобы увидеть на последовательной консоли все загрузочные сообщения.

Вы можете также указать внутреннюю консоль:

```
set console=vidconsole
```

Если вы не установите переменную загрузчика `console`, загрузчик, а затем и ядро будут использовать ту консоль, которая установлена параметром `-h` для загрузочного блока.

В версиях 3.2 или выше, вы можете указать консоль в `/boot/loader.conf.local` или `/boot/loader.conf` вместо `/boot/loader.rc`. С этим методом `/boot/loader.rc` должен выглядеть примерно так:

```
include /boot/loader.4th
start
```

Затем, создайте `/boot/loader.conf.local` и поместите туда следующую строку.

```
console=comconsole
```

или

```
console=vidconsole
```

Обращайтесь к loader.conf(5) за дополнительной информацией.

**Замечание:** На данный момент у загрузчика нет параметра, эквивалентного параметру `-P` загрузочного блока и нет способа автоматического выбора внутренней и последовательной консоли в зависимости от наличия клавиатуры.

#### 20.6.6.2. Использование для консоли отличного от `si00` последовательного порта

Вам потребуется перекомпилировать загрузчик для использования отличного от `si00` последовательного порта в качестве консоли. Следуйте процедуре, описанной в разделе Разд. 20.6.5.2.

#### 20.6.7. Предостережения

Идея в том, чтобы настроить выделенный сервер, который не требует графического оборудования или подсоединенной клавиатуры. К сожалению, хотя многие системы способны загрузиться без клавиатуры, есть совсем немного систем, способных загрузиться без графического адаптера. Компьютеры с AMI BIOS могут быть настроены для загрузки без графического адаптера простой установкой параметра настройки CMOS "graphics adapter" в значение "Not installed".

Однако, многие компьютеры не поддерживают этот параметр и не смогут загрузиться без графического оборудования. Для этих компьютеров вам потребуется оставить подключенной любую графическую карту (даже если это просто старая моно карта), хотя монитор и не подключен.

# Глава 21. PPP и SLIP

*Реструктурировал, реорганизовал и обновил Jim Mock. Перевод на русский язык: Денис Пеплин.*

## 21.1. Краткий обзор

В FreeBSD существует множество способов соединения одного компьютера с другим. Для установления соединения с отдельной сетью или интернет через обычный модем, или для открытия доступа к собственному компьютеру необходимо использование PPP или SLIP. В этой главе детально описана настройка таких модемных сервисов.

После прочтения этой главы вы будете знать:

- Как настроить PPP уровня пользователя (user PPP).
- Как настроить PPP уровня ядра (kernel PPP).
- Как настроить PPPoE (PPP over Ethernet).
- Как настроить PPPoA (PPP over ATM).
- Как настроить SLIP клиента и сервер.

Перед прочтением этой главы вам потребуется:

- Ознакомиться с основными сетевыми технологиями.
- Понимать основы и назначение модемного соединения и PPP и/или SLIP.

Вы возможно захотите узнать, в чем главное различие между PPP уровня пользователя и ядра. Ответ прост: PPP уровня пользователя обрабатывает входящие и исходящие данные в пространстве пользователя, а не в ядре. В терминах копирования данных между ядром и пространством пользователя это дорогостоящий путь, который однако позволяет значительно расширить возможности реализации PPP. PPP уровня пользователя для связи с внешним миром использует устройство `tun`, а PPP уровня ядра — устройство `ppp`.

**Замечание:** В дальнейшем в этой главе PPP уровня пользователя будет обозначаться просто как **ppp**, пока не потребуется различать его и любое другое программное обеспечение PPP, такое как **pppd**. Если не указано иначе, все команды, приведенные в этой главе, должны выполняться под `root`.

## 21.2. PPP уровня пользователя

*Обновил и расширил Tom Rhodes. Первоначально предоставил Brian Somers. Дополнительную*

информацию предоставили Nik Clayton, Dirk Frömberg, Peter Childs.

## 21.2.1. Настройка PPP уровня пользователя

### 21.2.1.1. Предположения

В этом документе предполагается, что у вас есть следующее:

- Учетная запись у провайдера интернет (Internet Service Provider, ISP), к которому вы подсоединяетесь, используя PPP.
- Модем или другое подключенное к системе и правильно настроенное устройство, позволяющее подключиться к провайдеру.
- Номер модемного пула провайдера.
- Имя пользователя (логин) и пароль (обычная UNIX пара логин/пароль, или PAP/CHAP пара логин/пароль).
- IP адреса одного или нескольких серверов имен. Обычно провайдер дает для этих целей два IP адреса. Если нет ни одного, вы можете использовать команду `enable dns` в `ppp.conf` и **ppp** настроит список серверов имен. Эта возможность зависит от наличия поддержки согласования DNS в реализации PPP провайдера.

Следующая информация может поставляться провайдером, но не является совершенно необходимой:

- IP адрес шлюза провайдера. Шлюз это компьютер, к которому вы подключитесь и который будет настроен в качестве *маршрута по умолчанию (default route)*. Если у вас нет этой информации, она может быть получена от PPP сервера после подключения.

Программой **ppp** этот IP адрес обозначается как HISADDR.

- Сетевая маска, которую вы должны использовать. Если провайдер не предоставил ее значение, вы можете использовать 255.255.255.255.
- Если провайдер предоставил статический IP и имя хоста, используйте их. Иначе позвольте удаленной стороне назначить свободный IP адрес.

Если у вас нет всей необходимой информации, свяжитесь с провайдером.

**Замечание:** В этом разделе строки файлов настройки из многих примеров пронумерованы. Эти номера приведены только для обсуждения настроек, они не должны помещаться в действующую настройку. Правильные отступы с табуляцией и пробелами также важны.

### 21.2.1.2. Создание файлов устройств PPP

В обычной ситуации, большинству пользователей нужно только одно устройство `tun (/dev/tun0)`. Ссылки на `tun0` ниже могут быть заменены на `tunN`, где *N* это любой номер устройства, соответствующий вашей системе.

Для систем FreeBSD без devfs(5) (FreeBSD 4.X более ранние), необходимо проверить устройство tun0 (это не требуется, если включена devfs(5), поскольку файлы устройств будут создаваться автоматически).

Простейший способ убедиться, что устройство tun0 настроено правильно, это пересоздать устройство. Для пересоздания устройства выполните следующее:

```
# cd /dev
# sh MAKEDEV tun0
```

Если вам необходимы 16 туннельных устройств, потребуется их создать. Это можно сделать, выполнив следующие команды:

```
# cd /dev
# sh MAKEDEV tun15
```

### 21.2.1.3. Автоматическая настройка PPP

И ppp и pppd (реализация PPP уровня ядра) используют файлы настройки, расположенные в каталоге /etc/ppp. Примеры для ppp уровня пользователя можно найти в /usr/share/examples/ppp/.

Настройка ppp требует редактирования нескольких файлов, в зависимости от ваших потребностей. То, что вы поместите в эти файлы, зависит в некоторой степени от того, предоставит ли провайдер статический IP адрес (т.е. вы получите один определенный IP адрес и будете использовать его постоянно) или динамический (т.е. ваш IP адрес будет изменяться при каждом подключении к провайдеру).

#### 21.2.1.3.1. PPP и статические IP адреса

Вам потребуется отредактировать файл настройки /etc/ppp/ppp.conf. Он похож на приведенный ниже пример.

**Замечание:** Строки, оканчивающиеся на :, вводятся без отступа в начале строки, остальные строки должны быть введены с отступом, как показано в примере.

```
1  default:
2      set log Phase Chat LCP IPCP CCP tun command
3      ident user-ppp VERSION (built COMPILATIONDATE)
4      set device /dev/cuaa0
5      set speed 115200
6      set dial "ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 5 \
7              \\\"\" AT OK-AT-OK ATE1Q0 OK \\dATDT\\\"T TIMEOUT 40 CONNECT"
8      set timeout 180
9      enable dns
10
11  provider:
12      set phone "(123) 456 7890"
13      set authname foo
14      set authkey bar
15      set login "TIMEOUT 10 \\\"\" \\\"\" gin:--gin: \\U word: \\P col: ppp"
```

```

16      set timeout 300
17      set ifaddr x.x.x.x y.y.y.y 255.255.255.255 0.0.0.0
18      add default HISADDR

```

Строка 1:

Начинает настройку по умолчанию (default). Команды этой настройки выполняются автоматически при запуске ppp.

Строка 2:

Включает параметры протоколирования. Когда настройка работает удовлетворительно, эта строка должна быть сокращена до следующей формы

```
set log phase tun
```

для предотвращения появления слишком больших лог файлов.

Строка 3:

Указывает PPP как идентифицировать себя на удаленной стороне. PPP идентифицирует себя на удаленной стороне если возникают проблемы согласования и установки соединения, предоставляя информацию, по которой администратор на удаленной стороне может воспользоваться для решения таких проблем.

Строка 4:

Указывает устройство, к которому подключен модем. COM1 это /dev/cuaa0, а COM2 это /dev/cuaa1.

Строка 5:

Устанавливает желаемую скорость подключения к модему. Если 115200 не работает (хотя должна работать для любого относительно нового модема) попробуйте 38400.

Строки 6 и 7:

Строка дозвола. PPP уровня пользователя применяет expect-send синтаксис, похожий на синтаксис chat(8). Обратитесь к странице справочника за информацией о возможностях этого языка.

Обратите внимание, что эта команда продолжается на следующей строке для улучшения читаемости. Любая команда в ppp.conf может быть продолжена на следующей строке, если последний символ предыдущей строки “\”.

Строка 8:

Устанавливает предельное время ожидания для соединения. Значение 180 секунд используется по умолчанию, так что строка с этим значением чисто косметическая.

Строка 9:

Указывает PPP запросить у удаленной стороны сервера имен. Если вы работаете с локальным сервером имен, эта строка должна быть закомментирована или удалена.

## Строка 10:

Пустая строка для улучшения читаемости. Пустые строки игнорируются PPP.

## Строка 11:

Определяет настройки для провайдера, называемого "provider". Это имя может быть изменено на имя вашего провайдера, чтобы в дальнейшем вы могли использовать `load provider` для начала соединения.

## Строка 12:

Определяет телефонный номер для этого провайдера. Несколько телефонных номеров могут быть указаны с помощью двоеточия (:) или символа канала (|) в качестве разделителя. Различия между двумя разделителями описаны в `ppp(8)`. Главным образом они заключаются в том, что если вы хотите перебирать номера, используйте двоеточие. Если вы хотите дозваниваться по первому номеру в первую очередь, и использовать другие номера только если дозвон по первому завершится неудачно, используйте символ канала. Всегда заключайте список номеров в кавычки, как показано в примере.

Вы должны включить телефонный номер в кавычки (") если в нем используются пробелы. Отсутствие кавычек может вызвать простую, но трудно обнаруживаемую ошибку.

## Строки 13 и 14:

Задают имя пользователя и пароль. При использовании приглашения `login` в стиле UNIX, эти значения используются командой `set login` через переменные `\U` и `\P`. При соединении с использованием PAP или CHAP, эти значения используются во время аутентификации.

## Строка 15:

Если вы используете PAP или CHAP, приглашение на вход не появится, и эта строка должна быть закомментирована или удалена. Обратитесь к странице аутентификация PAP и CHAP за дальнейшей информацией.

Строка для входа записана в том же `chat`-подобном синтаксисе, что и строка для дозвона. В этом примере строка работает для сервиса, сессия входа которого выглядит примерно так:

```
J. Random Provider
login: foo
password: bar
protocol: ppp
```

Вам потребуется изменить эту строку для использования с другим сервисом. При первом составлении скрипта убедитесь, что вы включили "chat" протоколирование, чтобы убедиться, что соединение происходит как ожидалось.

## Строка 16:

Установка максимального времени ожидания по умолчанию для соединения. В данном случае соединение будет разорвано автоматически после 300 секунд неактивности. Если вы не хотите,



чтобы соединение разрывалось, установите эту переменную в нуль, или используйте параметр командной строки `-ddial`.

#### Строка 17:

Устанавливает адрес интерфейса. Строка `x.x.x.x` должна быть заменена на IP адрес, который выделил вам провайдер. Строка `y.y.y.y` должна быть заменена на IP адрес шлюза провайдера (компьютер, к которому вы подключаетесь). Если провайдер не сообщил адрес шлюза, используйте `10.0.0.2/0`. Если вам требуется использовать "вычисленные" адреса, убедитесь, что создана запись в `/etc/ppp/ppp.linkup` в соответствии с инструкциями для PPP и динамических IP адресов. Если эта строка опущена, `ppp` не может быть запущен в режиме `-auto`.

#### Строка 18:

Добавляет маршрут по умолчанию к шлюзу провайдера. Специальное слово `HISADDR` заменяется адресом шлюза, указанном в строке 17. Важно, чтобы эта строка появилась после строки 17, до нее переменная `HISADDR` еще не инициализирована.

Если вы не будете запускать `ppp` с параметром `-auto`, эта строка должна быть перемещена в файл `ppp.linkup`.

Нет необходимости добавлять запись в `ppp.linkup`, если у вас статический IP адрес и `ppp` работает в режиме `-auto`, поскольку таблица маршрутизации настроена правильно еще до подключения. Однако, вы возможно захотите создать запись для запуска программ после соединения. Эта ситуация описана далее в примере по `sendmail`.

Примерные файлы настройки находятся в каталоге `/usr/share/examples/ppp/`.

#### 21.2.1.3.2. PPP и динамические IP адреса

Если провайдер не выделил статический IP адрес, `ppp` может быть настроен для определения локального и удаленного адреса. Это делается путем "вычисления" IP адреса и настройки его программой `ppp` с использованием IP Configuration Protocol (ICP) после установления соединения. Файл настройки `ppp.conf` тот же, что и в примере PPP и статические IP адреса, со следующим изменением:

```
17      set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.255
```

Как и раньше, не включайте номер строки, он используется только для ссылки на строку в этом примере. Требуется отступ хотя бы в один пробел.

#### Строка 17:

Номер после символа `/` это число бит в адресе, которые будут запрошены `ppp`. Вы можете использовать более подходящие вам IP адреса, но пример выше всегда будет работать.

Если вы не используете режим `-auto`, потребуются создать запись в `/etc/ppp/ppp.linkup`. Этот файл используется после установки соединения. На этот момент `ppp` уже настроит адреса интерфейсов и станет возможным добавление записей в таблицу маршрутизации:

```
1      provider:
2      add default HISADDR
```

Строка 1:

При установке соединения, `ppp` ищет запись в `ppp.linkup` по следующим правилам: сначала в соответствии с меткой, используемой в `ppp.conf`. Если это не сработает, ведется поиск записи для IP адреса шлюза. Это метка в IP записывается в виде IP адреса. Если запись все еще не найдена, используется запись `MYADDR`.

Строка 2:

Эта строка сообщает `ppp` добавить маршрут по умолчанию, указывающий на `HISADDR`. `HISADDR` будет заменен на IP адрес шлюза, определенного `IPCP`.

Детальный пример находится в записи `ppmdemand` файлов

`/usr/share/examples/ppp/ppp.conf.sample` и `/usr/share/examples/ppp/ppp.linkup.sample`.

#### 21.2.1.3.3. Прием входящих звонков

При настройке **ppp** для приема входящих звонков на компьютере, подключенном к локальной сети, вам необходимо решить, перенаправлять ли пакеты в локальную сеть. Если вы будете делать это, выделите удаленной стороне IP адрес из диапазона адресов локальной сети, и используйте команду `enable proxy` в файле `/etc/ppp/ppp.conf`. Вам потребуется также убедиться, что в файле `/etc/rc.conf` присутствует строка:

```
gateway_enable="YES"
```

#### 21.2.1.3.4. Какой *getty*?

Раздел Настройка FreeBSD для входящих соединений дает хорошее описание включения входящих соединений с использованием `getty(8)`.

Альтернатива `getty` это `mgetty` (<http://www.leo.org/~doering/mgetty/index.html>), более интеллектуальная версия `getty`, разработанная специально для приема входящих звонков..

Преимущество использование `mgetty` в том, что она активно *общается* с модемами, то есть если порт будет выключен в `/etc/ttys`, модем не будет отвечать на звонок.

Последние версии `mgetty` (от 0.99beta и выше) поддерживают также автоматическое определение потоков PPP, позволяя клиентам безкриптовое подключение к серверу.

Обратитесь к разделу Mgetty и AutoPPP за дальнейшей информацией по `mgetty`.

#### 21.2.1.3.5. Права **PPP**

Программа `ppp` обычно запускается с правами пользователя `root`. Если вы хотите разрешить запуск `ppp` в режиме сервера с правами обычного пользователя путем запуска `ppp` как описано ниже, этого пользователя необходимо добавить в группу `network` в файле `/etc/group`.

Вам также потребуется дать ему доступ к одному или более разделов конфигурации, используя команду `allow`:

```
allow users fred mary
```

Если эта команда используется в разделе `default`, она дает пользователю полный доступ.

#### 21.2.1.3.6. Оболочки PPP для пользователей с динамическими IP

Создайте файл, называющийся `/etc/ppp/ppp-shell` и содержащий следующее:

```
#!/bin/sh
IDENT='echo $0 | sed -e 's/^.*-\(.*\)$/\1/'`
CALLEDAS="$IDENT"
TTY='tty'

if [ x$IDENT = xdialup ]; then
    IDENT='basename $TTY'
fi

echo "PPP for $CALLEDAS on $TTY"
echo "Starting PPP for $IDENT"

exec /usr/sbin/ppp -direct $IDENT
```

Этот скрипт должен быть исполняемым. Теперь создайте на этот скрипт символическую ссылку с именем `ppp-dialup` с помощью следующей команды:

```
# ln -s ppp-shell /etc/ppp/ppp-dialup
```

Используйте этот скрипт в качестве *оболочки* для удаленных пользователей. Ниже приведен пример записи в `/etc/passwd` для удаленных пользователей PPP с именем пользователя `pchilds` (не забывайте использовать `vi` для редактирования файла паролей).

```
pchilds:*:1011:300:Peter Childs PPP:/home/ppp:/etc/ppp/ppp-dialup
```

Создайте каталог `/home/ppp`, который доступен для чтения и содержит следующие файлы нулевой длины:

```
-r--r--r-- 1 root wheel 0 May 27 02:23 .hushlogin
-r--r--r-- 1 root wheel 0 May 27 02:22 .rhosts
```

Это предотвратит отображение `/etc/motd`.

#### 21.2.1.3.7. Оболочки PPP для пользователей со статическими IP

Создайте файл `ppp-shell` как в примере выше, и для каждой учетной записи со статически назначаемым IP создайте символическую ссылку на `ppp-shell`.

Например, если у вас три пользователя удаленного доступа, `fred`, `sam` и `mary`, которые подключаются к вашей сети класса C, выполните следующее:

```
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-fred
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-sam
# ln -s /etc/ppp/ppp-shell /etc/ppp/ppp-mary
```

Оболочка каждого из этих пользователей удаленного доступа должна быть символической ссылкой, созданной выше (например, оболочка пользователя mary должна быть `/etc/ppp/ppp-mary`).

#### 21.2.1.3.8. Настройка `ppp.conf` для пользователей с динамическими IP

Файл `/etc/ppp/ppp.conf` должен содержать примерно такие строки:

```
default:
    set debug phase lcp chat
    set timeout 0

ttyd0:
    set ifaddr 203.14.100.1 203.14.100.20 255.255.255.255
    enable proxy

ttyd1:
    set ifaddr 203.14.100.1 203.14.100.21 255.255.255.255
    enable proxy
```

**Замечание:** Необходимо соблюдать отступы.

Раздел `default:` загружается для каждого соединения. Для каждой строки, включенной в `/etc/ttys`, создайте запись, подобную `ttyd0:` выше. Каждая строка должна содержать уникальные IP адреса из вашего пула IP адресов, выделенных пользователям с динамическими IP.

#### 21.2.1.3.9. Настройка `ppp.conf` для пользователей со статическими IP

Помимо содержимого, описанного в примере файла `/usr/share/examples/ppp/ppp.conf` выше, вам потребуется добавить раздел для каждого из пользователей со статическими IP. Мы продолжим использовать имена `fred`, `sam` и `mary` в качестве примера.

```
fred:
    set ifaddr 203.14.100.1 203.14.101.1 255.255.255.255

sam:
    set ifaddr 203.14.100.1 203.14.102.1 255.255.255.255

mary:
    set ifaddr 203.14.100.1 203.14.103.1 255.255.255.255
```

Если требуется, файл `/etc/ppp/ppp.linkup` должен также содержать информацию о маршрутизации для каждого пользователя со статическим IP. В строке ниже через клиентское соединение добавляется маршрут к сети класса C `203.14.101.0`.

```
fred:
    add 203.14.101.0 netmask 255.255.255.0 HISADDR

sam:
    add 203.14.102.0 netmask 255.255.255.0 HISADDR
```

```
mary:
  add 203.14.103.0 netmask 255.255.255.0 HISADDR
```

### 21.2.1.3.10. mgetty и AutoPPP

Настройка и компиляция mgetty с параметром AUTO\_PPP позволяет mgetty определять LCP фазу PPP соединений и автоматически порождать оболочку ppp. Однако, поскольку стандартный метод логин/пароль не используется, необходима аутентификация пользователей через PAP или CHAP.

В этом разделе предполагается, что пользователь успешно настроил, скомпилировал и установил версию mgetty с параметром AUTO\_PPP (v0.99beta или более поздняя).

Убедитесь, что в файле /usr/local/etc/mgetty+sendfax/login.config имеется следующая строка:

```
/AutoPPP/ - - /etc/ppp/ppp-pap-dialup
```

Это укажет mgetty запускать скрипт ppp-pap-dialup для обнаруженных соединений PPP.

Создайте файл /etc/ppp/ppp-pap-dialup, содержащий следующее (этот файл должен быть выполняемым):

```
#!/bin/sh
exec /usr/sbin/ppp -direct pap$IDENT
```

Для каждой линии, включенной в /etc/ttys, создайте соответствующую запись в /etc/ppp/ppp.conf. Она будет отлично сочетаться с тем, что было создано выше.

```
pap:
  enable pap
  set ifaddr 203.14.100.1 203.14.100.20-203.14.100.40
  enable proxy
```

Для каждого пользователя, входящего по этому методу, в файле /etc/ppp/ppp.secret должна присутствовать запись с логином/паролем, или, в качестве альтернативы, для аутентификации пользователей по PAP через /etc/passwd, необходимо использовать следующий параметр.

```
enable passwdauth
```

Если вы хотите присвоить некоторым пользователям статический IP, задайте его в качестве третьего аргумента в /etc/ppp/ppp.secret. Обратитесь к к /usr/share/examples/ppp/ppp.secret.sample за примерами.

### 21.2.1.3.11. MS расширения

Возможна настройка PPP для выдачи адресов DNS и NetBIOS по запросу.

Для включения этих расширений с PPP версии 1.x, необходимо добавить к соответствующему разделу /etc/ppp/ppp.conf следующие строки.

```
enable msext
set ns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

Для PPP версии 2 и выше:

```
accept dns
set dns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

Клиентам будут выдаваться адреса первичного и вторичного серверов имен, и адрес хоста NetBIOS.

Если в версии 2 и выше строка `set dns` будет опущена, PPP использует значения из `/etc/resolv.conf`.

#### 21.2.1.3.12. Аутентификация PAP и CHAP

Некоторые провайдеры настраивают систему так, что аутентификация производится с использованием либо PAP либо CHAP. В этом случае, приглашение `login:` при соединении не выдается и соединение PPP начинается сразу.

PAP менее безопасен, чем CHAP, но безопасность в данном случае не страдает, поскольку хотя пароли и передаются открытым текстом, они передаются только по модемной линии. У кракеров не так много возможностей для кражи паролей.

В примерах выше PPP и статические IP адреса или PPP и динамические IP адреса должны быть сделаны следующие изменения:

```
13      set authname MyUserName
14      set authkey MyPassword
15      set login
```

Строка 13:

Эта строка указывает имя пользователя PAP/CHAP. Вам потребуется заменить *MyUserName* на правильное значение.

Строка 14:

Эта строка указывает пароль PAP/CHAP. Вам потребуется заменить *MyPassword* на правильное значение. Вы можете также добавить дополнительную строку, такую как:

```
16      accept PAP
```

или

```
16      accept CHAP
```

для явного указания протокола, но и PAP и CHAP поддерживаются по умолчанию.

Строка 15:

Ваш провайдер обычно не потребует входа на сервер при использовании PAP или CHAP. Следовательно, вы должны отключить строку `"set login"`.

### 21.2.1.3.13. Изменение настроек `ppp` "на лету"

Возможно изменение настроек `ppp` программы во время ее работы в фоновом режиме, но только если открыт соответствующий диагностический порт. Для его открытия добавьте в настройку следующую строку:

```
set server /var/run/ppp-tun%d DiagnosticPassword 0177
```

С этой директивой `ppp` будет прослушивать заданный UNIX сокет, запрашивая у клиентов пароль перед выдачей доступа. Символы `%d` заменяются на номер используемого устройства `tun`.

Как только сокет открыт, в скриптах, выполняющих настройку работающей программы, можно использовать `pppctl(8)`.

### 21.2.1.4. Использование NAT

PPP имеет возможность использовать встроенный NAT без преобразования пакетов в ядре. Эта возможность может быть включена следующей строкой в `/etc/ppp/ppp.conf`:

```
nat enable yes
```

Альтернативно, PPP NAT может быть включен параметром командной строки `-nat`. Существует также переменная `/etc/rc.conf ppp_nat`, которая включена по умолчанию.

Если вы используете эту возможность, вы также можете найти полезными параметры `/etc/ppp/ppp.conf` для включения пересылки входящих соединений:

```
nat port tcp 10.0.0.2:ftp ftp
nat port tcp 10.0.0.2:http http
```

или отключить все входящие соединения

```
nat deny_incoming yes
```

### 21.2.1.5. Завершающая настройка системы

Теперь, когда программа `ppp` настроена, осталось выполнить еще несколько действий прежде, чем все это заработает. Они выполняются путем редактирования файла `/etc/rc.conf`.

Просматривая этот файл, убедитесь, что добавлена строка `hostname=`, например:

```
hostname="foo.example.com"
```

Если провайдер предоставил вам статический IP адрес и имя, возможно лучше всего использовать это имя в качестве имени хоста.

Проверьте переменную `network_interfaces`. Если вы хотите настроить систему для дозвона по требованию, убедитесь, что устройство `tun0` добавлено в список, иначе удалите его.

```
network_interfaces="lo0 tun0"
ifconfig_tun0=
```

**Замечание:** Переменная `ifconfig_tun0` должна быть пуста, необходимо также создать файл `/etc/start_if.tun0`. В этом файле должна находиться строка:

```
ppp -auto mysystem
```

Этот скрипт выполняется во время настройки сети, запуская демона `ppp` в автоматическом режиме. Если эта машина является шлюзом для локальной сети, вы можете также использовать переключатель `-alias`. Обратитесь к странице справочника за дальнейшей информацией.

Убедитесь, что программа маршрутизации отключена переменной в файле `/etc/rc.conf`:

```
router_enable="NO"
```

Важно, чтобы демон `routed` не был запущен, поскольку он может удалить запись маршрута по умолчанию, создаваемую `ppp`.

Возможно, стоит обратить внимание на переменную `sendmail_flags` и убедиться, что она не включает параметр `-q`, иначе `sendmail` попытается сразу же обратиться к сети, и компьютер может начать дозвон. Вы можете попробовать:

```
sendmail_flags="-bd"
```

Обратная сторона этого решения в том, что необходим запуск очереди `sendmail` после поднятия соединения `ppp`:

```
# /usr/sbin/sendmail -q
```

Вы можете использовать команду `!bg` в файле `ppp.linkup` для автоматического выполнения этой задачи:

```
1 provider:
2 delete ALL
3 add 0 0 HISADDR
4 !bg sendmail -bd -q30m
```

Если вам это не нужно, возможна настройка "dfilter" для блокирования SMTP трафика. Обратитесь к файлам примеров за дальнейшей информацией.

Все, что осталось, это перезагрузить компьютер. После перезагрузки вы можете либо выполнить:

```
# ppp
```

и затем набрать `dial provider` для запуска сессии PPP, либо, если вы хотите, чтобы программа `ppp` начинала соединение автоматически при появлении исходящего трафика (и файл `start_if.tun0` не создан), выполните:

```
# ppp -auto provider
```

### 21.2.1.6. Итоги

Для первоначальной настройки `ppp` необходимо пройти следующие шаги:



Сторона клиента:

1. Убедитесь, что устройство `tun` встроено в ядро.
2. Убедитесь, что устройства `tunN` находятся в каталоге `/dev`.
3. Создайте запись в `/etc/ppp/ppp.conf`. Пример `pppdemand` должен подойти для большинства провайдеров.
4. Если у вас динамический IP адрес, создайте запись в `/etc/ppp/ppp.linkup`.
5. Обновите файл `/etc/rc.conf`.
6. Создайте скрипт `start_if.tun0`, если необходим дозвон по требованию.

Сторона сервера:

1. Убедитесь, что устройство `tun` встроено в ядро.
2. Убедитесь, что устройства `tunN` находятся в каталоге `/dev`.
3. Создайте запись в `/etc/passwd` (используя программу `vipw(8)`).
4. Создайте профиль в домашнем каталоге пользователя, запускающий `ppp -direct direct-server` или подобную команду.
5. Создайте запись в `/etc/ppp/ppp.conf`. Пример `direct-server` должен подойти.
6. Создайте запись в `/etc/ppp/ppp.linkup`.
7. Обновите файл `/etc/rc.conf`.

## 21.3. PPP уровня ядра

*Части документа первоначально предоставили Gennady B. Sorokopud, Robert Huff.*

### 21.3.1. Настройка PPP уровня ядра

Перед началом настройки PPP на вашем компьютере, убедитесь, что `pppd` находится в `/usr/sbin` и каталог `/etc/ppp` существует.

`pppd` может работать в двух режимах:

1. В качестве "клиента" — когда вы хотите подключить компьютер к внешнему миру через последовательное соединение PPP или модемную линию.
2. В качестве "сервера" — когда компьютер подключен к сети и используется для подключения других компьютеров через PPP.

В обоих случаях вам потребуется настроить файл параметров (`/etc/ppp/options` или `~/ppprc` если на вашем компьютере более одного пользователя работают с PPP).

Вам потребуется также программа для модемных/последовательных линий (предпочтительно `comms/kermit`) для дозвона и установки соединения с удаленным хостом.

### 21.3.2. Использование pppd в качестве клиента

*Основано на информации, предоставленной Trev Roydhouse.*

Для подключения к линии PPP терминального сервера Cisco может использоваться следующий файл /etc/ppp/options.

```
crtstcts # enable hardware flow control
modem    # modem control line
noipdefault # remote PPP server must supply your IP address
           # if the remote host does not send your IP during IPCP
           # negotiation, remove this option
passive   # wait for LCP packets
domain ppp.foo.com # put your domain name here

:<remote_ip> # put the IP of remote PPP host here
            # it will be used to route packets via PPP link
            # if you didn't specified the noipdefault option
            # change this line to <local_ip>:<remote_ip>

defaultroute # put this if you want that PPP server will be your
             # default router
```

Для подключения:

1. Дозвонитесь на удаленный хост, используя **kermit** (или любую другую модемную программу), и введите ваше имя пользователя и пароль (или те данные, которые требуются для установления PPP соединения с удаленным хостом).
2. Выйдите из **kermit** (без обрыва соединения).
3. Введите следующее:

```
# /usr/src/usr.sbin/pppd.new/pppd /dev/tty01 19200
```

Убедитесь, что выбраны подходящая скорость и правильное имя устройства.

Теперь ваш компьютер подключен по PPP. Если соединение не состоялось, вы можете добавить параметр `debug` к файлу /etc/ppp/options, и отследить проблему по сообщениям на консоли.

Следующий скрипт /etc/ppp/pprup проведет все 3 стадии в автоматическом режиме:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing pppd, PID=' ${pid}
    kill ${pid}
fi
ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing kermit, PID=' ${pid}
    kill -9 ${pid}
fi
```

```
ifconfig ppp0 down
ifconfig ppp0 delete
```

```
kermit -y /etc/ppp/kermit.dial
pppd /dev/tty01 19200
```

/etc/ppp/kermit.dial это скрипт **kermit**, который дозванивается до удаленного хоста и проходит необходимую авторизацию (пример такого скрипта находится в конце этого раздела).

Используйте следующий скрипт /etc/ppp/pppdown для отключения от PPP линии:

```
#!/bin/sh
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ X${pid} != "X" ] ; then
    echo 'killing pppd, PID=' ${pid}
    kill -TERM ${pid}
fi

ps ax |grep kermit |grep -v grep
pid=`ps ax |grep kermit |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing kermit, PID=' ${pid}
    kill -9 ${pid}
fi

/sbin/ifconfig ppp0 down
/sbin/ifconfig ppp0 delete
kermit -y /etc/ppp/kermit.hup
/etc/ppp/ppptest
```

Проверьте, запущен ли еще pppd, выполнив /usr/etc/ppp/ppptest, который выглядит примерно так:

```
#!/bin/sh
pid=`ps ax| grep pppd |grep -v grep|awk '{print $1;}'`
if [ X${pid} != "X" ] ; then
    echo 'pppd running: PID=' ${pid-NONE}
else
    echo 'No pppd running.'
fi
set -x
netstat -n -I ppp0
ifconfig ppp0
```

Для обрыва соединения, выполните /etc/ppp/kermit.hup, который должен содержать:

```
set line /dev/tty01 ; put your modem device here
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
```

```

set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATH0\13
echo \13
exit

```

Существует альтернативный метод, использующий chat вместо kermi:

Для установления соединения `pppd` достаточно двух файлов.

`/etc/ppp/options:`

`/dev/cuaa1 115200`

```

crtscts # enable hardware flow control
modem # modem control line
connect "/usr/bin/chat -f /etc/ppp/login.chat.script"
noipdefault # remote PPP server must supply your IP address
               # if the remote host doesn't send your IP during
               # IPCP negotiation, remove this option
passive      # wait for LCP packets
domain <your.domain> # put your domain name here

: # put the IP of remote PPP host here
   # it will be used to route packets via PPP link
   # if you didn't specified the noipdefault option
   # change this line to <local_ip>:<remote_ip>

defaultroute # put this if you want that PPP server will be
               # your default router

```

`/etc/ppp/login.chat.script:`

**Замечание:** Все это может быть расположено на одной строке.

```

ABORT BUSY ABORT 'NO CARRIER' "" AT OK ATDT<phone.number>
CONNECT "" TIMEOUT 10 ogin:-\\r-ogin: <login-id>
TIMEOUT 5 sword: <password>

```

Как только эти файлы будут созданы и отредактированы, необходимо только запустить `pppd`, вот так:

```
# pppd
```

### 21.3.3. Использование pppd в качестве сервера

/etc/ppp/options должен содержать примерно следующее:

```
crtscts                # Hardware flow control
netmask 255.255.255.0  # netmask (not required)
192.114.208.20:192.114.208.165 # IP's of local and remote hosts
                        # local ip must be different from one
                        # you assigned to the Ethernet (or other)
                        # interface on your machine.
                        # remote IP is IP address that will be
                        # assigned to the remote machine
domain ppp.foo.com     # your domain
passive                # wait for LCP
modem                  # modem line
```

Следующий скрипт /etc/ppp/pppserv укажет **pppd** работать в качестве сервера:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing pppd, PID=' ${pid}
    kill ${pid}
fi
ps ax |grep kermi |grep -v grep
pid=`ps ax |grep kermi |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing kermi, PID=' ${pid}
    kill -9 ${pid}
fi

# reset ppp interface
ifconfig ppp0 down
ifconfig ppp0 delete

# enable autoanswer mode
kermi -y /etc/ppp/kermi.ans

# run ppp
pppd /dev/tty01 19200
```

Используйте этот скрипт /etc/ppp/pppservdown для остановки сервера:

```
#!/bin/sh
ps ax |grep pppd |grep -v grep
pid=`ps ax |grep pppd |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
    echo 'killing pppd, PID=' ${pid}
    kill ${pid}
fi
ps ax |grep kermi |grep -v grep
pid=`ps ax |grep kermi |grep -v grep|awk '{print $1;}'`
if [ "X${pid}" != "X" ] ; then
```

```

        echo 'killing kermid, PID=' ${pid}
        kill -9 ${pid}
    fi
    ifconfig ppp0 down
    ifconfig ppp0 delete

    kermid -y /etc/ppp/kermid.noans

```

Следующий **kermid** скрипт (/etc/ppp/kermid.ans) включит/отключит режим ответа модема на входящие звонки. Он должен выглядеть примерно так:

```

set line /dev/tty01
set speed 19200
set file type binary
set file names literal
set win 8
set rec pack 1024
set send pack 1024
set block 3
set term bytesize 8
set command bytesize 8
set flow none

pau 1
out +++
inp 5 OK
out ATN0\13
inp 5 OK
echo \13
out ATS0=1\13    ; change this to out ATS0=0\13 if you want to disable
                  ; autoanswer mode

inp 5 OK
echo \13
exit

```

Скрипт, называющийся /etc/ppp/kermid.dial, используется для дозвона и аутентификации на удаленном хосте. Вам потребуется приспособить его под собственные нужды. Поместите ваш логин и пароль в этот скрипт; вам также потребуется изменить операторы input в зависимости от ответов от модема и удаленного хоста.

```

;
; put the com line attached to the modem here:
;
set line /dev/tty01
;
; put the modem speed here:
;
set speed 19200
set file type binary          ; full 8 bit file xfer
set file names literal
set win 8
set rec pack 1024
set send pack 1024

```

```

set block 3
set term bytesize 8
set command bytesize 8
set flow none
set modem hayes
set dial hangup off
set carrier auto           ; Then SET CARRIER if necessary,
set dial display on        ; Then SET DIAL if necessary,
set input echo on
set input timeout proceed
set input case ignore
def \%x 0                   ; login prompt counter
goto slhup

:slcmd                      ; put the modem in command mode
echo Put the modem in command mode.
clear                      ; Clear unread characters from input buffer
pause 1
output +++                 ; hayes escape sequence
input 1 OK\13\10           ; wait for OK
if success goto slhup
output \13
pause 1
output at\13
input 1 OK\13\10
if fail goto slcmd         ; if modem doesn't answer OK, try again

:slhup                      ; hang up the phone
clear                      ; Clear unread characters from input buffer
pause 1
echo Hanging up the phone.
output ath0\13             ; hayes command for on hook
input 2 OK\13\10
if fail goto slcmd         ; if no OK answer, put modem in command mode

:sldial                     ; dial the number
pause 1
echo Dialing.
output atdt9,550311\13\10   ; put phone number here
assign \%x 0               ; zero the time counter

:look
clear                      ; Clear unread characters from input buffer
increment \%x              ; Count the seconds
input 1 {CONNECT }
if success goto sllogin
reinput 1 {NO CARRIER\13\10}
if success goto sldial
reinput 1 {NO DIALTONE\13\10}
if success goto slnodial
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}

```

```

if success goto slhup
if < \%x 60 goto look
else goto slhup

:sllogin                                ; login
assign \%x 0                            ; zero the time counter
pause 1
echo Looking for login prompt.

:slloop
increment \%x                            ; Count the seconds
clear                                    ; Clear unread characters from input buffer
output \13
;
; put your expected login prompt here:
;
input 1 {Username: }
if success goto sluid
reinput 1 {\255}
if success goto slhup
reinput 1 {\127}
if success goto slhup
if < \%x 10 goto slloop                 ; try 10 times to get a login prompt
else goto slhup                         ; hang up and start again if 10 failures

:sluid
;
; put your userid here:
;
output ppp-login\13
input 1 {Password: }
;
; put your password here:
;
output ppp-password\13
input 1 {Entering SLIP mode.}
echo
quit

:slnodial
echo \7No dialtone. Check the telephone line!\7
exit 1

; local variables:
; mode: csh
; comment-start: ";" "
; comment-start-skip: ";" "
; end:

```



## 21.4. Решение проблем с соединениями PPP

*Предоставил Tom Rhodes.*

Этот раздел охватывает несколько вопросов, которые могут возникнуть при использовании PPP через модемные соединения. Например, предположим, что вам потребовалось узнать, какое именно приглашение отображает система, до которой вы звонитесь. Некоторые провайдеры выдают приглашение `ssword`, другие `password`; если `ppp` скрипт не обрабатывает такие приглашения, попытка входа завершится неудачно. Наиболее общий способ отладки соединений `ppp` это подключение вручную. Ниже дана пошаговая информация по подключению вручную.

### 21.4.1. Проверьте файлы устройств

Если вы пересобирали ядро, проверьте устройство `sio`. Если вы не перенастраивали ядро, нет причин для беспокойства. Просто проверьте вывод `dmesg` для модемного устройства следующей командой:

```
#dmesg | grep sio
```

Вы должны получить информацию о устройствах `sio`. Это COM порты, которые нам необходимы. Если ваш модем работает как стандартный последовательный порт, вы увидите его на `sio1`, или `COM2`. Если это так, вам не требуется пересобирать ядро, необходимо лишь создать последовательное устройство. Вы можете сделать это, зайдя в `/dev` и запустив скрипт `MAKEDEV`. Создайте последовательные устройства:

```
# sh MAKEDEV cuaa0 cuaa1 cuaa2 cuaa3
```

Если модем находится на `sio1`, или `COM2` в DOS, модемным устройством будет `/dev/cuaa1`.

### 21.4.2. Подключение вручную

Подключение к интернет с контролем `ppp` вручную это быстрый, легкий и действенный способ отладки соединения или получения информации о обслуживании провайдером клиентского соединения `ppp`. Запустите **PPP** из командной строки. В примерах в качестве имени хоста мы будем использовать *example*. Запустите `ppp`, введя команду `ppp`:

```
# ppp
```

Теперь программа `ppp` запущена.

```
ppp ON example> set device /dev/cuaa1
```

Мы указали модемное устройство, в данном случае `cuaa1`.

```
ppp ON example> set speed 115200
```

Мы установили скорость подключения к модему, в данном случае 115,200 кбит/с.

```
ppp ON example> enable dns
```

Сообщает `ppp` настроить разрешение имен, добавив строки в `/etc/resolv.conf`. Если `ppp` не может определить имя хоста, его можно настроить позже вручную.

```
ppp ON example> term
```

Переключение в "терминальный" режим для контроля модема вручную.

```
deflink: Entering terminal mode on /dev/cuaa1
type '~h' for help
```

```
at
```

```
OK
```

```
atdt123456789
```

Использование команды `at` для инициализации модема, а затем `atdt` и номера провайдера для начала дозвола.

```
CONNECT
```

Подтверждение соединения, если есть проблемы с соединением, не вызванные проблемами в оборудовании, здесь мы попытаемся решить их.

```
ISP Login:myusername
```

Здесь вам предлагается ввести имя пользователя в приглашение, выводимое сервером провайдера.

```
ISP Pass:mypassword
```

Здесь предлагается ввести пароль, предоставленный провайдером. Как и при входе в FreeBSD, пароль не отображается.

```
Shell or PPP:ppp
```

В зависимости от вашего провайдера, это приглашение может не появиться. Здесь задается вопрос, хотите ли вы использовать оболочку на компьютере провайдера или запустить `ppp`. В этом примере мы выбрали `ppp`, поскольку хотим соединиться с интернет.

```
Ppp ON example>
```

Обратите внимание, что в этом примере первая буква `p` появилась в верхнем регистре. Это означает, что мы успешно подключились к провайдеру.

```
PPP ON example>
```

Мы успешно аутентифицировались у провайдера и ожидаем присвоения IP адреса.

```
PPP ON example>
```

Мы завершили согласование IP адресов, соединение успешно установлено.

```
PPP ON example>add default HISADDR
```

Это маршрут по умолчанию, его необходимо настроить, чтобы сделать возможной связь с внешним миром, поскольку на этот момент установлена связь только с удаленной стороной. Если маршрут не устанавливается из-за уже существующего маршрута, добавьте символ `!` перед `add`. Альтернативно вы можете сделать настройку перед установкой соединения.

Если все прошло удачно, на данный момент должно работать соединение с интернет, которое можно поместить в фоновый режим клавишами **CTRL+z**. Переход строки PPP в приглашении в нижний регистр (ppp) означает, что соединение было разорвано. Таким образом, символы P в верхнем регистре означают наличие соединения, а в нижнем — потерю соединения.

#### 21.4.2.1. Отладка

Если у вас выделенная линия и нет необходимости устанавливать соединение, выключите контроль передачи данных CTS/RTS командой `set ctsrts off`. Это обычно необходимо при подключении к поддерживающим PPP терминальным серверам, когда PPP прерывается при попытке записать данные в ваше соединение, и ожидает сигнала CTS, или Clear To Send, который не появляется. Если вы используете этот параметр, используйте также параметр `set assap`, который может быть необходим для устранения зависимости от оборудования путем пропуска определенных символов через соединение, обычно XON/XOFF. Обратитесь к странице справочника ppp(8) за более подробной информацией по этому параметру и его использованию.

Если у вас старый модем, может потребоваться использование `set parity even`. Проверка четности по умолчанию отключена, но она используется для устранения ошибок (с серьезным увеличением объема трафика) на старых модемах и у некоторых провайдеров.

PPP может не вернуться в командный режим, что обычно вызвано ошибкой согласования, когда провайдер ждет от вас начала процесса согласования. В этот момент использование команды `~r` заставит ppp начать отправку настроечной информации.

Если вы не получаете приглашения на вход, скорее всего вам требуется использование аутентификации PAP или CHAP вместо UNIX стиля как в примерах выше. Для использования PAP или CHAP просто добавьте следующие параметры PPP перед переходом в терминальный режим:

```
ppp ON example> set authname myusername
```

Необходимо заменить *myusername* на имя пользователя, выданное провайдером.

```
ppp ON example> set authkey mypassword
```

Где *mypassword* должен быть заменен на пароль, выданный провайдером.

Если вы подключаетесь, не можете определить ни одно доменное имя, попробуйте использовать ping(8) с каким-либо IP адресом и просмотрите выводимую информацию. Если 100 процентов (100%) пакетов теряются, скорее всего не назначен маршрут по умолчанию. Дважды проверьте, что параметр `add default HISADDR` установлен во время соединения. Если вы можете подключиться к удаленному IP адресу, возможно, что адрес сервера имен не был добавлен в `/etc/resolv.conf`. Этот файл должен выглядеть примерно так:

```
domain example.com
nameserver x.x.x.x
nameserver y.y.y.y
```

Где *x.x.x.x* и *y.y.y.y* должны быть заменены на IP адреса DNS серверов провайдера. Эта информация может предоставляться провайдером, вы можете выяснить это в службе поддержки.

Вы можете также настроить syslog(3) для протоколирования PPP соединения. Просто добавьте:

```
!ppp
```

```
*.* /var/log/ppp.log
```

в файл `/etc/syslog.conf`. В большинстве случаев эти строки уже присутствуют.

## 21.5. Использование PPP через Ethernet (PPPoE)

*Предоставил (с <http://node.to/freebsd/how-tos/how-to-freebsd-pppoe.html>) Jim Mock.*

Этот раздел описывает настройку PPP через Ethernet (PPP over Ethernet, PPPoE).

### 21.5.1. Настройка ядра

Для PPPoE более не требуется настройка ядра. Если необходимая поддержка netgraph не встроена в ядро, она будет динамически загружена **ppp**.

### 21.5.2. Настройка `ppp.conf`

Вот пример работающего `ppp.conf`:

```
default:
    set log Phase tun command # you can add more detailed logging if you wish
    set ifaddr 10.0.0.1/0 10.0.0.2/0

name_of_service_provider:
    set device PPPoE:x11 # replace x11 with your Ethernet device
    set authname YOURLOGINNAME
    set authkey YOURPASSWORD
    set dial
    set login
    add default HISADDR
```

### 21.5.3. Запуск `ppp`

Под пользователем `root` вы можете запустить:

```
# ppp -ddial name_of_service_provider
```

### 21.5.4. Запуск `ppp` при загрузке

Добавьте к `/etc/rc.conf` следующее:

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_nat="YES" # if you want to enable nat for your local network, otherwise NO
ppp_profile="name_of_service_provider"
```

### 21.5.5. Использование тега сервиса PPPoE

Иногда для установки соединения необходимо использовать тег сервиса. Теги сервисов используются для различения PPPoE серверов, подключенных к одной сети.

В документации, предоставляемой провайдером, должна находиться необходимая информация о теге сервиса. Если вы не можете ее обнаружить, обратитесь в службу поддержки.

В крайнем случае, вы можете попробовать метод, предложенный в программе Roaring Penguin PPPoE (<http://www.roaringpenguin.com/pppoe/>), которая находится в коллекции портов. Учтите, что этот метод может сделать ваш модем неработоспособным, так что дважды подумайте перед тем, как воспользоваться им. Просто установите программу, поставляемую с модемом вашим провайдером. Затем, войдите в меню **System** программы. Имя вашего профиля должно находиться здесь. Обычно это *ISP*.

Имя профиля (тег сервиса) будет использоваться в записи настройки PPPoE в файле `ppp.conf` (часть команды `set device`, относящаяся к провайдеру). Обратитесь к странице справочника `ppp(8)` за более подробной информацией. Эта команда должна выглядеть примерно так:

```
set device PPPoE:x11:ISP
```

Не забудьте изменить `x11` на имя устройства вашей Ethernet карты.

Не забудьте изменить `ISP` на профиль, определенный вами ранее.

Дополнительная информация:

- Cheaper Broadband with FreeBSD on DSL (<http://renaud.waldura.com/doc/freebsd/pppoe/>), опубликовал Renaud Waldura.
- Nutzung von T-DSL und T-Online mit FreeBSD (<http://www.ruhr.de/home/nathan/FreeBSD/tdsl-freebsd.html>) by Udo Erdelhoff (на немецком).

### 21.5.6. PPPoE с 3Com® HomeConnect® ADSL Modem Dual Link

Этот модем не поддерживает RFC 2516 (<http://www.faqs.org/rfcs/rfc2516.html>) (*Метод соединения PPP через Ethernet (PPPoE)*), написанный L. Mamakos, K. Lidl, J. Evarts, D. Carrel, D. Simone, и R. Wheeler). Вместо этого, для фреймов Ethernet используются различные коды типов пакетов. Сообщите 3Com (<http://www.3com.com/>) если считаете, что они должны соблюдать спецификации PPPoE.

Для включения поддержки этого нестандартного устройства, в FreeBSD необходимо установить переменную `sysctl`. Это может быть сделано автоматически, поместите в `/etc/sysctl.conf` следующую переменную:

```
net.graph.nonstandard_pppoe=1
```

или, для непосредственного включения, выполните команду `sysctl`

```
net.graph.nonstandard_pppoe=1.
```

К сожалению, поскольку эта настройка влияет на всю систему, невозможно одновременно взаимодействовать с нормальным PPPoE клиентом или сервером и 3Com HomeConnect® ADSL Modem.

## 21.6. Использование PPP через ATM (PPPoA)

Далее описано как настроить PPP через ATM (PPP over ATM, PPPoA). PPPoA популярен у европейских провайдеров DSL.

### 21.6.1. Использование PPPoA с Alcatel SpeedTouch™ USB

Поддержка PPPoA для этого устройства в FreeBSD поставляется в виде порта, поскольку встроенное программное обеспечение поставляется под лицензионным соглашением Alcatel ([http://www.speedtouchdsl.com/disclaimer\\_lx.htm](http://www.speedtouchdsl.com/disclaimer_lx.htm)) и не может свободно распространяться с основной системой FreeBSD.

Для установки этой программы, просто используйте коллекцию портов. Установите порт `net/pppoea` и следуйте инструкциям.

Как и многие устройства USB, Alcatel SpeedTouch™ USB должен загрузить встроенное программное обеспечение с компьютера. Возможна автоматизация этого процесса в FreeBSD, чтобы эта передача происходила при подключении устройства к порту USB. Для включения этой автоматической передачи необходимо добавить в `/etc/usb.d.conf` нижеприведенную информацию. Этот файл необходимо отредактировать под пользователем `root`.

```
device "Alcatel SpeedTouch USB"
    devname "ugen[0-9] +"
    vendor 0x06b9
    product 0x4061
    attach "/usr/local/sbin/modem_run -f /usr/local/libdata/mgmt.o"
```

Для включения демона USB, **usbd**, поместите в `/etc/rc.conf` следующую строку:

```
usbd_enable="YES"
```

Возможна также настройка **ppp** для дозвола при запуске. Для этого добавьте в `/etc/rc.conf` нижеприведенные строки. Опять же, для этого вам необходимо войти под пользователем `root`.

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_profile="adsl"
```

Необходимо также использовать пример файла `ppp.conf`, поставляемый с портом `net/pppoea`.

### 21.6.2. Использование mpd

Вы можете использовать **mpd** для подключения к различным сервисам, в частности к сервисам РРТР. Вы можете найти **mpd** в коллекции портов, `net/mpd`. Многие ADSL модемы требуют, чтобы РРТР туннель был создан между модемом и компьютером, один из таких модемов это Alcatel SpeedTouch Home.

Сначала установите порт, затем настройте **mpd** под собственные нужды и настройки провайдера. Порт помещает набор примеров настройки в каталог `PREFIX/etc/mpd/`. *PREFIX* означает каталог, в который устанавливаются порты, по умолчанию это `/usr/local/`. Полное руководство по настройке **mpd** доступно в HTML формате после установки порта. Оно находится в `PREFIX/share/doc/mpd/`.

Ниже находится пример настройки **mpd** для соединения с ADSL сервисом. Настройка разделена на два файла, первый это `mpd.conf`:

```
default:
    load adsl

adsl:
    new -i ng0 adsl adsl
    set bundle authname username ❶
    set bundle password password ❷
    set bundle disable multilink

    set link no pap acfcomp protocomp
    set link disable chap
    set link accept chap
    set link keep-alive 30 10

    set ipcp no vjcomp
    set ipcp ranges 0.0.0.0/0 0.0.0.0/0

    set iface route default
    set iface disable on-demand
    set iface enable proxy-arp
    set iface idle 0

    open
```

❶ Имя пользователя, используемое для аутентификации у провайдера.

❷ Пароль, используемый для аутентификации у провайдера.

Файл `mpd.links` содержит информацию о соединении или соединениях, которые вы планируете установить. Пример `mpd.links`, соответствующий приведенному выше примеру:

```
adsl:
    set link type pptp
    set pptp mode active
    set pptp enable originate outcall
    set pptp self 10.0.0.1 ❶
    set pptp peer 10.0.0.138 ❷
```

❶ IP адрес компьютера FreeBSD, с которого вы будете использовать **mpd**.

❷ IP адрес модема ADSL. Для Alcatel SpeedTouch Home этот адрес по умолчанию 10.0.0.138.

Инициализация соединения возможно простым выполнением следующей команды под `root`:

```
# mpd -b adsl
```

Вы можете просмотреть статус соединения с помощью следующей команды:

```
% ifconfig ng0
ng0: flags=88d1<UP,POINTOPOINT,RUNNING,NOARP,SIMPLEX,MULTICAST> mtu 1500
    inet 216.136.204.117 --> 204.152.186.171 netmask 0xffffffff
```

Использование **mpd** это рекомендуемый способ подключения к сервису ADSL из FreeBSD.

### 21.6.3. Использование **pptpclient**

Возможно также использование FreeBSD для подключения к сервисам PPPoA с помощью `net/pptpclient`.

Для подключения к сервису DSL с использованием `net/pptpclient`, установите порт или пакет и отредактируйте `/etc/ppp/ppp.conf`. Вам потребуется работать под `root`. Пример настройки `ppp.conf` дан ниже. За дальнейшей информацией по параметрам `ppp.conf` обратитесь к странице справочника **ppp**, `ppp(8)`.

```
adsl:
set log phase chat lcp ipcp ccp tun command
set timeout 0
enable dns
set authname username ❶
set authkey password ❷
set ifaddr 0 0
add default HISADDR
```

❶ Имя пользователя вашей учетной записи у провайдера DSL.

❷ Пароль для вашей учетной записи.

**Внимание:** Поскольку вам необходимо поместить пароль в незашифрованном виде в файл `ppp.conf`, убедитесь что никто другой не сможет прочесть содержимое этого файла. Следующая последовательность команд сделает этот файл доступным для чтения только пользователю `root`. Обратитесь к страницам справочника `chmod(1)` и `chown(8)` за дальнейшей информацией.

```
# chown root:wheel /etc/ppp/ppp.conf
# chmod 600 /etc/ppp/ppp.conf
```

Следующая команда откроет туннель для PPP сессии к вашему DSL маршрутизатору. Модемы Ethernet DSL поставляются с настроенным IP адресом локальной сети, к которому вы подключаетесь. У Alcatel SpeedTouch Home этот адрес `10.0.0.138`. В документации на ваш маршрутизатор должно быть указано, какой адрес используется. Для открытия туннеля и начала PPP сессии выполните:

```
# pptp address adsl
```

**Подсказка:** Чтобы вернуться в приглашение командной строки после выполнения этой команды, добавьте символ `"&"` в конец строки.

Устройство виртуального туннеля `tun` будет создано для взаимодействия между процессами **pptp** и **ppp**. Как только вы вернетесь в приглашение, или процесс **pptp** выполнит соединение, вы можете проверить туннель примерно такой командой:



```
% ifconfig tun0
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1500
    inet 216.136.204.21 --> 204.152.186.171 netmask 0xffffffff00
    Opened by PID 918
```

Если вы не сможете соединиться, проверьте настройку маршрутизатора, которая обычно доступна через **telnet** или через веб браузер. Если вы все еще не можете подключиться, проверьте вывод команды **pptr** и содержимое лог файла **ppp**, `/var/log/ppp.log`.

## 21.7. Использование SLIP

*Первоначально предоставил Satoshi Asami. Дополнительный материал предоставили Guy Helmer, Piero Serini.*

### 21.7.1. Настройка SLIP клиента

Ниже дан один из способов настройки FreeBSD для подключения к SLIP сети со статическим адресом. Для динамического подключения (адрес изменяется при каждом дозвоне) возможно потребуется более сложная настройка.

Сначала определите, к какому последовательному порту подключен модем. Многие создают символическую ссылку, такую как `/dev/modem`, на настоящий файл устройства, `/dev/cuaaN`. Это позволяет абстрагироваться от имени файла устройства, например если вы переносите модем на другой порт. Довольно сложно править множество файлов в `/etc` и `.kernrc` во всей системе!

**Замечание:** `/dev/cuaa0` это COM1, `cuaa1` это COM2, и т.д.

Убедитесь, что в вашем файле настройки ядра присутствует строка:

```
pseudo-device    sl      1
```

В FreeBSD 5.X, используйте вместо этой строки следующую:

```
device    sl
```

Эта строка включена в ядро GENERIC, так что если вы ее не удаляли, проблем быть не должно.

#### 21.7.1.1. То, что необходимо сделать только один раз

1. Добавьте ваш компьютер, шлюз и сервера имен в файл `/etc/hosts`. Вот пример такого файла:

```
127.0.0.1      localhost localhost
136.152.64.181 water.CS.Example.EDU water.CS water
136.152.64.1   inr-3.CS.Example.EDU inr-3  slip-gateway
128.32.136.9   ns1.Example.EDU ns1
128.32.136.12  ns2.Example.EDU ns2
```

2. Убедитесь, что в файле `/etc/host.conf` `hosts` находится перед `bind` (для FreeBSD версий до 5.0). Начиная с FreeBSD 5.0, система использует файл `/etc/nsswitch.conf`, убедитесь, что параметр `files` находится перед `dns` в строке `hosts` этого файла. Без этого параметра могут происходить странные вещи.
3. Отредактируйте файл `/etc/rc.conf`.

1. Установите имя хоста, настроив переменную `hostname`:

```
hostname="myname.my.domain"
```

Здесь необходимо использовать полное доменное имя вашего компьютера в интернет.

2. Добавьте `sl0` к списку сетевых интерфейсов, изменив переменную:

```
network_interfaces="lo0"
```

на:

```
network_interfaces="lo0 sl0"
```

3. Измените параметры `sl0`, добавив строку:

```
ifconfig_sl0="inet ${hostname} slip-gateway netmask 0xffffffff00 up"
```

4. Назначьте маршрутизатор по умолчанию, изменив строку:

```
defaultrouter="NO"
```

на:

```
defaultrouter="slip-gateway"
```

4. Создайте файл `/etc/resolv.conf`, содержащий:

```
domain CS.Example.EDU
nameserver 128.32.136.9
nameserver 128.32.136.12
```

Как вы видите, здесь указаны адреса серверов имен. Конечно, реальные имена доменов и адреса для вас будут другими.

5. Перегрузите компьютер и убедитесь, что его имя хоста настроено правильно.

### 21.7.1.2. Создание SLIP соединения

1. Дозвонитесь на удаленный сервер, введите `slip` в приглашение, имя своего компьютера и пароль. Все, что требуется ввести в вашем случае. Если вы используете `kermi`t, попробуйте такой скрипт:

```
# kermi t setup
set modem hayes
set line /dev/modem
set speed 115200
set parity none
```

```

set flow rts/cts
set terminal bytesize 8
set file type binary
# The next macro will dial up and login
define slip dial 643-9600, input 10 =>, if failure stop, -
output slip\x0d, input 10 Username:, if failure stop, -
output silvia\x0d, input 10 Password:, if failure stop, -
output ***\x0d, echo \x0aCONNECTED\x0a

```

Конечно, вам потребуется заменить имя хоста и пароль на ваши собственные. После этого, для подключения просто введите `slip` из приглашения `kermi`.

**Замечание:** Хранение пароля в любом месте файловой системы в незашифрованном виде это обычно плохая идея. Вы делаете это на свой риск.

2. Выйдите из `kermi` (вы можете приостановить его, нажав **Ctrl-z**) и введите под `root`:

```
# slattach -h -c -s 115200 /dev/modem
```

Если вы сможете выполнить `ping` для хостов по другую сторону маршрутизатора, вы подключились! Если это не работает, попробуйте параметр `slattach -a` вместо `-c`.

### 21.7.1.3. Как прервать соединение:

Сделайте следующее:

```
# kill -INT `cat /var/run/slattach.modem.pid`
```

для остановки `slattach`. Помните, что вы должны работать под `root` для выполнения этой команды. Затем вернитесь в `kermi` (запустив `fg`, если он приостановлен) и выйдите из него (`q`).

Страница справочника `slattach` сообщает, что для отключения интерфейса необходимо использовать `ifconfig s10 down`, но это похоже не играет никакой роли. (`ifconfig s10` сообщает о том же.)

Иногда модем может не сбросить соединение (это бывает довольно часто). В этом случае просто запустите `kermi` и выйдите из него еще раз. При второй попытке соединение обычно разрывается.

### 21.7.1.4. Решение проблем

Вот наиболее часто встречающиеся ситуации:

- Не используются параметры `slattach -c` или `-a` (это может быть не фатально, но иногда вызывает проблемы.)
- Используется `s10` вместо `s10` (с некоторыми шрифтами сложно увидеть разницу).
- Попробуйте использовать `ifconfig s10` для просмотра статуса интерфейса. Например, вы можете получить такую информацию:

```

# ifconfig s10
s10: flags=10<POINTOPOINT>
    inet 136.152.64.181 --> 136.152.64.1 netmask fffffff0

```

- Если вы получите сообщение no route to host от команды ping, возможно это проблема с таблицей маршрутизации. Используйте команду netstat -r для отображения существующих маршрутов:

```
# netstat -r
Routing tables

Destination      Gateway          Flags      Refs      Use  IfaceMTU    Rtt      Netmasks:

(root node)
(root node)

Route Tree for Protocol Family inet:
(root node) =>
default          inr-3.Example.EDU  UG          8    224515    sl0 -        -
localhost.Exampl localhost.Example. UH          5    42127    lo0 -        0.438
inr-3.Example.ED water.CS.Example.E UH          1         0    sl0 -        -
water.CS.Example localhost.Example. UGH         34  47641234    lo0 -        0.438
(root node)
```

Предыдущий пример получен на относительно загруженной системе. Числа в вашей системе будут сильно зависеть от загрузки сети.

## 21.7.2. Настройка SLIP сервера

Этот документ предоставляет решение для настройки SLIP сервера в системе FreeBSD, что обычно означает настройку системы для автоматического запуска соединений при удаленном входе SLIP клиентов.

### 21.7.2.1. Предварительные требования

Информация в этом разделе чисто техническая, поэтому требуются некоторые предварительные знания. Предполагается, что вы знакомы с сетевым протоколом TCP/IP, и в частности, с адресацией сетей и хостов, сетевыми масками, делением на подсети, маршрутизацией и протоколами маршрутизации, такими как RIP. Настройка SLIP сервисов на сервере удаленного доступа требует знания этих концепций, и если вы не знакомы с ними, прочтите или книгу *TCP/IP Network Administration* от Craig Hunt, опубликованную O'Reilly & Associates, Inc. (ISBN Number 0-937175-82-X), или книги Douglas Comer по протоколу TCP/IP.

В дальнейшем предполагается, что вы уже настроили ваш модем (модемы) и настроили соответствующие системные файлы для разрешения входа через них. Если вы еще не подготовили систему соответствующим образом, обратитесь к руководству по настройке сервисов удаленного входа; просмотрите список руководств на <http://www.FreeBSD.org/ru/docs.html>. Вы можете также обратиться к странице справочника sio(4) за информацией о драйвере последовательного порта и к страницам gettytab(5), getty(8) и init(8) за информацией по настройке системы для удаленного входа в систему через модемы, и возможно stty(1) за информацией о настройке параметров последовательных портов (таких как clocal для подключаемых непосредственно последовательных интерфейсов).

### 21.7.2.2. Краткий обзор

В типичной конфигурации FreeBSD работает в качестве SLIP сервера так: пользователь SLIP дозванивается на FreeBSD SLIP сервер и входит в систему со специальным SLIP логином, использующим `/usr/sbin/sliplogin` в качестве оболочки. Программа `sliplogin` просматривает файл `/etc/sliphome/slip.hosts` на предмет строки, соответствующей специальному пользователю, и если находит совпадение, подключает последовательную линию к доступному SLIP интерфейсу, а затем запускает shell скрипт `/etc/sliphome/slip.login` для настройки SLIP интерфейса.

#### 21.7.2.2.1. Пример входа на SLIP сервер

Например, идентификатор пользователя на SLIP сервере `Shelmerg`. Соответствующая запись в `/etc/master.passwd` будет выглядеть примерно так:

```
Shelmerg:password:1964:89::0:0:Guy Helmer - SLIP:/usr/users/Shelmerg:/usr/sbin/sliplogin
```

Когда `Shelmerg` входит в систему, `sliplogin` ищет строку в `/etc/sliphome/slip.hosts`, в которой находится соответствующий идентификатор пользователя; например, строка может быть такой:

```
Shelmerg dc-slip sl-helmer 0xffffffffc00 autocomp
```

После обнаружения этой строки `sliplogin` подключает последовательную линию к следующему доступному SLIP интерфейсу, а затем выполняет `/etc/sliphome/slip.login` примерно так:

```
/etc/sliphome/slip.login 0 19200 Shelmerg dc-slip sl-helmer 0xffffffffc00 autocomp
```

Если все проходит нормально, `/etc/sliphome/slip.login` вызовет `ifconfig` для SLIP интерфейса, к которому подключилась программа `sliplogin` (slip интерфейс 0 в примере выше, первый параметр в списке, задаваемом `slip.login`) для установки локального IP адреса ((`dc-slip`), удаленного IP адреса (`sl-helmer`), сетевой маски для SLIP интерфейса (`0xffffffffc00`), и любых дополнительных флагов (`autocomp`). Если что-то идет не так, `sliplogin` обычно протоколирует соответствующие сообщения в через уровень `daemon syslog`; эти сообщения как правило попадают в `/var/log/messages` (обратитесь к страницам справочника `syslogd(8)` и `syslog.conf(5)`, а также проверьте файл `/etc/syslog.conf`, чтобы выяснить, что протоколирует `syslogd` и куда помещается информация).

Достаточно примеров — давайте начнем настройку системы.

### 21.7.2.3. Настройка ядра

Стандартное ядро FreeBSD обычно поставляется с двумя SLIP интерфейсами ((`sl0` и `sl1`); вы можете использовать команду `netstat -i`, чтобы выяснить, определены ли эти интерфейсы в вашем ядре.

Пример вывода `netstat -i`:

Name	Mtu	Network	Address	Ipkts	Ierrs	Opkts	Oerrs	Coll
ed0	1500	<Link>	0.0.c0.2c.5f.4a	291311	0	174209	0	133
ed0	1500	138.247.224	ivory	291311	0	174209	0	133
lo0	65535	<Link>		79	0	79	0	0
lo0	65535	loop	localhost	79	0	79	0	0
sl0*	296	<Link>		0	0	0	0	0
sl1*	296	<Link>		0	0	0	0	0

Наличие в выводе `netstat -i` интерфейсов `sl0` и `sl1` означает, что SLIP интерфейсы встроены в ядро (символ `*` показывает неактивность интерфейсов).

Ядро FreeBSD по умолчанию не пересылает пакеты между интерфейсами (компьютер FreeBSD не работает как маршрутизатор), вследствие требований RFC (см. RFCs 1009 [Requirements for Internet Gateways], 1122 [Requirements for Internet Hosts — Communication Layers], и возможно 1127 [A Perspective on the Host Requirements RFCs]). Если вы хотите, чтобы FreeBSD SLIP работал в качестве маршрутизатора, отредактируйте файл `/etc/rc.conf` и присвойте переменной `gateway_enable` значение `YES`.

Для вступления изменений в силу потребуется перезагрузка.

В файле настройки стандартного ядра (`/sys/i386/conf/GENERIC`) находится строка:

```
pseudo-device sl 2
```

Она определяет число доступных устройств SLIP в ядре; Число в конце строки определяет максимально возможное количество одновременных SLIP соединений.

Обратитесь к Гл. 8 за информацией по настройке ядра FreeBSD.

#### 21.7.2.4. Настройка `sliplogin`

Как упоминалось ранее, в каталоге `/etc/sliphome` находятся три файла, являющиеся частью настройки для `/usr/sbin/sliplogin` (для `sliplogin` существует страница справочника, `sliplogin(8)`): `slip.hosts`, определяющий список пользователей SLIP и связанные с ними IP адреса; `slip.login`, который обычно всего лишь настраивает SLIP интерфейс; `slip.logout`, который восстанавливает состояние системы до запуска `slip.login` после завершения последовательного соединения.

##### 21.7.2.4.1. Настройка `slip.hosts`

`/etc/sliphome/slip.hosts` содержит строки, в которых находится как минимум четыре параметра, разделенных пробелами:

- ID пользователя SLIP
- Локальный адрес (локальный для SLIP сервера) SLIP соединения
- Удаленный адрес SLIP соединения
- Сетевая маска

Локальные и удаленные адреса могут быть именами хостов (разрешаемыми в IP адреса через файл `/etc/hosts` или через службу доменных имен, в зависимости от настроек в файле `/etc/nsswitch.conf` для FreeBSD 5.X, или `/etc/host.conf` для FreeBSD 4.X), а сетевая маска может быть именем, разрешаемым через файл `/etc/networks`. В системе, используемой в качестве примера, файл `/etc/sliphome/slip.hosts` выглядит так:

```
#
# login local-addr      remote-addr      mask          opt1    opt2
#                               (normal,compress,noicmp)
#
Shelmerg dc-slip        sl-helmerg      0xffffffff00   autocomp
```

В конце строки находятся один или более параметров.

- `normal` — нет сжатия заголовков
- `compress` — сжимать заголовки
- `autocomp` — сжимать заголовки, если удаленная сторона это позволяет
- `noismp` — запретить ICMP пакеты (любые "ping" пакеты будут отброшены и не станут помехой для другого трафика)

Выбор локального и удаленного адреса для SLIP соединений зависит от того, используете ли вы выделенную TCP/IP сеть, или используете на SLIP сервере "ARP прокси". (это не "настоящий" ARP прокси, но данная терминология используется в этом разделе). Если вы не уверены, какой метод выбрать, или как присвоить IP адреса, обратитесь к книгам по TCP/IP, упомянутым выше (Разд. 21.7.2.1).

Если вы собираетесь использовать отдельную подсеть для SLIP клиентов, потребуется выделить адреса за пределом адресов вашей сети и присвоить каждому SLIP клиенту IP адрес из данной подсети. Затем вам возможно потребуется настроить статический маршрут в используемую для SLIP подсеть через SLIP сервер на ближайшем IP маршрутизаторе.

Иначе, если вы будете использовать метод "проху ARP", потребуется присвоить SLIP клиентам IP адреса, не входящие в Ethernet подсеть сервера SLIP, а также настроить скрипты `/etc/sliphome/slip.login` и `/etc/sliphome/slip.logout`, чтобы использовать `arp(8)` для управления записями ARP прокси в таблице ARP сервера SLIP.

#### 21.7.2.4.2. Настройка `slip.login`

Типичный файл `/etc/sliphome/slip.login` выглядит примерно так:

```
#!/bin/sh -
#
#      @(#)slip.login  5.1 (Berkeley) 7/1/90
#
# generic login file for a slip line.  sliplogin invokes this with
# the parameters:
#      1      2      3      4      5      6      7-n
#  slipunit  ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
```

Этот файл `slip.login` всего лишь запускает `ifconfig` для соответствующего SLIP интерфейса с заданными локальным и удаленным адресом и сетевой маской.

Если вы решили использовать метод "ARP прокси" (вместо использования отдельной подсети для SLIP клиентов), ваш файл `/etc/sliphome/slip.login` должен выглядеть примерно так:

```
#!/bin/sh -
#
#      @(#)slip.login  5.1 (Berkeley) 7/1/90
#
# generic login file for a slip line.  sliplogin invokes this with
# the parameters:
```

```
#      1      2      3      4      5      6      7-n
#  slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 inet $4 $5 netmask $6
# Answer ARP requests for the SLIP client with our Ethernet addr
/usr/sbin/arp -s $5 00:11:22:33:44:55 pub
```

Дополнительная строка в этом `slip.login`, `arp -s $5 00:11:22:33:44:55 pub`, создает ARP запись в ARP таблице SLIP сервера. При соединении другого узла в Ethernet с IP адресом SLIP клиента, SLIP сервер выдает ответ с собственным Ethernet MAC адресом.

При использовании примера выше убедитесь, что заменили Ethernet MAC адрес (00:11:22:33:44:55) на MAC адрес Ethernet карты вашей системы, или ваш "ARP прокси" точно не будет работать! Вы можете определить Ethernet MAC адрес SLIP сервера, просмотрев вывод команды `netstat -i` выше; информация об адресе находится второй строке:

```
ed0    1500    <Link>0.2.c1.28.5f.4a          191923 0    129457      0    116
```

Это означает, что в данной системе Ethernet MAC адрес 00:02:c1:28:5f:4a — точки в MAC адресе, выдаваемые `netstat -i`, должны быть заменены на двоеточия, необходимо также добавить ноль в начало каждого односимвольного шестнадцатеричного номера для преобразования этого адреса в форму, пригодную для `arp(8)`; обратитесь к странице справочника `arp(8)` за полной информацией по использованию.

**Замечание:** При создании `/etc/sliphome/slip.login` и `/etc/sliphome/slip.logout`, должен быть установлен бит "выполнения" (`chmod 755 /etc/sliphome/slip.login /etc/sliphome/slip.logout`), или `sliplogin` не сможет их выполнить.

#### 21.7.2.4.3. Настройка `slip.logout`

`/etc/sliphome/slip.logout` не является совершенно необходимым (если только вы не реализуете "ARP прокси"), но если вы решили создать его, воспользуйтесь следующим примером:

```
#!/bin/sh -
#
#      slip.logout

#
# logout file for a slip line.  sliplogin invokes this with
# the parameters:
#      1      2      3      4      5      6      7-n
#  slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
```

Если вы используете "ARP прокси", потребуется удаление записи ARP для SLIP клиента через `/etc/sliphome/slip.logout`:

```
#!/bin/sh -
#
```



```
#      @(#)slip.logout

#
# logout file for a slip line.  sliplogin invokes this with
# the parameters:
#      1      2      3      4      5      6      7-n
#  slipunit ttyspeed loginname local-addr remote-addr mask opt-args
#
/sbin/ifconfig sl$1 down
# Quit answering ARP requests for the SLIP client
/usr/sbin/arp -d $5
```

Команда `arp -d $5` удаляет запись ARP, добавленную `slip.login` при входе SLIP клиента.

Повторяем: убедитесь, что на файл `/etc/sliphome/slip.logout` установлен бит выполнения (`chmod 755 /etc/sliphome/slip.logout`).

### 21.7.2.5. Соглашения о маршрутизации

Если вы не используете "ARP прокси" метод для маршрутизации пакетов между SLIP клиентами и остальной сетью (и возможно интернет), вам возможно потребуется статический маршрут (маршруты) до ближайшего шлюза (шлюзов) для маршрутизации подсети SLIP клиентов через SLIP сервер.

#### 21.7.2.5.1. Статические маршруты

Добавление статических маршрутов может стать для кого-то проблемой (это даже невозможно, если у вас нет соответствующих прав). Если в вашей организации сеть с несколькими маршрутизаторами, некоторые маршрутизаторы, например Cisco и Proteon, требуют не только настройки статического маршрута в подсеть SLIP, но и указания, о каких статических маршрутах сообщать другим маршрутизаторам, так что для наладки работоспособности статической маршрутизации может потребоваться некоторое исследование и отладка.

#### 21.7.2.5.2. Запуск **GateD®**

**Замечание:** **GateD®** это закрытое программно обеспечение, более недоступное в исходных текстах (дополнительная информация находится на вебсайте GateD (<http://www.gated.org/>)). Этот раздел существует лишь в целях обратной совместимости для тех, кто все еще использует старую версию.

Альтернатива головной боли со статическими маршрутами это установка **GateD** на FreeBSD SLIP сервере и настройка его для использования соответствующих протоколов маршрутизации (RIP/OSPF/BGP/EGP) для сообщения другим маршрутизаторам о вашей SLIP подсети. Вам потребуется создать `/etc/gated.conf` для настройки `gated`. Ниже дан пример:

```
#
# gated configuration file for dc.dsu.edu; for gated version 3.5alpha5
# Only broadcast RIP information for xxx.xxx.yy out the ed Ethernet interface
#
#
```

```

# tracing options
#
traceoptions "/var/tmp/gated.output" replace size 100k files 2 general ;

rip yes {
    interface sl noripout noripin ;
    interface ed ripin ripout version 1 ;
    traceoptions route ;
} ;

#
# Turn on a bunch of tracing info for the interface to the kernel:
kernel {
    traceoptions remnants request routes info interface ;
} ;

#
# Propagate the route to xxx.xxx.yy out the Ethernet interface via RIP
#

export proto rip interface ed {
    proto direct {
        xxx.xxx.yy mask 255.255.252.0 metric 1; # SLIP connections
    } ;
} ;

#
# Accept routes from RIP via ed Ethernet interfaces

import proto rip interface ed {
    all ;
} ;

```

В примере выше используется широковещательная рассылка информации о маршрутизации для подсети SLIP `xxx.xxx.yy` протоколом RIP на сеть Ethernet; если вы используете другой драйвер Ethernet вместо `ed`, потребуется соответственно изменить запись для `ed`. В этом примере отладочная информация переправляется в `/var/tmp/gated.output`; вы можете выключить отладку, если **GateD** работает. Вам потребуется заменить `xxx.xxx.yy` в сетевом адресе на вашу подсеть SLIP (убедитесь, что изменение сетевой маски в `proto direct` работает нормально).

Как только вы установили и настроили **GateD**, потребуется сообщить стартовым скриптам FreeBSD запускать его вместо **routed**. Простейший способ сделать это — установить переменные `router` и `router_flags` в `/etc/rc.conf`. Обратитесь к странице справочника **GateD** за информацией о параметрах командной строки.

# Глава 22. Электронная почта

*Оригинальную версию предоставил Bill Lloyd. Переписал Jim Mock. Перевод на русский язык: Алексей Докучаев, Денис Пеплин.*

## 22.1. Краткий обзор

“Электронная почта” называемая также email, является на сегодняшний день одним из самых популярных средств связи. Эта глава описывает основы работы с почтовым сервером в FreeBSD, а также введение в процесс отправки и получения почты в FreeBSD; однако, это не полноценный справочник и фактически в главу не вошло много важной информации. Более подробно эта тема рассмотрена во множестве прекрасных книг, список которых приведен в Прил. В.

После прочтения этой главы вы узнаете:

- Какие программные компоненты задействованы в отправке и получении электронной почты.
- Какие основные файлы настройки **sendmail** имеются в FreeBSD.
- Разницу между удаленными и локальными почтовыми ящиками.
- Как запретить спамерам использовать ваш почтовый сервер для пересылки почты.
- Как установить и настроить альтернативный агент передачи почты (Mail Transfer Agent, MTA), заменив им **sendmail**.
- Как разрешить наиболее часто встречающиеся проблемы с почтовым сервером.
- Как использовать SMTP с UUCP.
- Как настроить систему только для отправки почты.
- Как использовать почту с коммутируемым подключением к сети.
- Как настроить SMTP аутентификацию для дополнительной защиты.
- Как установить и настроить почтовый агент пользователя (Mail User Agent, MUA), например **mutt**, для отправки и получения почты.
- Как загрузить почту с удаленного POP или IMAP сервера.
- Как автоматически применять фильтры и правила к входящей почте.

Перед прочтением этой главы вам потребуется:

- Правильно настроить сетевое подключение (Гл. 24).
- Правильно настроить DNS для почтового сервера (Гл. 23).
- Знать как устанавливать дополнительное программное обеспечение сторонних разработчиков (Гл. 4).

## 22.2. Использование электронной почты

В работе почтовой системы задействованы пять основных частей: пользовательский почтовый клиент (Mail User Agent, MUA), почтовый сервис (демон) (Mail Transfer Agent, MTA), сервер DNS, удаленный или локальный почтовый ящик, и конечно сам почтовый сервер.

### 22.2.1. Пользовательский почтовый клиент

Обычно, это программа типа **mutt**, **pine**, **elm**, **mail**, а также программы с графическим интерфейсом, такие, как **balsa** или **xfmail**, или интегрированные приложения (например, какой-либо WWW браузер типа Netscape). Все эти программы общаются с локальным почтовым сервером, вызывая какой-либо демон, или напрямую по протоколу TCP.

### 22.2.2. Почтовый демон

FreeBSD по умолчанию поставляется с **sendmail**, но помимо того поддерживает множество других демонов почтового сервера, вот лишь некоторые из них:

- **exim**;
- **postfix**;
- **qmail**.

Почтовый демон выполняет только две функции: он отвечает за прием входящей почты и отправку исходящей. Он *не* отвечает за выдачу почты по протоколам POP или IMAP, и не обеспечивает подключения к локальным почтовым ящикам mbox или Maildir. Для этих целей вам может потребоваться дополнительный демон.

**Внимание:** Старые версии **sendmail** содержат некоторые серьезные ошибки безопасности, которые могут привести к получению атакующим локального и/или удаленного доступа к вашему компьютеру. Убедитесь, что вы работаете с современной версией, свободной от таких ошибок. Или установите альтернативный MTA из коллекции портов FreeBSD.

### 22.2.3. Email и DNS

Служба имен доменов (Domain Name System, DNS) и соответствующий ей демон named играют важную роль в доставке почты. Для доставки почты с вашего сайта другому, демон почтового сервера обратится к DNS для определения удаленного хоста, отвечающего за доставку почты по назначению. Тот же процесс происходит при доставке почты с удаленного хоста на ваш почтовый сервер.

DNS отвечает за сопоставления имен хостов IP адресам, как и за хранение информации, предназначенной для доставки почты, известной как MX записи. Запись MX (Mail eXchanger) определяет хост или хосты, которые будут получать почту для определенного домена. Если для вашего имени хоста или домена нет записи MX, почта будет доставлена непосредственно на ваш хост, IP адрес которого определен в записи A.

Вы можете просмотреть MX записи для любого домена с помощью команды `host(1)`, как показано в примере ниже:

```
% host -t mx FreeBSD.org
FreeBSD.org mail is handled (pri=10) by mx1.FreeBSD.org
```

## 22.2.4. Получение почты

Получение почты для вашего домена выполняет почтовый сервер. Он сохраняет отправленную в ваш домен почту в формате либо `mbox` (это метод по умолчанию), либо `Maildir`, в зависимости от настроек. После сохранения почты ее можно либо прочитать локально, используя такие приложения как `mail(1)`, **mutt**, или удаленно, по таким протоколам как POP или IMAP. Это означает, что для локального чтения почты вам не потребуется устанавливать сервер POP или IMAP.

### 22.2.4.1. Доступ к удаленным почтовым ящикам по протоколам POP и IMAP

Для удаленного доступа к почтовым ящикам вам потребуется доступ к POP или IMAP серверу. Хотя удаленный доступ обеспечивают оба протокола POP и IMAP, последний предоставляет множество дополнительных возможностей, вот некоторые из них:

- IMAP может как хранить сообщения на удаленном сервере, так и забирать их.
- IMAP поддерживает одновременные обновления.
- IMAP может быть очень полезен для низкоскоростных соединений, поскольку позволяет пользователям получить структуру сообщений без их загрузки; он также может использоваться для выполнения таких задач как поиск на сервере, для минимизации объема передаваемых между клиентом и сервером данных.

Для установки POP или IMAP сервера необходимо выполнить следующие действия:

1. Выберите IMAP или POP сервер, который подходит вам наилучшим образом. Следующие POP и IMAP серверы хорошо известны и могут быть приведены в качестве примера:
  - **qpopper**;
  - **teapop**;
  - **imap-uw**;
  - **courier-imap**;
2. Установите POP или IMAP даемон, выбранный из коллекции портов.
3. Если потребуется, настройте `/etc/inetd.conf` для запуска POP или IMAP сервера.

**Внимание:** Необходимо отметить, что и POP и IMAP серверы передают информацию, включая имя пользователя и пароль, в незашифрованном виде. Это означает, что если вы хотите защитить передачу информации по этим протоколам, потребуется использовать туннелирование сессий через `ssh(1)`. Туннелирование соединений описано в Разд. 14.12.7.

### 22.2.4.2. Доступ к локальным почтовым ящикам

Доступ к почтовым ящикам может быть осуществлен непосредственно путем использования MUA на сервере, где эти ящики расположены. Это можно сделать используя приложения вроде **mutt** или **mail(1)**.

### 22.2.5. Почтовый хост

Почтовый хост это сервер, который отвечает за отправку и получение почты для вашего компьютера, и возможно, для всей вашей сети.

## 22.3. Настройка **sendmail**

*Предоставил Christopher Shumway.*

В FreeBSD по умолчанию программой передачи почты (Mail Transfer Agent, MTA) является **sendmail(8)**. Работа **sendmail** заключается в приеме почты от почтовых программ пользователей (Mail User Agents, MUA) и отправке ее на соответствующий адрес, в соответствии с имеющимися настройками. **sendmail** может также принимать входящие соединения по сети и доставлять почту в локальные почтовые ящики или перенаправлять их другой программе.

**sendmail** использует следующие файлы настройки:

Имя файла	Назначение
<code>/etc/mail/access</code>	Файл базы данных доступа <b>sendmail</b>
<code>/etc/mail/aliases</code>	Синонимы почтовых ящиков
<code>/etc/mail/local-host-names</code>	Список хостов, для которых <b>sendmail</b> принимает почту
<code>/etc/mail/mailer.conf</code>	Настройки почтовой программы
<code>/etc/mail/mailertable</code>	Таблица доставки почтовой программы
<code>/etc/mail/sendmail.cf</code>	Основной файл настройки <b>sendmail</b>
<code>/etc/mail/virtusertable</code>	Таблицы виртуальных пользователей и доменов

### 22.3.1. `/etc/mail/access`

База данных доступа определяет список хостов или IP адресов, имеющих доступ к локальному почтовому серверу, а также тип предоставляемого доступа. Хосты могут быть перечислены как OK, REJECT, RELAY или просто переданы процедуре обработки ошибок **sendmail** с заданным сообщением об ошибке. Хостам, перечисленным с параметром по умолчанию OK, разрешено отправлять почты на этот хост, если адрес назначения почты принадлежит локальной машине. Все почтовые соединения от хостов, перечисленных с параметром REJECT, отбрасываются. Для хостов, перечисленных с параметром RELAY, разрешена передача через этот сервер почты с любым адресом назначения.

**Пример 22-1. Настройка базы данных доступа sendmail**

cyberspammer.com	550 We don't accept mail from spammers
FREE.STEALTH.MAILER@	550 We don't accept mail from spammers
another.source.of.spam	REJECT
okay.cyberspammer.com	OK
128.32	RELAY

В этом примере приведены пять записей. К отправителям, чей адрес соответствует записи в левой части таблицы, применяется правило записанное в правой части таблицы. В первых двух примерах код ошибки будет передан процедуре обработке ошибок **sendmail**. В этом случае на удаленном хосте будет получено соответствующее сообщение. В следующем примере почта отбрасывается почта от определенного хоста, `another.source.of.spam`. В четвертом примере разрешается прием почты от хоста `okay.cyberspammer.com`, имя которого более точно совпадает с этой записью, чем с `cyberspammer.com` в примере выше. При более точном совпадении правила перезаписываются. В последнем примере разрешается пересылка почты от хостов с IP адресами, начинающимися с 128.32. Эти хосты смогут отправлять почту через этот почтовый сервер для других почтовых серверов.

После изменения этого файла для обновления базы данных вам потребуется запустить `make` в каталоге `/etc/mail/`.

**22.3.2. /etc/mail/aliases**

База данных синонимов содержит список виртуальных почтовых ящиков, принадлежащих другим пользователям, файлам, программам, или другим синонимам. Вот несколько примеров, которые могут быть использованы для `/etc/mail/aliases`:

**Пример 22-2. Mail Aliases**

```
root: localuser
ftp-bugs: joe,eric,paul
bit.bucket: /dev/null
procmail: "|/usr/local/bin/procmail"
```

Формат файла прост; имя почтового ящика слева от двоеточия сопоставляется назначению(ям) справа. В первом примере производится простое сопоставление почтового ящика `root` почтовому ящику `localuser`, для которого затем опять будет произведен поиск в базе данных синонимов. Если совпадений не обнаружится, сообщение будет доставлено локальному пользователю `localuser`. В следующем примере приведен список рассылки. Почта на адрес `ftp-bugs` рассылается на три локальных почтовых ящика: `joe`, `eric` и `paul`. Обратите внимание, что удаленный почтовый ящик может быть задан в виде `user@example.com`. В следующем примере показана запись почты в файл, в данном случае `/dev/null`. И в последнем примере показано отправление почты программе, в данном случае почтовое сообщение переправляется через канал **UNIX** на стандартный вход `/usr/local/bin/procmail`.

После обновления этого файла вам потребуется запустить `make` в каталоге `/etc/mail/` для обновления базы данных.

### 22.3.3. /etc/mail/local-host-names

В этом файле находится список имен хостов, принимаемых программой `sendmail(8)` в качестве локальных. Поместите в этот файл любые домены или хосты, для которых **sendmail** должен принимать почту. Например, если этот почтовый сервер должен принимать почту для домена `example.com` и хоста `mail.example.com`, его файл `local-host-names` может выглядеть примерно так:

```
example.com
mail.example.com
```

После обновления этого файла необходимо перезапустить `sendmail(8)`, чтобы он смог перечитать изменения.

### 22.3.4. /etc/mail/sendmail.cf

Основной файл настройки **sendmail**, `sendmail.cf` управляет общим поведением **sendmail**, включая все, от перезаписи почтовых адресов до отправки удаленным серверам сообщений об отказе от пересылки почты. Конечно, файл настройки с таким многообразием возможностей очень сложен и подробное его описание выходит за рамки данного раздела. К счастью, для стандартных почтовых серверов изменять этот файл придется не часто.

Основной файл настройки **sendmail** может быть собран из макроса `m4(1)`, определяющего возможности и поведение **sendmail**. Подробнее этот процесс описан в файле `/usr/src/contrib/sendmail/cf/README`.

Для применения изменений после правки файла необходимо перезапустить **sendmail**.

### 22.3.5. /etc/mail/virtusertable

Файл `virtusertable` сопоставляет виртуальные почтовые домены и почтовые ящики реальным почтовым ящикам. Эти почтовые ящики могут быть локальными, удаленными, синонимами, определенными в `/etc/mail/aliases`, или файлами.

#### Пример 22-3. Пример таблицы виртуального домена

<code>root@example.com</code>	<code>root</code>
<code>postmaster@example.com</code>	<code>postmaster@noc.example.net</code>
<code>@example.com</code>	<code>joe</code>

В примере выше мы видим сопоставление адресов для домена `example.com`. Почта обрабатывается по первому совпадению с записью в этом файле. Первая запись сопоставляет адрес `root@example.com` локальному почтовому ящику `root`. Вторая запись сопоставляет `postmaster@example.com` локальному почтовому ящику `postmaster` на хосте `noc.example.net`. Наконец, до этого момента адрес в домене `example.com` не совпал ни с одним из предыдущих, будет применено последнее сопоставление, в котором соответствует всякое другое почтовое сообщение, отправленное на любой адрес в `example.com`. Это сообщение будет доставлено в локальный почтовый ящик `joe`.



## 22.4. Установка другой почтовой программы

*Написал Andrew Boothman. Информация получена из писем, написанных Gregory Neil Shapiro.*

Как уже упоминалось, FreeBSD поставляется с MTA (Mail Transfer Agent) **sendmail**. Следовательно, по умолчанию именно эта программа отвечает за вашу исходящую и входящую почту.

Однако, по различным причинам некоторые системные администраторы заменяют системный MTA. Эти причины варьируются от простого желания попробовать другой MTA до потребности в определенных возможностях пакета, основанного на другой почтовой программе. К счастью, вне зависимости от причины, в FreeBSD такая замена выполняется просто.

### 22.4.1. Установка нового MTA

Вам предоставлен широкий выбор MTA. Начните с поиска в коллекции портов FreeBSD, где их немало. Конечно, вы можете использовать любой MTA по желанию, взятый откуда угодно, если только сможете запустить его под FreeBSD.

Начните с установки нового MTA. После установки у вас будет возможность решить, действительно ли он подходит вашим нуждам, а также настроить новое программное обеспечение перед тем, как заменить им **sendmail**. При установке новой программы убедитесь, что она не пытается перезаписать системные файлы, такие как `/usr/bin/sendmail`. Иначе ваша новая почтовая программа фактически начнет работать до того, как вы ее настроите.

Обратитесь к документации на выбранный MTA за информацией по его настройке.

### 22.4.2. Отключение **sendmail**

Процедура, используемая для запуска **sendmail**, значительно изменилась между релизами 4.5-RELEASE и 4.6-RELEASE. Поэтому процедура, используемая для его отключения, немного отличается.

#### 22.4.2.1. FreeBSD 4.5-STABLE до 2002/4/4 и раньше (включая 4.5-RELEASE и более ранние)

Добавьте:

```
sendmail_enable="NO"
```

в `/etc/rc.conf`. Это отключит сервис входящей почты **sendmail**, но если `/etc/mail/mailer.conf` (см. ниже) не изменен, **sendmail** все еще будет использоваться для отправки почты.

#### 22.4.2.2. FreeBSD 4.5-STABLE после 2002/4/4 (включая 4.6-RELEASE и более поздние)

Для полного отключения **sendmail** используйте

```
sendmail_enable="NONE"
```

в `/etc/rc.conf`.

**Внимание:** Если вы отключите сервис исходящей почты **sendmail** таким способом, необходимо заменить его полностью работоспособной альтернативной системой доставки почты. Если вы не сделаете этого, системные

программы, такие как `periodic(8)`, не смогут отправлять сообщения по электронной почте как обычно. Многие программы в вашей системе могут требовать наличия функционирующей **sendmail**-совместимой системы. Если приложения будут продолжать использовать программу **sendmail**, пытаясь отправить почту после отключения почтовой системы, почта может попасть в неактивную очередь **sendmail** и никогда не будет доставлена.

Если вы хотите отключить только сервис входящей почты **sendmail**, установите

```
sendmail_enable="NO"
```

в `/etc/rc.conf`. Дополнительная информация о параметрах запуска **sendmail** доступна на странице справочника `rc.sendmail(8)`.

### 22.4.3. Запуск нового MTA при загрузке

Вы можете выбрать один из двух методов для запуска нового MTA при загрузке, опять же в зависимости от используемой версии FreeBSD.

#### 22.4.3.1. FreeBSD 4.5-STABLE до 2002/4/11 (включая 4.5-RELEASE и более ранние)

Добавьте в каталог `/usr/local/etc/rc.d/`, скрипт с расширением `.sh` и с правами выполнения пользователем `root`. Скрипт должен принимать параметры `start` и `stop`. Во время запуска системы он будет запущен командой

```
/usr/local/etc/rc.d/supermailer.sh start
```

которая также может использоваться для запуска сервера вручную. Во время завершения работы системы, системные скрипты используют параметр `stop`, выполнив команду

```
/usr/local/etc/rc.d/supermailer.sh stop
```

которая также может использоваться для остановки почтового сервера при работающей системе.

#### 22.4.3.2. FreeBSD 4.5-STABLE после 2002/4/11 (включая 4.6-RELEASE и более поздние)

С более поздними версиями FreeBSD, вы можете использовать метод, описанный выше, или добавить строку

```
mta_start_script="filename"
```

в `/etc/rc.conf`, где *filename* это имя скрипта, который вы хотите выполнить при загрузке для запуска MTA.

#### 22.4.4. Замещение **sendmail** как почтовой программы по умолчанию

Программа **sendmail** настолько распространена в качестве стандартной программы для систем UNIX, что многие программы считают, что она уже установлена и настроена. По этой причине многие альтернативные МТА предоставляют собственные совместимые реализации интерфейса командной строки **sendmail**; это облегчает их использование в качестве "прозрачной" замены **sendmail**.

Поэтому если вы используете альтернативную почтовую программу, потребуется убедиться, что когда программное обеспечение пытается выполнить стандартные исполняемые файлы **sendmail**, такие как `/usr/bin/sendmail`, на самом деле выполняются программы вновь установленной почтовой системы. К счастью, FreeBSD предоставляет систему, называемую `mailwrapper(8)`, которая выполняет эту работу за вас.

Когда установлен **sendmail**, файл `/etc/mail/mailer.conf` выглядит примерно так:

```
sendmail /usr/libexec/sendmail/sendmail
send-mail /usr/libexec/sendmail/sendmail
mailq /usr/libexec/sendmail/sendmail
newaliases /usr/libexec/sendmail/sendmail
hoststat /usr/libexec/sendmail/sendmail
purgestat /usr/libexec/sendmail/sendmail
```

Это означает, что когда выполняется какая-то из этих стандартных программ (например сам **sendmail**), система на самом деле вызывает копию `mailwrapper`, называемую **sendmail**, которая обращается к `mailer.conf` и выполняет вместо этого `/usr/libexec/sendmail/sendmail`. Такая схема делает простой замену программ, которые на самом деле выполняются, когда вызываются стандартные функции **sendmail**.

Поэтому если вы хотите выполнять `/usr/local/supermailer/bin/sendmail-compat` вместо **sendmail**, отредактируйте `/etc/mail/mailer.conf` так:

```
sendmail /usr/local/supermailer/bin/sendmail-compat
send-mail /usr/local/supermailer/bin/sendmail-compat
mailq /usr/local/supermailer/bin/mailq-compat
newaliases /usr/local/supermailer/bin/newaliases-compat
hoststat /usr/local/supermailer/bin/hoststat-compat
purgestat /usr/local/supermailer/bin/purgestat-compat
```

#### 22.4.5. Запуск новой почтовой программы

Как только вы все настроили, потребуется или уничтожить процесс **sendmail**, который уже не нужен и запустить новую почтовую программу, или просто перегрузить систему. Перезагрузка также даст вам возможность проверить, правильно ли настроена система для автоматического запуска МТА при загрузке.

## 22.5. Поиск и устранение неисправностей

### 1. Почему я должен использовать FQDN для хостов вне моей подсети?

Вы, видимо, обнаружили, что хост, к которому вы обратились, оказался на самом деле в другом домене; например, если вы находитесь в домене `foo.bar.edu` и хотите обратиться к хосту `tumble` в домене `bar.edu`, то должны указать его полное доменное имя, `tumble.bar.edu`, а не просто `tumble`.

Традиционно, программа разрешения имен BSD BIND позволяла это делать. Однако, текущая версия **BIND**, поставляемая с FreeBSD, больше не добавляет имена доменов, отличающихся от того, в котором вы находитесь, для не полностью указанных имен хостов. То есть, имя `tumble` будет опознан как `tumble.foo.bar.edu` или будет искаться в корневом домене.

Это отличается от предыдущего поведения, при котором поиск продолжался в доменах `tumble.bar.edu` и `tumble.edu`. Если вам интересны причины объявления такого поведения плохой практикой и даже ошибкой в безопасности, обратитесь к RFC 1535.

Хорошим решением будет поместить строку

```
search foo.bar.edu bar.edu
```

вместо ранее используемой:

```
domain foo.bar.edu
```

в файл `/etc/resolv.conf`. Однако удостоверьтесь, что порядок поиска не нарушает "границ полномочий между локальным и внешним администрированием", в терминологии RFC 1535.

### 2. **sendmail** выдает ошибку `mail loops back to myself`

В FAQ по **sendmail** дан следующий ответ:

Я получаю такие сообщения об ошибке:

```
553 MX list for domain.net points back to relay.domain.net
554 <user@domain.net>... Local configuration error
```

Как можно решить эту проблему?

Согласно записям MX, почта для домена `domain.net` перенаправляется на хост `relay.domain.net`, однако последний не распознается как `domain.net`. Добавьте `domain.net` в файл `/etc/mail/local-host-names` [известный как `/etc/sendmail.cw` до версии 8.10] (если вы используете `FUTURE(use_cw_file)`) или добавьте "`Cw domain.net`" в файл `/etc/mail/sendmail.cf`.

FAQ по **sendmail** можно найти на <http://www.sendmail.org/faq/> и рекомендуется прочесть его при желании произвести некоторые "усовершенствования" настроек почтовой системы.

### 3. Как организовать работу почтового сервера при коммутируемом соединении с Интернет?

Вы хотите подключить к интернет компьютер с FreeBSD, работающий в локальной сети. Компьютер с FreeBSD будет почтовым шлюзом для локальной сети. PPP соединение не выделенное.

Существует как минимум два пути, чтобы сделать это. Один способ это использование UUCP.

Другой способ это использование постоянно работающего интернет сервера для обеспечения вторичного MX сервиса вашего домена. Например, домен вашей компании `example.com`, и провайдер интернет настроил `example.net` для обеспечения вторичного MX сервиса:

<code>example.com.</code>	<code>MX</code>	<code>10</code>	<code>example.com.</code>
	<code>MX</code>	<code>20</code>	<code>example.net.</code>

Только один хост должен быть указан в качестве последнего получателя (добавьте запись `Cw example.com` в файл `/etc/mail/sendmail.cf` на машине `example.com`).

Когда программа `sendmail` (со стороны отправителя) "захочет" доставить почту, она попытается соединиться с вашим хостом (`example.com`) через модемное подключение. Скорее всего, ей это не удастся (вы, вероятнее всего, не будете подключены к интернет). Программа **sendmail** автоматически перейдет ко вторичному MX серверу, т.е. вашему провайдеру (`example.net`). Вторичный MX сервер будет периодически пытаться соединиться с вашим хостом и доставить почту на основной сервер MX (`example.com`).

Вы можете воспользоваться следующим сценарием, чтобы забирать почту каждый раз, когда вы входите в систему:

```
#!/bin/sh
# Put me in /usr/local/bin/pppmyisp
( sleep 60 ; /usr/sbin/sendmail -q ) &
/usr/sbin/ppp -direct pppmyisp
```

Если же вы хотите написать отдельный пользовательский скрипт, лучше воспользоваться командой `sendmail -qRexample.com` вместо вышеприведенного сценария, так как в этом случае вся почта в очереди для хоста `example.com` будет обработана немедленно.

Рассмотрим эту ситуацию подробнее:

Вот пример сообщения из `freebsd-isp` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-isp>).

```
> Мы предоставляем вторичный MX для наших клиентов. Вы соединяетесь
> с нашим сервером несколько раз в день, чтобы забрать почту для вашего
> первичного (главного) MX (мы не соединяемся с ним каждый раз, когда
> приходит новая почта для его доменов). Далее, sendmail отправляет
> почту, находящуюся в очереди каждые 30 минут, и клиент должен быть
> подключен к Интернет в течении 30 минут, чтобы удостовериться, что
> вся почта "ушла" на основной MX-сервер.
>
> Может быть, есть какая-либо команда, которая заставит sendmail
> немедленно отправить все почту, находящуюся в очереди? Естественно,
> пользователи не обладают какими-либо повышенными привилегиями на
> нашем сервере.
```

В разделе "privacy flags" файла

```
sendmail.cf, определяется опция
Opgoaway,restrictqrun
```

Уберите `restrictqrun`, чтобы разрешить рядовым пользователям инициировать работу с очередью. Вам также может понадобиться изменить порядок MX-серверов. Так, если вы предоставляете первый (основной) MX-сервер для ваших пользователей, мы указываем:

```
# If we are the best MX for a host, try directly instead of generating
# local config error.
OwTrue
```

Таким образом, удаленный хост будет доставлять почту непосредственно к вам, не пытаясь установить соединение с клиентом. Затем уже вы, в свою очередь, отправляете ее клиенту. Удостоверьтесь, что в DNS есть записи про `"customer.com"` и `"hostname.customer.com"`. Просто добавьте запись A в DNS для `"customer.com"`.

#### 4. Почему я продолжаю получать ошибки `Relaying Denied` при отправке почты через другие хосты?

В установке FreeBSD по умолчанию, **sendmail** настроен для отправки почты только от хоста, на котором он работает. Например, если доступен POP сервер, пользователи смогут проверять почту из школы, с работы или других удаленных точек, но не смогут отправлять письма. Обычно, через некоторое время после попытки будет отправлено письмо от **MAILER-DAEMON** с сообщением об ошибке 5.7 `Relaying Denied`.

Есть несколько путей разрешения этой ситуации. Самый прямой путь это использование адреса вашего провайдера в файле `relay-domains`, расположенном в `/etc/mail/relay-domains`. Быстрый способ сделать это:

```
# echo "your.isp.example.com" > /etc/mail/relay-domains
```

После создания или редактирования этого файла вы должны перезапустить **sendmail**. Это отлично работает, если вы администратор сервера и не хотите отправлять почту локально, или хотите воспользоваться почтовым клиентом/системой на другом компьютере или даже через другого провайдера. Это также очень полезно, если у вас настроены одна или две почтовые записи. Если необходимо добавить несколько адресов, вы можете просто открыть этот файл в текстовом редакторе и добавить домены, по одному на строку:

```
your.isp.example.com
other.isp.example.net
users-isp.example.org
www.example.org
```

Теперь будет отправляться любая почта, посылаемая через вашу систему любым хостом из этого списка (предоставляемого пользователем, имеющим учетную запись в вашей системе). Это отличный способ разрешить пользователям отправлять почту через вашу систему удаленно, одновременно он блокирует отправку спама.

## 22.6. Расширенное руководство

В следующем разделе рассматриваются более сложные темы, такие как настройка почты и включение почтовой системы для всего домена.

### 22.6.1. Базовая конфигурация

Изначально, вы можете отправлять почту "во внешний мир" если правильно составлен файл `/etc/resolv.conf` или запущен свой сервер имен. Если вы хотите, чтобы почта, предназначенная для хоста в вашем домене, доставлялась MTA (например, **sendmail**) на вашем хосте FreeBSD, есть два пути:

- Запустите свой собственный сервер DNS, тем самым организовав собственный домен, например, `FreeBSD.org`
- Получайте почту для вашего хоста непосредственно. Это работает при доставке почты непосредственно на DNS имя вашей машины. Например, `example.FreeBSD.org`.

Независимо от выбранного из предложенных выше вариантов, для доставки почты непосредственно на ваш хост у него должен быть постоянный IP адрес (а не динамический, как у большинства PPP соединений). Если вы находитесь за брандмауэром, то последний должен пропускать SMTP-пакеты. Если вы хотите, чтобы почта приходила непосредственно на ваш хост, необходимо убедиться в одном из двух:

- Убедитесь, что запись (с наименьшим номером) MX в DNS соответствует IP адресу вашего хоста.
- Убедитесь, что в DNS для вашего хоста вообще отсутствует MX-запись.

Выполнение любого из перечисленных условий обеспечит доставку почты для вашего хоста.

Попробуйте это:

```
# hostname
example.FreeBSD.org
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
```

Если вы это видите, то можно без проблем посылать почту на `<yourlogin@example.FreeBSD.org>` (предполагается, что **sendmail** на `example.FreeBSD.org` работает правильно).

Однако, если вы видите это:

```
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
example.FreeBSD.org mail is handled (pri=10) by hub.FreeBSD.org
```

то вся почта, посланная на `example.FreeBSD.org` будет собираться на `hub` (для того же пользователя), вместо того, чтобы быть отосланной непосредственно на ваш хост.

Эта информация обрабатывается вашим DNS сервером. Соответствующая запись DNS, указывающая, через какой хост будет проходить ваша почта, называется MX (*Mail eXchanger*). Если для хоста отсутствует такая запись, почта будет приходить прямо на этот хост.

Допустим, что запись MX для хоста `freefall.FreeBSD.org` в какой-то момент выглядела так:

```
freefall MX 30 mail.crl.net
```

```
freefall MX 40 agora.rdrop.com
freefall MX 10 freefall.FreeBSD.org
freefall MX 20 who.cdrom.com
```

Вы видите, что для хоста `freefall` существуют несколько MX-записей. Запись с наименьшим номером соответствует хосту, получающему почту непосредственно, если он доступен; если он недоступен по каким-то причинам, другие сервера (иногда называемые ("резервными MX") временно получают почту, и хранят ее пока не станут доступны хосты с меньшими номерами, в конечном итоге отправляя почту на эти хосты.

Чтобы альтернативные MX-хосты использовались наиболее эффективно, они должны быть независимо подключены к Интернет. Ваш провайдер (или дружественный сайт) скорее всего без проблем сможет оказать подобные услуги.

## 22.6.2. Почта для вашего домена

Для настройки "почтового хоста" (почтовый сервер) вам потребуется, чтобы почта, направляемая различным рабочим станциям, пересылалась этому хосту. Обычно вам необходима доставка всей почты для любого хоста вашего домена (в данном случае `*.FreeBSD.org`) на почтовый сервер, чтобы пользователи могли получать свою почту на с этого сервера.

Чтобы облегчить себе (и другим) жизнь, создайте на обеих машинах учетные записи с одинаковыми именами пользователей, например, с помощью команды `adduser(8)`.

Сервер, который вы будете использовать в качестве почтового, должен быть объявлен таковым для каждой машины в домене. Вот фрагмент примерной конфигурации:

```
example.FreeBSD.org A 204.216.27.XX ; Рабочая станция
MX 10 hub.FreeBSD.org ; Почтовый шлюз
```

Таким образом, вся корреспонденция, адресованная рабочей станции, будет обрабатываться вашим почтовым сервером, независимо от того, что указано в A-записи.

Все это можно реализовать только в том случае, если вы используете сервер DNS. Если вы по каким-либо причинам не имеете возможности установить свой собственный сервер имен, необходимо договориться с провайдером или теми, кто поддерживает ваш DNS.

Если вы хотите поддерживать несколько виртуальных почтовых серверов, может пригодиться следующая информация. Допустим, что ваш клиент зарезервировал домен, например, `customer1.org`, и вам требуется, чтобы почта, предназначенная для `customer1.org` приходила на ваш хост, например, `mail.myhost.com`. В таком случае, DNS должен выглядеть так:

```
customer1.org MX 10 mail.myhost.com
```

Заметьте, что если вам требуется только получать почту для домена, соответствующая A-запись *не* нужна.

**Замечание:** Помните, что если вы попытаетесь каким-либо образом обратиться к хосту `customer1.org`, у вас вряд ли что-либо получится, если нет A-записи для этого хоста.



Последнее, что вы должны сделать – это сказать программе **sendmail**, для каких доменов и/или хостов она должна принимать почту. Это можно сделать несколькими способами:

- Добавьте названия этих хостов в файл `/etc/mail/local-host-names`, если вы используете `FEATURE(use_cw_file)`. Если у вас **sendmail** версии ниже 8.10, необходимо отредактировать файл `/etc/sendmail.cw`.
- Добавьте строку `Cyour.host.com` в файл `/etc/sendmail.cf` или `/etc/mail/sendmail.cf` (если у вас **sendmail** версии 8.10 или более поздней).

## 22.7. SMTP через UUCP

Настройка поставляемого с FreeBSD **sendmail** предназначена для сайтов, подключенных к интернет непосредственно. Сайты, осуществляющие обмен почтой через UUCP, должны использовать другой файл настройки **sendmail**.

Редактирование `/etc/mail/sendmail.cf` вручную это сложная задача. **sendmail** версии 8 генерирует файлы настройки через препроцессор `m4(1)`, реально настройка выполняется на более высоком уровне абстракции. Файлы настройки `m4(1)` можно найти в `/usr/src/usr.sbin/sendmail/cf`.

Если вы не установили в систему все исходные тексты, пакет настройки **sendmail** можно найти в отдельном архиве исходных текстов. Если CDROM с исходными текстами FreeBSD смонтирован, выполните:

```
# cd /cdrom/src
# cat scontrib.?? | tar xzf - -C /usr/src/contrib/sendmail
```

Эта установка займет всего несколько сотен килобайт. Файл `README` в каталоге `cf` содержит введение в основы настройки `m4(1)`.

Лучшим способом настройки поддержки передачи по UUCP является использование возможности `mailertable`. При этом создается база данных, которая помогает **sendmail** решать вопросы маршрутизации.

Во-первых, создайте файл `.mc`. В каталоге `/usr/src/usr.sbin/sendmail/cf/cf` находятся несколько примеров. Возьмем для примера имя файла `foo.mc`. Все, что потребуется для преобразования его в `sendmail.cf`, это:

```
# cd /usr/src/usr.sbin/sendmail/cf/cf
# make foo.cf
# cp foo.cf /etc/mail/sendmail.cf
```

Типичный `.mc` файл может выглядеть примерно так:

```
VERSIONID('Your version number') OSTYPE(bsd4.4)

FEATURE(accept_unresolvable_domains)
FEATURE(nocanonify)
FEATURE(mailertable, 'hash -o /etc/mail/mailertable')

define('UUCP_RELAY', your.uucp.relay)
```

```
define('UUCP_MAX_SIZE', 200000)
define('confDONT_PROBE_INTERFACES')

MAILER(local)
MAILER(smtp)
MAILER(uucp)

Cw    your.alias.host.name
Cw    youruucpnodename.UUCP
```

Строки, содержащие `accept_unresolvable_domains`, `nocanonify`, и `confDONT_PROBE_INTERFACES`, предотвратят использование DNS для доставки почты. Пункт `UUCP_RELAY` необходим для поддержки доставки по UUCP. Просто поместите сюда имя хоста в интернет, способного работать с .UUCP адресами псевдо-доменов; скорее всего, вы введете сюда основной сервер пересылки почты провайдера.

Как только вы сделаете это, потребуется файл `/etc/mail/mailertable`. Если вы используете для всей почты только одно внешнее соединение, подойдет следующий файл:

```
#
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
.
    uucp-dom:your.uucp.relay
```

Более сложный пример может выглядеть так:

```
#
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
#
horus.interface-business.de    uucp-dom:horus
.interface-business.de        uucp-dom:if-bus
interface-business.de          uucp-dom:if-bus
.heep.sax.de                   smtp8:%1
horus.UUCP                     uucp-dom:horus
if-bus.UUCP                    uucp-dom:if-bus
.                               uucp-dom:
```

В первых трех строках обрабатываются специальные случаи, когда почта для домена должна отправляться не на маршрут по умолчанию, а на ближайшее соединение UUCP для сокращения пути доставки. Следующая строка обрабатывает почту, которая может быть доставлена по SMTP для локального Ethernet домена. Наконец, определены маршруты UUCP в нотации псевдо-доменов .UUCP, для включения перезаписи правил по умолчанию правилом `uucp-neighbor !recipient`. Последняя строка всегда содержит одиночную точку, означающую "все остальное", с отправкой через UUCP, являющимся универсальным почтовым шлюзом. Все имена узлов после ключевого слова `uucp-dom:` должны представлять существующие маршруты UUCP, проверить их можно с помощью команды `uname`.

Напоминаем, что этот файл должен быть преобразован в базу данных DBM перед использованием. Командную строку для этой задачи лучше всего поместить в качестве комментария в верхней части файла `mailertable`. Всегда выполняйте эту команду после правки файла `mailertable`.

И наконец: если вы не уверены, что некоторые отдельные почтовые маршруты будут работать, запомните параметр `sendmail -bt`. С этим параметром **sendmail** запускается в *режиме тестирования адреса*; просто введите 3, 0 и адрес, который вы хотите протестировать. В последней строке появится сообщение об используемом внутреннем почтовом агенте, хосте назначения, с которым вызывается

этот агент, и (возможно транслированный) адрес. Выход из этого режима происходит при нажатии **Ctrl+D**.

```
% sendmail -bt
ADDRESS TEST MODE (ruleset 3 NOT automatically invoked)
Enter <ruleset> <address>
> 3,0 foo@example.com
canonify          input: foo @ example . com
...
parse            returns: $# uucp-dom $@ your.uucp.relay $: foo < @ example . com . >
> ^D
```

## 22.8. Настройка почты только для отправки

*Предоставил Bill Moran.*

Существует множество случаев, когда может потребоваться только отправка почты через почтовый сервер. Вот отдельные примеры:

- У вас настольный компьютер, но вы хотите использовать такие программы как `send-pr(1)`. Для пересылки почты вам потребуется использовать почтовый сервер провайдера.
- Ваш компьютер является сервером, где почта не хранится локально, необходима только переправка всей почты через внешний почтовый сервер.

Практически любой МТА способен работать и в этих условиях. К сожалению, может быть очень сложно правильно настроить полноценный МТА для работы только с исходящей почтой. Такие программы, как **sendmail** и **postfix** слишком избыточны для этих целей.

К тому же, если вы используете обычные средства доступа в интернет, условий для запуска "почтового сервера" может быть недостаточно.

Простейшим способом удовлетворить имеющиеся потребности может быть установка порта `mail/ssmtp`. Выполните под `root` следующие команды:

```
# cd /usr/ports/mail/ssmtp
# make install replace clean
```

После установки потребуется настроить `mail/ssmtp` с помощью файла из четырех строк, расположенного в `/usr/local/etc/ssmtp/ssmtp.conf`:

```
root=yourrealemail@example.com
mailhub=mail.example.com
rewriteDomain=example.com
hostname=_HOSTNAME_
```

Убедитесь, что используете существующий почтовый адрес для `root`. Введите сервер вашего провайдера для пересылки исходящей почты вместо `mail.example.com` (некоторые провайдеры называют его "сервером исходящей почты" или "SMTP сервером").

Убедитесь, **sendmail** выключен путем добавления строки `sendmail_enable="NONE"` в `/etc/rc.conf`.

У пакета `mail/ssmtp` имеются и другие параметры. Обратитесь к файлу с примером настройки в `/usr/local/etc/ssmtp` или к странице справочника **ssmtp** за примерами и дополнительной информацией.

Установка **ssmtp** таким способом позволит правильно работать любым программам на вашем компьютере, которым требуется отправка почты, но не нарушит политику вашего провайдера и не позволит вашему компьютеру быть использованным спамерами.

## 22.9. Использование почты с коммутируемым соединением

Если у вас есть статический IP, настройки по умолчанию менять не потребуется. Установите имя хоста в соответствии с присвоенным именем интернет и **sendmail** будет делать свою работу.

Если у вас динамический IP адрес и используется коммутируемое PPP соединение с интернет, у вас возможно уже есть почтовый ящик на сервере провайдера. Предположим, что домен провайдера называется `example.net`, и что ваше имя пользователя `user`, ваш компьютер называется `bsd.home`, и провайдер сообщил вам, что возможно использование `relay.example.net` в качестве сервера для пересылки почты.

Для получения почты из почтового ящика необходима установка соответствующей программы. Хорошим выбором является утилита **fetchmail**, она поддерживает множество различных протоколов. Эта программа доступна в виде пакета или из коллекции портов (`mail/fetchmail`). Обычно провайдер предоставляет доступ по протоколу POP. Если вы работаете с пользовательским PPP, то можете автоматически забирать почту после установления соединения с интернет с помощью следующей записи в `/etc/ppp/ppp.linkup`:

```
MYADDR:
!bg su user -c fetchmail
```

Если вы используете **sendmail** (как показано ниже) для доставки почты к не-локальным учетным записям, вам возможно потребуется обработка почтовой очереди **sendmail** сразу после установки соединения с интернет. Для выполнения этой работы поместите в `/etc/ppp/ppp.linkup` следующую команду сразу после `fetchmail`:

```
!bg su user -c "sendmail -q"
```

Предполагается, что учетная запись для `user` существует на `bsd.home`. В домашнем каталоге `user` на `bsd.home`, создайте файл `.fetchmailrc`:

```
poll example.net protocol pop3 fetchall pass MySecret
```

Этот файл не должен быть доступен на чтение никому, кроме `user`, поскольку в нем находится пароль `MySecret`.

Для отправки почты с правильным заголовком `from:`, вам потребуется сообщить **sendmail** использовать `user@example.net` вместо `user@bsd.home`. Вы можете также указать **sendmail** отправлять почту через `relay.example.net`, для более быстрой пересылки почты.

Должен подойти следующий файл `.mc`:

```
VERSIONID('bsd.home.mc version 1.0')
OSTYPE(bsd4.4)dnl
```

```

FEATURE(nouucp)dnl
MAILER(local)dnl
MAILER(smtp)dnl
Cwlocalhost
Cwbsd.home
MASQUERADE_AS('example.net')dnl
FEATURE(allmasquerade)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(nocanonify)dnl
FEATURE(nodns)dnl
define('SMART_HOST', 'relay.example.net')
Dmbsd.home
define('confDOMAIN_NAME', 'bsd.home')dnl
define('confDELIVERY_MODE', 'deferred')dnl

```

Обратитесь к предыдущему разделу за информацией о том, как преобразовать этот файл .mc в файл `sendmail.cf`. Не забудьте также перезапустить **sendmail** после обновления `sendmail.cf`.

## 22.10. SMTP аутентификация

*Написал James Gorham.*

Наличие SMTP аутентификации на почтовом сервере дает множество преимуществ. SMTP аутентификация может добавить дополнительный уровень безопасности к **sendmail**, и позволяет мобильным пользователям, подключающимся к разным хостам, возможность использовать тот же почтовый сервер без необходимости перенастройки почтового клиента при каждом подключении.

1. Установите `security/cyrus-sasl` из портов. Вы можете найти этот порт в `security/cyrus-sasl`. В пакете `security/cyrus-sasl` есть множество параметров компиляции, и для используемого здесь метода убедитесь, что выбран параметр `pwcheck`.
2. После установки `security/cyrus-sasl`, отредактируйте `/usr/local/lib/sasl/Sendmail.conf` (или создайте его если он не существует) и добавьте следующую строку:

```
pwcheck_method: passwd
```

Этот метод включит аутентификацию **sendmail** через базу данных `passwd` FreeBSD. Это позволяет избежать проблем, связанных с созданием нового набора имен пользователей и паролей для каждого пользователя, которому необходима SMTP аутентификация, пароль для входа в систему и для отправки почты будет одним и тем же.

3. Теперь отредактируйте `/etc/make.conf` и добавьте следующие строки:

```

SENDMAIL_CFLAGS=-I/usr/local/include/sasl -DSASL
SENDMAIL_LDFLAGS=-L/usr/local/lib
SENDMAIL_LDADD=-lsasl

```

Эти параметры необходимы **sendmail** для подключения `cyrus-sasl` во время компиляции. Убедитесь, что `cyrus-sasl` был установлен до перекомпиляции **sendmail**.

4. Перекомпилируйте **sendmail**, выполнив следующие команды:

```

# cd /usr/src/usr.sbin/sendmail
# make cleandir

```

```
# make obj
# make
# make install
```

Компиляция **sendmail** должна пройти без проблем, если `/usr/src` не был сильно изменен и доступны необходимые разделяемые библиотеки.

- После компилирования и переустановки **sendmail**, отредактируйте файл `/etc/mail/freebsd.mc` (или тот файл, который используется в качестве `.mc`; многие администраторы используют в качестве имени этого файла вывод `hostname(1)` для обеспечения уникальности). Добавьте к нему следующие строки:

```
dnl set SASL options
TRUST_AUTH_MECH( 'GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN' )dnl
define( 'confAUTH_MECHANISMS', 'GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN' )dnl
define( 'confDEF_AUTH_INFO', '/etc/mail/auth-info' )dnl
```

Эти параметры настраивают различные методы, доступные **sendmail** для аутентификации пользователей. Если вы хотите использовать вместо **pwcheck** другой метод, обратитесь к прилагаемой документации.

- Наконец, запустите `make(1)` в каталоге `/etc/mail`. Из файла `.mc` будет создан файл `.cf`, называющийся `freebsd.cf` (или с тем именем, которое было использовано для файла `.mc`). Затем используйте команду `make install restart`, которая скопирует файл в `sendmail.cf`, и правильно перезапустит **sendmail**. Дополнительная информация об этом процессе находится в `/etc/mail/Makefile`.

Если все шаги пройдены успешно, введите информацию для аутентификации в настройки почтового клиента и отправьте тестовое сообщение. Для определения причин возможных ошибок установите параметр **sendmail** `LogLevel` в 13 и просмотрите `/var/log/maillog`.

Для включения сервиса после каждой загрузки системы вам может потребоваться добавление в `/etc/rc.conf` следующих строк:

```
sasl_pwcheck_enable="YES"
sasl_pwcheck_program="/usr/local/sbin/pwcheck"
```

Эти строки инициализируют `SMTP_AUTH` при загрузке системы.

За дальнейшей информацией обратитесь к странице **sendmail**, посвященной `SMTP` аутентификации (<http://www.sendmail.org/~ca/email/auth.html>).

## 22.11. Почтовые программы пользователей

*Предоставил Marc Silver.*

Почтовая программа пользователя (Mail User Agent, MUA) это приложение, используемое для отправки и получения почты. Кроме того, поскольку почта "эволюционирует" и становится более сложной, MUA совершенствуют свои функции по обработке почты, становятся более удобны в использовании. FreeBSD поддерживает множество различных пользовательских почтовых программ, каждая из которых может быть легко установлена из коллекции портов FreeBSD. Пользователи могут выбирать между графическими почтовыми клиентами, такими как **evolution** или **balsa**, консольными клиентами, такими как **mutt**, **pine** или **mail**, или веб интерфейсами, используемыми в некоторых больших организациях.

### 22.11.1. mail

В FreeBSD в качестве MUA по умолчанию используется `mail(1)`. Это консольный MUA, предоставляющий все основные функции, необходимые для отправки и получения текстовых сообщений, хотя его возможности по работе с вложениями ограничены и он может работать только с локальными почтовыми ящиками.

Хотя `mail` не поддерживает работу с серверами POP или IMAP, эти почтовые ящики могут быть загружены в локальный файл `mbox` с помощью **fetchmail**, который будет обсуждаться далее в этой главе (Разд. 22.12).

Для отправки и получения почты просто выполните команду `mail`, как в этом примере:

```
% mail
```

Содержимое почтового ящика в каталоге `/var/mail` будет автоматически прочитано утилитой `mail`. Если почтовый ящик пуст, утилита завершит работу с сообщением о том, что почта не была обнаружена. После чтения почтового ящика запустится интерфейс программы и будет отображен список сообщений. Сообщения нумеруются автоматически и будут выглядеть как в этом примере:

```
Mail version 8.1 6/6/93.  Type ? for help.
"/var/mail/marcs": 3 messages 3 new
>N  1 root@localhost      Mon Mar  8 14:05  14/510  "test"
   N  2 root@localhost      Mon Mar  8 14:05  14/509  "user account"
   N  3 root@localhost      Mon Mar  8 14:05  14/509  "sample"
```

Теперь сообщения могут быть прочитаны с помощью команды **t**, завершаемой номером сообщения, которое должно быть отображено. В этом примере мы прочтем первое сообщение:

```
& t 1
Message 1:
From root@localhost  Mon Mar  8 14:05:52 2004
X-Original-To: marcs@localhost
Delivered-To: marcs@localhost
To: marcs@localhost
Subject: test
Date: Mon,  8 Mar 2004 14:05:52 +0200 (SAST)
From: root@localhost (Charlie Root)
```

```
This is a test message, please reply if you receive it.
```

Как видно в примере выше, клавиша **t** выводит сообщение со всеми заголовками. Для повторного вывода списка сообщений необходимо использовать клавишу **h**.

Если требуется ответить на сообщение, используйте для ответа `mail`, нажав клавишу **R** или **r**. Клавиша **R** используется в `mail` для ответа только отправителю, а **r** для ответа и отправителю, и другим получателям сообщения. Вы можете также завершить эти команды номером письма, на которое хотите составить ответ. После этого необходимо ввести ответ, конец сообщения должен быть завершен символом `.` на новой строке. Пример можно увидеть ниже:

```
& R 1
To: root@localhost
Subject: Re: test
```

```
Thank you, I did get your email.
```

```
.
```

```
EOT
```

Для отправки нового сообщения используйте клавишу **m** и введите адрес получателя. Несколько получателей могут быть указаны через запятую. Введите тему сообщения и его содержимое. Конец сообщения отмечается помещением символа **.** на новой строке.

```
& mail root@localhost
```

```
Subject: I mastered mail
```

```
Now I can send and receive email using mail ... :)
```

```
.
```

```
EOT
```

В утилите **mail** для вызова справки в любой момент может быть использована команда **?**, для получения помощи по **mail** необходимо также обратиться к странице справочника **mail(1)**.

**Замечание:** Как упоминалось выше, команда **mail(1)** не была первоначально предназначена для работы с вложениями, и поэтому их поддержка довольно слабая. Современные MUA, такие как **mutt**, работают с вложениями гораздо более уверенно. Но если вы все же предпочитаете использовать **mail**, установите порт **converters/mpack**.

## 22.11.2. mutt

**mutt** это небольшая но очень мощная почтовая программа с отличными возможностями, в числе которых:

- Возможность сортировки сообщений по дискуссиям;
- Поддержка PGP для подписи и шифрования сообщений;
- Поддержка MIME;
- Поддержка Maildir;
- Широкие возможности настройки.

Все эти возможности делают **mutt** одним из самых лучших почтовых клиентов. Обратитесь к <http://www.mutt.org> за дополнительной информацией по **mutt**.

Стабильная версия **mutt** может быть установлена из порта **mail/mutt**. После установки порта, **mutt** может быть запущен следующей командой:

```
% mutt
```

**mutt** автоматически прочтет содержимое пользовательского почтового ящика в каталоге **/var/mail** и отобразит почту, если она имеется в наличии. Если почты в ящике пользователя нет, **mutt** будет ожидать команд от пользователя. В примере ниже показан **mutt** со списком сообщений:



```

q:Quit  d:Del  u:Undel  s:Save  m:Mail  r:Reply  g:Group  ?:Help
 1 N   Mar 09 Super-User   ( 1) test
 2 N   Mar 09 Super-User   ( 1) user account
 3 N   Mar 09 Super-User   ( 1) sample

-----
* Mutt: /var/mail/marcs [Msgs:3 New:3 1.6K]---(date/date)----- (all) -----

```

Для чтения почты просто выберите сообщение с помощью клавиш навигации и нажмите **Enter**. Пример **mutt**, отображающего сообщение, показан ниже:

```

i:Exit  -:PrevPg  <Space>:NextPg  u:View Attachm.  d:Del  r:Reply  j:Next  ?:Help
X-Original-To: marcs@localhost
Delivered-To: marcs@localhost
To: marcs@localhost
Subject: test
Date: Tue, 9 Mar 2004 10:28:36 +0200 (SAST)
From: Super-User <root@localhost>

This is a test message, please reply if you receive it.

-----
-N - 1/1: Super-User      test      -- (all) -----

```

Как и команда `mail(1)`, **mutt** позволяет пользователям отвечать как только отправителю, так и всем получателям. Для ответа только отправителю почты, используйте клавишу **r**. Для группового ответа и отправителю сообщения и всем получателям используйте клавишу **g**.

**Замечание:** **mutt** использует `vi(1)` в качестве редактора для создания писем и ответа на них. Редактор можно заменить путем создания или редактирования собственного `.muttrc` в своем домашнем каталоге и установки переменной `editor`.

Для создания нового почтового сообщения нажмите **m**. После введения темы **mutt** запустит `vi(1)` для создания письма. Как только письмо будет завершено, сохраните его и закройте `vi`, **mutt** продолжит работу, отобразив окно с сообщением, которое должно быть отправлено. Для отправки сообщения нажмите **y**. Пример окна с сообщением показан ниже:

```

y:Send q:Abort t:To c:CC s:Subj a:Attach file d:Descrip ?:Help
  From: Marc Silver <marcs@localhost>
  To: Super-User <root@localhost>
  Cc:
  Bcc:
  Subject: Re: test
  Reply-To:
  Fcc:
  Security: Clear

-- Attachments
- I      1 /tmp/mutt-bsd-c0hobscQ      [text/plain, 7bit, us-ascii, 1.1K]

-----
Mutt: Compose [Approx. msg size: 1.1K  Atts: 1]

```

**mutt** также содержит исчерпывающий справочник, к которому можно обратиться из большинства меню, нажав клавишу **?**. Верхняя строка также показывает клавиатурные сокращения, которые могут быть использованы.

### 22.11.3. pine

**pine** предназначен для начинающих пользователей, но включает некоторые дополнительные возможности.

**Внимание:** В программе **pine** ранее были обнаружены некоторые уязвимости, позволяющие удаленному взломщику выполнять произвольный код с правами пользователя локальной системы путем отправки специально подготовленного письма. Все эти *известные* проблемы были исправлены, но код **pine** написан в очень небезопасном стиле и офицеры безопасности FreeBSD считают, что возможно наличие других не обнаруженных уязвимостей. Имейте это ввиду при установке **pine**.

Текущая версия **pine** может быть установлена из порта mail/pine4. Как только порт установлен, **pine** можно запустить командой:

```
% pine
```

При первом запуске **pine** отображает страницу приветствия с кратким введением, а также просьбу команды разработчиков **pine** отправить анонимное почтовое сообщение, позволяющее им определить количество пользователей, работающих с их почтовым клиентом. Для отправки анонимного сообщения нажмите **Enter**, или **E** для выхода из приветствия без отправки анонимного сообщения. Пример приветствия показан ниже:

```

PINE 4.58  GREETING TEXT                                     No Messages

<<<This message will appear only once>>>

Welcome to Pine ... a Program for Internet News and Email

We hope you will explore Pine's many capabilities. From the Main Menu,
select Setup/Config to see many of the options available to you. Also
note that all screens have context-sensitive help text available.

SPECIAL REQUEST: This software is made available world-wide as a public
service of the University of Washington in Seattle. In order to justify
continuing development, it is helpful to have an idea of how many people
are using Pine. Are you willing to be counted as a Pine user? Pressing
Return will send an anonymous (meaning, your real email address will not
be revealed) message to the Pine development team at the University of
Washington for purposes of tallying.

Pine is a trademark of the University of Washington.

[ALL of greeting text]
? Help      E Exit this greeting      - PrevPage  Z Print
Ret [Be Counted!]      Spc NextPage

```

Затем отображается главное меню, перемещение по которому осуществляется с помощью клавиш навигации. В главном меню находятся ссылки для составления новых писем, просмотра почтовых каталогов, и даже управления адресной книгой. Ниже главного меню показаны клавиатурные сокращения, выполняющие соответствующие задачи.

По умолчанию **pine** открывает каталог **inbox**. Для просмотра списка сообщений нажмите **I**, или выберите **MESSAGE INDEX**, как показано ниже:

```

PINE 4.58  MAIN MENU                                         Folder: INBOX  3 Messages

?  HELP          - Get help using Pine
C  COMPOSE MESSAGE - Compose and send a message
I  MESSAGE INDEX  - View messages in current folder
L  FOLDER LIST    - Select a folder to view
A  ADDRESS BOOK   - Update address book
S  SETUP          - Configure Pine Options
Q  QUIT           - Leave the Pine program

Copyright 1989-2003.  PINE is a trademark of the University of Washington.

? Help      P PrevCnd      R RelNotes
O OTHER CMDS [Index]  N NextCnd  K KBLock

```

В списке показаны сообщения в текущем каталоге, они могут быть просмотрены с помощью клавиш навигации. Подсвеченные сообщения можно прочесть нажав **Enter**.

```

PINE 4.58  MESSAGE INDEX                               Folder: INBOX  Message 1 of 3 ANS
A  1 Mar  9 Super-User                                (471) test
A  2 Mar  9 Super-User                                (479) user account
A  3 Mar  9 Super-User                                (473) sample

? Help      < FldrList  P PrevMsg      = PrevPage D Delete  R Reply
0 OTHER CMDS > [ViewMsg] N NextMsg    Spc NextPage J Undelete F Forward

```

На снимке экрана ниже показан пример письма, отображаемого **pine**. Внизу экрана даны клавиатурные сокращения. Например, **r** используется для указания MUA ответить на отображаемое в данный момент сообщение.

```

PINE 4.58  MESSAGE TEXT                               Folder: INBOX  Message 1 of 3 ALL ANS
Date: Tue,  9 Mar 2004 10:28:36 +0200 (SAST)
From: Super-User <root@localhost>
To: marcs@localhost
Subject: test

This is a test message, please reply if you receive it.

[ALL of message]
? Help      < MsgIndex  P PrevMsg      = PrevPage D Delete  R Reply
0 OTHER CMDS > ViewAttch N NextMsg    Spc NextPage J Undelete F Forward

```

Ответ на письмо в **pine** осуществляется с помощью редактора **pico**, который устанавливается по умолчанию вместе с **pine**. **pico** упрощает навигацию в сообщении гораздо проще для новых пользователей, чем **vi(1)** или **mail(1)**. Как только ответ будет готов, сообщение можно отправить нажав **Ctrl+X**. **pine** запросит подтверждение.

```

PINE 4.58  COMPOSE MESSAGE REPLY  Folder: INBOX  3 Messages
To      : Super-User <root@localhost>
Cc      :
Atthmnt:
Subject : Re: test
----- Message Text -----

I did recieve your message...

^G Get Help  ^X Send      ^R Read File ^Y Prev Pg  ^K Cut Text  ^O Postpone
^C Cancel    ^J Justify   ^W Where is ^U Next Pg  ^U UnCut Text ^T To Spell

```

Программа **pine** может быть настроена через пункт **SETUP** главного меню. Обратитесь к странице <http://www.washington.edu/pine/> за дальнейшей информацией.

## 22.12. Использование fetchmail

*Предоставил Marc Silver.*

**fetchmail** это полноценный IMAP и POP клиент, позволяющий пользователям автоматически загружать почту с удаленных серверов IMAP и POP в локальные почтовые ящики; так доступ к почтовым ящикам упрощается. **fetchmail** может быть установлен из порта mail/fetchmail и предоставляет различные возможности, в том числе:

- Поддержка протоколов POP3, APOP, KPOP, IMAP, ETRN и ODMR.
- Возможность пересылки почты через SMTP, что позволяет использовать функции фильтрации, перенаправления и синонимов.
- Может быть запущен в режиме даемона для периодической проверки поступающих сообщений.
- Может забирать почту с нескольких почтовых ящиков и рассылать ее различным локальным пользователям в зависимости от настроек.

Описание всех возможностей **fetchmail** выходит за пределы этой главы, за дополнительной информацией обратитесь к документации по **fetchmail**. Утилита **fetchmail** требует наличия файла настройки `.fetchmailrc`. Этот файл включает информацию о сервере, а также информацию для аутентификации. Поскольку этот файл содержит важную информацию, правильно будет сделать его доступным для чтения только владельцем с помощью следующей команды:

```
% chmod 600 .fetchmailrc
```

В следующем примере файл `.fetchmailrc` предназначен для загрузки одного почтового ящика по протоколу POP. Этот файл указывает **fetchmail** соединиться с `example.com` с именем пользователя `joesoar` и паролем `xxx`. В примере подразумевается, что пользователь `joesoar` существует также и в локальной системе.

```
poll example.com protocol pop3 username "joesoap" password "XXX"
```

В следующем примере производится подключение к нескольким POP и IMAP серверам, при необходимости почта перенаправляется другим локальным пользователям:

```
poll example.com proto pop3:
user "joesoap", with password "XXX", is "jsoap" here;
user "andrea", with password "XXXX";
poll example2.net proto imap:
user "john", with password "XXXXX", is "myth" here;
```

Утилита **fetchmail** может работать в режиме демона с флагом `-d`, заданным с интервалом (в секундах), через который **fetchmail** должен опрашивать серверы, перечисленные в `.fetchmailrc`. В следующем примере **fetchmail** будет забирать почту каждые 60 секунд:

```
% fetchmail -d 60
```

Дополнительную информацию о **fetchmail** можно найти на сайте <http://www.catb.org/~esr/fetchmail/>.

## 22.13. Использование procmail

*Предоставил Marc Silver.*

Утилита **procmail** это невероятно мощное приложение, используемое для фильтрации входящей почты. Она позволяет пользователям определять "правила", которые могут быть сопоставлены входящим письмам для выполнения определенных действий или для перенаправления почты в альтернативные почтовые ящики и/или на почтовые адреса. **procmail** может быть установлен с помощью порта `mail/procmail`. После установки он может быть непосредственно интегрирован в большинство МТА; сверьтесь с документацией на ваш МТА. Другой способ интеграции **procmail** – добавление в файл `.forward`, находящийся в домашнем каталоге пользователя, следующей строки:

```
"|exec /usr/local/bin/procmail || exit 75"
```

В этом разделе будут показаны основы настройки правил **procmail**, а также краткое описание их действия. Эти и другие правила должны быть помещены в файл `.procmailrc`, который должен находиться в домашнем каталоге пользователя.

Большую часть этих правил также можно найти на странице справочника `procmailex(5)`.

Перенаправление всей почты от `user@example.com` на внешний адрес `goodmail@example2.com`:

```
:0
* ^From.*user@example.com
! goodmail@example2.com
```

Перенаправление всей почты объемом меньше 1000 байт на внешний адрес `goodmail@example2.com`:

```
:0
* < 1000
! goodmail@example2.com
```

Перенаправление всей почты, отправляемой на `alternate@example.com`, в почтовый ящик `alternate`:

```
:0
* ^TOalternate@example.com
alternate
```

Перенаправление всей почты с "Spam" в /dev/null:

```
:0
^Subject:.*Spam
/dev/null
```

Полезный пример, обрабатывающий входящую почту со списков рассылки FreeBSD.org и помещающий каждый список в отдельный почтовый ящик.

```
:0
* ^Sender:.owner-freebsd-\/[ ^@]+@FreeBSD.ORG
{
LISTNAME=${MATCH}
:0
* LISTNAME??^\/[ ^@]+
FreeBSD-${MATCH}
}
```

# Глава 23. Сетевые серверы

*Реорганизовал Murray Stokely. Перевод на русский язык: Андрей Захватов.*

## 23.1. Краткий обзор

Эта глава посвящена некоторым наиболее часто используемым сетевым службам систем UNIX. Мы опишем, как установить, настроить, протестировать и поддерживать многие различные типы сетевых сервисов. Для облегчения вашей работы в главу включены примеры конфигурационных файлов.

После чтения этой главы вы будете знать:

- Как управлять демоном **inetd**.
- Как настроить сетевую файловую систему.
- Как настроить сетевой сервер информации для совместного использования учётных записей пользователей.
- Как настроить автоматическое конфигурирование сетевых параметров при помощи DHCP.
- Как настроить сервер имён.
- Как настроить **Apache** HTTP сервер.
- Как настроить файловый и принт сервер для Windows клиентов с использованием **Samba**.
- Как синхронизировать дату и время, а также настроить сервер времени с протоколом NTP.

Перед чтением этой главы вы должны:

- Понимать основы работы скриптов `/etc/rc`.
- Свободно владеть основными сетевыми терминами.
- Знать как устанавливать дополнительные программы сторонних разработчиков (Гл. 4).

## 23.2. "Супер-сервер" **inetd**

*Текст предоставил Chern Lee.*

### 23.2.1. Обзор

**inetd(8)** называют также "супер-сервером Интернет", потому что он управляет соединениями к многим сервисам. Когда **inetd** принимает соединение, он определяет, для какой программы предназначено соединение, запускает соответствующий процесс и предоставляет ему сокет, ссылка на который передается процессу в качестве стандартных устройств ввода, вывода и сообщения об ошибках. Запуск одного экземпляра **inetd** уменьшает общую нагрузку на систему по сравнению с запуском каждого демона индивидуально в выделенном режиме.

В первую очередь **inetd** используется для вызова других демонов, но несколько простых протоколов, таких, как **chargen**, **auth** и **daytime**, обслуживаются непосредственно.



Этот раздел посвящен основам настройки **inetd** посредством его параметров командной строки и его конфигурационного файла, `/etc/inetd.conf`.

### 23.2.2. Настройки

**inetd** инициализируется посредством системы `/etc/rc.conf`. Параметр `inetd_enable` по умолчанию установлен в `NO`, однако часто включается утилитой **sysinstall** при выборе профиля среднего уровня безопасности. Указание

```
inetd_enable="YES"
```

или

```
inetd_enable="NO"
```

в файле `/etc/rc.conf` может задать или запретить запуск **inetd** во время загрузки.

Кроме того, через `inetd_flags` даемону **inetd** могут быть переданы различные параметры командной строки.

### 23.2.3. Параметры командной строки

Формат вызова **inetd**:

```
inetd [-d] [-l] [-w] [-W] [-c maximum] [-C rate] [-a address | hostname] [-p filename]
[-R rate] [configuration file]
```

**-d**

Включение отладочной информации.

**-l**

Включение регистрации успешных соединений.

**-w**

Включение механизма TCP Wrapping для внешних служб (по умолчанию включено).

**-W**

Включение механизма TCP Wrapping для внутренних служб, которые встроены в **inetd** (по умолчанию включено).

**-c maximum**

Определение максимального числа одновременных запусков каждой службы; по умолчанию не ограничено. Может быть переопределено индивидуально для каждой службы при помощи параметра `max-child`.

**-C rate**

Определение по умолчанию максимального количества раз, которое служба может быть вызвана с одного IP-адреса в минуту; по умолчанию не ограничено. Может быть переопределено для каждой службы параметром `max-connections-per-ip-per-minute`.

**-R rate**

Определяет максимальное количество раз, которое служба может быть вызвана в минуту; по умолчанию 256. Частота, равная 0, не ограничивает число вызовов.

**-a**

Задаёт один IP-адрес, к которому делается привязка. Альтернативно может быть указано имя хоста, и в этом случае используется соответствующий этому имени хоста адрес IPv4 или IPv6. Обычно имя хоста задаётся, когда **inetd** запускается в окружении `jail(8)`, и в этом случае имя хоста соответствует этому `jail(8)`-окружению.

Если используется формат с именем хоста и требуется привязка как для IPv4, так и для IPv6, то для каждой привязки требуется запись с соответствующим типом протокола для каждой службы в файле `/etc/inetd.conf`. К примеру, службе на основе TCP потребуется две записи, в одной для протокола используется `tcp4`, а в другой используется `tcp6`.

**-p**

Задаёт альтернативный файл для хранения ID процесса.

Эти параметры могут быть переданы в **inetd** при помощи `inetd_flags` в файле `/etc/rc.conf`. По умолчанию значение `inetd_flags` установлено в `-wW`, что включает механизм TCP wrapping для внутренних и внешних служб **inetd**. Новичкам эти параметры изменять и даже задавать их в файле `/etc/rc.conf` не нужно.

**Замечание:** Внешняя служба является демоном вне **inetd**, который запускается при получении соединения к нему. С другой стороны, внутренней службой является услуга, которую **inetd** предоставляет сам.

## 23.2.4. `inetd.conf`

Настройка **inetd** управляется через файл `/etc/inetd.conf`.

Если в файле `/etc/inetd.conf` делались изменения, то **inetd** можно заставить считать его конфигурационный файл повторно, послав сигнал `HangUP` процессу **inetd**, как показано здесь:

### Пример 23-1. Посылка сигнала `HangUP` процессу **inetd**

```
# kill -HUP `cat /var/run/inetd.pid`
```

В каждой строке конфигурационного файла описывается отдельный демон. Комментариям в файле предшествует знак `#`. Файл `/etc/inetd.conf` имеет такой формат:

```
service-name
```

```

socket-type
protocol
{wait|nowait}[/max-child[/max-connections-per-ip-per-minute]]
user[:group][[/login-class]]
server-program
server-program-arguments

```

Пример записи для демона **ftpd**, использующего IPv4:

```
ftp      stream  tcp      nowait  root    /usr/libexec/ftpd      ftpd -l
```

#### service-name

Это имя сервиса, предоставляемого конкретным демоном. Оно должно соответствовать сервису, указанному в файле `/etc/services`. Здесь определяется, какой порт должен обслуживать **inetd**. При создании нового сервиса он должен помещаться сначала в файл `/etc/services`.

#### socket-type

`stream`, `dgram`, `raw` либо `seqpacket`. `stream` должен использоваться для ориентированных на соединение демонов TCP, когда как `dgram` используется для демонов, использующих транспортный протокол UDP.

#### protocol

Одно из следующих:

Протокол	Описание
tcp, tcp4	TCP IPv4
udp, udp4	UDP IPv4
tcp6	TCP IPv6
udp6	UDP IPv6
tcp46	TCP как для IPv4, так и для v6
udp46	UDP как для IPv4, так и для v6

```
{wait|nowait}[/max-child[/max-connections-per-ip-per-minute]]
```

`wait|nowait` определяет, может ли демон, вызванный из **inetd**, работать с собственным сокетом, или нет. Сокеты типа `dgram` должны использовать параметр `wait`, когда как демоны с потоковыми сокетами, которые обычно многопоточны, должны использовать `nowait`. `wait` обычно передает много сокетов одному демону, когда как `nowait` порождает демон для каждого нового сокета.

Максимальное число порожденных демонов, которых может создать **inetd**, может быть задано параметром `max-child`. Если нужно ограничение в десять экземпляров некоторого демона, то после параметра `nowait` нужно задать `/10`.

Кроме `max-child`, может быть задействован другой параметр, ограничивающий максимальное число соединений от одного источника. `max-connections-per-ip-per-minute` служит именно для этого. Здесь значение, равное десяти, будет ограничивать любой заданный IP-адрес на

выполнение десяти попыток подключения к некоторому сервису в минуту. Это полезно для предотвращения намеренного или ненамеренного расходования ресурсов и атак типа Denial of Service (DoS) на машину.

В этом поле `wait` или `nowait` обязательны. `max-child` и `max-connections-per-ip-per-minute` опциональны.

Многопоточный даемон типа `stream` без ограничений `max-child` или `max-connections-per-ip-per-minute` будет определен просто вот так: `nowait`.

Тот же самый даемон с ограничением в максимум десять дамонов будет определен так: `nowait/10`.

Наконец, та же конфигурация с ограничением в двадцать соединений на IP-адрес в минуту и общим ограничением в максимум десять порожденных дамонов выглядит так: `nowait/10/20`.

Эти параметры, используемые все со значениями по умолчанию дамоном **fingerd**, имеют такой вид:

```
finger stream tcp      nowait/3/10 nobody /usr/libexec/fingerd fingerd -s
```

`user`

Это имя пользователя, под которым должен работать соответствующий даемон. Чаще всего даemons работают как пользователь `root`. Для обеспечения безопасности некоторые серверы запускаются как пользователь `daemon` или как пользователь с минимальными правами `nobody`.

`server-program`

Полный маршрут к даемону, который будет выполняться при установлении соединения. Если даемон является сервисом, предоставляемым самим **inetd**, то нужно задать ключевое слово `internal`.

`server-program-arguments`

Этот параметр работает вместе с параметром `server-program`, задавая параметры, начиная с `argv[0]`, передаваемые даемону при запуске. Если в командной строке задано `mydaemon -d`, то `mydaemon -d` будет являться значением для `server-program-arguments`. И снова, если даемон является внутренней службой, то здесь нужно использовать `internal`.

### 23.2.5. Безопасность

В зависимости от схемы безопасности, выбранной при установке, многие из дамонов **inetd** могут оказаться по умолчанию включенными. Если нет особой нужды в некотором даемоне, то выключите его! Поместите знак `#` перед ненужным дамоном в `/etc/inetd.conf` и пошлите сигнал для **inetd**. Некоторые даemons, такие, как **fingerd**, вообще нежелательны, потому что они дают атакующему слишком много информации.

Некоторые даemons не заботятся о безопасности и имеют большие таймауты для соединений или вообще их не имеют. Это позволяет атакующему неспешно устанавливать соединения к конкретному даемону, истощая имеющиеся ресурсы. Может оказаться полезным задать для некоторых дамонов ограничения `max-connections-per-ip-per-minute` и `max-child`.

По умолчанию механизм TCP wrapping включен. Обратитесь к справочной странице по `hosts_access(5)` для получения более подробной информации о задании ограничений TCP для различных демонов, запускаемых посредством **inetd**.

### 23.2.6. Разное

**daytime**, **time**, **echo**, **discard**, **chargen** и **auth** все являются услугами, предоставляемыми самим **inetd**.

Сервис **auth** предоставляет идентификационные сетевые услуги (**ident**, **identd**) и поддается настройке.

Обратитесь к справочной странице по `inetd(8)` для получения более подробной информации.

## 23.3. Network File System (NFS)

*Реорганизация и улучшения Tom Rhodes. Текст создал Bill Swingle.*

Кроме поддержки многих прочих типов файловых систем, во FreeBSD встроена поддержка сетевой файловой системы (Network File System), известной как NFS. NFS позволяет системе использовать каталоги и файлы совместно с другими машинами, посредством сети. Посредством NFS пользователи и программы могут получать доступ к файлам на удалённых системах точно так же, как если бы это были файлы на собственных дисках.

Вот некоторые из наиболее заметных преимуществ, которые даёт использование NFS:

- Отдельно взятые рабочие станции используют меньше собственного дискового пространства, так как совместно используемые данные могут храниться на одной отдельной машине и быть доступными для других машин в сети.
- Пользователям не нужно иметь домашние каталоги, отдельные для каждой машины в вашей сети. Домашние каталоги могут располагаться на сервере NFS и их можно сделать доступными отовсюду в сети.
- Устройства хранения информации, такие, как дискеты, приводы CD-ROM и устройства Zip®, могут использоваться другими машинами в сети. Это может привести к уменьшению переносимых устройств хранения информации в сети.

### 23.3.1. Как работает NFS

NFS строится по крайней мере из двух основных частей: сервера и одного или большего количества клиентов. Клиент обращается к данным, находящимся на сервере, в режиме удалённого доступа. Для того, чтобы это нормально функционировало, нужно настроить и запустить несколько процессов.

**Замечание:** В FreeBSD 5.X, утилита **portmap** была заменена на утилиту **rpcbind**. Таким образом, при использовании FreeBSD 5.X пользователю необходимо заменить в последующих примерах все команды **portmap** на **rpcbind**.

На сервере работают следующие демоны:

Даemon	Описание
<b>nfsd</b>	Даemon NFS, обслуживающий запросы от клиентов NFS.
<b>mountd</b>	Даemon монтирования NFS, который выполняет запросы, передаваемые ему от nfsd(8).
<b>portmap</b>	Даemon отображения портов позволяет клиентам NFS определить порт, используемый сервером NFS.

Клиент может запустить также даemon, называемый **nfsiod**. **nfsiod** обслуживает запросы, поступающие от сервера от сервера NFS. Он необязателен, увеличивает производительность, однако для нормальной и правильной работы не требуется. Для получения дополнительной информации обратитесь к разделу справочной системы о nfsiod(8).

### 23.3.2. Настройка NFS

Настройка NFS является достаточно незамысловатым процессом. Все процессы, которые должны быть запущены, могут быть запущены во время загрузки посредством нескольких модификаций в вашем файле `/etc/rc.conf`.

Проверьте, что на NFS-сервере в файле `/etc/rc.conf` имеются такие строки:

```
portmap_enable="YES"
nfs_server_enable="YES"
nfs_server_flags="-u -t -n 4"
mountd_flags="-r"
```

**mountd** запускается автоматически, если включена функция сервера NFS.

На клиенте убедитесь, что в файле `/etc/rc.conf` присутствует такой параметр:

```
nfs_client_enable="YES"
```

Файл `/etc/exports` определяет, какие файловые системы на вашем сервере NFS будут экспортироваться (иногда их называют "совместно используемыми"). Каждая строка в `/etc/exports` задаёт файловую систему, которая будет экспортироваться и какие машины будут иметь к ней доступ. Кроме машин, имеющих доступ, могут задаваться другие параметры, влияющие на характеристики доступа. Имеется полный набор параметров, которые можно использовать, но здесь пойдёт речь лишь о некоторых из них. Описания остальных параметров можно найти на страницах справочной системы по exports(5).

Вот несколько примерных строк из файла `/etc/exports`:

В следующих примерах даётся общая идея того, как экспортировать файловые системы, хотя конкретные параметры могут отличаться в зависимости от ваших условий и конфигурации сети. К примеру, чтобы экспортировать каталог `/cdrom` для трёх машин, находящихся в том же самом домене, что и сервер (поэтому отсутствует доменное имя для каждой машины) или для которых имеются записи в файле `/etc/hosts`. Флаг `-ro` указывает на использование экспортируемой файловой системы в режиме только чтения. С этим флагом удалённая система не сможет никоим образом изменить экспортируемую файловую систему.

```
/cdrom -ro host1 host2 host3
```

В следующей строке экспортируется файловая система /home, которая становится доступной трем хостам, указанным по их IP-адресам. Это полезно, если у вас есть собственная сеть без настроенного сервера DNS. Как вариант, файл /etc/hosts может содержать внутренние имена хостов; пожалуйста, обратитесь к справочную систему по hosts(5) для получения дополнительной информации. Флаг -alldirs позволяет рассматривать подкаталоги в качестве точек монтирования. Другими словами, это не монтирование подкаталогов, но разрешение клиентам монтировать только каталоги, которые им требуются или нужны.

```
/home -alldirs 10.0.0.2 10.0.0.3 10.0.0.4
```

В строке, приведённой ниже, файловая система /a экспортируется таким образом, что она доступна двум клиентам из других доменов. Параметр -maproot=root позволяет пользователю root удалённой системы осуществлять запись на экспортируемую файловую систему как пользователь root. Если параметр -maproot=root не задан, то даже если пользователь имеет права доступа root на удалённой системе, он не сможет модифицировать файлы на экспортированной файловой системе.

```
/a -maproot=root host.example.com box.example.org
```

Для того, чтобы клиент смог обратиться к экспортированной файловой системе, он должен иметь права сделать это. Проверьте, что клиент указан в вашем файле /etc/exports.

В файле /etc/exports каждая строка содержит информацию об экспортировании для отдельной файловой системы для отдельно взятого хоста. Удалённый хост может быть задан только один раз для каждой файловой системы, и может иметь только одну запись, используемую по умолчанию, для каждой локальной файловой системы. К примеру, предположим, что /usr является отдельной файловой системой. Следующий /etc/exports будет некорректен:

```
# Invalid when /usr is one file system
/usr/src client
/usr/ports client
```

Одна файловая система, /usr, имеет две строки, задающие экспортирование для одного и того же хоста, client. Правильный формат в этом случае таков:

```
/usr/src /usr/ports client
```

Свойства отдельной файловой системы, экспортируемой некоторому хосту, должны задаваться в одной строке. Строки без указания клиента воспринимаются как отдельный хост. Это ограничивает то, как вы можете экспортировать файловые системы, но для большинства это не проблема.

Ниже приведён пример правильного списка экспортирования, где /usr и /exports являются локальными файловыми системами:

```
# Экспортируем src и ports для client01 и client02, но
# только client01 имеет права пользователя root на них
/usr/src /usr/ports -maproot=root client01
/usr/src /usr/ports client02
# Клиентские машины имеют пользователя root и могут монтировать всё в
# каталоге /exports. Кто угодно может монтировать /exports/obj в режиме чтения
/exports -alldirs -maproot=root client01 client02
/exports/obj -ro
```

Вы должны перезапустить **mountd** после того, как изменили `/etc/exports`, чтобы изменения вступили в силу. Это может быть достигнуто посылкой сигнала HUP процессу `mountd`:

```
# kill -HUP `cat /var/run/mountd.pid`
```

Как вариант, при перезагрузке FreeBSD всё настроится правильно. Хотя выполнять перезагрузку вовсе не обязательно. Выполнение следующих команд пользователем `root` запустит всё, что нужно.

На сервере NFS:

```
# portmap
# nfsd -u -t -n 4
# mountd -r
```

На клиенте NFS:

```
# nfsiod -n 4
```

Теперь всё должно быть готово к реальному монтированию удалённой файловой системы. В приводимых примерах сервер будет носить имя `server`, а клиент будет носить имя `client`. Если вы только хотите временно смонтировать удалённую файловую систему, или всего лишь протестировать ваши настройки, то просто запустите команды, подобные приводимым здесь, работая как пользователь `root` на клиентской машине:

```
# mount server:/home /mnt
```

По этой команде файловая система `/home` на сервере будет смонтирована в каталог `/mnt` на клиенте. Если всё настроено правильно, вы сможете войти в каталог `/mnt` на клиенте и увидеть файлы, находящиеся на сервере.

Если вы хотите автоматически монтировать удалённую файловую систему при каждой загрузке компьютера, добавьте файловую систему в `/etc/fstab`. Вот пример:

```
server:/home    /mnt    nfs    rw    0    0
```

На страницах справочной системы по `fstab(5)` перечислены все доступные параметры.

### 23.3.3. Практическое использование

У NFS есть много вариантов практического применения. Ниже приводится несколько наиболее широко распространённых способов её использования:

- Настройка несколько машин для совместного использования CDROM или других носителей. Это более дешёвый и зачастую более удобный способ установки программного обеспечения на несколько машин.
- В больших сетях может оказаться более удобным настроить центральный сервер NFS, на котором размещаются все домашние каталоги пользователей. Эти домашние каталоги могут затем экспортироваться в сеть так, что пользователи всегда будут иметь один и тот же домашний каталог вне зависимости от того, на какой рабочей станции они работают.



- Несколько машин могут иметь общий каталог `/usr/ports/distfiles`. Таким образом, когда вам нужно будет установить порт на несколько машин, вы сможете быстро получить доступ к исходным текстам без их загрузки на каждой машине.

### 23.3.4. Автоматическое монтирование с **amd**

*Текст предоставил Wylie Stilwell. Текст перепечатал Chern Lee.*

**amd(8)** (демон автоматического монтирования) автоматически монтирует удалённую файловую систему, как только происходит обращение к файлу или каталогу в этой файловой системе. Кроме того, файловые системы, которые были неактивны некоторое время, будут автоматически размонтированы демоном **amd**. Использование **amd** является простой альтернативой статическому монтированию, так как в последнем случае обычно всё должно быть описано в файле `/etc/fstab`.

**amd** работает, сам выступая как сервер NFS для каталогов `/host` и `/net`. Когда происходит обращение к файлу в одном из этих каталогов, **amd** ищет соответствующий удалённый ресурс для монтирования и автоматически его монтирует. `/net` используется для монтирования экспортируемой файловой системы по адресу IP, когда как каталог `/host` используется для монтирования ресурса по удалённому имени хоста.

Обращение к файлу в каталоге `/host/foobar/usr` укажет **amd** на выполнение попытки монтирования ресурса `/usr`, который находится на хосте `foobar`.

#### Пример 23-2. Монтирование ресурса при помощи **amd**

Вы можете посмотреть доступные для монтирования ресурсы отдалённого хоста командой `showmount`. К примеру, чтобы посмотреть ресурсы хоста с именем `foobar`, вы можете использовать:

```
% showmount -e foobar
Exports list on foobar:
/usr      10.10.10.0
/a 10.10.10.0
% cd /host/foobar/usr
```

Как видно из примера, `showmount` показывает `/usr` как экспортируемый ресурс. При переходе в каталог `/host/foobar/usr` демон **amd** пытается разрешить имя хоста `foobar` и автоматически смонтировать требуемый ресурс.

**amd** может быть запущен из скриптов начальной загрузки, если поместить такую строку в файл `/etc/rc.conf`:

```
amd_enable="YES"
```

Кроме того, демону **amd** могут быть переданы настроечные флаги через параметр `amd_flags`. По умолчанию `amd_flags` настроен следующим образом:

```
amd_flags="-a /.amd_mnt -l syslog /host /etc/amd.map /net /etc/amd.map"
```

Файл `/etc/amd.map` задает опции, используемые по умолчанию при монтировании экспортируемых ресурсов. В файле `/etc/amd.conf` заданы настройки некоторых более сложных возможностей **amd**.

Обратитесь к справочным страницам по `amd(8)` и `amd.conf(5)` для получения более полной информации.

### 23.3.5. Проблемы взаимодействия с другими системами

*Текст предоставил John Lind.*

Некоторые сетевые адаптеры для систем PC с шиной ISA имеют ограничения, которые могут привести к серьезным проблемам в сети, в частности, с NFS. Эти проблемы не специфичны для FreeBSD, однако эту систему они затрагивают.

Проблема, которая возникает практически всегда при работе по сети систем PC (FreeBSD) с высокопроизводительными рабочими станциями, выпущенными такими производителями, как Silicon Graphics, Inc. и Sun Microsystems, Inc. Монтирование по протоколу NFS будет работать нормально, и некоторые операции также будут выполняться успешно, но неожиданно сервер окажется недоступным для клиент, хотя запросы к и от других систем будут продолжаться обрабатываться. Такое встречается с клиентскими системами, не зависимо от того, является ли клиент машиной с FreeBSD или рабочей станцией. Во многих системах при возникновении этой проблемы нет способа корректно завершить работу клиента. Единственным выходом зачастую является холодная перезагрузка клиента, потому что ситуация с NFS не может быть разрешена.

Хотя "правильным" решением является установка более производительного и скоростного сетевого адаптера на систему FreeBSD, имеется простое решение, приводящее к удовлетворительным результатам. Если система FreeBSD является *сервером*, укажите параметр `-w=1024` на клиенте при монтировании. Если система FreeBSD является *клиентом*, то смонтируйте файловую систему NFS с параметром `-r=1024`. Эти параметры могут быть заданы в четвертом поле записи в файле `fstab` клиента при автоматическом монтировании, или при помощи параметра `-o` в команде `mount(8)` при монтировании вручную.

Нужно отметить, что имеется также другая проблема, ошибочно принимаемая за приведенную выше, когда серверы и клиенты NFS находятся в разных сетях. Если это тот самый случай, *проверьте*, что ваши маршрутизаторы пропускают нужную информацию UDP, в противном случае вы ничего не получите, что бы вы ни предпринимали.

В следующих примерах `fastws` является именем хоста (интерфейса) высокопроизводительной рабочей станции, а `freebox` является именем хоста (интерфейса) системы FreeBSD со слабым сетевым адаптером. Кроме того, `/sharedfs` будет являться экспортируемой через NFS файловой системой (обратитесь к страницам справочной системы по команде `exports(5)`), а `/project` будет точкой монтирования экспортируемой файловой системы на клиенте. В любом случае, отметьте, что для вашего приложения могут понадобиться дополнительные параметры, такие, как `hard`, `soft` или `bg`.

Пример системы FreeBSD (`freebox`) как клиента в файле `/etc/fstab` на машине `freebox`:

```
fastws:/sharedfs /project nfs rw,-r=1024 0 0
```

Команда, выдаваемая вручную на машине `freebox`:

```
# mount -t nfs -o -r=1024 fastws:/sharedfs /project
```

Пример системы FreeBSD в качестве сервера в файле `/etc/fstab` на машине `fastws`:

```
freebox:/sharedfs /project nfs rw,-w=1024 0 0
```

Команда, выдаваемая вручную на машине `fastws`:

```
# mount -t nfs -o -w=1024 freebox:/sharedfs /project
```

Практически все 16-разрядные сетевые адаптеры позволят работать без указанных выше ограничений на размер блоков при чтении и записи.

Для тех, кто интересуется, ниже описывается, что же происходит при появлении этой ошибки, и объясняется, почему ее невозможно устранить. Как правило, NFS работает с "блоками" размером 8 килобайт (хотя отдельные фрагменты могут иметь меньшие размеры). Так, пакет Ethernet имеет максимальный размер около 1500 байт, то "блок" NFS разбивается на несколько пакетов Ethernet, хотя на более высоком уровне это все тот же единый блок, который должен быть принят, собран и *подтвержден* как один блок. Высокопроизводительные рабочие станции могут посылать пакеты, которые соответствуют одному блоку NFS, сразу друг за другом, насколько это позволяет делать стандарт. На слабых, низкопроизводительных адаптерах пакеты, пришедшие позже, накладываются поверх ранее пришедших пакетов того же самого блока до того, как они могут быть переданы хосту и блок как единое целое не может быть собран или подтвержден. В результате рабочая станция входит в ситуацию таймаута и пытается повторить передачу, но уже с полным блоком в 8 КБ, и процесс будет повторяться снова, до бесконечности.

Задав размер блока меньше размера пакета Ethernet, мы достигаем того, что любой полностью полученный пакет Ethernet может быть подтвержден индивидуально, и избежим тупиковую ситуацию.

Наложение пакетов может все еще проявляться, когда высокопроизводительные рабочие станции сбрасывают данные на PC-систему, однако повторение этой ситуации не обязательно с более скоростными адаптерами с "блоками" NFS. Когда происходит наложение, затронутые блоки будут переданы снова, и скорее всего, они будут получены, собраны и подтверждены.

## 23.4. Network Information System (NIS/YP)

*Текст создал Bill Swingle. Внёс добавления Eric Ogren, Udo Erdelhoff.*

### 23.4.1. Что это такое?

NIS, что является сокращением от Network Information Services (Сетевые Информационные Службы), которые были разработаны компанией Sun Microsystems для централизованного администрирования систем UNIX (изначально SunOS). В настоящее время эти службы практически стали промышленным стандартом; все основные UNIX-подобные системы (Solaris, HP-UX, AIX®, Linux, NetBSD, OpenBSD, FreeBSD и так далее) поддерживают NIS.

NIS первоначально назывались Yellow Pages (или yp), но из-за проблем с торговым знаком Sun изменила это название. Старое название (и yp) всё ещё часто употребляется.

Это система клиент/сервер на основе вызовов RPC, которая позволяет группе машин в одном домене NIS совместно использовать общий набор конфигурационных файлов. Системный администратор может настроить клиентскую систему NIS только с минимальной настроечной информацией, а затем добавлять, удалять и модифицировать настроечную информацию из одного места.

Это похоже на систему доменов Windows NT®; хотя их внутренние реализации не так уж и похожи, основные функции сравнимы.

### 23.4.2. Термины/программы, о которых вы должны знать

Существует несколько терминов и некоторое количество пользовательских программ, которые будут нужны, когда вы будете пытаться сделать NIS во FreeBSD, и в случае создания сервера, и в случае работы в качестве клиента NIS:

Термин	Описание
Имя домена NIS	Главный сервер NIS и все его клиенты (включая вторичные серверы), имеют доменное имя NIS. Как и в случае с именем домена Windows NT, имя домена NIS не имеет ничего общего с DNS.
<b>portmap</b>	Для обеспечения работы RPC (Remote Procedure Call, Удалённого Вызова Процедур, сетевого протокола, используемого NIS), должен быть запущен демон <b>portmap</b> . Если демон <b>portmap</b> не запущен, невозможно будет запустить сервер NIS, или работать как NIS-клиент.
<b>ybind</b>	”Связывает” NIS-клиента с его NIS-сервером. Он определяет имя NIS-домена системы, и при помощи RPC подключается к серверу. <b>ybind</b> является основой клиент-серверного взаимодействия в среде NIS; если на клиентской машине программа <b>ybind</b> перестанет работать, то эта машина не сможет получить доступ к серверу NIS.
<b>ypserv</b>	Программа <b>ypserv</b> , которая должна запускаться только на серверах NIS: это и есть сервер NIS. Если <b>ypserv(8)</b> перестанет работать, то сервер не сможет отвечать на запросы NIS (к счастью, на этот случай предусмотрен вторичный сервер). Есть несколько реализаций NIS (к FreeBSD это не относится), в которых не производится попыток подключиться к другому серверу, если ранее используемый сервер перестал работать. Зачастую единственным средством, помогающим в этой ситуации, является перезапуск серверного процесса (или сервера полностью) или процесса <b>ybind</b> на клиентской машине.
<b>rpc.yppasswdd</b>	Программа <b>rpc.yppasswdd</b> , другой процесс, который запускается только на главных NIS-серверах: это демон, позволяющий клиентам NIS изменять свои пароли NIS. Если этот демон не запущен, то пользователи должны будут входить на основной сервер NIS и там менять свои пароли.

### 23.4.3. Как это работает?

В системе NIS существует три типа хостов: основные (master) серверы, вторичные (slave) серверы и клиентские машины. Серверы выполняют роль централизованного хранилища информации о конфигурации хостов. Основные серверы хранят оригиналы этой информации, когда как вторичные серверы хранят ее копию для обеспечения избыточности. Клиенты связываются с серверами, чтобы предоставить им эту информацию.

Информация во многих файлах может совместно использоваться следующим образом. Файлы **master.passwd**, **group** и **hosts** используются совместно через NIS. Когда процессу, работающему на клиентской машине, требуется информация, как правило, находящаяся в этих файлах локально, то он делает запрос к серверу NIS, с которым связан.

### 23.4.3.1. Типы машин

- *Основной сервер NIS.* Такой сервер, по аналогии с первичным контроллером домена Windows NT, хранит файлы, используемые всеми клиентами NIS. Файлы `passwd`, `group` и различные другие файлы, используемые клиентами NIS, находятся на основном сервере.

**Замечание:** Возможно использование одной машины в качестве сервера для более чем одного домена NIS. Однако, в этом введении такая ситуация не рассматривается, и предполагается менее масштабное использование NIS.

- *Вторичные серверы NIS.* Похожие на вторичные контроллеры доменов Windows NT, вторичные серверы NIS содержат копии оригинальных файлов данных NIS. Вторичные серверы NIS обеспечивают избыточность, что нужно в критичных приложениях. Они также помогают распределять нагрузку на основной сервер: клиенты NIS всегда подключаются к тому серверу NIS, который ответил первым, в том числе и к вторичным серверам.
- *Клиенты NIS.* Клиенты NIS, как и большинство рабочих станций Windows NT, аутентифицируются на сервере NIS (или на контроллере домена Windows NT для рабочих станций Windows NT) во время входа в систему.

### 23.4.4. Использование NIS/YP

В этом разделе приводится пример настройки NIS.

**Замечание:** В этом разделе предполагается, что вы работаете с FreeBSD 3.3 или выше. Указания, приводимые здесь, скорее всего, *будут* работать с любой версией FreeBSD, выше, чем 3.0, однако нет гарантий, что это на самом деле так.

#### 23.4.4.1. Планирование

Давайте предположим, что вы являетесь администратором в маленькой университетской лаборатории. В настоящий момент в этой лаборатории с 15 машинами отсутствует единая точка администрирования; на каждой машине имеются собственные файлы `/etc/passwd` и `/etc/master.passwd`. Эти файлы синхронизируются друг с другом только вручную; сейчас, когда вы добавляете пользователя в лаборатории, вы должны выполнить команду `adduser` на всех 15 машинах. Понятно, что такое положение вещей нужно исправлять, так что вы решили перевести сеть на использование NIS, используя две машины в качестве серверов.

Итак, конфигурация лаборатории сейчас выглядит примерно так:

Имя машины	IP-адрес	Роль машины
ellington	10.0.0.2	Основной сервер NIS
coltrane	10.0.0.3	Вторичный сервер NIS

Имя машины	IP-адрес	Роль машины
basie	10.0.0.4	Факультетская рабочая станция
bird	10.0.0.5	Клиентская машина
cli[1-11]	10.0.0.[6-17]	Другие клиентские машины

Если вы определяете схему NIS первый раз, ее нужно хорошо обдумать. Вне зависимости от размеров вашей сети, есть несколько ключевых моментов, которые требуют принятия решений.

#### 23.4.4.1.1. Выбор имени домена NIS

Это имя не должно быть "именем домена", которое вы использовали. Более точно это имя называется "именем домена NIS". Когда клиент рассылает запросы на получение информации, он включает в них имя домена NIS, частью которого является. Таким способом многие сервера в сети могут указать, какой сервер на какой запрос должен отвечать. Думайте о домене NIS как об имени группы хостов, которые каким-то образом связаны.

Некоторые организации в качестве имени домена NIS используют свой домен Интернет. Это не рекомендуется, так как может вызвать проблемы в процессе решения сетевых проблем. Имя домена NIS должно быть уникальным в пределах вашей сети и хорошо, если оно будет описывать группу машин, которые представляет. Например, художественный отдел в компании Acme Inc. может находиться в домене NIS с именем "acme-art". В нашем примере положим, что мы выбрали имя test-domain.

Несмотря на это, некоторые операционные системы (в частности, SunOS) используют свое имя домена NIS в качестве имени домена Интернет. Если одна или более машин в вашей сети имеют такие ограничения, вы *обязаны* использовать имя домена Интернет в качестве имени домена NIS.

#### 23.4.4.1.2. Требования к серверу

Есть несколько вещей, которые нужно иметь в виду при выборе машины для использования в качестве сервера NIS. Одной из обескураживающей вещью, касающейся NIS, является уровень зависимости клиентов от серверов. Если клиент не может подключиться к серверу своего домена NIS, зачастую машину просто становится нельзя использовать. Отсутствие информации о пользователях и группах приводит к временной остановке работы большинства систем. Зная это, вы должны выбрать машину, которая не должна подвергаться частым перезагрузкам и не используется для разработки. Сервер NIS в идеале должен быть отдельно стоящей машиной, единственным целью в жизни которой является быть сервером NIS. Если вы работаете в сети, которая не так уж сильно загружена, то можно поместить сервер NIS на машине, на которой запущены и другие сервисы, просто имейте в виду, что если сервер NIS становится недоступным, то это негативно отражается на *всех* клиентах NIS.

#### 23.4.4.2. Серверы NIS

Оригинальные копии всей информации NIS хранятся на единственной машине, которая называется главным сервером NIS. Базы данных, которые используются для хранения информации, называются картами NIS. Во FreeBSD эти карты хранятся в /var/yp/[domainname], где [domainname] является именем обслуживаемого домена NIS. Один сервер NIS может поддерживать одновременно несколько доменов, так что есть возможность иметь несколько таких каталогов, по одному на каждый

обслуживаемый домен. Каждый домен будет иметь свой собственный независимый от других набор карт.

Основной и вторичный серверы обслуживают все запросы NIS с помощью демона `yrserv`. `yrserv` отвечает за получение входящих запросов от клиентов NIS, распознавание запрашиваемого домена и отображение имени в путь к соответствующему файлу базы данных, а также передаче информации из базы данных обратно клиенту.

#### 23.4.4.2.1. Настройка основного сервера NIS

Настройка основного сервера NIS может оказаться сравнительно простой, в зависимости от ваших потребностей. В поставку FreeBSD сразу включена поддержка NIS. Все, что вам нужно, это добавить следующие строки в файл `/etc/rc.conf`, а FreeBSD сделает за вас всё остальное..

1.

```
nisdomainname="test-domain"
```

В этой строке задается имя домена NIS, которое будет `test-domain`, еще до настройки сети (например, после перезагрузки).

2.

```
nis_server_enable="YES"
```

Здесь указывается FreeBSD на запуск процессов серверов NIS, когда дело доходит до сетевых настроек.

3.

```
nis_yppasswdd_enable="YES"
```

Здесь указывается на запуск демона `rpc.yppasswdd`, который, как это отмечено выше, позволит пользователям менять свой пароль NIS с клиентской машины.

**Замечание:** В зависимости от ваших настроек NIS, вам могут понадобиться дополнительные строки. Обратитесь к разделу о серверах NIS, которые являются и клиентами NIS ниже для получения подробной информации.

А теперь всё, что вам нужно сделать, это запустить команду `/etc/netstart`, работая как администратор. По ней произойдет настройка всего, при этом будут использоваться значения, заданные в файле `/etc/rc.conf`.

#### 23.4.4.2.2. Инициализация карт NIS

*Карты NIS* являются файлами баз данных, которые хранятся в каталоге `/var/yp`. Они генерируются из конфигурационных файлов, находящихся в каталоге `/etc` основного сервера NIS, за одним исключением: файл `/etc/master.passwd`. На это есть весомая причина, вам не нужно распространять пароли пользователя `root` и других административных пользователей на все серверы в домене NIS. По этой причине, прежде чем инициализировать карты NIS, вы должны сделать вот что:

```
# cp /etc/master.passwd /var/yp/master.passwd
```

```
# cd /var/yp
# vi master.passwd
```

Вы должны удалить все записи, касающиеся системных пользователей (bin, tty, kmem, games и так далее), а также записи, которые вы не хотите распространять клиентам NIS (например, root и другие пользователи с UID, равным 0 (администраторы)).

**Замечание:** Проверьте, чтобы файл /var/yp/master.passwd был недоступен для записи ни для группы, ни для остальных пользователей (режим доступа 600)! Воспользуйтесь командой `chmod`, если это нужно.

Когда с этим будет покончено, самое время инициализировать карты NIS! В поставку FreeBSD включен скрипт с именем `ypinit`, который делает это (обратитесь к его справочной странице за дополнительной информацией). Отметьте, что этот скрипт имеется в большинстве операционных систем UNIX, но не во всех. В системе Digital Unix/Compaq Tru64 UNIX он называется `ypsetup`. Так как мы генерируем карты для главного сервера NIS, то при вызове программы `ypinit` мы передаем ей параметр `-m`. Для генерации карт NIS в предположении, что вы уже сделали шаги, описанные выше, выполните следующее:

```
ellington# ypinit -m test-domain
Server Type: MASTER Domain: test-domain
Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
Do you want this procedure to quit on non-fatal errors? [y/n: n] n
Ok, please remember to go back and redo manually whatever fails.
If you don't, something might not work.
At this point, we have to construct a list of this domains YP servers.
rod.darktech.org is already known as master server.
Please continue to add any slave servers, one per line. When you are
done with the list, type a <control D>.
master server : ellington
next host to add: coltrane
next host to add: ^D
The current list of NIS servers looks like this:
ellington
coltrane
Is this correct? [y/n: y] y
```

[..вывод при генерации карт..]

```
NIS Map update completed.
ellington has been setup as an YP master server without any errors.
```

Программа `ypinit` должна была создать файл /var/yp/Makefile из /var/yp/Makefile.dist. При создании этого файла предполагается, что вы работаете в окружении с единственным сервером NIS и только с машинами FreeBSD. Так как в домене test-domain имеется также и вторичный сервер, то вы должны отредактировать файл /var/yp/Makefile:

```
ellington# vi /var/yp/Makefile
```

Вы должны закомментировать строку, в которой указано



```
NOPUSH = "True"
```

(она уже не раскомментирована).

#### 23.4.4.2.3. Настройка вторичного сервера NIS

Настройка вторичного сервера NIS осуществляется ещё проще, чем настройка главного сервера. Войдите на вторичный сервер и отредактируйте файл `/etc/rs.conf` точно также, как вы делали это ранее. Единственным отличием является то, что при запуске программы `ypinit` мы теперь должны использовать опцию `-s`. Применение опции `-s` требует также указание имени главного сервера NIS, так что наша команда должна выглядеть так:

```
coltrane# ypinit -s ellington test-domain
```

```
Server Type: SLAVE Domain: test-domain Master: ellington
```

```
Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
```

```
Do you want this procedure to quit on non-fatal errors? [y/n: n]  n
```

```
Ok, please remember to go back and redo manually whatever fails.
```

```
If you don't, something might not work.
```

```
There will be no further questions. The remainder of the procedure
should take a few minutes, to copy the databases from ellington.
```

```
Transferring netgroup...
```

```
ypxfr: Exiting: Map successfully transferred
```

```
Transferring netgroup.byuser...
```

```
ypxfr: Exiting: Map successfully transferred
```

```
Transferring netgroup.byhost...
```

```
ypxfr: Exiting: Map successfully transferred
```

```
Transferring master.passwd.byuid...
```

```
ypxfr: Exiting: Map successfully transferred
```

```
Transferring passwd.byuid...
```

```
ypxfr: Exiting: Map successfully transferred
```

```
Transferring passwd.byname...
```

```
ypxfr: Exiting: Map successfully transferred
```

```
Transferring group.bygid...
```

```
ypxfr: Exiting: Map successfully transferred
```

```
Transferring group.byname...
```

```
ypxfr: Exiting: Map successfully transferred
```

```
Transferring services.byname...
```

```
ypxfr: Exiting: Map successfully transferred
```

```
Transferring rpc.bynumber...
```

```
ypxfr: Exiting: Map successfully transferred
```

```
Transferring rpc.byname...
```

```
ypxfr: Exiting: Map successfully transferred
```

```
Transferring protocols.byname...
```

```
ypxfr: Exiting: Map successfully transferred
```

```
Transferring master.passwd.byname...
```

```
ypxfr: Exiting: Map successfully transferred
```

```
Transferring networks.byname...
```

```

ypxfr: Exiting: Map successfully transferred
Transferring networks.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring netid.byname...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring ypservers...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byname...
ypxfr: Exiting: Map successfully transferred

```

coltrane has been setup as an YP slave server without any errors.  
Don't forget to update map ypservers on ellington.

Теперь у вас должен быть каталог с именем `/var/yp/test-domain`. Копии карт главного сервера NIS должны быть в этом каталоге. Вы должны удостовериться, что этот каталог обновляется. Следующие строки в `/etc/crontab` вашего вторичного сервера должны это делать:

```

20 * * * * root    /usr/libexec/ypxfr passwd.byname
21 * * * * root    /usr/libexec/ypxfr passwd.byuid

```

Эти две строки заставляют вторичный сервер синхронизировать свои карты с картами главного сервера. Хотя эти строчки не обязательны, так как главный сервер делает попытки передать все изменения в своих картах NIS на свои вторичные серверы, но из-за того, что информация для входа в систему настолько жизненно важна для систем, зависящих от сервера, что выполнение регулярных обновлений является совсем не плохой идеей. Это ещё более важно в загруженных сетях, в которых обновления карт могут не всегда завершаться успешно.

А теперь точно также запустите команду `/etc/netstart` на вторичном сервере, по которой снова выполнится запуск сервера NIS.

#### 23.4.4.3. Клиенты NIS

Клиент NIS выполняет так называемую привязку к конкретному серверу NIS при помощи демона `yrbind`. `yrbind` определяет домен, используемый в системе по умолчанию (тот, который устанавливается по команде `domainname`), и начинает широковещательную рассылку запросов RPC в локальной сети. В этих запросах указано имя домена, к серверу которого `yrbind` пытается осуществить привязку. Если сервер, который был настроен для обслуживания запрашиваемого домена, получит широковещательный запрос, он ответит `yrbind`, который, в свою очередь запомнит адрес сервера. Если имеется несколько серверов (например, главный и несколько вторичных), то `yrbind` будет использовать адрес первого ответившего. С этого момента клиентская система будет направлять все свои запросы NIS на этот сервер. Время от времени `yrbind` будет "пинать" сервер для проверки его работоспособности. Если на один из тестовых пакетов не удастся получить ответа за разумное время, то `yrbind` пометит этот домен как домен, с которым связь разорвана, и снова начнет процесс посылки широковещательных запросов в надежде найти другой сервер.

### 23.4.4.3.1. Настройка клиента NIS

Настройка машины с FreeBSD в качестве клиента NIS достаточно проста.

1. Отредактируйте файл `/etc/rc.conf`, добавив туда следующие строки для того, чтобы задать имя домена NIS и запустить `yrbind` во время запуска сетевых служб:

```
nisdomainname="test-domain"
nis_client_enable="YES"
```

2. Для импортирования всех возможных учётных записей от сервера NIS, удалите все записи пользователей из вашего файла `/etc/master.passwd` и воспользуйтесь командой `vipw` для добавления следующей строки в конец файла:

```
+:::~::~:
```

**Замечание:** Эта строчка даст всем пользователям с корректной учетной записью в картах учетных баз пользователей доступ к этой системе. Есть множество способов настроить ваш клиент NIS, изменив эту строку. Посмотрите ниже текст, касающийся сетевых групп, чтобы получить более подробную информацию. Дополнительная информация для изучения находится в книге издательства O'Reilly под названием *Managing NFS and NIS*.

**Замечание:** Вы должны оставить хотя бы одну локальную запись (то есть не импортировать ее через NIS) в вашем `/etc/master.passwd` и эта запись должна быть также членом группы `wheel`. Если с NIS что-то случится, эта запись может использоваться для удаленного входа в систему, перехода в режим администратора и исправления неисправностей.

3. Для импортирования всех возможных записей о группах с сервера NIS, добавьте в ваш файл `/etc/group` такую строчку:

```
+:*:::
```

После завершения выполнения этих шагов у вас должно получиться запустить команду `yrcat passwd` и увидеть карту учетных записей сервера NIS.

## 23.4.5. Безопасность NIS

В общем-то любой пользователь, зная имя вашего домена, может выполнить запрос RPC к `yrserv(8)` и получить содержимое ваших карт NIS. Для предотвращения такого неавторизованного обмена `yrserv(8)` поддерживает так называемую систему "securenets", которая может использоваться для ограничения доступа к некоторой группе хостов. При запуске `yrserv(8)` будет пытаться загрузить информацию, касающуюся `securenets`, из файла `/var/yp/securenets`.

**Замечание:** Имя каталога зависит от параметра, указанного вместе с опцией `-p`. Этот файл содержит записи, состоящие из указания сети и сетевой маски, разделенных пробелом. Строчки, начинающиеся со знака "#", считаются комментариями. Примерный файл `securenets` может иметь примерно такой вид:

```
# allow connections from local host -- mandatory
127.0.0.1      255.255.255.255
# allow connections from any host
# on the 192.168.128.0 network
192.168.128.0 255.255.255.0
# allow connections from any host
# between 10.0.0.0 to 10.0.15.255
# this includes the machines in the testlab
10.0.0.0      255.255.240.0
```

Если `ypserv(8)` получает запрос от адреса, который соответствует одному из этих правил, он будет обрабатывать запрос обычным образом. Если же адрес не подпадает ни под одно правило, запрос будет проигнорирован и в журнал будет записано предупреждающее сообщение. Если файл `/var/yp/securenets` не существует, `ypserv` будет обслуживать соединения от любого хоста.

Программа `ypserv` также поддерживает пакет программ **tcpwrapper** от Wietse Venema. Это позволяет администратору для ограничения доступа вместо `/var/yp/securenets` использовать конфигурационные файлы **tcpwrapper**.

**Замечание:** Хотя оба этих метода управления доступом обеспечивают некоторую безопасность, они, как основанные на проверке привилегированного порта, оба подвержены атакам типа "IP spoofing". Весь сетевой трафик, связанный с работой NIS, должен блокироваться вашим брандмауэром.

Серверы, использующие файл `/var/yp/securenets`, могут быть не в состоянии обслуживать старых клиентов NIS с древней реализацией протокола TCP/IP. Некоторые из этих реализаций при рассылке широковещательных запросов устанавливают все биты машинной части адреса в ноль и/или не в состоянии определить маску подсети при вычислении адреса широковещательной рассылки. Хотя некоторые из этих проблем могут быть решены изменением конфигурации клиента, другие могут привести к отказу от использования `/var/yp/securenets`.

Использование `/var/yp/securenets` на сервере с такой архаичной реализацией TCP/IP является весьма плохой идеей, и приведёт к потере работоспособности NIS в большей части вашей сети.

Использование пакета **tcpwrapper** увеличит время отклика вашего сервера NIS. Дополнительной задержки может оказаться достаточно для возникновения таймаутов в клиентских программах, особенно в загруженных сетях или с медленными серверами NIS. Если одна или более ваших клиентских систем страдают от таких проблем, вы должны преобразовать такие клиентские системы во вторичные серверы NIS и сделать принудительную их привязку к самим себе.

### 23.4.6. Запрет входа некоторых пользователей

В нашей лаборатории есть машина `basie`, о которой предполагается, что она является исключительно факультетской рабочей станцией. Мы не хотим исключать эту машину из домена NIS, однако файл `passwd` на главном сервере NIS содержит учетные записи как для работников факультета, так и студентов. Что мы можем сделать?

Есть способ ограничить вход некоторых пользователей на этой машине, даже если они присутствуют в базе данных NIS. Чтобы это сделать, вам достаточно добавить `-username` в конец файла `/etc/master.passwd` на клиентской машине, где `username` является именем пользователя, которому

вы хотите запретить вход. Рекомендуется сделать это с помощью утилиты `vipw`, так как `vipw` проверит ваши изменения в `/etc/master.passwd`, а также автоматически перестроит базу данных паролей по окончании редактирования. Например, если мы хотим запретить пользователю `bill` осуществлять вход на машине `basie`, то мы сделаем следующее:

```
basie# vipw
[add -bill to the end, exit]
vipw: rebuilding the database...
vipw: done

basie# cat /etc/master.passwd

root:[password]:0:0:0:0:The super-user:/root:/bin/csh
toor:[password]:0:0:0:0:The other super-user:/root:/bin/sh
daemon:*:1:1:0:0:Owner of many system processes:/root:/sbin/nologin
operator:*:2:5:0:0:System &:/sbin/nologin
bin:*:3:7:0:0:Binaries Commands and Source,,:/sbin/nologin
tty:*:4:65533:0:0:Tty Sandbox:/sbin/nologin
kmem:*:5:65533:0:0:KMem Sandbox:/sbin/nologin
games:*:7:13:0:0:Games pseudo-user:/usr/games:/sbin/nologin
news:*:8:8:0:0:News Subsystem:/sbin/nologin
man:*:9:9:0:0:Mister Man Pages:/usr/share/man:/sbin/nologin
bind:*:53:53:0:0:Bind Sandbox:/sbin/nologin
uucp:*:66:66:0:0:UUCP pseudo-user:/var/spool/uucppublic:/usr/libexec/uucp/uucico
xten:*:67:67:0:0:X-10 daemon:/usr/local/xten:/sbin/nologin
pop:*:68:6:0:0:Post Office Owner:/nonexistent:/sbin/nologin
nobody:*:65534:65534:0:0:Unprivileged user:/nonexistent:/sbin/nologin
+:::
-bill

basie#
```

### 23.4.7. Использование сетевых групп

*Текст предоставил Udo Erdelhoff.*

Способ, описанный в предыдущем разделе, работает достаточно хорошо, если вам нужны особые правила для очень малой группы пользователей или машин. В более крупных сетях вы *забудете* о запрете входа определенных пользователей на важные машины или даже будете настраивать каждую машину по отдельности, теряя таким образом главное преимущество использования NIS: *централизованное* администрирование.

Ответом разработчиков NIS на эту проблему являются *сетевые группы*. Их назначение и смысл можно сравнить с обычными группами, используемыми в файловых системах UNIX. Главное отличие заключается в отсутствии числового идентификатора и возможности задать сетевую группу включением как пользователей, так и других сетевых групп.

Сетевые группы были разработаны для работы с большими, сложными сетями с сотнями пользователей и машин. С одной стороны, хорошо, если вам приходится с такой ситуацией. С другой стороны, эта сложность делает невозможным описание сетевых групп с помощью простых примеров. Пример, используемый в дальнейшем, демонстрирует эту проблему.

Давайте предположим, что успешное внедрение системы NIS в вашей лаборатории заинтересовало ваше руководство. Вашим следующим заданием стало расширение домена NIS для включения в него некоторых других машин студенческого городка. В двух таблицах перечислены имена новых машин и пользователей, а также их краткое описание.

Имена пользователей	Описание
alpha, beta	Обычные служащие IT-департамента
charlie, delta	Практиканты IT-департамента
echo, foxtrott, golf, ...	Обычные сотрудники
able, baker, ...	Проходящие интернатуру

Имена машин	Описание
war, death, famine, pollution	Ваши самые важные серверы. Только служащим IT позволяется входить на эти машины.
pride, greed, envy, wrath, lust, sloth	Менее важные серверы. Все сотрудники департамента IT могут входить на эти машины.
one, two, three, four, ...	Обычные рабочие станции. Только <i>реально нанятым</i> служащим позволяется использовать эти машины.
trashcan	Очень старая машина без каких-либо критичных данных. Даже проходящим интернатуру разрешено ее использовать.

Если вы попытаетесь реализовать эти требования, ограничивая каждого пользователя по отдельности, то вам придется добавить на каждой машине в файл `passwd` по одной строчке `-user` для каждого пользователя, которому запрещено входить на эту систему. Если вы забудете даже одну строчку, у вас могут начаться проблемы. Гораздо проще делать это правильно во время начальной установки, однако вы постепенно *будете забывать* добавлять строчки для новых пользователей во время повседневной работы. В конце концов, Мерфи был оптимистом.

Использование в этой ситуации сетевых групп дает несколько преимуществ. Нет необходимости описывать по отдельности каждого пользователя; вы ставите в соответствие пользователю одну или несколько сетевых групп и разрешаете или запрещаете вход всем членам сетевой группы. Если вы добавляете новую машину, вам достаточно определить ограничения на вход для сетевых групп. Если добавляется новый пользователь, вам достаточно добавить его к одной или большему числу сетевых групп. Эти изменения независимы друг от друга: нет больше комбинаций "для каждого пользователя и машины". Если настройка вашей системы NIS тщательно спланирована, то для разрешения или запрещения доступа к машинам вам нужно будет модифицировать единственный конфигурационный файл.

Первым шагом является инициализация карты NIS по имени `netgroup`. Программа `ypinit(8)` во FreeBSD по умолчанию этой карты не создаёт, хотя реализация NIS будет её поддерживать, как только она будет создана. Чтобы создать пустую карту, просто наберите

```
ellington# vi /var/yp/netgroup
```

и начните добавлять содержимое. Например, нам нужно по крайней мере четыре сетевых группы: сотрудники IT, практиканты IT, обычные сотрудники и интернатура.

```
IT_EMP (,alpha,test-domain) (,beta,test-domain)
IT_APP (,charlie,test-domain) (,delta,test-domain)
USERS (,echo,test-domain) (,foxtrott,test-domain) \
(,golf,test-domain)
INTERNS (,able,test-domain) (,baker,test-domain)
```

IT\_EMP, IT\_APP и так далее являются именами сетевых групп. Несколько слов в скобках служат для добавления пользователей в группу. Три поля внутри группы обозначают следующее:

1. Имя хоста или хостов, к которым применимы последующие записи. Если имя хоста не указано, то запись применяется ко всем хостам. Если же указывается имя хоста, то вы получите мир темноты, ужаса и страшной путаницы.
2. Имя учетной записи, которая принадлежит этой сетевой группе.
3. Домен NIS для учетной записи. Вы можете импортировать в вашу сетевую группу учетные записи из других доменов NIS, если вы один из тех несчастных, имеющих более одного домена NIS.

Каждое из этих полей может содержать шаблоны, подробности даны в странице справочника по netgroup(5).

**Замечание:** Не нужно использовать имена сетевых групп длиннее 8 символов, особенно если в вашем домене NIS имеются машины, работающие под управлением других операционных систем. Имена чувствительны к регистру; использование заглавных букв для имен сетевых групп облегчает распознавание пользователей, имен машин и сетевых групп.

Некоторые клиенты NIS (отличные от FreeBSD) не могут работать с сетевыми группами, включающими большое количество записей. Например, в некоторых старых версиях SunOS возникают проблемы, если сетевая группа содержит более 15 *записей*. Вы можете обойти это ограничение, создав несколько подгрупп с 15 или меньшим количеством пользователей и настоящую сетевую группу, состоящую из подгрупп:

```
BIGGRP1 (,joe1,domain) (,joe2,domain) (,joe3,domain) [...]
BIGGRP2 (,joe16,domain) (,joe17,domain) [...]
BIGGRP3 (,joe31,domain) (,joe32,domain)
BIGGROUP BIGGRP1 BIGGRP2 BIGGRP3
```

Вы можете повторить этот процесс, если вам нужно иметь более 225 пользователей в одной сетевой группе.

Активация и распространение вашей карты NIS проста:

```
ellington# cd /var/yp
ellington# make
```

Это приведет к созданию трех карт NIS netgroup, netgroup.byhost и netgroup.byuser. Воспользуйтесь утилитой ypcat(1) для проверки доступности ваших новых карт NIS:

```
ellington% ypcat -k netgroup
ellington% ypcat -k netgroup.byhost
ellington% ypcat -k netgroup.byuser
```

Вывод первой команды должен соответствовать содержимому файла /var/yp/netgroup. Вторая команда не выведет ничего, если вы не зададите сетевые группы, специфичные для хоста. Третья команда может использоваться пользователем для получения списка сетевых групп.

Настройка клиента достаточно проста. Чтобы настроить сервер war, вам достаточно запустить `virw(8)` и заменить строку

$$+ \begin{array}{cccccccc} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{array}$$

Ha

+@IT\_EMP: : : : : : : :

Теперь только данные, касающиеся пользователей, определенных в сетевой группе IT\_ЕМР, импортируются в базу паролей машины war и только этим пользователям будет разрешен вход.

К сожалению, это ограничение также касается и функции `~` командного процессора и всех подпрограмм, выполняющих преобразование между именами пользователей и их числовыми ID. Другими словами, команда `cd ~user` работать не будет, команда `ls -l` будет выдавать числовые идентификаторы вместо имён пользователей, а `find . -user joe -print` работать откажется, выдавая сообщение `No such user`. Чтобы это исправить, вам нужно будет выполнить импорт всех записей о пользователях *без разрешения на вход на ваши серверы*.

Это можно сделать, добавив еще одну строку в файл `/etc/master.passwd`. Эта строка должна содержать:

+:::/:sbin/nologin, что означает "Произвести импортирование всех записей с заменой командного процессора на /sbin/nologin в импортируемых записях". Вы можете заменить любое поле в строке с паролем, указав значение по умолчанию в вашем /etc/master.passwd.

**Внимание:** Проверьте, что строка `++++++:/sbin/nologin` помещена после `@IT_EMP:++++++`. В противном случае все пользовательские записи, импортированные из NIS, будут иметь `/sbin/nologin` в качестве оболочки.

После этого изменения при появлении нового сотрудника ИТ вам будет достаточно изменять только одну карту NIS. Вы можете применить подобный метод для менее важных серверов, заменяя старую строку +::: в их файлах `/etc/master.passwd` на нечто, подобное следующему:

```
+@IT_EMP:::::::::
+@IT_APP:::::::::
+:::::::::/sbin/nologin
```

Соответствующие строки для обычных рабочих станций могут иметь такой вид:

```
+@IT_EMP:::::::::
+@USERS:::::::::
+:::::::::/sbin/nologin
```

И все было прекрасно до того момента, когда через несколько недель изменилась политика: Департамент ИТ начал нанимать интернатуру. Интернатуре в ИТ позволили использовать обычные рабочие станции и менее важные серверы; практикантам позволили входить на главные серверы. Вы создали новую сетевую группу IT\_INTERN, добавили в нее новую интернатуру и начали изменять настройки на всех и каждой машине... Как говорит старая мудрость: "Ошибки в централизованном планировании приводят к глобальному хаосу".



Возможность в NIS создавать сетевые группы из других сетевых групп может использоваться для предотвращения подобных ситуаций. Одним из вариантов является создание сетевых групп на основе ролей. Например, вы можете создать сетевую группу с именем BIGSRV для задания ограничений на вход на важные серверы, другую сетевую группу с именем SMALLSRV для менее важных серверов и третью сетевую группу под названием USERBOX для обычных рабочих станций. Каждая из этих сетевых групп содержит сетевые группы, которым позволено входить на эти машины. Новые записи для вашей карты NIS сетевой группы должны выглядеть таким образом:

```
BIGSRV    IT_EMP    IT_APP
SMALLSRV  IT_EMP    IT_APP    ITINTERN
USERBOX   IT_EMP    ITINTERN  USERS
```

Этот метод задания ограничений на вход работает весьма хорошо, если вы можете выделить группы машин с одинаковыми ограничениями. К сожалению, такая ситуация может быть исключением, но не правилом. В большинстве случаев вам нужна возможность определять ограничения на вход индивидуально для каждой машины.

Задание сетевых групп в зависимости от машин является другой возможностью, которой можно воспользоваться при изменении политики, описанной выше. При таком развитии событий файл `/etc/master.passwd` на каждой машине содержит две строки, начинающиеся с `+`. Первая из них добавляет сетевую группу с учётными записями, которым разрешено входить на эту машину, а вторая добавляет все оставшиеся учетные записи с `/sbin/nologin` в качестве командного процессора. Хорошей идеей является использование **ИМЕНИ МАШИНЫ** заглавными буквами для имени сетевой группы. Другими словами, строки должны иметь такой вид:

```
+@BOXNAME:::::::::
+:::::::::/sbin/nologin
```

Как только вы завершите эту работу для всех ваших машин, вам не нужно будет снова модифицировать локальные версии `/etc/master.passwd`. Все будущие изменения могут быть выполнены изменением карты NIS. Вот пример возможной карты сетевой группы для этого случая с некоторыми полезными дополнениями:

```
# Сначала определяем группы пользователей
IT_EMP    (,alpha,test-domain)    (,beta,test-domain)
IT_APP    (,charlie,test-domain)   (,delta,test-domain)
DEPT1     (,echo,test-domain)      (,foxtrott,test-domain)
DEPT2     (,golf,test-domain)       (,hotel,test-domain)
DEPT3     (,india,test-domain)      (,juliet,test-domain)
ITINTERN  (,kilo,test-domain)       (,lima,test-domain)
D_INTERNS (,able,test-domain)       (,baker,test-domain)
#
# Теперь задаем несколько групп на основе ролей
USERS     DEPT1     DEPT2     DEPT3
BIGSRV    IT_EMP    IT_APP
SMALLSRV  IT_EMP    IT_APP    ITINTERN
USERBOX   IT_EMP    ITINTERN  USERS
#
# И группы для специальных задач
# Открыть пользователям echo и golf доступ к антивирусной машине
SECURITY  IT_EMP    (,echo,test-domain) (,golf,test-domain)
#
```

```
# Сетевые группы, специфичные для машин
# Наши главные серверы
WAR    BIGSRV
FAMINE  BIGSRV
# Пользователю india необходим доступ к этому серверу
POLLUTION BIGSRV (,india,test-domain)
#
# Этот очень важен и ему требуются большие ограничения доступа
DEATH   IT_EMP
#
# Антивирусная машина, упомянутая выше
ONE     SECURITY
#
# Ограничить машину единственным пользователем
TWO     (,hotel,test-domain)
# [...]далее следуют другие группы]
```

Если вы используете какие-либо базы данных для управления учетными записями ваших пользователей, вы должны смочь создать первую часть карты с помощью инструментов построения отчетов вашей базы данных. В таком случае новые пользователи автоматически получают доступ к машинам.

И последнее замечание: Не всегда бывает разумно использовать сетевые группы на основе машин. Если в студенческих лабораториях вы используете несколько десятков или даже сотен одинаковых машин, то вам нужно использовать сетевые группы на основе ролей, а не основе машин, для того, чтобы размеры карты NIS оставались в разумных пределах.

### 23.4.8. Важные замечания

Есть некоторые действия, которые нужно будет выполнять по-другому, если вы работаете с NIS.

- Каждый раз, когда вы собираетесь добавить пользователя в лаборатории, вы должны добавить его *только* на главном сервере NIS и *обязательно перестроить карты NIS*. Если вы забудете сделать это, то новый пользователь не сможет нигде войти, кроме как на главном сервере NIS. Например, если в лаборатории нам нужно добавить нового пользователя jsmith, мы делаем вот что:

```
# pw useradd jsmith
# cd /var/yp
# make test-domain
```

Вместо `pw useradd jsmith` вы можете также запустить команду `adduser jsmith`.

- *Не помещайте административные учетные записи в карты NIS*. Вам не нужно распространять административных пользователей и их пароли на машины, которые не должны иметь доступ к таким учётным записям.
- *Сделайте главный и вторичные серверы NIS безопасными и минимизируйте их время простоя*. Если кто-то либо взломает, либо просто отключит эти машины, то люди без права входа в лабораторию с легкостью получат доступ.

Это основное уязвимое место в любой централизованно администрируемой системе. Если вы не защищаете ваши серверы NIS, вы будете иметь дело с толпой разозлённых пользователей!

### 23.4.9. Совместимость с NIS v1

**ypserv** из поставки FreeBSD имеет встроенную поддержку для обслуживания клиентов NIS v1. Реализация NIS во FreeBSD использует только протокол NIS v2, хотя другие реализации имеют поддержку протокола v1 для совместимости со старыми системами. Демоны **ypbind**, поставляемые с такими системами, будут пытаться осуществить привязку к серверу NIS v1, даже если это им не нужно (и они будут постоянно рассылать широковещательные запросы в поиске такого сервера даже после получения ответа от сервера v2). Отметьте, что хотя имеется поддержка обычных клиентских вызовов, эта версия **ypserv** не обрабатывает запросы на передачу карт v1; следовательно, она не может использоваться в качестве главного или вторичного серверов вместе с другими серверами NIS, поддерживающими только протокол v1. К счастью, скорее всего, в настоящий момент такие серверы практически не используются.

### 23.4.10. Серверы NIS, которые также являются клиентами NIS

Особое внимание следует уделить использованию **ypserv** в домене со многими серверами, когда серверные машины являются также клиентами NIS. Неплохо бы заставить серверы осуществить привязку к самим себе, запретив рассылку запросов на привязку и возможно, перекрестную привязку друг к другу. Если один сервер выйдет из строя, а другие будут зависеть от него, то в результате могут возникнуть странные ситуации. Постепенно все клиенты попадут в таймаут и попытаются привязаться к другим серверам, но полученная задержка может быть значительной, а странности останутся, так как серверы снова могут привязаться друг к другу.

Вы можете заставить хост выполнить привязку к конкретному серверу, запустив команду `ypbind` с флагом `-S`. Если вы не хотите делать это вручную каждый раз при перезагрузке вашего сервера NIS, то можете добавить в файл `/etc/rc.conf` такие строки:

```
nis_client_enable="YES"    # run client stuff as well
nis_client_flags="-S NIS domain,server"
```

Дополнительную информацию можно найти на странице справки по `ypbind`(8).

### 23.4.11. Форматы паролей

Одним из общих вопросов, которые возникают в начале работы с NIS, является вопрос совместимости форматов паролей. Если ваш сервер NIS использует пароли, зашифрованные алгоритмом DES, то он будет поддерживать только тех клиентов, что также используют DES. К примеру, если в вашей сети имеются клиенты NIS, использующие Solaris, то вам, скорее всего, необходимо использовать пароли с шифрованием по алгоритму DES.

Чтобы понять, какой формат используют ваши серверы и клиенты, загляните в файл `/etc/login.conf`. Если хост настроен на использование паролей, зашифрованных по алгоритму DES, то класс `default` будет содержать запись вроде следующей:

```
default:\
:passwd_format=des:\
:copyright=/etc/COPYRIGHT:\
[Последующие строки опущены]
```

Другими возможными значениями для `passwd_format` являются `blf` и `md5` (для паролей, шифруемых по стандартам Blowfish и MD5 соответственно).

Если вы внесли изменения в файл `/etc/login.conf`, то вам также нужно перестроить базу данных параметров входа в систему, что достигается запуском следующей команды пользователем `root`:

```
# cap_mkdb /etc/login.conf
```

**Замечание:** Формат паролей, которые уже находятся в файле `/etc/master.passwd`, не будет изменён до тех пор, пока пользователь не сменит свой пароль *после* перестроения базы данных параметров входа в систему.

После этого, чтобы удостовериться в том, что пароли зашифрованы в том формате, который выбран вами, нужно проверить, что строка `crypt_default` в `/etc/auth.conf` указывает предпочтение выбранного вами формата паролей. Для этого поместите выбранный формат первым в списке. Например, при использовании DES-шифрования паролей строка будет выглядеть так:

```
crypt_default = des blf md5
```

Выполнив вышеперечисленные шаги на каждом из серверов и клиентов NIS, работающих на FreeBSD, вы можете обеспечить их согласованность относительно используемого в вашей сети формата паролей. Если у вас возникли проблемы с аутентификацией клиента NIS, начать её решать определённо стоит отсюда. Запомните: если вы хотите использовать сервер NIS в гетерогенной сети, вам, наверное, нужно будет использовать DES на всех системах в силу того, что это минимальный общий стандарт.

## 23.5. Автоматическая настройка сети (DHCP)

*Текст написал Greg Sutter.*

### 23.5.1. Что такое DHCP?

DHCP, или Dynamic Host Configuration Protocol (Протокол Динамической Конфигурации Хостов), описывает порядок, по которому система может подключиться к сети и получить необходимую информацию для работы в ней. Во FreeBSD используется реализация DHCP от ISC (Internet Software Consortium), так что вся информация, описывающая особенности, зависящие от реализации, относится к дистрибутиву ISC.

### 23.5.2. Что описывается в этом разделе

В этом разделе описываются компоненты системы ISC DHCP как со стороны клиента, так и со стороны сервера. Программа, работающая на клиентской стороне, `dhclient`, интегрирована в поставку FreeBSD, а серверная часть доступна в виде порта `net/isc-dhcp3-server`. Кроме ссылок ниже, много

полезной информации находится на страницах справочной системы, описывающих `dhclient(8)`, `dhcp-options(5)` и `dhclient.conf(5)`.

### 23.5.3. Как это работает

Когда на клиентской машине выполняется программа `dhclient`, являющаяся клиентом DHCP, она начинает широковещательную рассылку запросов на получение настроечной информации. По умолчанию эти запросы делаются на 68 порт UDP. Сервер отвечает на UDP 67, выдавая клиенту адрес IP и другую необходимую информацию, такую, как сетевую маску, маршрутизатор и серверы DNS. Вся эта информация даётся в форме "аренды" DHCP и верна только определенное время (что настраивается администратором сервера DHCP). При таком подходе устаревшие адреса IP тех клиентов, которые больше не подключены к сети, могут автоматически использоваться повторно.

Клиенты DHCP могут получить от сервера очень много информации. Подробный список находится в странице Справочника `dhcp-options(5)`.

### 23.5.4. Интеграция с FreeBSD

Клиент DHCP от ISC, `dhclient`, полностью интегрирован во FreeBSD. Поддержка клиента DHCP есть как в программе установки, так и в самой системе, что исключает необходимость в знании подробностей конфигурации сети в любой сети, имеющей сервер DHCP. Утилита `dhclient` включена во все версии FreeBSD, начиная с 3.2.

DHCP поддерживается утилитой **sysinstall**. При настройке сетевого интерфейса из программы **sysinstall** первый вопрос, который вам задается: "Do you want to try DHCP configuration of this interface?" ("Хотите ли вы попробовать настроить этот интерфейс через DHCP?"). Утвердительный ответ приведёт к запуску программы `dhclient`, и при удачном его выполнении к автоматическому заданию информации для настройки интерфейса.

Есть две вещи, которые вы должны сделать для того, чтобы ваша система использовала DHCP при загрузке:

- Убедитесь, что устройство `bpf` включено в компиляцию вашего ядра. Чтобы это сделать, добавьте строчку `device bpf` (`pseudo-device bpf` в FreeBSD 4.X) в конфигурационный файл ядра и перестройте ядро. Более подробная информация о построении ядер имеется в Гл. 8.

Устройство `bpf` уже является частью ядра `GENERIC`, которое поставляется вместе с FreeBSD, так что, если вы не используете другое ядро, то вам и не нужно его делать для того, чтобы работал DHCP.

**Замечание:** Те, кто беспокоится о безопасности, должны иметь в виду, что устройство `bpf` является также тем самым устройством, которое позволяет работать программам-снифферам пакетов (хотя для этого они должны быть запущены пользователем `root`). Наличие устройства `bpf` необходимо для использования DHCP, но если вы чересчур беспокоитесь о безопасности, то вам нельзя добавлять устройство `bpf` в ядро только для того, чтобы в неопределённом будущем использовать DHCP.

- Отредактируйте ваш файл `/etc/rc.conf`, включив в него следующее:

```
ifconfig_fxp0="DHCP"
```

**Замечание:** Обязательно замените `fxp0` на имя интерфейса, который вы хотите настраивать динамически, как это описано в Разд. 11.8.

Если `dhclient` в вашей системе находится в другом месте или если вы хотите задать дополнительные параметры для `dhclient`, то также укажите следующее (изменив так, как вам нужно):

```
dhcp_program="/sbin/dhclient"
dhcp_flags=" "
```

Сервер DHCP, **dhcpcd**, включён как часть порта `net/isc-dhcp3-server` в коллекцию портов. Этот порт содержит DHCP-сервер от ISC и документацию.

### 23.5.5. Файлы

- `/etc/dhclient.conf`

`dhclient` требует наличия конфигурационного файла, `/etc/dhclient.conf`. Как правило, файл содержит только комментарии, а настройки по умолчанию достаточно хороши. Этот настроечный файл описан на страницах справочной системы по `dhclient.conf(5)`.

- `/sbin/dhclient`

`dhclient` скомпилирован статически и находится в каталоге `/sbin`. На страница Справочника `dhclient(8)` дается более подробная информация о `dhclient`.

- `/sbin/dhclient-script`

`dhclient-script` является специфичным для FreeBSD скриптом настройки клиента DHCP. Он описан в `dhclient-script(8)`, но для нормального функционирования никаких модификаций со стороны пользователя не требуется.

- `/var/db/dhclient.leases`

В этом файле клиент DHCP хранит базу данных выданных к использованию адресов в виде журнала. На странице `dhclient.leases(5)` дается гораздо более подробное описание.

### 23.5.6. Дополнительная литература

Полное описание протокола DHCP дается в RFC 2131 (<http://www.freessoft.org/CIE/RFC/2131/>). Кроме того, дополнительная информация есть на сервере <http://www.dhcp.org/>.

## 23.5.7. Установка и настройка сервера DHCP

### 23.5.7.1. Чему посвящён этот раздел

Этот раздел даёт информацию о том, как настроить систему FreeBSD для работы в качестве сервера DHCP на основе реализации пакета DHCP от ISC (Internet Software Consortium).

Серверная часть пакета не поставляется как часть FreeBSD, так что вам потребуется установить порт `net/isc-dhcp3-relay` для получения этого сервиса. Обратитесь к Гл. 4 для получения более полной информации об использовании коллекции портов.

### 23.5.7.2. Установка сервера DHCP

Для того, чтобы настроить систему FreeBSD на работу в качестве сервера DHCP, вам необходимо обеспечить присутствие устройства `bpf(4)`, вкомпилированного в ядро. Для этого добавьте строку `device bpf` (`pseudo-device bpf` в FreeBSD 4.X) в файл конфигурации вашего ядра. Для получения более полной информации о построении ядер, обратитесь к Гл. 8.

Устройство `bpf` уже входит в состав ядра `GENERIC`, поставляемого с FreeBSD, так что вам не нужно создавать собственное ядро для обеспечения работы DHCP.

**Замечание:** Те, кто обращает особое внимание на вопросы безопасности, должны заметить, что `bpf` является тем устройством, что позволяет нормально работать sniffерам пакетов (хотя таким программам требуются привилегированный доступ). Наличие устройства `bpf` *обязательно* для использования DHCP, но если вы очень обеспокоены безопасностью, наверное, вам не нужно включать `bpf` в ваше ядро только потому, что в отдалённом будущем вы собираетесь использовать DHCP.

Следующим действием, которое вам нужно выполнить, является редактирование примерного `dhcpd.conf`, который устанавливается в составе порта `net/isc-dhcp3-server`. По умолчанию это файл `/usr/local/etc/dhcpd.conf.sample`, и вы должны скопировать его в файл `/usr/local/etc/dhcpd.conf` перед тем, как его редактировать.

### 23.5.7.3. Настройка сервера DHCP

`dhcpd.conf` состоит из деклараций относительно подсетей и хостов, и проще всего описывается на примере:

```
option domain-name "example.com";❶
option domain-name-servers 192.168.4.100;❷
option subnet-mask 255.255.255.0;❸

default-lease-time 3600;❹
max-lease-time 86400;❺
ddns-update-style none;❻

subnet 192.168.4.0 netmask 255.255.255.0 {
    range 192.168.4.129 192.168.4.254;❼
    option routers 192.168.4.1;❽
}
```

```
host mailhost {
    hardware ethernet 02:03:04:05:06:07; ❹
    fixed-address mailhost.example.com; ❿
}
```

- ❶ Этот параметр задаёт домен, который будет выдаваться клиентам в качестве домена, используемого по умолчанию при поиске. Обратитесь к страницам справочной системы по resolv.conf(5) для получения дополнительной информации о том, что это значит.
- ❷ Этот параметр задаёт список разделённых запятыми серверов DNS, которые должен использовать клиент.
- ❸ Маска сети, которая будет выдаваться клиентам.
- ❹ Клиент может запросить определённое время, которое будет действовать выданная информация. В противном случае сервер выдаст настройки с этим сроком (в секундах).
- ❺ Это максимальное время, на которое сервер будет выдавать конфигурацию. Если клиент запросит больший срок, он будет подтверждён, но будет действовать только `max-lease-time` секунд.
- ❻ Этот параметр задаёт, будет ли сервер DHCP пытаться обновить DNS при выдаче или освобождении конфигурационной информации. В реализации ISC этот параметр является *обязательным*.
- ❼ Это определение того, какие IP-адреса должны использоваться в качестве резерва для выдачи клиентам. IP-адреса между и включая границы, будут выдаваться клиентам.
- ❽ Объявление маршрутизатора, используемого по умолчанию, который будет выдаваться клиентам.
- ❾ Аппаратный MAC-адрес хоста (чтобы сервер DHCP мог распознать хост, когда тот делает запрос).
- (10) Определение того, что хосту всегда будет выдаваться один и тот же IP-адрес. Заметьте, что указание здесь имени хоста корректно, так как сервер DHCP будет разрешать имя хоста самостоятельно до того, как выдать конфигурационную информацию.

Как только вы закончите составлять свой `dhcpd.conf`, вы можете продолжить работу запуском сервера при помощи следующей команды:

```
# /usr/local/etc/rc.d/isc-dhcpd.sh start
```

Если в будущем вам понадобится сделать изменения в настройке вашего сервера, то важно заметить, что посылка сигнала `SIGUP` приложению **dhcpd** *не приведёт* к перезагрузке настроек, как это бывает для большинства демонов. Вам нужно послать сигнал `SIGTERM` для остановки процесса, а затем перезапустить его при помощи вышеприведённой команды.

#### 23.5.7.4. Файлы

- `/usr/local/sbin/dhcpd`

**dhcpd** скомпонован статически и расположен в каталоге `/usr/local/sbin`. Страницы справочной системы `dhcpd(8)`, устанавливаемые портом, содержат более полную информацию о **dhcpd**.



- `/usr/local/etc/dhcpd.conf`

**dhcpd** требует наличия конфигурационного файла, `/usr/local/etc/dhcpd.conf`, до того, как он будет запущен и начнёт предоставлять сервис клиентам. Необходимо, чтобы этот файл содержал все данные, которая будет выдаваться обслуживаемым клиентам, а также информацию о работе сервера. Этот конфигурационный файл описывается на страницах справочной системы `dhcpd.conf(5)`, которые устанавливаются портом.

- `/var/db/dhcpd.leases`

Сервер DHCP ведёт базу данных выданной информации в этом файле, который записывается в виде протокола. Страницы справочной системы `dhcpd.leases(5)`, устанавливаемые портом, дают гораздо более подробное описание.

- `/usr/local/sbin/dhcrelay`

**dhcrelay** используется в сложных ситуациях, когда сервер DHCP пересылает запросы от клиента другому серверу DHCP в отдельной сети. Если вам нужна такая функциональность, то установите порт `net/isc-dhcp3-server`. На страницах справочной системы `dhcrelay(8)`, которые устанавливаются портом, даётся более полное описание.

## 23.6. Domain Name System (DNS)

*Текст предоставил Chern Lee.*

### 23.6.1. Обзор

По умолчанию во FreeBSD используется одна из версий программы BIND (Berkeley Internet Name Domain), являющейся самой распространённой реализацией протокола DNS. DNS - это протокол, при помощи которого имена преобразуются в IP-адреса и наоборот. Например, в ответ на запрос о `www.FreeBSD.org` будет получен IP-адрес веб-сервера Проекта FreeBSD, а запрос о `ftp.FreeBSD.org` возвратит IP-адрес соответствующей машины с FTP-сервером. Точно также происходит и обратный процесс. Запрос, содержащий IP-адрес машины, возвратит имя хоста. Для выполнения запросов к DNS вовсе не обязательно иметь в системе работающий сервер имён.

В сети Интернет DNS управляется через достаточно сложную систему авторизованных корневых серверов имён, и других менее крупных серверов имён, которые содержат и кэшируют информацию о конкретных доменах.

В этом документе рассматривается BIND 8.x, так как это стабильная версия, используемая во FreeBSD. BIND 9.x может быть установлен как порт `net/bind9`.

Протокол DNS стандартизован в RFC1034 и RFC1035.

На данный момент пакет BIND поддерживается Internet Software Consortium <http://www.isc.org/>.

## 23.6.2. Используемая терминология

Для понимания этого документа нужно понимать значения некоторых терминов, связанных с работой DNS.

Термин	Определение
Прямой запрос к DNS (forward DNS)	Преобразование имён хостов в адреса IP
Ориджин (origin)	Обозначает домен, покрываемый конкретным файлом зоны
<b>named</b> , bind, сервер имён	Общепотребительные названия для обозначения пакета BIND, обеспечивающего работу сервера имён во FreeBSD.
Резолвер	Системный процесс, посредством которого машина обращается к серверу имён для получения информации о зоне
Обратный DNS (reverse DNS)	Операция, обратная прямому запросу к DNS; преобразование адресов IP в имена хостов
Корневая зона	Начало иерархии зон Интернет. Все зоны находятся под корневой зоной, подобно тому, как все файлы располагаются ниже корневого каталога.
Зона	Отдельный домен, поддомен или часть DNS, управляемая одним сервером.

Примеры зон:

- `.` является корневой зоной
- `org.` является зоной ниже корневой зоны
- `example.org` является зоной под зоной `org.`
- `foo.example.org.` является поддоменом, зоной под зоной `example.org.`
- `1.2.3.in-addr.arpa` является зоной, в которую включены все IP-адреса, формирующие пространство адресов `3.2.1.*`.

Как можно видеть, уточняющая часть имени хоста появляется слева. Например, `example.org.` более точен, чем `org.`, также, как `org.` более точен, чем корневая зона. Расположение каждой части имени хоста сильно похоже на файловую систему: каталог `/dev` расположен в корневой файловой системе, и так далее.

## 23.6.3. Причины, по которым вам может понадобится сервер имён

Сервера имён обычно используются в двух видах: авторитетный сервер имён и кэширующий сервер имён.

Авторитетный сервер имён нужен, когда:

- нужно предоставлять информацию о DNS остальному миру, отвечая на запросы авторизованно.
- зарегистрирован домен, такой, как `example.org` и в этом домене требуется поставить имена машин в соответствие с их адресами IP.
- блоку адресов IP требуется обратные записи DNS (IP в имена хостов).

- резервный (slave) сервер имён должен отвечать на запросы о домене, когда основной не работает или не доступен.

Кэширующий сервер имён нужен, когда:

- локальный сервер DNS может кэшировать информацию и отвечать на запросы быстрее, чем это происходит при прямом опросе внешнего сервера имён.
- требуется уменьшение общего сетевого трафика (DNS составляет около 5% всего трафика Интернет, или чуть больше).

Например, когда кто-нибудь запрашивает информацию о `www.FreeBSD.org`, то обычно резолвер обращается к серверу имён вашего провайдера, посылает запрос и ожидает ответа. С локальным кэширующим сервером DNS запрос во внешний мир будет делаться всего один раз. Каждый дополнительный запрос не будет посылаться за пределы локальной сети, потому что информация уже имеется в кэше.

### 23.6.4. Как это работает

Во FreeBSD даемон BIND, по очевидным причинам, называется **named**.

Файл	Описание
<b>named</b>	даемон BIND
<b>ndc</b>	программа управления даемоном сервера имён
<b>/etc/namedb</b>	каталог, в котором располагается вся информация о зонах BIND
<b>/etc/namedb/named.conf</b>	конфигурационный файл для даемона

Файлы зон обычно располагаются в каталоге `/etc/namedb` и содержат информацию о зоне DNS, за которую отвечает сервер имён.

### 23.6.5. Запуск BIND

Так как сервер имён BIND устанавливается по умолчанию, его настройка сравнительно проста.

Чтобы даемон **named** запускался во время загрузки, поместите в `/etc/rc.conf` следующую строку:

```
named_enable="YES"
```

Для запуска даемона вручную (после его настройки):

```
# ndc start
```

### 23.6.6. Конфигурационные файлы

#### 23.6.6.1. Использование `make-localhost`

Обязательно выполните следующие команды:

```
# cd /etc/namedb
# sh make-localhost
```

для того, чтобы правильно создать файл /etc/namedb/localhost.rev локальной обратной зоны для loopback-интерфейса.

### 23.6.6.2. /etc/namedb/named.conf

```
// $FreeBSD$
//
// Refer to the named(8) manual page for details.  If you are ever going
// to setup a primary server, make sure you've understood the hairy
// details of how DNS is working.  Even with simple mistakes, you can
// break connectivity for affected parties, or cause huge amount of
// useless Internet traffic.

options {
    directory "/etc/namedb";

    // In addition to the "forwarders" clause, you can force your name
    // server to never initiate queries of its own, but always ask its
    // forwarders only, by enabling the following line:
    //
    //     forward only;

    // If you've got a DNS server around at your upstream provider, enter
    // its IP address here, and enable the line below.  This will make you
    // benefit from its cache, thus reduce overall DNS traffic in the
    Internet.
    /*
    forwarders {
        127.0.0.1;
    };
    */
```

Как и говорится в комментариях, если вы хотите получить эффект от использования кэша провайдера, то можно включить раздел `forwarders`. В обычном случае сервер имён будет рекурсивно опрашивать определённые серверы имён Интернет до тех пор, пока не получит ответ на свой запрос. При включении этого раздела он будет автоматически опрашивать сервер имён вашего провайдера (или тот, который здесь указан), используя преимущества его кэша. наличия нужной информации. Если соответствующий сервер имён провайдера работает быстро и имеет хороший канал связи, то в результате такой настройки вы можете получить хороший результат.

**Внимание:** 127.0.0.1 здесь работать *не будет*. Измените его на IP-адрес сервера имён провайдера.

```
/*
 * If there is a firewall between you and name servers you want
 * to talk to, you might need to uncomment the query-source
 * directive below. Previous versions of BIND always asked
 * questions using port 53, but BIND 8.1 uses an unprivileged
```

```
* port by default.
*/
// query-source address * port 53;

/*
 * If running in a sandbox, you may have to specify a different
 * location for the dumpfile.
 */
// dump-file "s/named_dump.db";
};

// Note: the following will be supported in a future release.
/*
host { any; } {
topology {
127.0.0.0/8;
};
};
*/

// Setting up secondaries is way easier and the rough picture for this
// is explained below.
//
// If you enable a local name server, don't forget to enter 127.0.0.1
// into your /etc/resolv.conf so this server will be queried first.
// Also, make sure to enable it in /etc/rc.conf.

zone "." {
type hint;
file "named.root";
};

zone "0.0.127.IN-ADDR.ARPA" {
type master;
file "localhost.rev";
};

zone
"0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.INT" {
type master;
file "localhost.rev";
};

// NB: Do not use the IP addresses below, they are faked, and only
// serve demonstration/documentation purposes!
//
// Example secondary config entries. It can be convenient to become
// a secondary at least for the zone where your own domain is in. Ask
// your network administrator for the IP address of the responsible
// primary.
//
// Never forget to include the reverse lookup (IN-ADDR.ARPA) zone!
// (This is the first bytes of the respective IP address, in reverse
```

```
// order, with ".IN-ADDR.ARPA" appended.)
//
// Before starting to setup a primary zone, better make sure you fully
// understand how DNS and BIND works, however. There are sometimes
// unobvious pitfalls. Setting up a secondary is comparably simpler.
//
// NB: Don't blindly enable the examples below. :-) Use actual names
// and addresses instead.
//
// NOTE!!! FreeBSD runs BIND in a sandbox (see named_flags in rc.conf).
// The directory containing the secondary zones must be write accessible
// to BIND. The following sequence is suggested:
//
//      mkdir /etc/namedb/s
//      chown bind:bind /etc/namedb/s
//      chmod 750 /etc/namedb/s
```

Дополнительная информация о запуске BIND в ограниченном окружении находится в соответствующем разделе.

```
/*
zone "example.com" {
type slave;
file "s/example.com.bak";
masters {
192.168.1.1;
};
};

zone "0.168.192.in-addr.arpa" {
type slave;
file "s/0.168.192.in-addr.arpa.bak";
masters {
192.168.1.1;
};
};
*/
```

Это примеры описаний прямой и обратной зон из файла `named.conf` для вторичных серверов.

Для каждого новой зоны, которую будет обслуживать сервер имён, в файл `named.conf` должна быть добавлена запись.

К примеру, самая простая запись для домена `example.org` может выглядеть вот так:

```
zone "example.org" {
type master;
file "example.org";
};
```

Зона является первичной, что отражается в поле `type`, и информация о зоне хранится в файле `/etc/namedb/example.org`, что указывается в поле `file`.

```
zone "example.org" {
type slave;
file "example.org";
};
```

В случае вторичной зоны информация о ней передается с основного сервера имён для заданной зоны и сохраняется в указанном файле. Если и когда основной сервер имён выходит из строя или недостижим, то скачанная информация о зоне будет находиться на вторичных серверах и они смогут обслуживать эту зону.

### 23.6.6.3. Файлы зон

Пример файла зоны `example.org` для основного сервера (распологающийся в файле `/etc/namedb/example.org`) имеет такой вид:

```
$TTL 3600

example.org. IN SOA ns1.example.org. admin.example.org. (
5          ; Serial
10800      ; Refresh
3600       ; Retry
604800     ; Expire
86400 )    ; Minimum TTL

; DNS Servers
@          IN NS      ns1.example.org.
@          IN NS      ns2.example.org.

; Machine Names
localhost  IN A       127.0.0.1
ns1        IN A       3.2.1.2
ns2        IN A       3.2.1.3
mail       IN A       3.2.1.10
@          IN A       3.2.1.30

; Aliases
www        IN CNAME   @

; MX Record
@          IN MX      10      mail.example.org.
```

Заметьте, что все имена хостов, оканчивающиеся на `""`, задают полное имя, тогда как все имена без символа `""` на конце считаются заданными относительно `origin`. Например, `www` преобразуется в `www.origin`. В нашем воображаемом файле ориджином является `example.org.`, так что `www` преобразуется в `www.example.org.`

Файл зоны имеет следующий формат:

```
recordname      IN recordtype  value
```

Наиболее часто используемые записи DNS:

## SOA

начало зоны ответственности

## NS

авторитативный сервер имен

## A

адрес хоста

## CNAME

каноническое имя для алиаса

## MX

обмен почтой

## PTR

указатель на доменное имя (используется в обратных зонах DNS)

```
example.org. IN SOA ns1.example.org. admin.example.org. (
5          ; Serial
10800      ; Refresh after 3 hours
3600       ; Retry after 1 hour
604800     ; Expire after 1 week
86400 )    ; Minimum TTL of 1 day
```

example.org.

имя домена, а также ориджин для этого файла зоны.

ns1.example.org.

основной/авторитативный сервер имён для этой зоны.

admin.example.org.

человек, отвечающий за эту зону, адрес электронной почты с подменённым символом "@".  
(<admin@example.org> становится admin.example.org)

5

последовательный номер файла. При каждом изменении файла зоны это число должно увеличиваться. В настоящее время для нумерации многие администраторы предпочитают формат **ггггммддвв**. 2001041002 будет означать, что файл последний раз изменялся 10.04.2001, а последнее число 02 означает, что это была вторая модификация файла за день. Последовательный номер важен, так как он служит для того, чтобы вторичные серверы узнавали об обновлении зоны.

```
@          IN NS      ns1.example.org.
```

Это NS-запись. Такие записи должны иметься для всех серверов имён, которые будут отвечать за зону. Символ @, используемый здесь, преобразуется в example.org. Этот символ @ соответствует ориджину.



```
localhost      IN A      127.0.0.1
ns1            IN A      3.2.1.2
ns2            IN A      3.2.1.3
mail           IN A      3.2.1.10
@              IN A      3.2.1.30
```

Записи типа A служат для обозначения имён машин. Как это видно выше, имя `ns1.example.org` будет преобразовано в `3.2.1.2`. И снова здесь используется символ ориджина `@`, обозначая, что `example.org` будет преобразовано в `3.2.1.30`.

```
www            IN CNAME  @
```

Записи с каноническими именами обычно используются для присвоения машинам псевдонимов. В этом примере `www` является псевдонимом для машины, соответствующей ориджину, то есть `example.org` (`3.2.1.30`). Записи **CNAME** могут использоваться для присвоения псевдонимов именам хостов или для использования одного имени несколькими машинами по очереди.

```
@              IN MX      10      mail.example.org.
```

**MX**-запись указывает, какие почтовые серверы отвечают за обработку входящей электронной почты для зоны. `mail.example.org` является именем почтового сервера, а `10` обозначает приоритет этого почтового сервера.

Можно иметь несколько почтовых серверов с приоритетами 3, 2 и 1. Почтовый сервер, пытающийся доставить почту для `example.org`, сначала попытается связаться с машиной, имеющей **MX**-запись с самым большим приоритетом, затем с приоритетом поменьше и так далее, до тех пор, пока почта не будет отправлена.

Для файлов зон `in-addr.arpa` (обратные записи **DNS**) используется тот же самый формат, отличающийся только использованием записей **PTR** вместо **A** или **CNAME**.

```
$TTL 3600
```

```
1.2.3.in-addr.arpa. IN SOA ns1.example.org. admin.example.org. (
5                ; Serial
10800            ; Refresh
3600             ; Retry
604800           ; Expire
3600 )           ; Minimum
```

```
@              IN NS      ns1.example.org.
@              IN NS      ns2.example.org.
```

```
2              IN PTR      ns1.example.org.
3              IN PTR      ns2.example.org.
10             IN PTR      mail.example.org.
30             IN PTR      example.org.
```

В этом файле дается полное соответствие имён хостов **IP**-адресам в нашем описанном ранее вымышленном домене.

### 23.6.7. Кэширующий сервер имён

Кэширующий сервер имён - это сервер имён, не отвечающий ни за какую зону. Он просто выполняет запросы от своего имени и сохраняет результаты для последующего использования. Для настройки такого сервера достаточно исключить все описания зон из стандартной конфигурации сервера имён.

### 23.6.8. Запуск **named** в песочнице

Для дополнительной безопасности вам может потребоваться запускать **named(8)** с правами непривилегированного пользователя и настроить его на выполнение **chroot(8)** в каталог-песочницу. Это позволит сделать недоступным для демона **named** все, что расположено вне песочницы. Если **named** будет взломан, то это поможет уменьшить возможный ущерб. По умолчанию во FreeBSD имеются пользователь и группа с именами **bind**, которые предназначены именно для такого использования.

**Замечание:** Многие рекомендуют вместо настройки **named** на использование **chroot**, запускать **named** внутри **jail(8)**. В этом разделе такой подход не рассматривается.

Так как **named** не сможет обратиться ни к чему вне песочницы (например, совместно используемым библиотекам, сокетам протоколов и так далее), то нужно выполнить несколько шагов, чтобы **named** смог работать нормально. В следующем списке предполагается, что каталогом песочницы является **/etc/namedb** и что вы не делали никаких изменений в содержимом этого каталога. Выполните следующие шаги, работая как пользователь **root**:

- Создайте все каталоги, которые ожидает увидеть **named**:

```
# cd /etc/namedb
# mkdir -p bin dev etc var/tmp var/run master slave
# chown bind:bind slave var/*❶
```

- ❶ Программе **named** нужен доступ с правом записи в эти каталоги, так что это все, что мы ей предоставим.

- Измените и создайте базовые файлы зоны и настроек:

```
# cp /etc/localtime etc❶
# mv named.conf etc && ln -sf etc/named.conf
# mv named.root master
# sh make-localhost && mv localhost.rev localhost-v6.rev master
# cat > master/named.localhost
$ORIGIN localhost.
$TTL 6h
@ IN SOA localhost. postmaster.localhost. (
1 ; serial
3600 ; refresh
1800 ; retry
604800 ; expiration
3600 ) ; minimum
IN NS localhost.
IN A 127.0.0.1
```

^D

❶ Это позволит программе **named** протоколировать правильное время в syslogd(8).

- Если вы используете FreeBSD версии ранее 4.9-RELEASE, то постройте статически скомпонованную копию **named-xfer** и скопируйте её в песочницу:

```
# cd /usr/src/lib/libisc
# make cleandir && make cleandir && make depend && make all
# cd /usr/src/lib/libbind
# make cleandir && make cleandir && make depend && make all
# cd /usr/src/libexec/named-xfer
# make cleandir && make cleandir && make depend && make NOSHARED=yes all
# cp named-xfer /etc/namedb/bin && chmod 555 /etc/namedb/bin/named-xfer❶
```

После установки статически скомпонованного **named-xfer**, во избежание появления старых копий библиотек и программ в дереве исходного кода, требуется некоторая зачистка:

```
# cd /usr/src/lib/libisc
# make cleandir
# cd /usr/src/lib/libbind
# make cleandir
# cd /usr/src/libexec/named-xfer
# make cleandir
```

❶ Иногда при выполнении этого шага возникают ошибки. Если это случилось, выполните такую команду:

```
# cd /usr/src && make cleandir && make cleandir

и удалите ваше дерево /usr/obj:

# rm -fr /usr/obj && mkdir /usr/obj
```

При этом из вашего дерева исходных текстов будет удалён весь "мусор", и повторение вышеописанных шагов должно выполняться успешно.

Если вы используете FreeBSD 4.9-RELEASE или более позднюю версию, то копия **named-xfer** в каталоге **/usr/libexec** по умолчанию является статически скомпонованной, и вы можете просто скопировать её в песочницу при помощи команды **cp**(1).

- Создайте файл устройства **dev/null**, который **named** может видеть и писать в него:

```
# cd /etc/namedb/dev && mknod null c 2 2
# chmod 666 null
```

- Создайте символическую ссылку **/var/run/ndc** на **/etc/namedb/var/run/ndc**:

```
# ln -sf /etc/namedb/var/run/ndc /var/run/ndc
```

**Замечание:** Это просто для того, чтобы не задавать опцию **-c** при каждом запуске **ndc**(8). Так как содержимое каталога **/var/run** удаляется при загрузке, и если это показалось вам полезным, то вы можете добавить эту команду в **crontab** для **root** с использованием параметра **@reboot**. Обратитесь к справочной странице по **crontab**(5) для получения более полной информации относительно этого.



```

zone "." IN {
    type hint;
    file "master/named.root";
};
zone "private.example.net" in {
    type master;
    file "master/private.example.net.db";
    allow-transfer { 192.168.10.0/24; };
};
zone "10.168.192.in-addr.arpa" in {
    type slave;
    masters { 192.168.10.2; };
    file "slave/192.168.10.db";❹
};

```

- ❶ В директиве `directory` указан каталог `/`, так как все файлы, которые нужны для **named**, находятся внутри этого каталога (вспомните, что это равнозначно "обычному" пользовательскому `/etc/namedb`).
- ❷ Задаёт полный путь к двоичному исполняемому файлу `named-xfer` (внутри границ видимости **named**). Это необходимо, так как **named** компилируется с тем, чтобы брать `named-xfer` по умолчанию из `/usr/libexec`.
- ❸ Задаёт имя файла (относительно директивы `directory` выше), в котором **named** может найти файл зоны для этой зоны.
- ❹ Задаёт имя файла (относительно директивы `directory` выше), в котором **named** должен записывать копию файла зоны для этой зоны после успешной передачи ее с основного сервера. Вот почему нам нужно изменить владельца каталога `slave` на `bind` на этапах настроек выше.

После выполнения шагов выше либо перезагрузите ваш сервер, либо перезапустите `syslogd(8)` и запустите `named(8)`, не забыв использовать новые опции, заданные в `syslogd_flags` и `named_flags`. Теперь **named** должен заработать в песочнице!

### 23.6.9. Безопасность

Хотя BIND является самой распространенной реализацией DNS, всегда стоит вопрос об обеспечении безопасности. Время от времени обнаруживаются возможные и реальные бреши в безопасности.

Весьма полезно прочесть сообщения безопасности CERT (<http://www.cert.org/>) и подписаться на Список рассылки FreeBSD, посвящённый срочным сообщениям, связанным с безопасностью (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-security-notifications>) для того, чтобы быть в курсе текущих проблем с обеспечением безопасности Internet и FreeBSD.

**Подсказка:** Если возникают проблемы, то наличие последних исходных текстов и свежескомпилированного **named** не помешает.

### 23.6.10. Дополнительная литература

Справочная информация по BIND/**named**: `ndc(8)`, `named(8)` и `named.conf(5)`

- Официальная страница ISC BIND (<http://www.isc.org/products/BIND/>)
- FAQ по BIND (<http://www.nominum.com/getOpenSourceResource.php?id=6>)
- Книга издательства O'Reilly DNS and BIND 4th Edition (<http://www.oreilly.com/catalog/dns4/>)
- RFC1034 - Domain Names - Concepts and Facilities (<ftp://ftp.isi.edu/in-notes/rfc1034.txt>)
- RFC1035 - Domain Names - Implementation and Specification (<ftp://ftp.isi.edu/in-notes/rfc1035.txt>)

## 23.7. BIND9 и FreeBSD

*Автор: Tom Rhodes. Перевод на русский язык: Dmitry Morozovsky.*

FreeBSD начиная с версии 5.3 содержит в базовой поставке сервер DNS BIND9. Среди прочих нововведений стоит отметить новые возможности, связанные с безопасностью, новую организацию файлов конфигурации и автоматическое использование опции `chroot(8)`. Первая часть этой главы посвящена обсуждению новых возможностей и их использования, вторая описывает процесс обновления сервера DNS при переходе на FreeBSD 5.3. Мы будем ссылаться на сервер BIND как `named(8)`. Мы подразумеваем, что читатель знаком с терминологией, описываемой в предыдущей главе.

Файлы конфигурации для `named` в настоящее время располагаются в каталоге `/var/named/etc/namedb/` и требуют модификации перед началом использования сервера. Большая часть работы по конфигурации сервера DNS происходит именно в этом каталоге.

### 23.7.1. Конфигурация основной (Master) зоны

Для создания основной зоны перейдите в каталог `/var/named/etc/namedb/` и выполните команду

```
# sh make-localhost
```

При удачном стечении обстоятельств в каталоге `master` появится два новых файла: `localhost.rev` для локального домена и `localhost-v6.rev` для локальной зоны IPv6. В поставляемом стандартном файле конфигурации `named.conf` уже есть ссылки на оба файла зон.

### 23.7.2. Конфигурация зависимой (Slave) Зоны

Дополнительные домены и поддомены могут быть добавлены как зависимые зоны. В большинстве случаев достаточно просто скопировать файл `master/localhost.rev` в каталог `slave` и изменить копию. После изменения файлов зон нужно добавить ссылки на них в основной файл конфигурации `named.conf`, например, так:

```
zone "example.com" {
    type slave;
    file "slave/example.com";
    masters {
```

```

10.0.0.1;
};
};

zone "0.168.192.in-addr.arpa" {
type slave;
file "slave/0.168.192.in-addr.arpa";
masters {
10.0.0.1;
};
};

```

Отметим, что использованный в этом примере IP адрес является адресом, с которого будут копироваться файлы зон; этот адрес не обязательно является публичным сервером DNS.

### 23.7.3. Стартовая конфигурация системы

Для автоматического запуска демона `named` при загрузке системы добавьте в файл стартовой конфигурации `rc.conf` строку

```
named_enable="YES"
```

Есть и другие параметры конфигурации, но эта строка — минимально необходима. Список параметров можно найти на странице справочника `rc.conf(5)`. Если запуск `named` не разрешен в `rc.conf`, сервер можно запустить вручную командой

```
# /etc/rc.d/named start
```

### 23.7.4. BIND9: вопросы безопасности

FreeBSD автоматически запускает `named` в ограниченном пространстве `chroot(8)`; кроме того, существует еще несколько средств, помогающих защитить сервис DNS от возможных атак.

#### 23.7.4.1. Списки ограничения доступа

Для ограничения доступа к зонам могут применяться списки доступа. Для этого опишите для домена список IP адресов, которым разрешены запросы к содержимому доменной зоны, и используйте имя этого списка в конфигурации зоны. Вот пример описания списка ограничения доступа:

```

acl "example.com" {
192.168.0.0/24;
};

zone "example.com" {
type slave;
file "slave/example.com";
masters {
10.0.0.1;
};
allow-query { example.com; };

```

```
};

zone "0.168.192.in-addr.arpa" {
type slave;
file "slave/0.168.192.in-addr.arpa";
masters {
10.0.0.1;
};
allow-query { example.com; };
};
```

#### 23.7.4.2. Версия сервера

Разрешать всем запросы о версии вашего DNS сервера может быть не самой лучшей идеей: возможный злоумышленник может воспользоваться ей для выяснения известных ошибок и уязвимостей и применить их вашему серверу. Для предотвращения этого в раздел `options` файла конфигурации `named.conf` можно вписать строку, скрывающую версию сервера:

```
options {
directory          "/etc/namedb";
pid-file           "/var/run/named/pid";
dump-file          "/var/dump/named_dump.db";
statistics-file    "/var/stats/named.stats";
version "None of your business";
```

## 23.8. Apache HTTP сервер

*Предоставил Murray Stokely.*

### 23.8.1. Обзор

FreeBSD используется в качестве платформы для многих из самых нагруженных серверов в мире. Большинство серверов в интернет используют **Apache HTTP сервер**. Пакеты **Apache** должны быть включены в поставку FreeBSD. Если вы не установили их во время установки системы, воспользуйтесь портами `www/apache13` или `www/apache2`.

Как только **Apache** был успешно установлен, его необходимо настроить.

**Замечание:** В этом разделе рассказывается о версии 1.3.X **Apache HTTP сервера**, поскольку эта версия наиболее широко используется в FreeBSD. **Apache 2.X** содержит много новых технологий, но здесь они не обсуждаются. За дополнительной информацией о **Apache 2.X**, обращайтесь к <http://httpd.apache.org/>.



## 23.8.2. Настройка

В FreeBSD основной файл настройки **Apache HTTP сервера** устанавливается в `/usr/local/etc/apache/httpd.conf`. Это обычный текстовый UNIX файл настройки с строками комментариев, начинающимися с символа `#`. Исчерпывающее описание всех возможных параметров настройки находится за пределом рассмотрения этой книги, поэтому здесь будут описаны только наиболее часто модифицируемые директивы.

```
ServerRoot "/usr/local"
```

Указывает верхний каталог установки **Apache** по умолчанию. Бинарные файлы находятся в `bin` и `sbin`, подкаталоги расположены относительно корневого каталога сервера, файлы настройки находятся в `etc/apache`.

```
ServerAdmin you@your.address
```

Адрес, на который должны будут отправляться сообщения о проблемах с сервером. Этот адрес выводится на некоторые генерируемые сервером страницы, например с сообщениями об ошибках.

```
ServerName www.example.com
```

`ServerName` позволяет вам устанавливать имя хоста, которое отправляется обратно клиентам, если оно отличается от того, с которым настроен хост (например, использование `www` вместо реального имени хоста).

```
DocumentRoot "/usr/local/www/data"
```

`DocumentRoot`: Каталог, внутри которого будут храниться документы. По умолчанию, все запросы обрабатываются внутри этого каталога, но символические ссылки и синонимы могут использоваться для указания на другие каталоги.

Хорошей идеей будет сделать резервные копии настроек **Apache** перед внесением изменений. Как только вы будете удовлетворены первоначальной настройкой, можно запускать **Apache**.

## 23.8.3. Запуск Apache

**Apache** не запускается из `inetd`, как это делают многие другие сетевые серверы. Он настроен для автономного запуска, чтобы обеспечивать большую производительность при обработке HTTP запросов от браузеров клиентов. Для упрощения запуска, остановки и перезапуска сервера существует shell скрипт. Для запуска **Apache** в первый раз просто выполните:

```
# /usr/local/sbin/apachectl start
```

Вы можете остановить сервер в любой момент, выполнив:

```
# /usr/local/sbin/apachectl stop
```

После внесения любых изменений в файл настроек, вам потребуется перезапустить сервер:

```
# /usr/local/sbin/apachectl restart
```

Для перезапуска **Apache** без прерывания имеющихся соединений, выполните:

```
# /usr/local/sbin/apachectl graceful
```

Дополнительная информация находится на странице справочного руководства `apachectl(8)`.

Для запуска **Apache** при старте системы, добавьте в `/etc/rc.conf` следующую строку:

```
apache_enable="YES"
```

Если вы хотите передать программе **Apache** `httpd` дополнительные параметры командной при загрузке системы, они могут быть помещены в дополнительную строку `rc.conf`:

```
apache_flags=" "
```

Теперь, когда веб сервер запущен, вы можете просмотреть свой веб сайт, задав в строке браузера адрес `http://localhost/`. По умолчанию отображается веб страница `/usr/local/www/data/index.html`.

#### 23.8.4. Виртуальный хостинг

**Apache** поддерживает два различных типа виртуального хостинга (Virtual Hosting). Первый метод основан на именах (Name-based Virtual Hosting). Он использует полученные от клиента заголовки HTTP/1.1 для определения имени хоста. Это позволяет многим различным доменам использовать один и тот же IP адрес.

Для настройки **Apache** на использование этого типа хостинга добавьте в `httpd.conf` запись подобную следующей:

```
NameVirtualHost *
```

Если веб сервер назывался `www.domain.tld` и вы хотите настроить виртуальный домен для `www.someotherdomain.tld`, необходимо добавить в `httpd.conf` следующие записи:

```
<VirtualHost *>
ServerName www.domain.tld
DocumentRoot /www/domain.tld
</VirtualHost>

<VirtualHost *>
ServerName www.someotherdomain.tld
DocumentRoot /www/someotherdomain.tld
</VirtualHost>
```

Замените адреса и пути к документам на те, что вы будете использовать.

За дополнительной информацией по настройке виртуальных хостов обращайтесь к официальной документации **Apache**: <http://httpd.apache.org/docs/vhosts/>

#### 23.8.5. Модули Apache

Существуют множество различных модулей **Apache**, которые добавляют функциональность к основному серверу. Коллекция портов FreeBSD предоставляет простой способ установки **Apache** с некоторыми наиболее популярными дополнительными модулями.

### 23.8.5.1. mod\_ssl

Модуль **mod\_ssl** использует библиотеку OpenSSL для сильной криптографии через протоколы Secure Sockets Layer (SSL v2/v3) и Transport Layer Security (TLS v1). Этот модуль содержит все необходимое для запроса подписанного сертификата из центра сертификации для защищенного веб сервера на FreeBSD.

Если вы еще не установили **Apache**, версия **Apache** с **mod\_ssl** может быть установлена через порт `www/apache13-modssl`. Поддержка SSL также присутствует и включена по умолчанию для порта `www/apache2`.

### 23.8.5.2. mod\_perl

Проект интеграции **Apache**/Perl объединяет мощь языка программирования Perl и HTTP сервера **Apache**. С модулем **mod\_perl** возможно написание модулей **Apache** полностью на Perl. Кроме того, постоянно запущенный встроенный в сервер интерпретатор позволяет не тратить ресурсы на запуск внешнего интерпретатора и время на запуск Perl.

Если вы еще не установили **Apache**, его версия с модулем **mod\_perl** может быть установлена через порт `www/apache13-modperl`.

### 23.8.5.3. PHP

PHP, сокращение от "PHP: Hypertext Preprocessor" это широко используемый Open Source скриптовый язык общего назначения, который в основном предназначен для веб разработки и может быть встроен в HTML. Его синтаксис был взят от C, Java, и Perl и язык легок в изучении. Основное преимущество языка в том, что он позволяет веб разработчикам создавать веб страницы быстро, но есть и множество других возможностей.

PHP может быть установлен из порта `lang/php5`.

## 23.9. Файл сервер и печать для Microsoft® Windows клиентов (Samba)

*Предоставил Murray Stokely.*

### 23.9.1. Обзор

**Samba** это популярный пакет программ с открытыми исходными текстами, которая предоставляет файловые и принт-сервисы Microsoft Windows клиентам. Эти клиенты могут подключаться и использовать файловое пространство FreeBSD, как если бы это был локальный диск, или принтеры FreeBSD, как если бы это были локальные принтеры.

Пакет **Samba** должен быть включен в поставку FreeBSD. Если вы не установили **Samba** при первой установке системы, ее можно установить из порта или пакета `net/samba3`.

## 23.9.2. Настройка

Файл настройки **Samba** по умолчанию устанавливается в `/usr/local/etc/smb.conf.default`. Этот файл необходимо скопировать в `/usr/local/etc/smb.conf` и отредактировать перед использованием **Samba**.

В файле `smb.conf` находится информация, необходимая для работы **Samba**, например определение принтеров и "общих каталогов", которые будут использоваться совместно с Windows клиентами. В пакет **Samba** входит программа с веб интерфейсом, называемая **swat**, которая дает простой способ редактирования файла `smb.conf`.

### 23.9.2.1. Использование Samba Web Administration Tool (SWAT)

Программа веб администрирования Samba (Samba Web Administration Tool, SWAT) запускается как даемон из **inetd**. Следовательно, в `/etc/inetd.conf` необходимо снять комментарий перед тем, как использовать **swat** для настройки **Samba**:

```
swat    stream  tcp      nowait/400    root    /usr/local/sbin/swat
```

Как описано в Прим. 23-1, после изменения настроек **inetd** необходимо отправить HangUP сигнал.

Как только **swat** был включен `inetd.conf`, вы можете использовать браузер для подключения к `http://localhost:901`. Сначала необходимо зарегистрироваться с системной учетной записью `root`.

После успешного входа на основную страницу настройки **Samba**, вы можете просмотреть документацию или начать настройку, нажав на кнопку **Globals**. Раздел **Globals** соответствует переменным, установленным в разделе `[global]` файла `/usr/local/etc/smb.conf`.

### 23.9.2.2. Глобальные настройки

Независимо от того, используете ли вы **swat**, или редактируете `/usr/local/etc/smb.conf` непосредственно, первые директивы, которые вы скорее всего встретите при настройке **Samba**, будут следующими:

```
workgroup
```

Имя домена или рабочей группы NT для компьютеров, которые будут получать доступ к этому серверу.

```
netbios name
```

Устанавливает имя NetBIOS, под которым будет работать **Samba** сервер. По умолчанию оно устанавливается равным первому компоненту DNS имени хоста.

```
server string
```

Устанавливает строку, которая будет показана командой `net view` и некоторыми другими сетевыми инструментами, которые отображают строку описания сервера.

### 23.9.2.3. Настройки безопасности

Две из наиболее важных настроек в `/usr/local/etc/smb.conf` отвечают за выбор модели безопасности и за формат паролей для клиентов. Эти параметры контролируются следующими директивами:

`security`

Два наиболее часто используемых параметра это `security = share` и `security = user`. Если имена пользователей для клиентов совпадают с их именами на компьютере FreeBSD, вы возможно захотите включить безопасность уровня пользователя (`user`). Это политика безопасности по умолчанию, она требует, чтобы клиент авторизовался перед доступом к совместно используемым ресурсам.

На уровне безопасности `share` клиенту не требуется входить на сервер перед подключением к ресурсу. Эта модель безопасности использовалась по умолчанию в старых версиях **Samba**.

`passdb backend`

**Samba** поддерживает несколько различных подсистем аутентификации. Вы можете аутентифицировать клиентов с помощью LDAP, NIS+, базы данных SQL, или через модифицированный файл паролей. Метод аутентификации по умолчанию `smbpasswd`, и здесь рассматривается только он.

Предполагая, что используется подсистема по умолчанию `smbpasswd`, необходимо создать файл `/usr/local/private/smbpasswd`, чтобы **Samba** могла аутентифицировать клиентов. Если вы хотите разрешить всем учетным записям UNIX доступ с Windows клиентов, используйте следующую команду:

```
# grep -v "^#" /etc/passwd | make_smbpasswd > /usr/local/private/smbpasswd
# chmod 600 /usr/local/private/smbpasswd
```

Обратитесь к документации на **Samba** за дополнительной информацией о параметрах настройки. Основные настройки, рассмотренные здесь, достаточны для первого запуска **Samba**.

### 23.9.3. Запуск Samba

Для запуска **Samba** при загрузке системы, добавьте в `/etc/rc.conf` следующую строку:

```
samba_enable="YES"
```

Затем вы можете запустить **Samba** в любой момент, набрав:

```
# /usr/local/etc/rc.d/samba.sh start
Starting SAMBA: removing stale tdb's :
Starting nmbd.
Starting smbd.
```

**Samba** состоит из трех отдельных демонов. Вы можете видеть, что **nmbd** и **smbd** запускаются скриптом `samba.sh`. Если вы включили сервис разрешения имен `winbind` в `smb.conf`, то увидите также запуск демона **winbindd**.

Вы можете остановить **Samba** в любой момент, набрав:

```
# /usr/local/etc/rc.d/samba.sh stop
```

**Samba** это сложный программный набор с функциональностью, позволяющей полную интеграцию в сети Microsoft Windows. За дальнейшей информацией о функциях, выходящих за рамки описанной здесь базовой установки, обращайтесь к <http://www.samba.org>.

## 23.10. Протокол передачи файлов (FTP)

*Предоставил Murray Stokely.*

### 23.10.1. Обзор

Протокол передачи файлов (File Transfer Protocol, FTP) дает пользователям простой путь передачи файлов на и с FTP сервера. В FreeBSD серверная программа FTP, **ftpd**, включена в базовую систему. Это упрощает настройку и администрирование FTP сервера в FreeBSD.

### 23.10.2. Настройка

Наиболее важный шаг заключается в определении того, каким учетным записям будет позволено получать доступ к FTP серверу. В обычной системе FreeBSD есть множество системных учетных записей, используемых различными даемонами, но пользователям должно быть запрещен вход с использованием этих учетных записей. В файле `/etc/ftpusers` находится список пользователей, которым запрещен доступ по FTP. По умолчанию он включает упомянутые системные учетные записи, но в него можно добавить и определенных пользователей, которым будет запрещен доступ по FTP.

Вам может понадобиться ограничить доступ определенных пользователей без полного запрета использования FTP. Это можно сделать через файл `/etc/ftpchroot`. В нем находится список пользователей и групп, к которым применяется ограничение доступа. На странице справочника `ftpchroot(5)` дана подробная информация, и она не будет дублироваться здесь.

Если вы захотите разрешить анонимный FTP доступ на сервер, в системе FreeBSD необходимо создать пользователя `ftp`. Этот пользователь сможет входить на FTP сервер с именем пользователя `ftp` или `anonymous`, с любым паролем (существует соглашение об использовании почтового адреса пользователя в качестве пароля). FTP сервер выполнит `chroot(2)` при входе пользователя `anonymous` для ограничения доступа только домашним каталогом пользователя `ftp`.

Существуют два текстовых файла, определяющих сообщение, отправляемое FTP клиентам. Содержимое файла `/etc/ftpwelcome` будет выведено пользователям перед приглашением на вход. После успешного входа будет выведено содержимое файла `/etc/ftpmotd`. Обратите внимание, что путь к этому файлу задается относительно домашнего каталога пользователя, так что анонимным пользователям будет отправляться `~ftp/etc/ftpmotd`.

Как только FTP сервер был правильно настроен, он должен быть включен в `/etc/inetd.conf`. Все, что необходимо, это удалить символ комментария `#` из начала существующей строки **ftpd**:

```
ftp stream tcp nowait root /usr/libexec/ftpd ftpd -l
```

Как описано в Прим. 23-1, сигнал HangUP должен быть отправлен **inetd** после того, как этот файл настройки был изменен.

Теперь вы можете войти на FTP сервер, введя:

```
% ftp localhost
```

### 23.10.3. Поддержка

Для протоколирования даемон **ftpd** использует сообщения syslog(3). По умолчанию, syslog(3) поместит сообщения, относящиеся к FTP, в файл `/var/log/xferlog`. Местоположение лог файла FTP может быть изменено путем изменения следующей строки в файле `/etc/syslog.conf`:

```
ftp.info          /var/log/xferlog
```

Учитывайте потенциальные проблемы, возникающие с анонимным FTP сервером. В частности, вы должны дважды подумать, прежде чем позволить анонимным пользователям загружать файлы на сервер. Вы можете обнаружить, что FTP сайт стал форумом, на котором происходит обмен нелицензионным коммерческим программным обеспечением или чем-то еще хуже. Если вам необходимо разрешить анонимную выгрузку файлов на FTP, права должны быть настроены таким образом, чтобы эти файлы не могли прочитать другие анонимные пользователи до их рассмотрения администратором.

## 23.11. Синхронизация часов через NTP

*Текст предоставил Том Хукис.*

### 23.11.1. Обзор

С течением времени часы компьютера имеют тенденцию отставать. Network Time Protocol - Сетевой Протокол Времени (NTP) является одним из способов вести точное время.

Многие сервисы Интернет опираются или сильно зависят от точности часов компьютеров. К примеру, веб-сервер может получать запрос на посылку файла, который был недавно модифицирован. В локальной сети необходимо, чтобы часы компьютеров, совместно использующих файлы, были синхронизированы, чтобы время модификации файлов устанавливалось правильно. Такие службы, как cron(8), также зависят от правильности установки системных часов, поскольку запускают команды в определенное время.

FreeBSD поставляется с сервером NTP `ntpd(8)`, который можно использовать для опроса других серверов NTP для установки часов на вашей машине или предоставления услуг точного времени.

### 23.11.2. Выбор подходящих серверов NTP

Для синхронизации ваших часов вам нужно найти для использования один или большее количество серверов NTP. Ваш сетевой администратор или провайдер могут иметь сервер NTP для этой цели—обратитесь к ним, так ли это в вашем случае. Существует онлайн список общедоступных

серверов NTP (<http://ntp.isc.org/bin/view/Servers/WebHome>), которым можно воспользоваться для поиска ближайшего к вам сервера NTP. Не забудьте выяснить политику выбранного вами сервера и спросить разрешения, если это требуется.

Выбор нескольких несвязанных серверов NTP является хорошей идеей в том случае, если один из используемых вами серверов станет недоступным или его часы неточны. `ntpd(8)` использует ответы, которые он получает от других серверов с умом—он делает предпочтение надежным серверам.

### 23.11.3. Настройка вашей машины

#### 23.11.3.1. Базовая конфигурация

Если вам нужно только синхронизировать ваши часы при загрузке машины, вы можете воспользоваться утилитой `ntpdate(8)`. Это может подойти для некоторых настольных машин, которые часто перезагружаются и только требуют изредка синхронизироваться, но на большинстве машин должен работать `ntpd(8)`.

Использование `ntpdate(8)` при загрузке также хорошо для машин, на которых запущен демон `ntpd(8)`. Программа `ntpd(8)` изменяет время постепенно, тогда как `ntpdate(8)` устанавливает время вне зависимости от того, насколько велика разница между текущим временем машины и точным временем.

Для включения `ntpdate(8)` во время загрузки, добавьте строчку `ntpdate_enable="YES"` в файл `/etc/rc.conf`. Вам также потребуется указать все серверы, с которыми вы хотите синхронизироваться, и все параметры, которые передаются в `ntpdate(8)`, в `ntpdate_flags`.

#### 23.11.3.2. Общие настройки

NTP настраивается в файле `/etc/ntp.conf`, формат которого описан в `ntp.conf(5)`. Вот простой пример:

```
server ntplocal.example.com prefer
server timeserver.example.org
server ntp2a.example.net

driftfile /var/db/ntp.drift
```

Параметр `server` задает, какие серверы будут использоваться, по одному в каждой строке. Если сервер задан с аргументом `prefer`, как `ntplocal.example.com`, то этому серверу отдается предпочтение перед остальными. Ответ от предпочтительного сервера будет отброшен, если он значительно отличается от ответов других серверов, в противном случае он будет использоваться безотносительно к другим ответам. Аргумент `prefer` обычно используется для серверов NTP, о которых известно, что они очень точны, такими, на которых используется специальное оборудование точного времени.

Параметр `driftfile` задает файл, который используется для хранения смещения частоты системных часов. Программа `ntpd(8)` использует его для автоматической компенсации естественного смещения часов, позволяя ему поддерживать достаточно правильную настройку, даже если он на некоторый период отключается от внешнего источника информации о времени.

Параметр `driftfile` задает, какой файл используется для сохранения информации о предыдущих ответах от серверов NTP, которые вы используете. Этот файл содержит внутреннюю информацию для NTP. Он не должен изменяться никакими другими процессами.



### 23.11.3.3. Управление доступом к вашему серверу

По умолчанию ваш сервер NTP будет доступен всем хостам в Интернет. Параметр `restrict` в файле `/etc/ntp.conf` позволяет вам контролировать, какие машины могут обращаться к вашему серверу.

Если вы хотите запретить всем машинам обращаться к вашему серверу NTP, добавьте следующую строку в файл `/etc/ntp.conf`:

```
restrict default ignore
```

Если вы хотите разрешить синхронизировать свои часы с вашим сервером только машинам в вашей сети, но запретить им настраивать сервер или быть равноправными участниками синхронизации времени, то вместо указанной добавьте строчку

```
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```

где `192.168.1.0` является адресом IP вашей сети, а `255.255.255.0` её сетевой маской.

`/etc/ntp.conf` может содержать несколько директив `restrict`. Для получения подробной информации обратитесь к подразделу `Access Control Support` (Поддержка Управления Доступом) в `ntp.conf(5)`.

### 23.11.4. Запуск сервера NTP

Для того, чтобы сервер NTP запускался при загрузке, добавьте строку `xntpd_enable="YES"` в файл `/etc/rc.conf`. Если вы хотите передать дополнительные опции в `ntpd(8)`, то отредактируйте параметр `xntpd_flags` в файле `/etc/rc.conf`.

Для запуска сервера без перезагрузки вашей машины, выполните команду `ntpd`, не забыв задать дополнительные параметры из переменной `xntpd_flags` в файле `/etc/rc.conf`. К примеру:

```
# ntpd -p /var/run/ntpd.pid
```

**Замечание:** Во FreeBSD 5.X различные параметры из `/etc/rc.conf` были переименованы. В частности, в списке параметров вам необходимо заменить каждую строчку `xntpd` на `ntpd`.

### 23.11.5. Использование `ntpd` с временным подключением к Интернет

Для нормальной работы программе `ntpd(8)` не требуется постоянное подключение к Интернет. Однако если ваше временное подключение к Интернет настроено для дозвона по требованию, хорошо бы запретить трафику NTP вызывать дозвон или поддерживать соединение постоянно. Если вы используете пользовательский PPP, то можете воспользоваться директивами `filter` в файле `/etc/ppp/ppp.conf`. К примеру:

```
set filter dial 0 deny udp src eq 123
# Prevent NTP traffic from initiating dial out
set filter dial 1 permit 0 0
set filter alive 0 deny udp src eq 123
```

```
# Prevent incoming NTP traffic from keeping the connection open
set filter alive 1 deny udp dst eq 123
# Prevent outgoing NTP traffic from keeping the connection open
set filter alive 2 permit 0/0 0/0
```

Более подробную информацию можно найти в разделе `PACKET FILTERING` (ФИЛЬТРАЦИЯ ПАКЕТОВ) в `ppp(8)`, а примеры в `/usr/share/examples/ppp/`.

**Замечание:** Некоторые провайдеры Интернет блокируют трафик по портам с маленькими номерами, что приводит к неработоспособности NTP, так как ответы никогда не достигают вашей машины.

### 23.11.6. Дополнительная литература

Документация по серверу NTP может быть найдена в каталоге `/usr/share/doc/ntp/` в формате HTML.

# Глава 24. Сложные вопросы работы в сети

*Перевод на русский язык: Андрей Захватов.*

## 24.1. Краткий обзор

Эта глава охватывает множество различных сетевых тематик повышенной сложности.

После чтения этой главы вы будете знать:

- Основные понятия о маршрутизации и маршрутах.
- Как настроить IEEE 802.11 и Bluetooth®.
- Как заставить FreeBSD работать в качестве сетевого моста.
- Как настроить загрузку по сети для бездисковой машины.
- Как настроить трансляцию сетевых адресов.
- Как соединить два компьютера посредством PLIP.
- Как настроить IPv6 на машине FreeBSD.
- Как настроить ATM в FreeBSD 5.X.

Перед чтением этой главы вы должны:

- Понимать основы работы скриптов `/etc/rc`.
- Свободно владеть основными сетевыми терминами.
- Знать как настраивать и устанавливать новое ядро FreeBSD (Гл. 8).
- Знать как устанавливать дополнительное программное обеспечение сторонних разработчиков (Гл. 4).

## 24.2. Сетевые шлюзы и маршруты

*Текст предоставил Coranth Gryphon.*

Чтобы некоторая машина могла найти в сети другую, должен иметься механизм описания того, как добраться от одной машине к другой. Такой механизм называется *маршрутизацией*. "Маршрут" задаётся парой адресов: "адресом назначения" (destination) и "сетевым шлюзом" (gateway). Эта пара указывает на то, что если Вы пытаетесь соединиться с *адресом назначения*, то вам нужно устанавливать связь через "сетевой шлюз". Существует три типа адресов назначения: отдельные хосты, подсети и "маршрут по умолчанию" (default). "Маршрут по умолчанию" (default route) используется, если не подходит ни один из других маршрутов. Мы поговорим немного подробнее о маршрутах по умолчанию позже. Также имеется и три типа сетевых шлюзов: отдельные хосты, интерфейсы (также называемые "подключениями" (links)) и аппаратные адреса Ethernet (MAC-адреса).

### 24.2.1. Пример

Для иллюстрации различных аспектов маршрутизации мы будем использовать следующий пример использования команды `netstat`:

```
% netstat -r
Routing tables
```

Destination	Gateway	Flags	Refs	Use	Netif	Expire
default	outside-gw	UGSc	37	418	ppp0	
localhost	localhost	UH 0	181	lo0		
test0	0:e0:b5:36:cf:4f	UHLW 5	63288	ed0	77	
10.20.30.255	link#1	UHLW 1	2421			
example.com	link#1	UC 0 0				
host1	0:e0:a8:37:8:1e	UHLW 3	4601	lo0		
host2	0:e0:a8:37:8:1e	UHLW 0 5	lo0 =>			
host2.example.com	link#1	UC 0 0				
224	link#1	UC 0 0				

В первых двух строках задаются маршрут по умолчанию (который будет описан в следующем разделе) и маршрут на `localhost`.

Интерфейс (колонокка `Netif`), который указан в этой таблице маршрутов для использования с `localhost` и который назван `lo0`, имеет также второе название, устройство `loopback`. Это значит сохранение всего трафика для указанного адреса назначения внутри, без отправки его по сети, так как он все равно будет направлен туда, где был создан.

Следующими выделяющимися адресами являются адреса, начинающиеся с `0:e0:...` Это аппаратные адреса Ethernet, или MAC-адреса. FreeBSD будет автоматически распознавать любой хост (в нашем примере это `test0`) в локальной сети Ethernet и добавит маршрут для этого хоста, указывающий непосредственно на интерфейс Ethernet, `ed0`. С этим типом маршрута также связан параметр таймаута (колонокка `Expire`), используемый в случае неудачной попытки услышать этот хост в течении некоторого периода времени. Если такое происходит, то маршрут до этого хоста будет автоматически удалён. Такие хосты поддерживаются при помощи механизма, известного как RIP (Routing Information Protocol), который вычисляет маршруты к хостам локальной сети при помощи определения кратчайшего расстояния.

FreeBSD добавит также все маршруты к подсетям для локальных подсетей (`10.20.30.255` является широковещательным адресом для подсети `10.20.30`, а имя `example.com` является именем домена, связанным с этой подсетью). Назначение `link#1` соответствует первому адаптеру Ethernet в машине. Отметьте отсутствие дополнительного интерфейса для этих строк.

В обеих этих группах (хосты и подсети локальной сети) маршруты конфигурируются автоматически демоном, который называется **routed**. Если он не запущен, то будут существовать только статически заданные (то есть введенные явно) маршруты.

Строка `host1` относится к нашему хосту, который известен по адресу Ethernet. Так как мы являемся посылающим хостом, FreeBSD знает, что нужно использовать `loopback`-интерфейс (`lo0`) вместо того, чтобы осуществлять отсылку в интерфейс Ethernet.

Две строки `host2` являются примером того, что происходит при использовании алиасов в команде `ifconfig(8)` (обратитесь к разделу об Ethernet для объяснения того, почему мы это делаем). Символ `=>` после интерфейса `lo0` указывает на то, что мы используем не просто интерфейс `loopback` (так как это

адрес, обозначающий локальный хост), но к тому же это алиас. Такие маршруты появляются только на хосте, поддерживающем алиасы; для всех остальных хостов в локальной сети для таких маршрутов будут показаны просто строчки link#1.

Последняя строчка (подсеть назначения 224) имеет отношение к многоадресной посылке, которая будет рассмотрена в другом разделе.

И наконец, различные атрибуты каждого маршрута перечисляются в колонке `Flags`. Ниже приводится краткая таблица некоторых из этих флагов и их значений:

U	Up: Маршрут актуален.
H	Host: Адресом назначения является отдельный хост.
G	Gateway: Посылать все для этого адреса назначения на указанную удаленную систему, которая будет сама определять дальнейший путь прохождения информации.
S	Static: Маршрут был настроен вручную, а не автоматически сгенерирован системой.
C	Clone: Новый маршрут сгенерирован на основе указанного для машин, к которым мы подключены. Такой тип маршрута обычно используется для локальных сетей.
W	WasCloned: Указывает на то, что маршрут был автоматически сконфигурирован на основе маршрута в локальной сети (Clone).
L	Link: Маршрут включает ссылку на аппаратный адрес Ethernet.

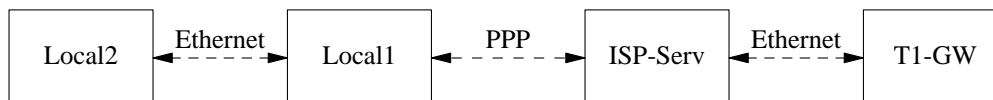
### 24.2.2. Маршруты по умолчанию

Когда локальной системе нужно установить соединение с удаленным хостом, она обращается к таблице маршрутов для того, чтобы определить, существует ли такой маршрут. Если удаленный хост попадает в подсеть, для которой известен способ ее достижения (маршруты типа Cloned), то система определяет возможность подключиться к ней по этому интерфейсу.

Если все известные маршруты не подходят, у системы имеется последняя возможность: маршрут "default". Это маршрут с особым типом сетевого шлюза (обычно единственным, присутствующим в системе), и в поле флагов он всегда помечен как `s`. Для хостов в локальной сети этот сетевой шлюз указывает на машину, имеющую прямое подключение к внешнему миру (неважно, используется ли связь по протоколу PPP, канал DSL, кабельный модем, T1 или какой-то другой сетевой интерфейс).

Если вы настраиваете маршрут по умолчанию на машине, которая сама является сетевым шлюзом во внешний мир, то маршрутом по умолчанию будет являться сетевой шлюз у Вашего провайдера Интернет (ISP).

Давайте взглянем на примеры маршрутов по умолчанию. Вот типичная конфигурация:



Хосты Local1 и Local2 находятся в нашей сети. Local1 подключён к ISP через коммутируемое соединение по протоколу PPP. Этот компьютер с сервером PPP подключён посредством локальной сети к другому шлюзовому компьютеру через внешний интерфейс самого ISP к Интернет.

Маршруты по умолчанию для каждой из ваших машин будут следующими:

Хост	Маршрут по умолчанию	Интерфейс
Local2	Local1	Ethernet
Local1	T1-GW	PPP

Часто задаётся вопрос "Почему (или каким образом) в качестве шлюза по умолчанию для машины Local1 мы указываем T1-GW, а не сервер провайдера, к которому подключаемся?".

Запомните, что из-за использования PPP-интерфейсом адреса в сети провайдера Интернет с вашей стороны соединения, маршруты для всех других машин в локальной сети провайдера будут сгенерированы автоматически. Таким образом, вы уже будете знать, как достичь машины T1-GW, так что нет нужды в промежуточной точке при посылке трафика к серверу ISP.

В локальных сетях адрес X.X.X.1 часто используется в качестве адреса сетевого шлюза. Тогда (при использовании того же самого примера) если пространство адресов класса C вашей локальной сети было задано как 10.20.30, а ваш провайдер использует 10.9.9, то маршруты по умолчанию будут такие:

Хост	Маршрут по умолчанию
Local2 (10.20.30.2)	Local1 (10.20.30.1)
Local1 (10.20.30.1, 10.9.9.30)	T1-GW (10.9.9.1)

Вы можете легко задать используемый по умолчанию маршрутизатор посредством файла `/etc/rc.conf`. В нашем примере на машине Local2 мы добавили такую строку в файл `/etc/rc.conf`:

```
defaultrouter="10.20.30.1"
```

Это также возможно сделать и непосредственно из командной строки при помощи команды `route(8)`:

```
# route add default 10.20.30.1
```

Для получения дополнительной информации об управлении таблицами маршрутизации, обратитесь к справочной странице по команде `route(8)`.

### 24.2.3. Хосты с двойным подключением

Есть еще один тип подключения, который мы должны рассмотреть, и это случай, когда хост находится в двух различных сетях. Технически, любая машина, работающая как сетевой шлюз (в примере выше использовалось PPP-соединение), считается хостом с двойным подключением. Однако этот термин реально используется для описания машины, находящейся в двух локальных сетях.

В одном случае у машины имеется два адаптера Ethernet, каждый имеющий адрес в разделенных подсетях. Как альтернативу можно рассмотреть вариант с одним Ethernet-адаптером и использованием алиасов в команде `ifconfig(8)`. В первом случае используются два физически разделённые сети Ethernet, в последнем имеется один физический сегмент сети, но две логически разделённые подсети.

В любом случае таблицы маршрутизации настраиваются так, что для каждой подсети эта машина определена как шлюз (входной маршрут) в другую подсеть. Такая конфигурация, при которой машина выступает в роли маршрутизатора между двумя подсетями, часто используется, если нужно

реализовать систему безопасности на основе фильтрации пакетов или функций брандмауэра в одном или обоих направлениях.

Если вы хотите, чтобы эта машина действительно перемещала пакеты между двумя интерфейсами, то вам нужно указать FreeBSD на включение этой функции. Обратитесь к следующей главе, чтобы узнать, как это сделать.

#### 24.2.4. Построение маршрутизатора

Сетевой маршрутизатор является обычной системой, которая пересылает пакеты с одного интерфейса на другой. Стандарты Интернет и хорошая инженерная практика не позволяют Проекту FreeBSD включать эту функцию по умолчанию во FreeBSD. Вы можете включить эту возможность, изменив значение следующей переменной в YES в файле rc.conf(5):

```
gateway_enable=YES          # Set to YES if this host will be a gateway
```

Этот параметр изменит значение sysctl(8)-переменной `net.inet.ip.forwarding` в 1. Если вам временно нужно выключить маршрутизацию, вы можете на время сбросить это значение в 0.

Вашему новому маршрутизатору нужна информация о маршрутах для того, чтобы знать, куда пересылать трафик. Если ваша сеть достаточно проста, то вы можете использовать статические маршруты. С FreeBSD также поставляется стандартный демон BSD для маршрутизации `routed(8)`, который умеет работать с RIP (как версии 1, так и версии 2) и IRDP. Поддержка BGP v4, OSPF v2 и других сложных протоколов маршрутизации имеется в пакете `net/zebra`. Также существуют и коммерческие продукты, применяемые как более комплексное решение проблемы маршрутизации в сети, такие как **GateD**.

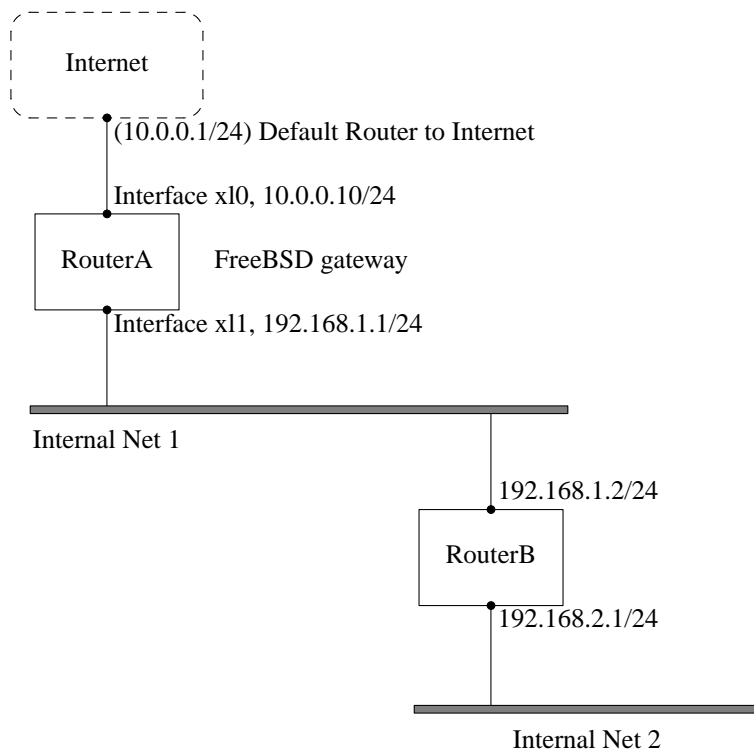
Даже когда FreeBSD настроена таким образом, она не полностью соответствует стандартным требованиям Интернет для маршрутизаторов. Однако для обычного использования такое неполное соответствие достаточно.

#### 24.2.5. Настройка статических маршрутов

*Предоставил Al Hoang.*

##### 24.2.5.1. Ручная настройка

Предположим, что у нас есть следующая сеть:



В этом сценарии, RouterA это наш компьютер с FreeBSD, который выступает в качестве маршрутизатора в сеть Интернет. Его маршрут по умолчанию настроен на 10.0.0.1, что позволяет ему соединяться с внешним миром. Мы будем предполагать, что RouterB уже правильно настроен и знает все необходимые маршруты (на этом рисунке все просто; добавьте на RouterB маршрут по умолчанию, используя 192.168.1.1 в качестве шлюза).

Если мы посмотрим на таблицу маршрутизации RouterA, то увидим примерно следующее:

```
% netstat -nr
Routing tables
```

```
Internet:
Destination      Gateway          Flags    Refs      Use  Netif  Expire
default          10.0.0.1        UGS      0        49378   x10
127.0.0.1        127.0.0.1       UH       0          6    lo0
10.0.0/24        link#1          UC       0          0    x10
192.168.1/24     link#2          UC       0          0    x11
```

С текущей таблицей маршрутизации RouterA не сможет достичь внутренней сети 2 (Internal Net 2). Один из способов обхода этой проблемы — добавление маршрута вручную. Следующая команда добавляет внутреннюю сеть 2 к таблице маршрутизации RouterA с 192.168.1.2 в качестве следующего узла:

```
# route add -net 192.168.2.0/24 192.168.1.2
```

Теперь RouterA сможет достичь любого хоста в сети 192.168.2.0/24.



### 24.2.5.2. Постоянная конфигурация

Предыдущий пример прекрасно подходит для настройки статического маршрута в работающей системе. Однако, проблема заключается в том, что маршрутная информация не сохранится после перезагрузки FreeBSD. Способ сохранения добавленного маршрута заключается в добавлении его в файл `/etc/rc.conf`:

```
# Добавление статического маршрута в Internal Net 2
static_routes="internalnet2"
route_internalnet2="-net 192.168.2.0/24 192.168.1.2"
```

В переменной `static_routes` находятся строки, разделенные пробелами. Каждая строка означает имя маршрута. В примере выше в `static_routes` есть только одна строка, это `internalnet2`. Затем мы добавили переменную `route_internalnet2`, куда помещены все параметры, которые необходимо передать команде `route(8)`. В примере выше была использована команда:

```
# route add -net 192.168.2.0/24 192.168.1.2
```

поэтому нам потребуется `"-net 192.168.2.0/24 192.168.1.2"`.

Как было сказано выше, мы можем добавить в `static_routes` более чем одну строку. Это позволит создать несколько статических маршрутов. В следующем примере показано добавление маршрутов для сетей `192.168.0.0/24` и `192.168.1.0/24` (этот маршрутизатор не показан на рисунке выше:

```
static_routes="net1 net2"
route_net1="-net 192.168.0.0/24 192.168.0.1"
route_net2="-net 192.168.1.0/24 192.168.1.1"
```

### 24.2.6. Распространение маршрутов

Мы уже говорили о том, как мы задаем наши маршруты во внешний мир, но не упоминали о том, как внешний мир находит нас.

Мы уже знаем, что таблицы маршрутизации могут быть настроены так, что весь трафик для некоторого диапазона адресов (в нашем примере это подсеть класса C) может быть направлен заданному хосту в той сети, которая будет перенаправлять входящие пакеты дальше.

При получении адресного пространства, выделенного Вашей сети, Ваш провайдер настроит свои таблицы маршрутизации так, что весь трафик для Вашей подсети будет пересылаться по PPP-соединению к Вашей сети. Но как серверы по всей стране узнают, что Ваш трафик нужно посылать Вашему ISP?

Существует система (подобная распределению информации DNS), которая отслеживает все назначенные пространства адресов и определяет точку подключения к магистрали Интернет. "Магистралью" называют главные каналы, по которым идет трафик Интернет внутри страны и по всему миру. Каждая магистральная машина имеет копию основного набора таблиц, согласно которой трафик для конкретной сети направляется по конкретному магистральному каналу, и затем, передаваясь по цепочке провайдеров, он достигает вашей сети.

Задачей вашего провайдера является объявить на магистрали о том, что он отвечает за подключение (и поэтому на него указывает маршрут) вашей сети. Этот процесс называется распространением маршрута.

### 24.2.7. Устранение неполадок

Иногда с распространением маршрута возникают проблемы, и некоторые сайты не могут к вам подключиться. Наверное, самой полезной командой для определения точки неверной работы маршрутизации является `tracroute(8)`. Она также полезна и когда вы сами не можете подключиться к удаленной машине (то есть команда `ping(8)` не срабатывает).

Команда `tracroute(8)` запускается с именем удаленного хоста, с которым вы хотите установить соединение, в качестве параметра. Она показывает промежуточные сетевые шлюзы по пути следования, в конце концов достигая адрес назначения или прерывая свою работу из-за отсутствия соединения.

За дополнительной информацией обратитесь к странице Справочника по `tracroute(8)`.

### 24.2.8. Маршрутизация многоадресного трафика

FreeBSD изначально поддерживает как приложения, работающие с многоадресным трафиком, так и его маршрутизацию. Такие приложения не требуют особой настройки FreeBSD; обычно они работают сразу. Для маршрутизации многоадресного трафика требуется, чтобы поддержка этого была включена в ядро:

```
options MROUTING
```

Кроме того, демон многоадресной маршрутизации, `mROUTED(8)`, должен быть настроен посредством файла `/etc/mROUTED.conf` на использование туннелей и DVMRP. Дополнительную информацию о настройке многоадресного трафика можно найти на страницах справочной системы, посвящённых даемону `mROUTED(8)`.

## 24.3. Беспроводные сети

*Текст предоставил Перевёл на русский язык Eric Anderson, Андрей Захватов.*

### 24.3.1. Введение

Было бы весьма полезным иметь возможность использовать компьютер без хлопот, связанных с постоянно подключенным сетевым кабелем. FreeBSD может использоваться как клиент беспроводной сети, и даже в качестве "точки доступа" к ней.

### 24.3.2. Режимы работы беспроводной связи

Существуют два варианта конфигурации устройств беспроводного доступа 802.11: BSS и IBSS.

#### 24.3.2.1. Режим BSS

Режим BSS является наиболее часто используемым. Режим BSS также называют режимом инфраструктуры. В этом режиме несколько точек доступа беспроводной сети подключаются к

проводной сети передачи данных. Каждое беспроводная сеть имеет собственное имя. Это имя является идентификатором **SSID** сети.

Клиенты беспроводной сети подключаются к этим точкам доступа беспроводной сети. Стандарт IEEE 802.11 определяет протокол, используемый для связи в беспроводных сетях. Клиент сети беспроводного доступа может подключаться к некоторой сети, если задан её **SSID**. Клиент может также подключаться к любой сети, если **SSID** не задан.

#### 24.3.2.2. Режим IBSS

Режим IBSS, также называемый ad-hoc, предназначен для соединений точка-точка. На самом деле существуют два типа режима ad-hoc. Один из них является режимом IBSS, называемый также режимом ad-hoc или IEEE ad-hoc. Этот режим определён стандартами IEEE 802.11. Второй режим называется демонстрационным режимом ad-hoc, или Lucent ad-hoc (или, иногда неправильно, режимом ad-hoc). Это старый, существовавший до появления 802.11, режим ad-hoc, и он должен использоваться только для старых сетей. В дальнейшем мы не будем рассматривать ни один из режимов ad-hoc.

### 24.3.3. Режим инфраструктуры

#### 24.3.3.1. Точки доступа

Точки доступа представляют собой беспроводные сетевые устройства, позволяющие одному или большему количеству клиентов беспроводной сети использовать эти устройства в качестве центрального сетевого концентратора. При использовании точки доступа все клиенты работают через неё. Зачастую используются несколько точек доступа для полного покрытия беспроводной сетью некоторой зоны, такой, как дом, офис или парк.

Точки доступа обычно имеют несколько подключений к сети: адаптер беспроводной связи и один или большее количество сетевых ethernet-адаптеров для подключения к остальной части сети.

Точки доступа могут быть либо приобретены уже настроенными, либо вы можете создать собственную при помощи FreeBSD и поддерживаемого адаптера беспроводной связи. Несколько производителей выпускают точки беспроводного доступа и адаптеры беспроводной связи с различными возможностями.

#### 24.3.3.2. Построение точки доступа с FreeBSD

##### 24.3.3.2.1. Требования

Для того, чтобы создать беспроводную точку доступа на FreeBSD, вам нужно иметь совместимый адаптер беспроводной связи. На данный момент поддерживаются адаптеры только на основе набора микросхем Prism. Вам также потребуется поддерживаемый FreeBSD адаптер проводной сети (найти такой будет нетрудно, FreeBSD поддерживает множество различных устройств). В этом руководстве мы будем полагать, что вы будете строить сетевой мост (bridge(4)) для пропуска всего трафика между устройством беспроводной связи и сетью, подключенной к обычному Ethernet-адаптеру.

Функциональность hostap, которая используется FreeBSD для организации точки доступа, работает лучше всего с некоторыми версиями микрокода. Адаптеры Prism 2 должны использовать микрокод

версии 1.3.4 или более новый. Адаптеры Prism 2.5 и Prism 3 должны использовать микрокод версии 1.4.9. Более старые версии микрокода могут работать нормально, а могут и некорректно. В настоящее время единственным способом обновления адаптеров является использование утилит обновления для Windows, которые можно получить у производителя ваших адаптеров.

#### 24.3.3.2.2. Настройка

Первым делом убедитесь, что ваша система распознаёт адаптер беспроводной связи:

```
# ifconfig -a
wi0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet6 fe80::202:2dff:fe2d:c938%wi0 prefixlen 64 scopeid 0x7
    inet 0.0.0.0 netmask 0xff000000 broadcast 255.255.255.255
    ether 00:09:2d:2d:c9:50
    media: IEEE 802.11 Wireless Ethernet autoselect (DS/2Mbps)
    status: no carrier
    ssid ""
    stationname "FreeBSD Wireless node"
    channel 10 authmode OPEN powersavemode OFF powersavesleep 100
    wepmode OFF weptxkey 1
```

На данном этапе не беспокойтесь о деталях, просто убедитесь, что выдаётся нечто, указывающее на установленный адаптер беспроводной связи. Если при этом у вас есть проблемы с недоступностью интерфейса беспроводной связи, и вы используете PC Card, то обратитесь к страницам справочной системы, описывающим `pccardc(8)` и `pccardd(8)` для получения более полной информации.

Теперь вам нужно загрузить модуль для подготовки той части FreeBSD, что отвечает за организацию сетевых мостов, для работы с точкой доступа. Для загрузки модуля `bridge(4)` просто выполните следующую команду:

```
# kldload bridge
```

При загрузке модуля никаких сообщений об ошибках быть не должно. Если это всё же произошло, вам может потребоваться вкомпилировать код для модуля `bridge(4)` в ядро. В этом вам должен помочь раздел этого Руководства об организации сетевых мостов.

Теперь, когда вы завершили с той частью, что касается организации сетевого моста, нам нужно указать ядру FreeBSD, какие интерфейсы должны объединяться в сетевом мосте. Это мы делаем при помощи `sysctl(8)`:

```
# sysctl net.link.ether.bridge=1
# sysctl net.link.ether.bridge_cfg="wi0,xl0"
# sysctl net.inet.ip.forwarding=1
```

Во FreeBSD 5.2-RELEASE и последующих версиях нужно использовать вместо указанных следующие параметры:

```
# sysctl net.link.ether.bridge.enable=1
# sysctl net.link.ether.bridge.config="wi0,xl0"
# sysctl net.inet.ip.forwarding=1
```

Теперь необходимо настроить адаптер беспроводной сети. Следующая команда заставит адаптер работать в режиме точки доступа:

```
# ifconfig wi0 ssid my_net channel 11 media DS/11Mbps mediaopt hostap up stationname "FreeBSD AP"
```

Строчка `ifconfig(8)` активизирует интерфейс `wi0`, конфигурирует его SSID как `my_net`, а имя станции как `FreeBSD AP`. `media DS/11Mbps` переводит адаптер в режим 11Mbps и нужен только для того, чтобы сработал параметр `mediaopt`. Параметр `mediaopt hostap` переводит интерфейс в режим точки доступа. Параметр `channel 11` задаёт использование канала 802.11b. Страница справки по команде `wicontrol(8)` перечисляет корректные значения каналов для ваших нужд.

Теперь у вас должна получиться полнофункциональная работающая точка доступа. Настоятельно советуем прочесть страницы справочной по `wicontrol(8)`, `ifconfig(8)`, и `wi(4)` для получения дополнительной информации.

Также полагаем, что вы прочтёте следующий раздел о шифровании.

#### 24.3.3.2.3. Информация о состоянии

После того, как точка доступа сконфигурирована и начала свою работу, операторам может понадобиться видеть клиентов, связанных с этой точкой. В любой момент оператор может набрать:

```
# wicontrol -l
1 station:
00:09:b7:7b:9d:16 asid=04c0, flags=3<ASSOC,AUTH>, caps=1<ESS>, rates=f<1M,2M,5.5M,11M>, sig=38/15
```

Это показывает, что имеется одна связанная станция с перечисленными характеристиками. Выдаваемое значение сигнала должно использоваться только как сравнительный индикатор его силы. Его перевод в dBm или другие единицы измерения различаются в разных версиях микрокода.

#### 24.3.3.3. Клиенты

Клиент в беспроводной сети представляет собой систему, которая обращается к точке доступа или непосредственно к другому клиенту.

Как правило, клиенты беспроводной сети имеют только один сетевой адаптер, а именно адаптер беспроводной сети.

Существует несколько различных способов конфигурации клиента беспроводной сети. Они основаны на различных режимах работы в беспроводной сети, обычно BSS (режим инфраструктуры, который требует точки доступа) или IBSS (ad-hoc или режим одноранговой сети). В нашем примере мы будем использовать самый популярный из них, режим BSS, для связи с точкой доступа.

##### 24.3.3.3.1. Требования

Существует только одно жёсткое условие для настройки FreeBSD в качестве клиента беспроводной сети. Вам нужен адаптер беспроводной связи, поддерживаемый FreeBSD.

### 24.3.3.3.2. Конфигурация FreeBSD как клиента беспроводной сети

Перед тем, как подключиться к беспроводной сети, вам нужно будет узнать о ней несколько вещей. В этом примере мы подключаемся к сети, которая называется *my\_net*, и шифрование в ней отключено.

**Замечание:** В этом примере мы не используем шифрование, но это небезопасно. В следующем разделе вы узнаете, как её включить, почему это так важно, и почему некоторые технологии шифрования всё же не могут полностью обеспечить вашу информационную безопасность.

Удостоверьтесь, что ваш адаптер распознаётся во FreeBSD:

```
# ifconfig -a
wi0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet6 fe80::202:2dff:fe2d:c938%wi0 prefixlen 64 scopeid 0x7
    inet 0.0.0.0 netmask 0xff000000 broadcast 255.255.255.255
    ether 00:09:2d:2d:c9:50
    media: IEEE 802.11 Wireless Ethernet autoselect (DS/2Mbps)
    status: no carrier
    ssid ""
    stationname "FreeBSD Wireless node"
    channel 10 authmode OPEN powersavemode OFF powersavesleep 100
    wepmode OFF weptxkey 1
```

Теперь мы можем изменить настройки адаптера на те, что соответствуют нашей сети:

```
# ifconfig wi0 inet 192.168.0.20 netmask 255.255.255.0 ssid my_net
```

Замените 192.168.0.20 и 255.255.255.0 на правильные IP-адрес и сетевую маску в вашей проводной сети. Запомните, что наша точка доступа выступает в роли моста для данных между беспроводной и проводной сетями, так что они будут доступны для других устройств, находящихся в сети, как будто они тоже находятся в проводной сети.

Как только вы это выполнили, то сможете получить ping от хостов в проводной сети, как будто вы подключены посредством обычных проводов.

Если вы столкнулись с проблемами при работе в беспроводной сети, удостоверьтесь, что вы ассоциированы (подключены) с точкой доступа:

```
# ifconfig wi0
```

должна выдать некоторую информацию, и вы должны увидеть:

```
status: associated
```

Если статус не будет соответствовать *associated*, это может значить, что вы оказались вне зоны досягаемости точки доступа, включили шифрование или, возможно, имеются проблемы с конфигурацией.

#### 24.3.3.4. Шифрование

Шифрование в беспроводной сети имеет важное значение, потому что у вас нет больше возможности ограничить сеть хорошо защищённой областью. Данные вашей беспроводной сети вещаются по всей окрестности, так что любой заинтересовавшийся может их считать. Вот здесь используется шифрование. Шифруя данные, посылаемые в эфир, вы делаете их прямой перехват гораздо более сложным для всех любопытных.

Двумя наиболее широко применяемыми способами шифрации данных между вашим клиентом и точкой доступа являются WEP и ipsec(4).

##### 24.3.3.4.1. WEP

WEP является сокращением от Wired Equivalency Protocol (Протокол Соответствия Проводной сети). WEP является попыткой сделать беспроводные сети такими же надёжными и безопасными, как проводные. К сожалению, он был взломан и сравнительно легко поддаётся вскрытию. Это означает также, что он не тот протокол, на который следует опираться, когда речь идёт о шифровании критически важных данных.

Он лучше, чем ничего, так что используйте следующую команду для включения WEP в вашей новой точке доступа FreeBSD:

```
# ifconfig wi0 inet up ssid my_net wepmode on wepkey 0x1234567890 media DS/11Mbps mediaopt hostap
```

Вы можете включить WEP на клиенте следующей командой:

```
# ifconfig wi0 inet 192.168.0.20 netmask 255.255.255.0 ssid my_net wepmode on wepkey 0x1234567890
```

Отметьте, что вы должны заменить `0x1234567890` на более уникальный ключ.

##### 24.3.3.4.2. IPsec

ipsec(4) является гораздо более надёжным и мощным средством шифрования данных в сети. Этот метод определённо является предпочтительным для шифрования данных в беспроводной сети. Более детально ознакомиться с безопасностью и применением ipsec(4) вы можете в разделе об IPsec этого Руководства.

#### 24.3.3.5. Утилиты

Имеется несколько утилит, которые можно использовать для настройки и отладки вашей беспроводной сети, и здесь мы попытаемся описать некоторые из них и что они могут делать.

##### 24.3.3.5.1. Пакет **bsd-airtools**

Пакет **bsd-airtools** представляет собой полный набор инструментов, включая инструменты для проверки беспроводной сети на предмет взлома WEP-ключа, обнаружения точки доступа и тому подобное.

Утилиты **bsd-airtools** можно установить из порта `net/bsd-airtools`. Информацию об установке портов можно найти в Главе Гл. 4 этого Руководства.

Программа `dstumbler` является инструментом, предназначенным для обнаружения точки доступа и выдачи отношения уровня сигнала к шуму. Если у вас с трудом получается запустить точку доступа, `dstumbler` может помочь вам начать.

Для тестирования информационной безопасности вашей беспроводной сети, вы можете воспользоваться набором `"dweputils"` (`dwepcrack`, `dwepdump` и `dwekeygen`), который может помочь понять, является ли WEP подходящим решением для обеспечения ваших потребностей в информационной безопасности.

#### 24.3.3.5.2. Утилиты `wicontrol`, `ancontrol` и `raycontrol`

Это инструменты, которые могут быть использованы для управления поведением адаптера беспроводной связи в сети. В примере выше мы выбирали `wicontrol(8)`, так как нашим адаптером беспроводной сети был интерфейс `wi0`. Если у вас установлено устройство беспроводного доступа от Cisco, этим интерфейсом будет `an0`, и тогда вы будете использовать `ancontrol(8)`.

#### 24.3.3.5.3. Команда `ifconfig`

Команда `ifconfig(8)` может использоваться для установки многих из тех параметров, что задаёт `wicontrol(8)`, однако работа с некоторыми параметрами в ней отсутствует. Обратитесь к `ifconfig(8)` для выяснения параметров и опций командной строки.

### 24.3.3.6. Поддерживаемые адаптеры

#### 24.3.3.6.1. Точки доступа

Единственными адаптерами, которые на данный момент поддерживаются в режиме BSS (как точка доступа), являются те устройства, что сделаны на основе набора микросхем Prism 2, 2.5 или 3). Полный список можно увидеть в `wi(4)`.

#### 24.3.3.6.2. Клиенты 802.11b

Практически все адаптеры беспроводной связи 802.11b на данный момент во FreeBSD поддерживаются. Большинство адаптеров, построенных на основе Prism, Spectrum24, Hermes, Aironet и Raylink, будут работать в качестве адаптера беспроводной сети в режиме IBSS (ad-hoc, одноранговая сеть и BSS).

#### 24.3.3.6.3. Клиенты 802.11a и 802.11g

Драйвер устройства `ath(4)` поддерживает 802.11a и 802.11g. Если ваша карта основана на чипсете Atheros, вы можете использовать этот драйвер.

К сожалению, все еще много производителей, не предоставляющих схематику своих драйверов сообществу open source, поскольку эта информация считается торговым секретом. Следовательно, у разработчиков FreeBSD и других операционных систем остается два варианта: разработать драйверы долгим и сложным методом обратного инжиниринга, или использовать существующие драйверы для платформ Microsoft Windows. Большинство разработчиков FreeBSD выбрали второй способ.



Благодаря усилиям Билла Паула (wpaul), начиная с FreeBSD 5.3-RELEASE существует "прозрачная" поддержка Network Driver Interface Specification (NDIS). FreeBSD NDISulator (известный также как Project Evil) преобразует бинарный драйвер Windows так, что он работает так же как и в Windows. Эта возможность все еще относительно новая, но в большинстве тестов она работает адекватно.

Для использования NDISulator потребуются три вещи:

1. Исходные тексты ядра
2. Бинарный драйвер Windows XP (расширение .SYS)
3. Файл конфигурации бинарного драйвера Windows XP (расширение .INF)

Вам может потребоваться компиляция драйвера оболочки мини порта ndis(4). Под root:

```
# cd /usr/src/sys/modules/ndis
# make && make install
```

Определите местоположение файлов для вашей карты. Обычно их можно найти на входящем в комплект CD или на веб сайте поставщика. В нашем примере используются файлы W32DRIVER.SYS и W32DRIVER.INF.

Следующий шаг это компиляция бинарного драйвера в загружаемый модуль ядра. Чтобы сделать это, сначала зайдите в каталог модуля if\_ndis и с правами root скопируйте туда драйверы Windows:

```
# cd /usr/src/sys/modules/if_ndis
# cp /path/to/driver/W32DRIVER.SYS ./
# cp /path/to/driver/W32DRIVER.INF ./
```

Теперь используйте утилиту ndiscvt для создания заголовка определения драйвера ndis\_driver\_data.h перед сборкой модуля:

```
# ndiscvt -i W32DRIVER.INF -s W32DRIVER.SYS -o ndis_driver_data.h
```

Параметры -i и -s задают соответственно файл настройки и бинарный файл. Мы используем параметр -o ndis\_driver\_data.h, поскольку Makefile при создании модуля будет обращаться именно к этому файлу.

**Замечание:** Некоторым драйверам Windows для работы требуются дополнительные файлы. Вы можете включить их параметром ndiscvt -f. Обратитесь к странице справочной системы ndiscvt(8) за дополнительной информацией.

Наконец, соберите и установите модуль драйвера:

```
# make && make install
```

Для использования драйвера необходимо загрузить соответствующие модули:

```
# kldload ndis
# kldload if_ndis
```

Первая команда загружает оболочку драйвера минипорта NDIS, вторая загружает собственно сетевой интерфейс. Проверьте `dmesg(8)` на предмет ошибок загрузки. Если все прошло хорошо, вывод должен быть примерно таким:

```
ndis0: <Wireless-G PCI Adapter> mem 0xf4100000-0xf4101fff irq 3 at device 8.0 on pci1
ndis0: NDIS API version: 5.0
ndis0: Ethernet address: 0a:b1:2c:d3:4e:f5
ndis0: 11b rates: 1Mbps 2Mbps 5.5Mbps 11Mbps
ndis0: 11g rates: 6Mbps 9Mbps 12Mbps 18Mbps 36Mbps 48Mbps 54Mbps
```

Начиная с этого момента вы можете использовать устройство `ndis0` как любое другое беспроводное устройство (например, `wi0`); в этой ситуации применима информация, приведенная в начале этой главы.

## 24.4. Bluetooth

*Текст предоставил Pav Lucistnik.*

### 24.4.1. Введение

Bluetooth является беспроводной технологией для создания персональных сетей на расстоянии не более 10 метров, работающей на частоте 2.4 ГГц, которая не подлежит лицензированию. Обычно такие сети формируются из портативных устройств, таких, как сотовые телефоны, КПК и ноутбуки. В отличие от Wi-Fi, другой популярной беспроводной технологии, Bluetooth предоставляет более высокий уровень сервиса, например, файловые серверы типа FTP, передачу файлов, голоса, эмуляцию последовательного порта и другие.

Стек протоколов Bluetooth во FreeBSD реализован на основе технологии Netgraph (обратитесь к `netgraph(4)`). Широкий спектр USB-устройств Bluetooth поддерживается драйвером `ng_ubt(4)`. Устройства Bluetooth на основе набора микросхем Broadcom BCM2033 поддерживаются драйвером `ng_bt3c(4)`. Устройства Bluetooth, работающие через последовательные и UART-порты, поддерживаются драйверами `sio(4)`, `ng_h4(4)` и `hseriald(8)`. В этом разделе описывается использование Bluetooth-устройств, подключаемых через USB. Поддержка Bluetooth имеется во FreeBSD 5.0 и более новых версиях системы.

### 24.4.2. Подключение устройства

По умолчанию драйверы устройств Bluetooth поставляются в виде модулей ядра. Перед подключением устройства вам необходимо подгрузить драйвер в ядро:

```
# kldload ng_ubt
```

Если Bluetooth-устройство в момент запуска системы подключено, то загружайте модуль из файла `/boot/loader.conf`:

```
ng_ubt_load="YES"
```

Подключите ваше USB-устройство. На консоли (или в журнале syslog) появится примерно такое сообщение:

```
ubt0: vendor 0x0a12 product 0x0001, rev 1.10/5.25, addr 2
ubt0: Interface 0 endpoints: interrupt=0x81, bulk-in=0x82, bulk-out=0x2
ubt0: Interface 1 (alt.config 5) endpoints: isoc-in=0x83, isoc-out=0x3,
      wMaxPacketSize=49, nframes=6, buffer size=294
```

Скопируйте файл `/usr/share/examples/netgraph/bluetooth/rc.bluetooth` в какое-нибудь подходящее место, например, в файл `/etc/rc.bluetooth`. Этот скрипт используется для запуска и остановки работы Bluetooth-стека. Перед отключением устройства рекомендуется остановить его работы, хотя (обычно) это не фатально. При запуске стека вы получите сообщения, подобные следующим:

```
# /etc/rc.bluetooth start ubt0
BD_ADDR: 00:02:72:00:d4:1a
Features: 0xff 0xff 0xf 00 00 00 00 00
<3-Slot> <5-Slot> <Encryption> <Slot offset>
<Timing accuracy> <Switch> <Hold mode> <Sniff mode>
<Park mode> <RSSI> <Channel quality> <SCO link>
<HV2 packets> <HV3 packets> <u-law log> <A-law log> <CVSD>
<Paging scheme> <Power control> <Transparent SCO data>
Max. ACL packet size: 192 bytes
Number of ACL packets: 8
Max. SCO packet size: 64 bytes
Number of SCO packets: 8
```

### 24.4.3. Host Controller Interface (HCI)

Host Controller Interface (HCI) предоставляет интерфейс для управления контроллером передатчика и менеджером соединений, а также доступ к данным о состоянии оборудования и его управляющим регистрам. Этот интерфейс предоставляет унифицированный метод доступа к передающим возможностям Bluetooth. Уровень HCI на управляющей машине обменивается данными и командами с микрокодом HCI в оборудовании Bluetooth. Драйвер для Host Controller Transport Layer (то есть физической шины) предоставляет обоим слоям HCI возможность обмениваться данными друг с другом.

Для одного Bluetooth-устройства создаётся один узел Netgraph типа *hci*. HCI-узел обычно подключается к узлу драйвера устройства Bluetooth (входящий поток) и к узлу L2CAP (исходящий поток). Все операции с HCI должны выполняться на узле HCI, но не на узле драйвера устройства. В качестве имени по умолчанию для узла HCI используется `"devicehci"`. Дополнительные подробности можно найти на справочной странице `ng_hci(4)`.

Одной из самой часто выполняемой задач является обнаружение Bluetooth-устройств в радиусе RF-доступности. Эта операция называется *опросом* (*inquiry*). Опрос и другие операции, связанные с HCI, выполняются при помощи утилиты `hccontrol(8)`. Пример ниже показывает, как найти доступные устройства Bluetooth. Список таких устройств должен быть получен в течение нескольких секунд. Заметьте, что удалённые устройства будут отвечать на опрос, если только они находятся в режиме *обнаруживаемости* (*discoverable*).

```
% hccontrol -n ubt0hci inquiry
```

```

Inquiry result, num_responses=1
Inquiry result #0
    BD_ADDR: 00:80:37:29:19:a4
    Page Scan Rep. Mode: 0x1
    Page Scan Period Mode: 00
    Page Scan Mode: 00
    Class: 52:02:04
    Clock offset: 0x78ef
Inquiry complete. Status: No error [00]

```

BD\_ADDR является уникальным адресом устройства Bluetooth, вроде MAC-адресов сетевых адаптеров. Этот адрес необходим для дальнейшей работы с устройством. Адресу BD\_ADDR можно присвоить удобное для чтения имя. Файл `/etc/bluetooth/hosts` содержит информацию об известных хостах Bluetooth. В следующем примере показано, как получить имя, назначенное удалённому устройству:

```

% hccontrol -n ubt0hci remote_name_request 00:80:37:29:19:a4
BD_ADDR: 00:80:37:29:19:a4
Name: Pav's T39

```

Если вы выполните опрос на другом Bluetooth-устройстве, но ваш компьютер будет опознан как `'your.host.name (ubt0)'`. Имя, назначаемое локальному устройству, может быть в любой момент изменено.

Система Bluetooth предоставляет услуги по соединениям типа точка-точка (при этом задействованы только два устройства Bluetooth) или точка-ко-многим-точкам. В последнем случае соединение используется совместно несколькими устройствам Bluetooth. В следующем примере показывается, как получить список активных для локального устройства соединений:

```

% hccontrol -n ubt0hci read_connection_list
Remote BD_ADDR    Handle Type Mode Role Encrypt Pending Queue State
00:80:37:29:19:a4    41  ACL    0 MAST    NONE      0      0 OPEN

```

Идентификатор соединения (*connection handle*) полезен, когда необходимо прекратить соединение. Заметьте, что обычно нет нужды делать это вручную. Стек будет автоматически разрывать неактивные соединения.

```

# hccontrol -n ubt0hci disconnect 41
Connection handle: 41
Reason: Connection terminated by local host [0x16]

```

Обратитесь к помощи посредством `hccontrol help` для получения полного списка доступных HSI-команд. Большинство команд HSI для выполнения не требуют прав администратора системы.

#### 24.4.4. Logical Link Control and Adaptation Protocol (L2CAP)

Протокол L2CAP (Logical Link Control and Adaptation Protocol) предоставляет услуги по работе с данными, как ориентированные на соединения, так и без ориентации на них, протоколам более высокого уровня с возможностями мультиплексирования и обеспечением операций по сегментации и обратной сборке. L2CAP позволяет протоколам более высокого уровня и приложениям передавать и получать пакеты данных L2CAP длиной до 64 Кбайт.

L2CAP основан на концепции *каналов*. Каналом является логическое соединение поверх соединения по радиоканалу. Каждый канал привязан к некоторому протоколу по принципу многие-к-одному. Несколько каналов могут быть привязаны к одному и тому же протоколу, но канал не может быть привязан к нескольким протоколам. Каждый пакет L2CAP, получаемый каналом, перенаправляется к соответствующему протоколу более высокого уровня. Несколько каналов могут совместно использовать одно и то же радиосоединение.

Для одного Bluetooth-устройства создается один узел Netgraph типа *l2cap*. Узел L2CAP обычно подключается к узлу Bluetooth HCI (нижестоящий) и узлам Bluetooth-сокетов (вышестоящие). По умолчанию для узла L2CAP используется имя "device12cap". Для получения дополнительной информации обратитесь к справочной странице по `ng_l2cap(4)`.

Полезной является программа `l2ping(8)`, которая может использоваться для проверки связи с другими устройствами. Некоторые реализации Bluetooth могут не возвращать все данные, посылаемые им, так что 0 bytes в следующем примере - это нормально.

```
# l2ping -a 00:80:37:29:19:a4
0 bytes from 0:80:37:29:19:a4 seq_no=0 time=48.633 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=1 time=37.551 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=2 time=28.324 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=3 time=46.150 ms result=0
```

Утилита `l2control(8)` используется для выполнения различных операций с узлами L2CAP. В этом примере показано, как получить список логических соединений (каналов) и перечень радиосоединений локального устройства:

```
% l2control -a 00:02:72:00:d4:1a read_channel_list
L2CAP channels:
Remote BD_ADDR      SCID/ DCID   PSM  IMTU/ OMTU State
00:07:e0:00:0b:ca   66/   64     3   132/  672 OPEN
% l2control -a 00:02:72:00:d4:1a read_connection_list
L2CAP connections:
Remote BD_ADDR      Handle Flags Pending State
00:07:e0:00:0b:ca   41  0           0 OPEN
```

Ещё одним диагностическим инструментом является `btsockstat(1)`. Она выполняет действия, подобные тем, что обычно выполняет `netstat(1)`, но со структурами данных, связанных с работой в сети Bluetooth. В примере ниже описывается то же самое логическое соединение, что и с `l2control(8)` выше.

```
% btsockstat
Active L2CAP sockets
PCB      Recv-Q Send-Q Local address/PSM      Foreign address  CID   State
c2afe900    0      0 00:02:72:00:d4:1a/3    00:07:e0:00:0b:ca 66    OPEN
Active RFCOMM sessions
L2PCB    PCB      Flag MTU   Out-Q DLCs State
c2afe900 c2b53380 1    127    0     Yes  OPEN
Active RFCOMM sockets
PCB      Recv-Q Send-Q Local address      Foreign address  Chan DLCI State
c2e8bc80    0    250 00:02:72:00:d4:1a  00:07:e0:00:0b:ca 3     6    OPEN
```

### 24.4.5. Протокол RFCOMM

Протокол RFCOMM эмулирует последовательные порты поверх протокола L2CAP. Он основан на ETSI-стандарте TS 07.10. RFCOMM представляет собой простой транспортный протокол, с дополнительными возможностями по эмуляции 9 цепей последовательных портов RS-232 (EIA/TIA-232-E). Протокол RFCOMM поддерживает одновременно до 60 соединений (каналов RFCOMM) между двумя устройствами Bluetooth.

В рамках RFCOMM полный коммуникационный маршрут включает два приложения, работающие на разных устройствах (конечные коммуникационные точки) с коммуникационным сегментом между ними. RFCOMM предназначен для сокрытия приложений, использующих последовательные порты устройств, в которых они расположены. Коммуникационный сегмент по сути является Bluetooth-связью от одного устройства к другому (прямое соединение).

RFCOMM имеет дело с соединением между устройствами в случае прямого соединения, или между устройством и модемом в сетевом случае. RFCOMM может поддерживать и другие конфигурации, такие, как модули, работающие через беспроводную технологию Bluetooth с одной стороны и предоставляющие проводное соединение с другой стороны.

Во FreeBSD протокол RFCOMM реализован на уровне сокетов Bluetooth.

### 24.4.6. Pairing of Devices

По умолчанию связь Bluetooth не аутентифицируется, поэтому любое устройство может общаться с любым другим. Устройство Bluetooth (например, сотовый телефон) может задать обязательность аутентификации для предоставления определённого сервиса (в частности, услугу доступа по коммутируемой линии). Bluetooth-аутентификация обычно выполняется через *PIN-коды*. PIN-код представляет из себя ASCII-строку длиной до 16 символов. Пользователь обязан ввести один и тот же PIN-код на обоих устройствах. Как только он введёт PIN-код, оба устройства сгенерируют *ключ связи*. После этого ключ может быть сохранён либо в самом устройстве, либо на постоянном носителе. В следующий раз оба устройства будут использовать ранее сгенерированный ключ соединения. Процедура, описанная выше, носит название *подгонки пары* (pairing). Заметьте, что если ключ связи потерян любой из сторон, то подбор пары должен быть повторен.

За обработку всех запросов на Bluetooth-аутентификацию отвечает демон hcsecd(8). По умолчанию файл конфигурации называется /etc/bluetooth/hcsecd.conf. Пример раздела, содержащего информацию о сотовом телефоне с явно заданным PIN-кодом "1234" приведен ниже:

```
device {
    bdaddr    00:80:37:29:19:a4;
    name      "Pav's T39";
    key       nokey;
    pin       "1234";
}
```

Кроме длины, на PIN-коды не накладывается никаких ограничений. Некоторые устройства (например, Bluetooth-гарнитуры) могут иметь фиксированный встроенный PIN-код. Параметр -d позволяет запустить hcsecd(8) как фоновый процесс, что облегчает просмотр происходящих событий. Задайте получение парного ключа на удалённом устройстве и иницируйте Bluetooth-соединение с этим устройством. Удалённое устройство должно подтвердить получение пары и запросить PIN-код. Введите тот же самый код, что находится в hcsecd.conf. Теперь ваш ПК и удалённое устройство

спарены. Альтернативным способом является инициация процесса создания пары на удалённом устройстве. Ниже даётся пример выдачи протокола команды `hcsecd`:

```
hcsecd[16484]: Got Link_Key_Request event from 'ubt0hci', remote bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39', link key does
hcsecd[16484]: Sending Link_Key_Negative_Reply to 'ubt0hci' for remote bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Got PIN_Code_Request event from 'ubt0hci', remote bdaddr 0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39', PIN code exist
hcsecd[16484]: Sending PIN_Code_Reply to 'ubt0hci' for remote bdaddr 0:80:37:29:19:a4
```

### 24.4.7. Service Discovery Protocol (SDP)

Протокол обнаружения сервисов SDP даёт возможность клиентским приложениям осуществлять поиск услуг, предоставляемых серверными приложениями, а также характеристик этих услуг. В перечень атрибутов сервиса включается тип класса предлагаемого сервиса и информация о механизме или протоколе, требуемом для использования сервиса.

SDP подразумевает коммуникации между SDP-сервером и SDP-клиентом. Сервер поддерживает список сервисов, в котором описываются параметры сервисов, связанных с сервером. Каждая запись об услуге содержит информацию об одном сервисе. Клиент может запросить информацию об определённом сервисе, обслуживаемом SDP-сервером, выдавая SDP-запрос. Если клиент или приложение, связанное с клиентом, решат воспользоваться сервисом, то для его использования необходимо открыть отдельное соединение к устройству, предоставляющему сервис. SDP предоставляет механизм обнаружения услуг и их параметров, но не даёт механизма использования этих сервисов.

Обычно SDP-клиент выполняет поиск услуг на основе некоторых желаемых характеристик услуг. Однако иногда возникает необходимость выяснить полный перечень типов услуг, предоставляемых SDP-сервером, не имея никакой информации об имеющихся сервисах. Такой процесс всех предлагаемых сервисов называется *обзором* (browsing).

Bluetooth SDP сервер `sdpd(8)` и клиент с интерфейсом командной строки `sdpcontrol(8)` включены в стандартную поставку FreeBSD. В следующем примере показано, как выполнять запрос на SDP-обзор.

```
% sdpcontrol -a 00:01:03:fc:6e:ec browse
Record Handle: 00000000
Service Class ID List:
    Service Discovery Server (0x1000)
Protocol Descriptor List:
    L2CAP (0x0100)
        Protocol specific parameter #1: u/int/uuid16 1
        Protocol specific parameter #2: u/int/uuid16 1

Record Handle: 0x00000001
Service Class ID List:
    Browse Group Descriptor (0x1001)

Record Handle: 0x00000002
Service Class ID List:
    LAN Access Using PPP (0x1102)
Protocol Descriptor List:
    L2CAP (0x0100)
    RFCOMM (0x0003)
```

```

Protocol specific parameter #1: u/int8/bool 1
Bluetooth Profile Descriptor List:
  LAN Access Using PPP (0x1102) ver. 1.0

```

... и так далее. Заметьте, что каждый сервис имеет перечень атрибутов (например, канал RFCOMM). В зависимости от сервиса вам может потребоваться где-то сохранить эти атрибуты. Некоторые реализации Bluetooth не поддерживают просмотр сервисов и могут возвращать пустой список. В этом случае возможен поиск конкретной услуги. В примере ниже показано, как выполнить поиск службы OBEX Object Push (OPUSH):

```
% sdpcontrol -a 00:01:03:fc:6e:ec search OPUSH
```

Во FreeBSD предоставление сервисов клиентам Bluetooth осуществляется сервером sdpd(8):

```
# sdpd
```

Приложение на локальном сервере, желающее предоставить сервис Bluetooth удаленным клиентам, регистрирует сервис через локального демона SDP. Пример такого приложения — rfcomm\_pppd(8). После запуска оно регистрирует Bluetooth LAN сервис через локального демона SDP.

Список сервисов, зарегистрированных через локальный SDP сервер, может быть получен путем выдачи запроса на просмотр SDP через локальный контрольный канал:

```
# sdpcontrol -l browse
```

#### 24.4.8. Доступ к сети по коммутируемой линии связи (DUN) и по протоколу PPP (LAN)

Модуль работы с коммутируемым доступом к сети (DUN - Dial-Up Networking) в большинстве случаев используется с модемами и сотовыми телефонами. Этот модуль покрывает следующие случаи:

- сотовый телефон или модем используется вместе с компьютером в качестве беспроводного модема для подключения к серверу коммутируемого доступа в Интернет, или другой коммутируемой услуге;
- сотовый телефон или модем используется компьютером для приёма входящих соединений.

Модуль доступа к сети по протоколу PPP (Network Access with PPP - LAN) может использоваться в следующих ситуациях:

- доступ к ЛВС для одного Bluetooth-устройства;
- доступ к ЛВС для нескольких Bluetooth-устройств;
- связь между двумя ПК (при помощи протокола PPP поверх эмулируемого последовательного канала связи).

Во FreeBSD оба случая реализуются при помощи сервисных программ ppp(8) и rfcomm\_pppd(8) - это обработчик, преобразующий RFCOMM-соединения Bluetooth в нечто, с чем может работать PPP. Перед тем, как использовать любой модуль, в файле /etc/ppp/ppp.conf должна быть создана новая PPP-метка. Примеры использования можно найти в справочной странице к rfcomm\_pppd(8).



В следующем примере `rfcomm_pppd(8)` будет использоваться для открытия RFCOMM-соединения к удалённому устройству с `BD_ADDR 00:80:37:29:19:a4` на DUN RFCOMM-канале. Реальный номер RFCOMM-канала будет получаться с удалённого устройства через SDP. Возможно указать RFCOMM-канал вручную, и в этом случае `rfcomm_pppd(8)` не будет выполнять SDP-запрос. Для нахождения RFCOMM-канала на удалённом устройстве используйте утилиту `sdpcontrol(8)`.

```
# rfcomm_pppd -a 00:80:37:29:19:a4 -c -C dun -l rfcomm-dialup
```

Для того, чтобы организовать сервис Network Access with PPP (LAN), необходимо запустить сервер `sdpd(8)`. В файле `/etc/ppp/ppp.conf` должна быть создана новая запись для клиентов LAN. Примеры можно найти в справке по `rfcomm_pppd(8)`. Наконец, запустите RFCOMM PPP сервер на существующем номере канала RFCOMM. Сервер RFCOMM PPP автоматически регистрирует Bluetooth LAN сервис через локальный SDP даемон. В примере ниже показано, как запустить сервер RFCOMM PPP.

```
# rfcomm_pppd -s -C 7 -l rfcomm-server
```

### 24.4.9. OBEX Object Push (OPUSH) Profile

OBEX является широко используемым протоколом для простой передачи файлов между мобильными устройствами. В основном он используется в коммуникациях через инфракрасный порт для передачи файлов между ноутбуками или КПК, а также для пересылки визитных карточек или календарных планов между сотовыми телефонами и другими устройствами с персональными информационными менеджерами.

Сервер и клиент OBEX реализованы в виде пакета стороннего разработчика **obexapp**, который доступен в виде порта `comms/obexapp`.

Клиент OBEX используется для отправки или приёма объектов с сервера OBEX. Объектом, к примеру, может быть визитная карточка или указание. Клиент OBEX может получить номер RFCOMM-канала, указав вместо него имя сервиса. Поддерживаются следующие имена сервиса: `IrMC`, `FTRN` и `OPUSH`. Канал RFCOMM можно задать его номером. Ниже даётся пример сеанса OBEX, где с сотового телефона забирается объект с информацией об устройстве, а новый объект (визитная карточка) передаётся в каталог сотового телефона.

```
% obexapp -a 00:80:37:29:19:a4 -C IrMC
obex> get telecom/devinfo.txt devinfo-t39.txt
Success, response: OK, Success (0x20)
obex> put new.vcf
Success, response: OK, Success (0x20)
obex> di
Success, response: OK, Success (0x20)
```

Для того, чтобы предоставить сервис OBEX Push, должен быть запущен сервер `sdpd(8)`. Должен быть создан корневой каталог, в котором будут сохраняться все поступающие объекты. По умолчанию корневым каталогом является `/var/spool/obex`. Наконец, запустите OBEX сервер на существующем номере канала RFCOMM. OBEX сервер автоматически регистрирует сервис OBEX Object Push через локального даемона SDP. В примере ниже показано, как запустить OBEX-сервер.

```
# obexapp -s -C 10
```

### 24.4.10. Профиль последовательного порта (SPP)

Профиль последовательного порта (SPP - Serial Port Profile) позволяет Bluetooth-устройствам осуществлять эмуляцию последовательного порта RS232 (или подобного). Этот профиль покрывает случаи, касающиеся работы унаследованных приложений с Bluetooth в качестве замены кабельному соединению, при это используется абстракция виртуального последовательного порта.

Утилита `rfcomm_sppd(1)` реализует профиль последовательного порта. В качестве виртуального последовательного порта используется псевдотерминал. В примере ниже показано, как подключиться к сервису Serial Port удалённого устройства. Заметьте, что вы не указываете RFCOMM-канал - `rfcomm_sppd(1)` может получить его с удалённого устройства через SDP. Если вы хотите переопределить это, укажите RFCOMM-канал явно в командной строке.

```
# rfcomm_sppd -a 00:07:E0:00:0B:CA -t /dev/tty6
rfcomm_sppd[94692]: Starting on /dev/tty6...
```

После подключения псевдотерминал можно использовать как последовательный порт:

```
# cu -l tty6
```

### 24.4.11. Решение проблем

#### 24.4.11.1. Удалённое устройство не подключается

Некоторые старые Bluetooth-устройства не поддерживают переключение ролей. По умолчанию, когда FreeBSD подтверждает новое соединение, она пытается выполнить переключение роли и стать ведущим устройством. Устройства, которые это не поддерживают, не смогут подключиться. Заметьте, что переключение ролей выполняется при установлении нового соединения, поэтому невозможно выяснить, поддерживает ли удалённое устройство переключение ролей. На локальной машине имеется возможность отключить переключение ролей при помощи HCI-параметра:

```
# hccontrol -n ubt0hci write_node_role_switch 0
```

#### 24.4.11.2. Что-то идёт не так, можно ли посмотреть, что в точности происходит?

Да, можно. Воспользуйтесь пакетом стороннего разработчика, **hcidump** который доступен в виде порта `comms/hcidump`. Утилита **hcidump** похожа на `tcpdump(1)`. Она может быть использована для вывода на терминал содержимого Bluetooth-пакетов и сбрасывать пакеты Bluetooth в файл.

## 24.5. Мосты

*Текст создал Steve Peterson.*

### 24.5.1. Введение

Иногда полезно разделить одну физическую сеть (такую, как сегмент Ethernet) на два отдельных

сегмента сети без необходимости создания подсетей IP и использования маршрутизатора для соединения сегментов. Устройство, которое соединяет две сети на такой манер, называется "сетевым мостом" ("bridge"). Система FreeBSD с двумя сетевыми адаптерами может выступать в роли моста.

Мост работает на основе изучения адресов уровня MAC (адресов Ethernet) устройств на каждом из своих сетевых интерфейсах. Он перенаправляет трафик между двумя сетями, только когда адреса отправителя и получателя находятся в разных сетях.

По многим параметрам мост работает также, как коммутатор Ethernet с малым количеством портов.

## 24.5.2. Ситуации, когда можно использовать мосты

На сегодняшний день есть две ситуации, когда можно использовать мост.

### 24.5.2.1. Большой трафик в сегменте

Первая ситуация возникает, когда ваша физическая сеть перегружена трафиком, но по каким-то соображениям вы не хотите разделять сеть на подсети и соединять их с помощью маршрутизатора.

Давайте рассмотрим в качестве примера газету, в которой редакторский и производственный отделы находятся в одной и той же подсети. Пользователи в редакторском отделе все используют сервер А для служб доступа к файлам, а пользователи производственного отдела используют сервер В. Для объединения всех пользователей используется сеть Ethernet, а высокая нагрузка на сеть замедляет работу.

Если пользователи редакторского отдела могут быть собраны в одном сегменте сети, а пользователи производственного отдела в другом, то два сетевых сегмента можно объединить мостом. Только сетевой трафик, предназначенный для интерфейсов с "другой" стороны моста, будет посылаться в другую сеть, тем самым снижая уровень нагрузки на каждый сегмент сети.

### 24.5.2.2. Брандмауэр с возможностями фильтрации/ограничения пропускной способности трафика

Второй распространенной ситуацией является необходимость в обеспечении функций брандмауэра без трансляции сетевых адресов (NAT).

Для примера можно взять маленькую компанию, которая подключена к своему провайдеру по каналу DSL или ISDN. Для неё провайдер выделил 13 глобально доступных IP-адресов для имеющихся в сети 10 персональных компьютеров. В такой ситуации использование брандмауэра на основе маршрутизатора затруднено из-за проблем с разделением на подсети.

Брандмауэр на основе моста может быть настроен и включен между маршрутизаторами DSL/ISDN без каких-либо проблем с IP-адресацией.

## 24.5.3. Настройка моста

### 24.5.3.1. Выбор сетевого адаптера

Для работы моста требуются по крайней мере два сетевых адаптера. К сожалению, не все сетевые

адаптеры во FreeBSD 4.0 поддерживают функции моста. Прочтите страницу Справочника по `bridge(4)` для выяснения подробностей о поддерживаемых адаптерах.

Перед тем, как продолжить, сначала установите и протестируйте два сетевых адаптера.

### 24.5.3.2. Изменения в конфигурации ядра

Для включения поддержки функций моста в ядре, добавьте строчку

```
options BRIDGE
```

в файл конфигурации вашего ядра, и перестройте ядро.

### 24.5.3.3. Поддержка функций брандмауэра

Если вы планируете использовать мост в качестве брандмауэра, вам нужно также добавить опцию `IPFIREWALL`. Прочтите Разд. 14.9, содержащий общую информацию о настройке моста в качестве брандмауэра.

Если вам необходимо обеспечить прохождение не-IP пакетов (таких, как ARP) через мост, то имеется опция брандмауэра, которую можно задать. Это опция `IPFIREWALL_DEFAULT_TO_ACCEPT`. Заметьте, что при этом правило, используемое брандмауэром по умолчанию, меняется на разрешительное для всех пакетов. Перед тем, как задавать эту опцию, убедитесь, что вы понимаете работу вашего набора правил.

### 24.5.3.4. Поддержка функций ограничения пропускной способности

Если вы хотите использовать мост в качестве машины, ограничивающей пропускную способность, то добавьте в файл конфигурации ядра опцию `DUMMYNET`. Дополнительную информацию можно почерпнуть из страницы Справочника по `dummynet(4)`.

## 24.5.4. Включение функций моста

Добавьте строку

```
net.link.ether.bridge=1
```

в файл `/etc/sysctl.conf` для включения функций моста во время работы системы, и строку:

```
net.link.ether.bridge_cfg=if1,if2
```

для включения функций моста для указанных интерфейсов (замените `if1` и `if2` на имена двух ваших сетевых интерфейсов). Если вы хотите, чтобы проходящие через мост пакеты фильтровались посредством `ipfw(8)`, вы должны добавить строчку:

```
net.link.ether.bridge_ipfw=1
```

Во FreeBSD 5.2-RELEASE и последующих версиях нужно использовать вместо указанных следующие строки:

```
net.link.ether.bridge.enable=1
net.link.ether.bridge.config=if1,if2
net.link.ether.bridge.ipfw=1
```

### 24.5.5. Дополнительные замечания

Если вы хотите осуществлять удалённый доступ на мост через ssh(1) из сети, то корректно назначить одному из сетевых адаптеров IP-адрес. Общепринято, что назначение адреса обоим сетевым адаптерам является не самой хорошей идеей.

Если в вашей сети присутствует несколько мостов, не должно быть более одного маршрута между любыми двумя рабочими станциями. С технической точки зрения это означает отсутствие поддержки протокола spanning tree.

Сетевой мост может увеличить задержки в замерах командой ping(8), особенно для трафика между двумя разными сегментами.

## 24.6. Работа с бездисковыми станциями

*Текст обновил Jean-François Dockès. Реорганизовал и улучшил Alex Dupre.*

Машина с FreeBSD может загружаться по сети и работать без наличия локального диска, используя файловые системы, монтируемые с сервера NFS. Кроме стандартных конфигурационных файлов, не нужны никакие модификации в системе. Такую систему легко настроить, потому что все необходимые элементы уже готовы:

- Имеется по крайней мере два возможных способа загрузки ядра по сети:
  - PXE: Система Intel Preboot eXecution Environment является формой загрузочного ПЗУ, встроенного в некоторые сетевые адаптеры или материнские платы. Обратитесь к справочной странице по pxeboot(8) для получения более полной информации.
  - Порт **Etherboot** (net/etherboot) генерирует код, который может применяться в ПЗУ для загрузки ядра по сети. Код может быть либо прошит в загрузочный PROM на сетевом адаптере, либо загружен с локальной дискеты (или винчестера), или с работающей системы MS-DOS. Поддерживаются многие сетевые адаптеры.
- Примерный скрипт (/usr/share/examples/diskless/clone\_root) облегчает создание и поддержку корневой файловой системы рабочей станции на сервере. Скрипт, скорее всего, потребует некоторых настроек, но он позволит вам быстро начать работу.
- Стандартные файлы начального запуска системы, располагающиеся в /etc, распознают и поддерживают загрузку системы в бездисковом варианте.
- Подкачка, если она нужна, может выполняться через файл NFS либо на локальный диск.

Существует много способов настройки бездисковой рабочей станции. При этом задействованы многие компоненты, и большинство из них могут быть настроены для удовлетворения ваших вкусов. Далее будет описаны варианты полной настройки системы, при этом упор будет делаться на простоту и

совместимость с стандартной системой скриптов начальной загрузки FreeBSD. Описываемая система имеет такие характеристики:

- Бездисковые рабочие станции совместно используют файловую систему / в режиме только чтения, а также используют /usr совместно тоже в режиме только чтения.

Корневая файловая система является копией стандартной корневой системы FreeBSD (обычно сервера), с некоторыми настроечными файлами, измененными кем-то специально для бездисковых операций или, возможно, для рабочей станции, которой она предназначена.

Части корневой файловой системы, которые должны быть доступны для записи, перекрываются файловыми системами mfs(8) (FreeBSD 4.X) или md(4) (FreeBSD 5.X). Любые изменения будут потеряны при перезагрузках системы.

- Ядро передается и загружается посредством **Etherboot** или PXE, и в некоторых ситуациях может быть использован любой из этих методов.

**Предостережение:** Как описано, эта система не защищена. Она должна располагаться в защищенной части сети, а другие хосты не должны на нее полагаться.

Вся информация этого раздела была протестирована с релизами FreeBSD 4.9-RELEASE и 5.2.1-RELEASE. Текст структурирован преимущественно для использования с 4.X. Отличия для 5.X упоминаются особо.

### 24.6.1. Общая информация

Настройка бездисковых рабочих станций относительно проста, но в то же время легко сделать ошибку. Иногда сложно диагностировать эти ошибки по нескольким причинам. Например:

- Параметры компиляции могут по-разному проявлять себя во время работы.
- Сообщения об ошибках бывают загадочны или вовсе отсутствуют.

В данной ситуации некоторые знания, касающиеся используемых внутренних механизмов, очень полезны при разрешении проблем, которые могут возникнуть.

Для выполнения успешной загрузки необходимо произвести несколько операций:

- Компьютеру необходимо получить начальные параметры, такие как собственный IP адрес, имя исполняемого файла, корневой каталог. Для этого используются протоколы DHCP или BOOTP. DHCP это совместимое расширение BOOTP, используются те же номера портов и основной формат пакетов.

Возможна настройка системы для использования только BOOTP. Серверная программа bootpd(8) включена в основную систему FreeBSD.

Тем не менее, у DHCP есть множество преимуществ над BOOTP (лучше файлы настройки, возможность использования PXE, плюс многие другие преимущества, не относящиеся непосредственно к бездисковым операциям), и мы в основном будем описывать настройку DHCP, с эквивалентными примерами для bootpd(8), когда это возможно. Пример конфигурации будет использовать пакет **ISC DHCP** (релиз 3.0.1.r12 был установлен на тестовом сервере).

- Компьютеру требуется загрузить в локальную память одну или несколько программ. Используются TFTP или NFS. Выбор между TFTP или NFS производится во время компилирования в нескольких местах. Часто встречающаяся ошибка это указание имен файлов для другого протокола: TFTP обычно загружает все файлы с одного каталога сервера, и принимает имена файлов относительно этого каталога. NFS нужны абсолютные пути к файлам.
- Необходимо инициализировать и выполнить возможные промежуточные программы загрузки и ядро. В этой области существует несколько важных вариаций:
  - PXE загрузит pxeboot(8), являющийся модифицированной версией загрузчика третьей стадии FreeBSD. loader(8) получит большинство параметров, необходимых для старта системы, и оставит их в окружении ядра до контроля передачи. В этом случае возможно использование ядра GENERIC.
  - **Etherboot**, непосредственно загрузит ядро, с меньшей подготовкой. Вам потребуется собрать ядро со специальными параметрами.

PXE и **Etherboot** работают одинаково хорошо с системами 4.X. Поскольку ядро 5.X обычно позволяет loader(8) выполнить больше предварительной работы, метод PXE на системах 5.X предпочтителен.

Если ваш BIOS и сетевые карты поддерживают PXE, используйте его. Однако, все же возможен запуск системы 5.X с **Etherboot**.

- Наконец, компьютеру требуется доступ к файловым системам. NFS используется во всех случаях. Обратитесь также к странице справочника diskless(8).

## 24.6.2. Инструкции по настройке

### 24.6.2.1. Конфигурация с использованием ISC DHCP

Сервер **ISC DHCP** может обрабатывать как запросы BOOTP, так и запросы DHCP.

Начиная с релиза 4.9, **ISC DHCP 3.0** не включается в поставку системы. Сначала вам нужно будет установить порт `net/isc-dhcp3-server` или соответствующий пакет.

После установки **ISC DHCP** ему для работы требуется конфигурационный файл (обычно называемый `/usr/local/etc/dhcpd.conf`). Вот прокомментированный пример, где хост `margaux` использует **Etherboot**, а хост `corbieres` использует PXE:

```
default-lease-time 600;
max-lease-time 7200;
authoritative;

option domain-name "example.com";
option domain-name-servers 192.168.4.1;
option routers 192.168.4.1;

subnet 192.168.4.0 netmask 255.255.255.0 {
    use-host-decl-names on; ❶
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.4.255;
```

```

host margaux {
    hardware ethernet 01:23:45:67:89:ab;
    fixed-address margaux.example.com;
    next-server 192.168.4.4; ❶
    filename "/data/misc/kernel.diskless"; ❷
    option root-path "192.168.4.4:/data/misc/diskless"; ❸
}
host corbieres {
    hardware ethernet 00:02:b3:27:62:df;
    fixed-address corbieres.example.com;
    next-server 192.168.4.4;
    filename "pxeboot";
    option root-path "192.168.4.4:/data/misc/diskless";
}
}

```

- ❶ Этот параметр указывает **dhcpd** посылать значения деклараций `host` как имя хоста для бездисковой машины. Альтернативным способом было бы добавление `option host-name margaux` внутри объявлений `host`.
- ❷ Директива `next-server` определяет сервер TFTP или NFS, используемый для получения загрузчика или файла ядра (по умолчанию используется тот же самый хост, на котором расположен сервер DHCP).
- ❸ Директива `filename` определяет файл, который **Etherboot** или PXE будут загружать для следующего шага выполнения. Он должен быть указан в соответствии с используемым методом передачи. **Etherboot** может быть скомпилирован для использования NFS или TFTP. FreeBSD порт по умолчанию использует NFS. PXE использует TFTP, поэтому здесь применяются относительные пути файлов (это может зависеть от настроек TFTP сервера, но обычно довольно типично). Кроме того, PXE загружает `pxeboot`, а не ядро. Существуют другие интересные возможности, такие как загрузка `pxeboot` из каталога `/boot` FreeBSD CD-ROM (поскольку `pxeboot(8)` может загружать GENERIC ядро, это делает возможной загрузку с удаленного CD-ROM).
- ❹ Параметр `root-path` определяет путь к корневой файловой системе, в обычной нотации NFS. При использовании PXE, можно оставить IP хоста отключенным, если параметр ядра `BOOTP` не используется. Затем NFS сервер может использоваться так же, как и TFTP.

#### 24.6.2.2. Настройка с использованием BOOTP

Далее описана эквивалентная конфигурация с использованием **bootpd** (для одного клиента). Она будет располагаться в `/etc/bootptab`.

Пожалуйста, отметьте, что **Etherboot** должен быть откомпилирован с нестандартной опцией `NO_DHCP_SUPPORT` для того, чтобы можно было использовать `BOOTP`, и что для работы PXE *необходим* DHCP. Единственным очевидным преимуществом **bootpd** является его наличие в поставке системы.

```

.def100:\
:hn:ht=1:sa=192.168.4.4:vm=rflc1048:\
:sm=255.255.255.0:\
:ds=192.168.4.1:\

```



```
:gw=192.168.4.1:\
:hd="/tftpboot":\
:bf="/kernel.diskless":\
:rp="192.168.4.4:/data/misc/diskless":
```

```
margaux:ha=0123456789ab:tc=.def100
```

### 24.6.2.3. Подготовка программы загрузки при помощи Etherboot

Сайт Etherboot (<http://etherboot.sourceforge.net>) содержит подробную документацию (<http://etherboot.sourceforge.net/doc/html/userman/t1.html>), в основном предназначенную для систем Linux, но несомненно, она полезна. Далее будет просто кратко описано, как вы должны использовать **Etherboot** в системе FreeBSD.

Сначала вы должны установить пакет или порт `net/etherboot`.

Вы можете изменить настройку **Etherboot** (например, для использования TFTP вместо NFS) путем редактирования файла `Config` в каталоге исходных текстов **Etherboot**.

В нашей ситуации мы будем использовать загрузочную дискету. Для других методов (PROM или программа MS-DOS) пожалуйста, обратитесь к документации по **Etherboot**.

Для создания загрузочной дискеты, вставьте дискету в дисковод на машине, где установлен **Etherboot**, затем перейдите в каталог `src` в дереве **Etherboot** и наберите:

```
# gmake bin32/devicetype.fd0
```

`devicetype` зависит от типа адаптера Ethernet на бездискковой рабочей станции. Обратитесь к файлу `NIC` в том же самом каталоге для определения правильного значения для `devicetype`.

### 24.6.2.4. Загрузка с PXE

По умолчанию, `pxeboot(8)` загружает ядро через NFS. Он может быть скомпилирован для использования вместо него TFTP путем указания параметра `LOADER_TFTP_SUPPORT` в `/etc/make.conf`. Смотрите комментарии в `/etc/defaults/make.conf` (или `/usr/share/examples/etc/make.conf` систем 5.X) с инструкциями.

Есть два не документированных параметра `make.conf`, которые могут быть полезны для настройки бездисккового компьютера с последовательной консолью: `BOOT_PXELDR_PROBE_KEYBOARD`, и `BOOT_PXELDR_ALWAYS_SERIAL` (последняя существует только в FreeBSD 5.X).

Для использования PXE при загрузке компьютера вам обычно потребуется выбрать параметр `Boot from network` (загрузка по сети) в настройках BIOS, или нажать функциональную клавишу во время загрузки PC.

### 24.6.2.5. Настройка серверов TFTP и NFS

Если вы используете PXE или **Etherboot**, настроенные для использования TFTP, вам нужно включить `tftpd` на файловом сервере:

1. Создайте каталог, файлы которого будут обслуживать **tftpd**, например, `/tftpboot`.
2. Добавьте в ваш `/etc/inetd.conf` такую строчку:

```
tftp dgram udp wait root /usr/libexec/tftpd tftpd -l -s /tftpboot
```

**Замечание:** Бывает, что некоторым версиям PXE требуется TCP-вариант TFTP. В таком случае добавьте вторую строчку, заменяющую `dgram udp` на `stream tcp`.

3. Укажите **inetd** на повторное чтение своего конфигурационного файла:

```
# kill -HUP `cat /var/run/inetd.pid`
```

Вы можете поместить каталог `tftpboot` в любом месте на сервере. Проверьте, что это местоположение указано как в `inetd.conf`, так и в `dhcpd.conf`.

Во всех случаях, вам также нужно включить NFS и экспортировать соответствующую файловую систему на сервере NFS.

1. Добавьте следующее в `/etc/rc.conf`:

```
nfs_server_enable="YES"
```

2. Экспортируйте файловую систему, в которой расположен корневой каталог для бездисковой рабочей станции, добавив следующую строку в `/etc/exports` (подправьте точку монтирования и замените *margaux corbieres* именами бездисковых рабочих станций):

```
/data/misc -alldirs -ro margaux corbieres
```

3. Укажите **mountd** на повторное чтение настроечного файла. На самом деле если вам потребовалось на первом шаге включить NFS в `/etc/rc.conf`, то вам нужно будет выполнить перезагрузку.

```
# kill -HUP `cat /var/run/mountd.pid`
```

#### 24.6.2.6. Построение ядра для бездисковой рабочей станции

При использовании **Etherboot**, вам потребуется создать конфигурационный файл ядра для бездискового клиента со следующими параметрами (вдобавок к обычным):

```
options      BOOTP          # Use BOOTP to obtain IP address/hostname
options      BOOTP_NFSROOT  # NFS mount root filesystem using BOOTP info
```

Вам может потребоваться использовать `BOOTP_NFSV3`, `BOOT_COMPAT` и `BOOTP_WIRED_TO` (посмотрите `LINT` в 4.X или `NOTES` в 5.X).

Эти имена параметров сложились исторически, и могут немного ввести в заблуждение, поскольку включают необязательное использование **DHCP** и **BOOTP** в ядре (возможно включение обязательного использования **BOOTP** или **DHCP** use).

Постройте ядро (обратитесь к Гл. 8) и скопируйте его в каталог, указанный в `dhcpd.conf`.

**Замечание:** При использовании PXE, сборка ядра с вышеприведенными параметрами не является совершенно необходимой (хотя желательна). Включение этих параметров приведет к выполнению

большинства DHCP запросов во время загрузки ядра, с небольшим риском несоответствия новых значений и значений, полученных rxeboot(8) в некоторых особых случаях. Преимущество использования в том, что в качестве побочного эффекта будет установлено имя хоста. Иначе вам потребуется установить имя хоста другим методом, например в клиент-специфичном файле `rc.conf`.

**Замечание:** Для включения возможности загрузки с **Etherboot**, в ядро 5.X необходимо включить устройство `hints`. Вам потребуется установить в файле конфигурации следующий параметр (см. файл комментариев NOTES):

```
hints "GENERIC.hints"
```

#### 24.6.2.7. Подготовка корневой файловой системы

Вам нужно создать корневую файловую систему для бездисковых рабочих станций, в местоположении, заданном как `root-path` в `dhcpcd.conf`. В следующем разделе описаны два способа, чтобы сделать это.

##### 24.6.2.7.1. Использование скрипта `clone_root`

Это самый простой способ создания корневой файловой системы, но на данный момент он не поддерживается в FreeBSD 4.X. Этот shell скрипт находится в `/usr/share/examples/diskless/clone_root`, и требует настройки, по крайней мере, задания того места, где будет создана файловая система (переменная `DEST`).

Прочтите комментарии в начале скрипта для получения указаний. Там описано, как строится основная файловая система, и как файлы могут быть выборочно заменены версиями, предназначенными для работы без диска, для подсети или для отдельной рабочей станции. Также здесь даются примеры бездисковых файлов `/etc/fstab` и `/etc/rc.conf`.

Файлы README в `/usr/share/examples/diskless` много интересной информации, но вместе с другими примерами из каталога `diskless` они на самом деле описывают метод настройки, который отличается от того, что используется в `clone_root` и стартовых скриптах системы из `/etc`, этим несколько запутывая дело. Используйте их только для справки, за исключением того случая, когда вы выберете метод, ими описываемый, и тогда вам нужны исправленные скрипты `rc`.

##### 24.6.2.7.2. Использование стандартной процедуры `make world`

Этот метод может быть применен к FreeBSD 4.X или 5.X и установит новую систему (не только корневую) в `DESTDIR`. Все, что вам потребуется сделать, это просто выполнить следующий скрипт:

```
#!/bin/sh
export DESTDIR=/data/misc/diskless
mkdir -p ${DESTDIR}
cd /usr/src; make world && make kernel
cd /usr/src/etc; make distribution
```

Как только это будет сделано, вам может потребоваться настроить `/etc/rc.conf` и `/etc/fstab`, помещенные в `DESTDIR`, в соответствии с вашими потребностями.

#### 24.6.2.8. Настройка области подкачки

Если это нужно, то файл подкачки, расположенный на сервере, можно использовать посредством NFS. Один из методов, используемых для этого, не поддерживается в релизах 5.X.

##### 24.6.2.8.1. Подкачка по NFS в FreeBSD 4.X

Местоположение и размер файла подкачки могут быть указаны FreeBSD-специфичными параметрами `BOOTP/DHCP` 128 и 129. Примеры файлов настройки для **ISC DHCP 3.0** или **bootpd** приведены ниже:

1. Добавьте следующие строки в `dhcpd.conf`:

```
# Global section
option swap-path code 128 = string;
option swap-size code 129 = integer 32;

host margaux {
    ... # Standard lines, see above
    option swap-path "192.168.4.4:/netswapvolume/netswap";
    option swap-size 64000;
}
```

`swap-path` это путь к каталогу, где находятся файлы подкачки. Название каждого файла имеет вид `swap.client-ip`.

Старые версии **dhcpd** использовали синтаксис `option option-128 "...`, который больше не поддерживается.

Во `/etc/bootptab` будет использоваться такой синтаксис:

```
T128="192.168.4.4:/netswapvolume/netswap":T129=0000fa00
```

**Замечание:** В файле `/etc/bootptab` размер файла подкачки должен быть записан в шестнадцатеричном формате.

2. На файловом сервере NFS создайте файл (или файлы) подкачки:

```
# mkdir /netswapvolume/netswap
# cd /netswapvolume/netswap
# dd if=/dev/zero bs=1024 count=64000 of=swap.192.168.4.6
# chmod 0600 swap.192.168.4.6
```

`192.168.4.6` является IP-адресом бездискового клиента.

3. На файловом сервере NFS, в `/etc/exports` добавьте такую строку:

```
/netswapvolume -maproot=0:10 -alldirs margaux corbieres
```

Затем укажите **mountd** на повторное чтение файла `exports`, как описано ранее.

#### 24.6.2.8.2. Подкачка по NFS в FreeBSD 4.X

Положение и размер файла подкачки могут быть указаны в FreeBSD-специфичных параметрах BOOTP/DHCP с номерами 128 и 129. Ниже приведены примеры файлов настройки для **ISC DHCP 3.0** или **bootpd**:

1. Добавьте следующие строки к `dhcpd.conf`:

```
# Global section
option swap-path code 128 = string;
option swap-size code 129 = integer 32;

host margaux {
    ... # Standard lines, see above
    option swap-path "192.168.4.4:/netswapvolume/netswap";
    option swap-size 64000;
}
```

`swap-path` это путь к каталогу, где расположены файлы подкачки. Файлы называются `swap.client-ip`.

Старые версии **dhcpcd** используют синтаксис `option option-128 "...`, которые более не поддерживаются.

`/etc/bootptab` вместо этого использует следующий синтаксис:

```
T128="192.168.4.4:/netswapvolume/netswap":T129=0000fa00
```

**Замечание:** В `/etc/bootptab`, размер подкачки должен вычисляться в шестнадцатеричном формате.

2. Создайте на NFS сервере с файлами подкачки файлы:

```
# mkdir /netswapvolume/netswap
# cd /netswapvolume/netswap
# dd if=/dev/zero bs=1024 count=64000 of=swap.192.168.4.6
# chmod 0600 swap.192.168.4.6
```

`192.168.4.6` это IP адрес бездискового клиента.

3. На файловом сервере NFS с файлами подкачки добавьте следующую строку к `/etc/exports`:

```
/netswapvolume -maproot=0:10 -alldirs margaux corbieres
```

Затем заставьте **mountd** перечитать конфигурационные файлы как было показано выше.

### 24.6.2.9. Различные проблемы

#### 24.6.2.9.1. Работа с `/usr`, доступной только для чтения

Если бездисковая рабочая станция настроена на запуск X, вам нужно подправить настроечный файл для **XDM**, который по умолчанию помещает протокол ошибок в `/usr`.

#### 24.6.2.9.2. Использование не-FreeBSD сервера

Если сервер с корневой файловой системой работает не под управлением FreeBSD, вам потребуется создать корневую файловую систему на машине FreeBSD, а затем скопировать ее в нужное место, при помощи `tar` или `cpio`.

В такой ситуации иногда возникают проблемы со специальными файлами в `/dev` из-за различной разрядности целых чисел для старшего/младшего чисел. Решением этой проблемы является экспортирование каталога с не-FreeBSD сервера, монтирование его на машине с FreeBSD и запуск скрипта `MAKEDEV` на машине с FreeBSD для создания правильных файлов устройств (во FreeBSD 5.0 и более поздних версиях используется `devfs(5)` для создания файлов устройств прозрачно для пользователя, запуск `MAKEDEV` в этих версиях бессмысленно).

## 24.7. ISDN

Полезным источником информации о технологии ISDN и его аппаратном обеспечении является Страница Дэна Кегела (Dan Kegel) об ISDN (<http://www.alumni.caltech.edu/~dank/isdn/>).

Быстрое введение в ISDN:

- Если вы живёте в Европе, то вам может понадобиться изучить раздел об ISDN-адаптерах.
- Если вы планируете использовать ISDN в основном для соединений с Интернет через провайдера по коммутируемому, невыделенному соединению, рекомендуется посмотреть информацию о терминальных адаптерах. Это даст вам самую большую гибкость и наименьшее количество проблем при смене провайдера.
- Если вы объединяете две локальные сети или подключаетесь к Интернет через постоянное ISDN-соединение, рекомендуем остановить свой выбор на отдельном мосте/маршрутизаторе.

Стоимость является важным фактором при выборе вашего решения. Далее перечислены все возможности от самого дешевого до самого дорогого варианта.

### 24.7.1. Адаптеры ISDN

*Текст предоставил Hellmuth Michaelis.*

Реализация ISDN во FreeBSD поддерживает только стандарт DSS1/Q.931 (или Евро-ISDN) при помощи пассивных адаптеров. Начиная с FreeBSD 4.4 поддерживаются некоторые активные адаптеры, прошивки которых поддерживают также другие сигнальные протоколы; также сюда впервые включена поддержка адаптеров ISDN Primary Rate (PRI).

Пакет программ **isdn4bsd** позволяет вам подключаться к другим маршрутизаторам ISDN при помощи IP поверх DHLC, либо при помощи синхронного PPP; либо при помощи PPP на уровне ядра с `isppp`, модифицированного драйвера `sppp(4)`, или при помощи пользовательского `ppp(8)`. При использовании пользовательского `ppp(8)` возможно использование двух и большего числа В-каналов ISDN. Также имеется приложение, работающее как автоответчик, и много утилит, таких, как программный модем на 300 Бод.

Во FreeBSD поддерживается все возрастающее число адаптеров ISDN для ПК, и сообщения показывают, что они успешно используются по всей Европе и других частях света.

Из пассивных адаптеров ISDN поддерживаются в основном те, которые сделаны на основе микросхем Infineon (бывший Siemens) ISAC/HSCX/IPAC ISDN, а также адаптеры ISDN с микросхемами от Cologne Chip (только для шины ISA), адаптеры PCI с микросхемами Winbond W6692, некоторые адаптеры с набором микросхем Tiger300/320/ISAC и несколько адаптеров, построенных на фирменных наборах микросхем, такие, как AVM Fritz!Card PCI V.1.0 и AVM Fritz!Card PnP.

На данный момент из активных адаптеров ISDN поддерживаются AVM B1 (ISA и PCI) адаптеры BRI и AVM T1 PCI адаптеры PRI.

Документацию по **isdn4bsd** можно найти в каталоге `/usr/share/examples/isdn/` вашей системы FreeBSD или на домашней странице **isdn4bsd** (<http://www.freebsd-support.de/i4b/>), на которой также размещены ссылки на советы, замечания по ошибкам и более подробную информацию, например, на руководство по **isdn4bsd** (<http://people.FreeBSD.org/~hm/>).

Если вы заинтересованы в добавлении поддержки для различных протоколов ISDN, не поддерживаемых на данный момент адаптеров ISDN для PC или каких-то других усовершенствованиях **isdn4bsd**, пожалуйста, свяжитесь с Hellmuth Michaelis <[hm@FreeBSD.org](mailto:hm@FreeBSD.org)>.

Для обсуждения вопросов, связанных с установкой, настройкой и устранением неисправностей **isdn4bsd**, имеется список рассылки **freebsd-isdn** (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-isdn>).

`subscribe freebsd-isdn`

## 24.7.2. Терминальные адаптеры ISDN

Терминальные адаптеры (ТА) для ISDN выполняют ту же роль, что и модемы для обычных телефонных линий.

Большинство ТА используют стандартный набор AT-команд Hayes-модемов, и могут использоваться в качестве простой замены для модемов.

ТА будут работать точно так же, как и модемы, за исключением скорости соединения и пропускной способности, которые будут гораздо выше, чем у вашего старого модема. Вам потребуется настроить PPP точно также, как и в случае использования модема. Проверьте, что вы задали скорость работы последовательного порта максимально высокой.

Главным преимуществом использования ТА для подключения к провайдеру Интернет является возможность использования динамического PPP. Так как пространство адресов IP истощается все больше, большинство провайдеров не хочет больше выдавать вам статический IP-адрес. Большинство же маршрутизаторов не может использовать динамическое выделение IP-адресов.

ТА полностью полагаются на демон PPP, который используете из-за его возможностей и стабильности соединения. Это позволяет вам при использовании FreeBSD легко заменить модем на ISDN, если у вас

уже настроено соединение PPP. Однако, в тоже время любые проблемы, которые возникают с программой PPP, отражаются и здесь.

Если вы хотите максимальной надёжности, используйте PPP на уровне параметра ядра, а не пользовательский PPP.

Известно, что следующие TA работают с FreeBSD:

- Motorola BitSurfer и Bitsurfer Pro
- Adtran

Большинство остальных TA, скорее всего, тоже будут работать, производители TA прилагают все усилия для обеспечения поддержки практически всего набора стандартных AT-команд модема.

Как и в случае модемов проблемой использования внешнего TA является потребность в хорошем последовательном адаптере на вашем компьютере.

Вы должны прочесть учебник Последовательные устройства во FreeBSD ([http://www.FreeBSD.org/doc/en\\_US.ISO8859-1/articles/serial-uart/index.html](http://www.FreeBSD.org/doc/en_US.ISO8859-1/articles/serial-uart/index.html)) для того, чтобы в деталях понять работу последовательных устройств и осознать различие между асинхронными и синхронными последовательными портами.

TA, работающий со стандартным последовательным (асинхронным) портом PC, ограничивает вас скоростью 115.2 Кбит/с, хотя реально у вас соединение на скорости 128 Кбит/с. Чтобы использовать 128 Кбит/с, которые обеспечивает ISDN, полностью, вы должны подключить TA к синхронному последовательному адаптеру.

Не обманывайте себя, думая, что покупка встроенного TA поможет избежать проблемы синхронности/асинхронности. Встроенные TA просто уже имеют внутри стандартный последовательный порт PC. Все, что при этом достигается - это экономия дополнительных последовательного кабеля и электрической розетки.

Синхронный адаптер с TA по крайней мере так же быстр, как и отдельный маршрутизатор, а если он работает под управлением машины класса 386 с FreeBSD, то это гораздо более гибкое решение.

Выбор между использованием синхронного адаптера/TA или отдельного маршрутизатора в большей степени является религиозным вопросом. По этому поводу в списках рассылки была некоторая дискуссия. Рекомендуем поискать в архивах (<http://www.FreeBSD.org/search/index.html>) обсуждение полностью.

### 24.7.3. Отдельные мосты/маршрутизаторы ISDN

Мосты или маршрутизаторы ISDN не так уж специфичны для FreeBSD или для любой другой операционной системы. Для более подробного описания технологий маршрутизации и работы мостов, пожалуйста, обратитесь к справочникам по сетевым технологиям.

В контексте этого раздела термины маршрутизатор и сетевой мост будут использоваться как взаимозаменяемые.

Вместе с падением цен на простые мосты/маршрутизаторы ISDN, они становятся все более популярными. Маршрутизатор ISDN представляет собой маленькую коробочку, которая подключается непосредственно в вашу сеть Ethernet, и поддерживает связь с другим



мостом/маршрутизатором. Всё программное обеспечение для работы по PPP и другим протоколам встроено в маршрутизатор.

Маршрутизатор обладает гораздо большей пропускной способностью, чем стандартный ТА, так как он использует полное синхронное соединение ISDN.

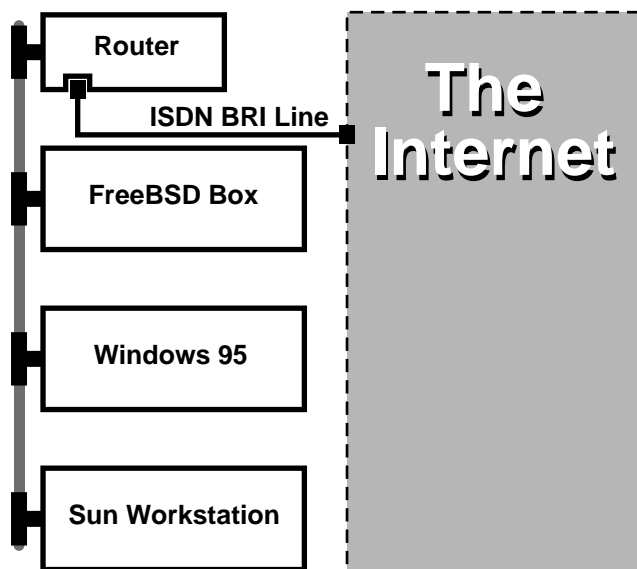
Основной проблемой с маршрутизаторами и мостами ISDN является то, что их совместная работа с оборудованием других производителей может оказаться под вопросом. Если вы собираетесь подключаться к провайдеру, то вы должны обсудить с ним то, что вам нужно.

Если вы планируете объединить два сегмента локальной сети, например, домашнюю сеть с сетью офиса, это самое простое решение с минимальными издержками на обслуживание. Так как вы покупаете оборудование для обеих сторон соединения, то можете быть уверены, что связь будет работать нормально.

Например, для соединения домашнего компьютера или сети подразделения к сети центрального офиса, может использоваться такая настройка:

#### Пример 24-1. Офис подразделения или домашняя сеть

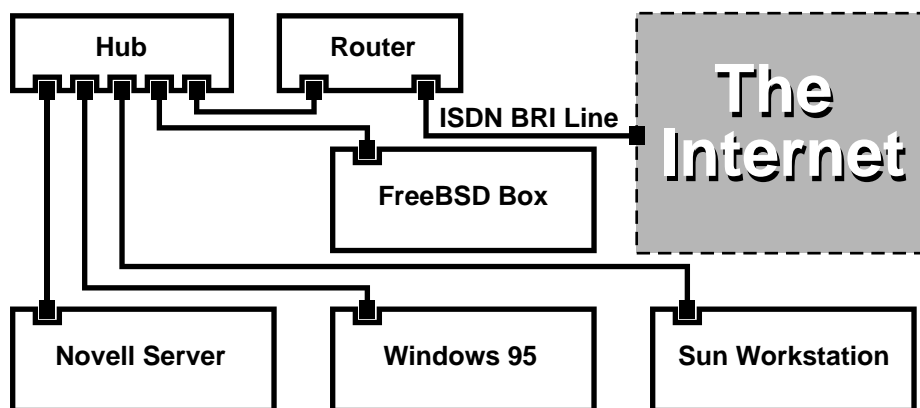
Сеть построена в топологии общей шины на основе 10 base 2 Ethernet ("thinnet" - "тонкий Ethernet"). Подключите маршрутизатор к сетевому кабелю с помощью трансивера AUI/10BT, если это нужно.



Если ваш домашний или удаленный офис представляет собой один компьютер, то для непосредственного подключения к маршрутизатору вы можете использовать витую пару с перекрестным соединением.

#### Пример 24-2. Центральный офис или другая локальная сеть

Сеть построена в топологии звезды на основе 10 Base T Ethernet ("витая пара").



Одним большим преимуществом большинства маршрутизаторов/мостов является то, что они позволяют иметь 2 *отдельных независимых* соединения PPP к 2 различным сайтам *одновременно*. Это не поддерживается в большинстве ТА, кроме специальных (обычно дорогих) моделей, имеющих по два последовательных порта. Не путайте это с балансировкой нагрузки, MPP и так далее.

Это может оказаться весьма полезной особенностью, например, если у вас имеется постоянное ISDN-соединение в вашем офисе, и вы хотите им воспользоваться, но не хотите задействовать дополнительный канал ISDN на работе. Маршрутизатор, расположенный в офисе, может использовать выделенное соединение по каналу В (64 Кбит/с) для Интернет, и одновременно другой канал В для отдельного соединения для передачи данных. Второй канал В может использоваться для входящих, исходящих и динамически распределяемых соединений (MPP и так далее) совместно с первым каналом В для повышения пропускной способности.

Мост Ethernet также позволяет вам передавать больше, чем просто трафик IP. Вы сможете передавать IPX/SPX и любые другие протоколы, которые вы используете.

## 24.8. Даemon преобразования сетевых адресов (natd)

*Текст предоставил Chern Lee.*

### 24.8.1. Обзор

Даemon преобразования сетевых адресов (Network Address Translation) во FreeBSD, широко известный как natd(8), является даемоном, который принимает входящие IP-пакеты, изменяет адрес отправителя на адрес локальной машины и повторно отправляет эти пакеты в потоке исходящих пакетов. natd(8) делает это, меняя IP-адрес отправителя и порт таким образом, что когда данные принимаются обратно, он может определить расположение источника начальных данных и переслать их машине, которая запрашивала данные изначально.

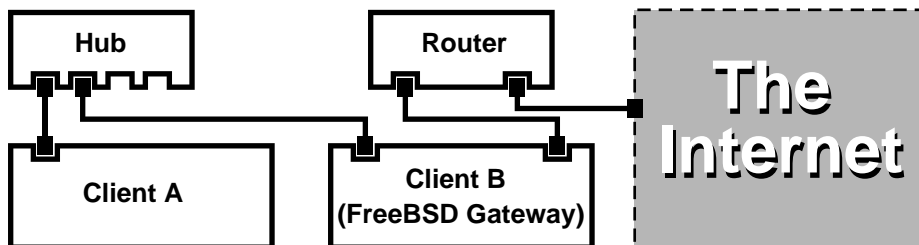
Чаще всего NAT используется для организации так называемого Совместного Использования Интернет.

## 24.8.2. Настройка

Из-за исчерпания пространства адресов в IPv4 и увеличения количества пользователей высокоскоростных каналов связи, таких, как кабельное подключение или DSL, необходимость в решении по Совместному Использованию Интернет растёт. Возможность подключить несколько компьютеров через единственное соединение и IP-адрес делает natd(8) подходящим решением.

Чаше всего у пользователя имеется машина, подключенная к кабельному каналу или каналу DSL с одним IP-адресом и есть желание использовать этот единственный подключенный компьютер для организации доступа в Интернет другим компьютерам в локальной сети.

Для этого машина FreeBSD, находящаяся в Интернет, должна выступать в роли шлюза. Эта шлюзовая машина должна иметь два сетевых адаптера—один для подключения к маршрутизатору Интернет, а другой для подключения к ЛВС. Все машины в локальной сети подключаются через сетевой концентратор или коммутатор.



Подобная конфигурация часто используется для совместного использования доступа в Интернет. Одна из подключенных к локальной сети машин подключается к Интернет. Остальные машины работают с Интернет посредством этой "шлюзовой" машины.

## 24.8.3. Настройка

В файле конфигурации ядра должны присутствовать следующие параметры:

```
options IPFIREWALL
options IPDIVERT
```

Дополнительно, если это нужно, можно добавить следующее:

```
options IPFIREWALL_DEFAULT_TO_ACCEPT
options IPFIREWALL_VERBOSE
```

В файле /etc/rc.conf должны быть такие строки:

```
gateway_enable="YES" ❶
firewall_enable="YES" ❷
firewall_type="OPEN" ❸
natd_enable="YES"
natd_interface="fxp0" ❹
natd_flags="" ❺
```

- ❶ Указывает машине выступать в качестве шлюза. Выполнение команды `sysctl net.inet.ip.forwarding=1` приведёт к тому же самому результату.
- ❷ При загрузке включает использование правил брандмауэра из файла `/etc/rc.firewall`.
- ❸ Здесь задается предопределенный набор правил брандмауэра, который разрешает все. Посмотрите файл `/etc/rc.firewall` для нахождения дополнительных типов.
- ❹ Указывает, через какой интерфейс передавать пакеты (интерфейс, подключенный к Интернет).
- ❺ Любые дополнительные параметры, передаваемые при запуске даемону `natd(8)`.

При использовании вышеуказанных параметров в файле `/etc/rc.conf` при загрузке будет запущена команда `natd -interface fxp0`. Эту команду можно запустить и вручную.

**Замечание:** Если для передачи `natd(8)` набирается слишком много параметров, возможно также использовать конфигурационный файл. В этом случае имя настроечного файла должно быть задано добавлением следующей строки в `/etc/rc.conf`:

```
natd_flags="-f /etc/natd.conf"
```

Файл `/etc/natd.conf` будет содержать перечень конфигурационных параметров, по одному в строке. К примеру, для примера из следующего раздела будет использоваться такой файл:

```
redirect_port tcp 192.168.0.2:6667 6667
redirect_port tcp 192.168.0.3:80 80
```

Для получения более полной информации о конфигурационном файле прочтите страницу справки по `natd(8)` относительно параметра `-f`.

Каждой машине и интерфейсу в ЛВС должен быть назначен IP-адрес из адресного пространства частных сетей, как это определено в RFC 1918 (<ftp://ftp.isi.edu/in-notes/rfc1918.txt>), а в качестве маршрутизатора по умолчанию должен быть задан IP-адрес машины с **natd** из внутренней сети.

Например, клиенты А и В в ЛВС имеют IP-адреса 192.168.0.2 и 192.168.0.3, а интерфейс машины с **natd** в локальной сети имеет IP-адрес 192.168.0.1. Маршрутизатором по умолчанию для клиентов А и В должна быть назначена машина с **natd**, то есть 192.168.0.1. Внешний, или Интернет-интерфейс машины с **natd** не требует особых настроек для работы `natd(8)`.

#### 24.8.4. Перенаправление портов

Минусом использования `natd(8)` является то, что машины в локальной сети недоступны из Интернет. Клиенты в ЛВС могут выполнять исходящие соединения во внешний мир, но не могут обслуживать входящие. Это является проблемой при запуске служб Интернет на клиентских машинах в локальной сети. Простым решением является перенаправление некоторых портов Интернет машины с **natd** на клиента локальной сети.

Пусть, к примеру, сервер IRC запущен на клиенте А, а веб-сервер работает на клиенте В. Чтобы это работало, соединения, принимаемые на портах 6667 (IRC) и 80 (веб), должны перенаправляться на соответствующие машины.

Программе `natd(8)` должна быть передана команда `-redirect_port` с соответствующими параметрами. Синтаксис следующий:

```
-redirect_port proto targetIP:targetPORT[-targetPORT]
    [_aliasIP:]aliasPORT[-aliasPORT]
    [remoteIP[:remotePORT[-remotePORT]]]
```

В примере выше аргументы должны быть такими:

```
-redirect_port tcp 192.168.0.2:6667 6667
-redirect_port tcp 192.168.0.3:80 80
```

При этом будут перенаправлены соответствующие порты *tcp* на клиентские машины в локальной сети.

Аргумент `-redirect_port` может использоваться для указания диапазонов портов, а не конкретного порта. Например, `tcp 192.168.0.2:2000-3000 2000-3000` будет перенаправлять все соединения, принимаемые на портах от 2000 до 3000, на порты от 2000 до 3000 клиента А.

Эти параметры можно указать при непосредственном запуске `natd(8)`, поместить их в параметр `natd_flags=" "` файла `/etc/rc.conf`, либо передать через конфигурационный файл.

Для получения информации о других параметрах настройки обратитесь к справочной странице по `natd(8)`.

### 24.8.5. Перенаправление адреса

Перенаправление адреса полезно, если имеется несколько адресов IP, и они должны быть на одной машине. В этой ситуации `natd(8)` может назначить каждому клиенту ЛВС свой собственный внешний IP-адрес. Затем `natd(8)` преобразует исходящие от клиентов локальной сети пакеты, заменяя IP-адреса на соответствующие внешние, и перенаправляет весь трафик, входящий на некоторый IP-адрес, обратно конкретному клиенту локальной сети. Это также называют статическим NAT. К примеру, пусть IP-адреса 128.1.1.1, 128.1.1.2 и 128.1.1.3 принадлежат шлюзовой машине **natd**. 128.1.1.1 может использоваться в качестве внешнего IP-адреса шлюзовой машины **natd**, тогда как 128.1.1.2 и 128.1.1.3 будут перенаправляться обратно к клиентам ЛВС А и В.

Синтаксис для `-redirect_address` таков:

```
-redirect_address localIP publicIP
```

localIP

Внутренний IP-адрес клиента локальной сети.

publicIP

Внешний IP, соответствующий клиенту локальной сети.

В примере этот аргумент будет выглядеть так:

```
-redirect_address 192.168.0.2 128.1.1.2
-redirect_address 192.168.0.3 128.1.1.3
```

Как и для `-redirect_port`, эти аргументы также помещаются в строку `natd_flags=" "` файла `/etc/rc.conf` или передаются через конфигурационный файл. При перенаправлении адресов нет нужды в перенаправлении портов, потому что перенаправляются все данные, принимаемые для конкретного IP-адреса.

Внешние IP-адреса машины с **natd** должны быть активизированы и являться синонимами для внешнего интерфейса. Обратитесь к `rc.conf(5)`, чтобы это сделать.

## 24.9. IP по параллельному порту (PLIP)

PLIP позволяет нам работать с TCP/IP по параллельному порту. Это полезно для машин без сетевых адаптеров или для установки на ноутбуки. В этом разделе мы обсудим:

- создание кабеля для параллельного порта (laplink).
- Соединение двух компьютеров посредством PLIP.

### 24.9.1. Создание параллельного кабеля

Вы можете приобрести кабель для параллельного порта в большинстве магазинов, торгующих комплектующими. Если вы его не найдете, или же просто хотите знать, как он делается, то следующая таблица поможет вам сделать такой кабель из обычного принтерного кабеля для параллельного порта.

**Таблица 24-1.** Распайка кабеля для параллельного порта для сетевой работы

A-name	A-End	B-End	Описание	Post/Bit
DATA0	2	15	Data	0/0x01
-ERROR	15	2		1/0x08
DATA1	3	13	Data	0/0x02
+SLCT	13	3		1/0x10
DATA2	4	12	Data	0/0x04
+PE	12	4		1/0x20
DATA3	5	10	Strobe	0/0x08
-ACK	10	5		1/0x40
DATA4	6	11	Data	0/0x10
BUSY	11	6		1/0x80
GND	18-25	18-25	GND	-

### 24.9.2. Настройка PLIP

Прежде всего вы должны найти laplink-кабель. Затем удостоверьтесь, что на обоих компьютерах в ядро включена поддержка драйвера lpt(4):

```
# grep lp /var/run/dmesg.boot
lpt0: <Printer> on ppbus0
lpt0: Interrupt-driven port
```

Управление параллельным портом должно выполняться по прерываниям. Во FreeBSD 4.X в файле конфигурации ядра должна присутствовать строка, подобная следующей:

```
device ppc0 at isa? irq 7
```

Во FreeBSD 5.X файл /boot/device.hints должен содержать следующие строки:

```
hint.ppc.0.at="isa"
hint.ppc.0.irq="7"
```

Затем проверьте, что файл конфигурации ядра имеет строку `device plip`, или загружен ли модуль ядра `plip.ko`. В обоих случаях интерфейс работы с сетью по параллельному порту должен присутствовать на момент прямого использования команды `ifconfig` (8). Во FreeBSD 4.X это должно быть примерно так:

```
# ifconfig lp0
lp0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
```

а для FreeBSD 5.X:

```
# ifconfig plip0
plip0: flags=8810<POINTOPOINT,SIMPLEX,MULTICAST> mtu 1500
```

**Замечание:** Имя устройства, используемого для параллельного интерфейса, во FreeBSD 4.X (`lpX`) и FreeBSD 5.X (`plipX`).

Подключите кабель `laplink` к параллельным интерфейсам на обоих компьютерах.

Настройте параметры сетевого интерфейса с обеих сторон, работая как пользователь `root`. К примеру, если вы хотите соединить хост `host1`, на котором работает FreeBSD 4.X, с хостом `host2` под управлением FreeBSD 5.X:

```
                host1 <-----> host2
IP Address      10.0.0.1      10.0.0.2
```

Настройте интерфейс на машине `host1`, выполнив:

```
# ifconfig plip0 10.0.0.1 10.0.0.2
```

Настройте интерфейс на машине `host2`, выполнив:

```
# ifconfig lp0 10.0.0.2 10.0.0.1
```

Теперь вы должны получить работающее соединение. Пожалуйста, прочтите страницы руководства по `lp(4)` и `lpt(4)` для выяснения деталей.

Вы должны также добавить оба хоста в `/etc/hosts`:

```
127.0.0.1          localhost.my.domain localhost
10.0.0.1           host1.my.domain host1
10.0.0.2           host2.my.domain
```

Чтобы проверить работу соединения, перейдите к каждому хосту и выполните тестирование соединения с другой машиной посредством команды `ping`. К примеру, на машине `host1`:

```
# ifconfig lp0
lp0: flags=8851<UP,POINTOPOINT,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    inet 10.0.0.1 --> 10.0.0.2 netmask 0xff000000
# netstat -r
Routing tables
```

```
Internet:
```

```

Destination      Gateway          Flags          Refs          Use          Netif  Expire
host2             host1           UH              0              0             lp0
# ping -c 4 host2
PING host2 (10.0.0.2): 56 data bytes
64 bytes from 10.0.0.2: icmp_seq=0 ttl=255 time=2.774 ms
64 bytes from 10.0.0.2: icmp_seq=1 ttl=255 time=2.530 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=255 time=2.556 ms
64 bytes from 10.0.0.2: icmp_seq=3 ttl=255 time=2.714 ms

--- host2 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.530/2.643/2.774/0.103 ms

```

## 24.10. IPv6

*Первоначальный текст написал Aaron Kaplan. Реструктуризацию и добавления внёс Tom Rhodes. Расширил Brad Davis.*

IPv6 (также называемый IPng "IP next generation" - следующее поколение IP) является новой версией широко известного протокола IP (называемого также IPv4). Как и другие современные системы \*BSD, FreeBSD включает эталонную реализацию IPv6 от KAME. Так что система FreeBSD поставляется со всем, что вам нужно для экспериментирования с IPv6. Этот раздел посвящён настройке и запуску в работу IPv6.

В начале 1990-х люди стали беспокоиться о быстро иссякающем адресном пространстве IPv4. Принимая во внимание темпы роста Интернет, имелись основные проблемы:

- Нехватка адресов. Сегодня это не такая большая проблема, так как стали применяться адресные пространства для частных сетей (10.0.0.0/8, 192.168.0.0/24 и так далее) и технология преобразования сетевых адресов (NAT - Network Address Translation).
- Таблицы маршрутов становятся чересчур большими. Это всё ещё является проблемой сегодня.

IPv6 решает эти и многие другие вопросы:

- 128-битное адресное пространство. Другими словами, теоретически доступны 340,282,366,920,938,463,374,607,431,768,211,456 адреса. Это означает плотность примерно в  $6.67 \cdot 10^{27}$  адресов IPv6 на квадратный метр нашей планеты.
- Маршрутизаторы будут хранить в своих таблицах только агрегированные адреса сетей, что уменьшает средний размер таблицы маршрутизации до 8192 записей.

Имеется также множество других полезных особенностей IPv6, таких, как:

- Автоматическая настройка адреса (RFC2462 (<http://www.ietf.org/rfc/rfc2462.txt>))
- Групповые адреса ("один к нескольким из многих")
- Обязательные адреса множественной рассылки
- IPsec (IP security - безопасный IP)



- Упрощённая структура заголовка
- Мобильный IP
- Механизмы преобразования IPv6-в-IPv4

Для получения дополнительной информации посмотрите:

- Обзор IPv6 на сайте playground.sun.com (<http://playground.sun.com/pub/ipng/html/ipng-main.html>)
- KAME.net (<http://www.kame.net>)
- 6bone.net (<http://www.6bone.net>)

### 24.10.1. Основы адресации IPv6

Существуют различные типы адресов IPv6: одноадресные (Unicast), групповые (Anycast) и многоадресные (Multicast).

Адреса типа Unicast хорошо всем известны. Пакет, посланный на такой адрес, достигает в точности интерфейса, который этому адресу соответствует.

Адреса типа Anycast синтаксически неотличимы от адресов Unicast, но они адресуют группу интерфейсов. Пакет, направленный такому адресу, попадёт в ближайший (согласно метрике маршрутизатора) интерфейс. Адреса Anycast могут использоваться только маршрутизаторами.

Адреса типа Multicast идентифицируют группу интерфейсов. Пакет, посланный на такой адрес, достигнет всех интерфейсов, привязанных к группе многоадресного вещания.

**Замечание:** Широковещательные адреса IPv4 (обычно xxx.xxx.xxx.255) выражаются адресами многоадресного вещания IPv6.

**Таблица 24-2. Зарезервированные адреса IPv6**

IPv6 адрес	Длина префикса (биты)	Описание	Заметки
::	128 бит	нет описания	cf. 0.0.0.0 в IPv4
:::1	128 бит	loopback адрес	cf. 127.0.0.1 в IPv4
::00:xx:xx:xx:xx	96 бит	встроенный IPv4	Нижние 32 бита это адрес IPv4. Также называется "IPv4 совместимым IPv6 адресом"
::ff:xx:xx:xx:xx	96 бит	Адрес IPv6, отображенный на IPv4	Нижние 32 бита это адрес IPv4. Для хостов, не поддерживающих IPv6.
fe80:: - feb::	10 бит	link-local	cf. loopback адрес в IPv4
fec0:: - fef::	10 бит	site-local	
ff::	8 бит	широковещательный	

IPv6 адрес	Длина префикса (биты)	Описание	Заметки
001 (основание 2)	3 бит	global unicast	Все global unicast адреса присваиваются из этого пула. Первые три бита "001".

### 24.10.2. Чтение адресов IPv6

Каноническая форма представляется в виде `x:x:x:x:x:x:x`, где каждый символ "x" является 16-битовым шестнадцатеричным числом. К примеру, `FEBC:A574:382B:23C1:AA49:4592:4EFE:9982`

Часто в адресе присутствуют длинные строчки, заполненные нулями, поэтому одна такая последовательность на адрес может быть сокращена до "::". Кроме того, до трех ведущих "0" на шестнадцатеричную четверку могут быть пропущены. К примеру, `fe80::1` соответствует канонической форме `fe80:0000:0000:0000:0000:0000:0000:0001`.

В третьей форме последние 32 бита записываются в широко известном (десятичном) стиле IPv4 с точками "." в качестве разделителей. Например, `f2002::10.0.0.1` соответствует (шестнадцатеричному) каноническому представлению `2002:0000:0000:0000:0000:0000:0a00:0001`, которое, в свою очередь, равнозначно записи `2002::a00:1`.

Теперь читатель должен понять следующую запись:

```
# ifconfig
```

```
rl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
    inet 10.0.0.10 netmask 0xfffff00 broadcast 10.0.0.255
    inet6 fe80::200:21ff:fe03:8e1%rl0 prefixlen 64 scopeid 0x1
    ether 00:00:21:03:08:e1
    media: Ethernet autoselect (100baseTX )
    status: active
```

`fe80::200:21ff:fe03:8e1%rl0` является автоматически настроенным локальным адресом. Он генерируется из MAC адреса в процессе автоматической конфигурации.

Для получения дополнительной информации о структуре адресов IPv6 обратитесь к RFC3513 (<http://www.ietf.org/rfc/rfc3513.txt>).

### 24.10.3. Настройка подключения

На данный момент существуют четыре способа подключиться к другим хостам и сетям IPv6:

- Подключиться к экспериментальному bbone
- Получить сеть IPv6 от вышестоящего провайдера. Для получения рекомендаций обратитесь к вашему провайдеру Интернет.
- Туннелировать посредством 6-в-4 (RFC3068 (<http://www.ietf.org/rfc/rfc3068.txt>))
- Использовать порт `net/freenet6`, если вы используете коммутируемое соединение.

Здесь мы будем рассматривать подключение к bbone, так как на данный момент это является самым популярным способом.

Сначала взгляните на сайт bbone (<http://www.bbone.net/>) и найдите ближайшую к вам точку подключения к bbone. Напишите ответственному и при некоторой удаче вам дадут инструкции по настройке соединения. Обычно это касается настройки туннеля GRE (gif).

Вот типичный пример настройки туннеля gif(4):

```
# ifconfig gif0 create
# ifconfig gif0
gif0: flags=8010<POINTOPOINT,MULTICAST> mtu 1280
# ifconfig gif0 tunnel MY_IPv4_ADDR HIS_IPv4_ADDR
# ifconfig gif0 inet6 alias MY_ASSIGNED_IPv6_TUNNEL_ENDPOINT_ADDR
```

Замените слова, написанные заглавными буквами, информацией, которую вам дал вышестоящий узел bbone.

При этом установится туннель. Проверьте работу туннеля утилитой ping6(8) с адресом ff02::1%gif0. Вы должны получить два положительных ответа.

**Замечание:** Если вы заинтригованы адресом ff02::1%gif0, скажем, что это адрес многоадресного вещания. %gif0 указывает на использование такого адреса с сетевым интерфейсом gif0. Так как мы выполняем ping над адресом многоадресного вещания, то другая сторона туннеля также должна ответить.

Теперь настройка маршрута к вашей вышестоящей точке подключения bbone должна быть весьма проста:

```
# route add -inet6 default -interface gif0
# ping6 -n MY_UPLINK

# traceroute6 www.jp.FreeBSD.org
(3ffe:505:2008:1:2a0:24ff:fe57:e561) from 3ffe:8060:100::40:2, 30 hops max, 12 byte packets
 1  atnet-meta6  14.147 ms  15.499 ms  24.319 ms
 2  6bone-gw2-ATNET-NT.ipv6.tilab.com  103.408 ms  95.072 ms  *
 3  3ffe:1831:0:ffff::4  138.645 ms  134.437 ms  144.257 ms
 4  3ffe:1810:0:6:290:27ff:fe79:7677  282.975 ms  278.666 ms  292.811 ms
 5  3ffe:1800:0:ff00::4  400.131 ms  396.324 ms  394.769 ms
 6  3ffe:1800:0:3:290:27ff:fe14:cdee  394.712 ms  397.19 ms  394.102 ms
```

Эта выдача будет отличаться от машины к машине. Теперь вы должны суметь достигнуть сайта IPv6 [www.kame.net](http://www.kame.net) (<http://www.kame.net>) и увидеть танцующую черепаху — в случае, если ваш браузер поддерживает IPv6, как, например, [www/mozilla](http://www.mozilla) или **Konqueror**, который входит в x11/kdebase3, или [www/epiphany](http://www/epiphany).

#### 24.10.4. DNS в мире IPv6

Для IPv6 использовались два типа записей DNS. IETF объявил записи A6 устаревшими. Стандартом на данный момент являются записи AAAA.

Использование записей AAAA достаточно просто. Назначение вашему имени хоста нового адреса IPv6 достигается просто добавлением:

```
MYHOSTNAME          AAAA      MYIPv6ADDR
```

к вашему первичному файлу DNS зоны. В случае, если вы не обслуживаете собственные зоны DNS, обратитесь к вашему провайдеру DNS. Имеющиеся версии **bind** (версий 8.3 и 9) и `dns/djbdns` (с патчем IPv6) поддерживают записи AAAA.

## 24.10.5. Внесение необходимых изменений в `/etc/rc.conf`

### 24.10.5.1. Настройки клиентов IPv6

Эти установки помогут вам настроить компьютер, который будет работать в сети как клиент, а не как маршрутизатор. Для включения настройки интерфейсов через `rtol(8)` при загрузке, все, что вам потребуется, это добавить следующую строку:

```
ipv6_enable="YES"
```

Для статического присвоения IP адреса, такого как `2001:471:1f11:251:290:27ff:fee0:2093`, интерфейсу `fxp0`, добавьте:

```
ipv6_ifconfig_fxp0="2001:471:1f11:251:290:27ff:fee0:2093"
```

Для назначения маршрутизатором по умолчанию `2001:471:1f11:251::1`, добавьте следующую строку к `/etc/rc.conf`:

```
ipv6_defaultrouter="2001:471:1f11:251::1"
```

### 24.10.5.2. Настройки маршрутизатора/шлюза IPv6

Этот раздел поможет вам использовать инструкции, которые выдал провайдер туннеля, например, `bone` (<http://www.bone.net/>), и сделать эти настройки постоянными. Для восстановления туннеля при загрузке системы используйте в `/etc/rc.conf` нижеприведенные настройки.

Задайте список туннельных интерфейсов (Generic Tunneling interfaces), которые необходимо настроить, например `gif0`:

```
gif_interfaces="gif0"
```

Для настройки интерфейса с локальным подключением на `MY_IPv4_ADDR` к удаленной точке `REMOTE_IPv4_ADDR`:

```
gifconfig_gif0="MY_IPv4_ADDR REMOTE_IPv4_ADDR"
```

Для включения IPv6 адреса, который был вам присвоен для использования в подключении к туннелю IPv6, добавьте:

```
ipv6_ifconfig_gif0="MY_ASSIGNED_IPv6_TUNNEL_ENDPOINT_ADDR"
```

Затем все, что вам потребуется сделать, это добавить маршрут по умолчанию для IPv6. Это другая сторона туннеля IPv6:

```
ipv6_defaultrouter="MY_IPv6_REMOTE_TUNNEL_ENDPOINT_ADDR"
```

### 24.10.6. Распространение маршрутов и автоматическая настройка хостов

Этот раздел поможет вам настроить `rtadvd(8)` для распространения маршрута IPv6 по умолчанию.

Для включения `rtadvd(8)` вам понадобится добавить в `/etc/rc.conf` следующую строку:

```
rtadvd_enable="YES"
```

Важно указать интерфейс, на котором выполняется запрос маршрутизатора IPv6. Например, для указания `rtadvd(8)` использовать `fxp0`:

```
rtadvd_interfaces="fxp0"
```

Теперь мы должны создать файл настройки, `/etc/rtadvd.conf`. Вот пример:

```
fxp0:\n:addr$#1:addr="2001:471:1f11:246::":prefixlen#64:tc=ether:
```

Замените `fxp0` на интерфейс, который вы будете использовать.

Затем, замените `2001:471:1f11:246::` на префикс вашего размещения.

Если у вас выделенная подсеть /64, больше ничего менять не потребуется. Иначе, вам потребуется изменить `prefixlen#` на корректное значение.

## 24.11. Асинхронный режим передачи (ATM) в FreeBSD 5.X

*Предоставил Harti Brandt.*

### 24.11.1. Классическая настройка IP через ATM (PVC)

Классический IP через ATM (CLIP) это простейший метод использования асинхронного режима передачи (Asynchronous Transfer Mode, ATM) с IP. Он может быть использован с коммутируемыми подключениями (switched connections, SVC) и с постоянными подключениями (permanent connections, PVC). В этом разделе будет описано как настроить сеть на основе PVC.

#### 24.11.1.1. Полностью объединенные конфигурации

Первый метод для настройки CLIP с PVC это подключение каждого компьютера к каждому в сети с выделенным PVC. Хотя настройка проста, она непрактична для большого количества компьютеров. В примере предполагается, что в сети есть четыре компьютера, каждый подключенный к ATM сети с помощью карты ATM адаптера. Первый шаг это планирование IP адресов и ATM подключений между компьютерами. Мы используем:

Хост	IP адрес
hostA	192.168.173.1
hostB	192.168.173.2
hostC	192.168.173.3
hostD	192.168.173.4

Для сборки полностью объединенной сети нам потребуется по одному ATM соединению между каждой парой компьютеров:

Компьютеры	VPI.VCI соединение
hostA - hostB	0.100
hostA - hostC	0.101
hostA - hostD	0.102
hostB - hostC	0.103
hostB - hostD	0.104
hostC - hostD	0.105

Значения VPI и VCI на каждом конце соединения конечно могут отличаться, но для упрощения мы предполагаем, что они одинаковы. Затем нам потребуется настроить ATM интерфейсы на каждом хосте:

```
hostA# ifconfig hatm0 192.168.173.1 up
hostB# ifconfig hatm0 192.168.173.2 up
hostC# ifconfig hatm0 192.168.173.3 up
hostD# ifconfig hatm0 192.168.173.4 up
```

предполагая, что ATM интерфейс называется hatm0 на всех хостах. Теперь PVC необходимо настроить на hostA (мы предполагаем, что ATM коммутаторы уже настроены, вам необходимо свериться с руководством на коммутатор за информацией по настройке).

```
hostA# atmconfig natm add 192.168.173.2 hatm0 0 100 llc/snap ubr
hostA# atmconfig natm add 192.168.173.3 hatm0 0 101 llc/snap ubr
hostA# atmconfig natm add 192.168.173.4 hatm0 0 102 llc/snap ubr

hostB# atmconfig natm add 192.168.173.1 hatm0 0 100 llc/snap ubr
hostB# atmconfig natm add 192.168.173.3 hatm0 0 103 llc/snap ubr
hostB# atmconfig natm add 192.168.173.4 hatm0 0 104 llc/snap ubr

hostC# atmconfig natm add 192.168.173.1 hatm0 0 101 llc/snap ubr
hostC# atmconfig natm add 192.168.173.2 hatm0 0 103 llc/snap ubr
hostC# atmconfig natm add 192.168.173.4 hatm0 0 105 llc/snap ubr

hostD# atmconfig natm add 192.168.173.1 hatm0 0 102 llc/snap ubr
hostD# atmconfig natm add 192.168.173.2 hatm0 0 104 llc/snap ubr
hostD# atmconfig natm add 192.168.173.3 hatm0 0 105 llc/snap ubr
```

Конечно, вместо UBR может быть использован другой тип, если ATM адаптер поддерживает это. В этом случае имя типа дополняется параметрами трафика. Помощь по atmconfi g(8) может быть получена командой:

```
# atmconfig help natm add
```

или на странице справочника `atmconfig(8)`.

Та же настройка может быть выполнена через `/etc/rc.conf`. Для `hostA` это будет выглядеть примерно так:

```
network_interfaces="lo0 hatm0"  
ifconfig_hatm0="inet 192.168.173.1 up"  
natm_static_routes="hostB hostC hostD"  
route_hostB="192.168.173.2 hatm0 0 100 llc/snap ubr"  
route_hostC="192.168.173.3 hatm0 0 101 llc/snap ubr"  
route_hostD="192.168.173.4 hatm0 0 102 llc/snap ubr"
```

Текущий статус всех маршрутов `CLIP` может быть получен командой:

```
hostA# atmconfig natm show
```

## V. Приложения



# Приложение А. Получение FreeBSD

## А.1. Издатели CDROM и DVD

### А.1.1. Коробочная версия

FreeBSD доступна в коробочной версии (FreeBSD CD диски, дополнительное программное обеспечение, печатная документация) от нескольких поставщиков:

- CompUSA  
WWW: <http://www.compusa.com/>
- Frys Electronics  
WWW: <http://www.frys.com/>

### А.1.2. Наборы CD и DVD

Наборы FreeBSD CD и DVD доступны у множества онлайн поставщиков:

- BSD Mall (подразделение Daemon News)  
PO Box 161  
Nauvoo, IL 62354  
USA  
Телефон: +1 866 273-6255  
Факс: +1 217 453-9956  
Email: [<sales@bsdmail.com>](mailto:sales@bsdmail.com)  
WWW: <http://www.bsdmail.com/freebsd1.html>
- BSD-Systems  
Email: [<info@bsd-systems.co.uk>](mailto:info@bsd-systems.co.uk)  
WWW: <http://www.bsd-systems.co.uk>
- fastdiscs.com  
6 Eltham Close  
Leeds, LS6 2TY  
United Kingdom  
Телефон: +44 870 1995 171  
Email: [<sales@fastdiscs.com>](mailto:sales@fastdiscs.com)  
WWW: <http://fastdiscs.com/freebsd/>
- FreeBSD Mall, Inc.  
3623 Sanford Street  
Concord, CA 94520-1405  
США  
Телефон: +1 925 674-0783  
Факс: +1 925 674-0821  
Email: [<info@freebsdmail.com>](mailto:info@freebsdmail.com)

WWW: <http://www.freebsdmail.com/>

- FreeBSD Services Ltd  
11 Lapwing Close  
Bicester  
OX26 6XR  
Великобритания  
WWW: <http://www.freebsd-services.com/>
- Ingram Micro  
1600 E. St. Andrew Place  
Santa Ana, CA 92705-4926  
США  
Телефон: 1 (800) 456-8000  
WWW: <http://www.ingrammicro.com/>
- Hinner EDV  
St. Augustinus-Str. 10  
D-81825 München  
Германия  
Телефон: (089) 428 419  
WWW: <http://www.hinner.de/linux/freebsd.html>
- Ikarios  
22-24 rue Voltaire  
92000 Nanterre  
Франция  
WWW: <http://ikarios.com/form/#freebsd>
- JMC Software  
Ирландия  
Телефон: 353 1 6291282  
WWW: <http://www.thelinuxmall.com>
- The Linux Emporium  
Hilliard House, Lester Way  
Wallingford  
OX10 9TA  
Великобритания  
Телефон: +44 1491 837010  
Факс: +44 1491 837016  
WWW: <http://www.linuxemporium.co.uk/products/freebsd/>
- Linux System Labs Australia  
21 Ray Drive  
Balwyn North  
VIC - 3104  
Австралия  
Телефон: +61 3 9857 5918  
Факс: +61 3 9857 8974  
WWW: <http://www.lsl.com.au>
- LinuxCenter.Ru

ул. Галерная, 55  
 Санкт-Петербург  
 190000  
 Россия  
 Телефон: +7-812-3125208  
 Email: <info@linuxcenter.ru>  
 WWW: <http://linuxcenter.ru/freebsd>

- UNIXDVD.COM LTD  
 57 Primrose Avenue  
 Sheffield  
 S5 6FS  
 Великобритания  
 WWW: <http://www.unixdvd.com/>

### **A.1.3. Распространители**

Если вы продавец и хотите заниматься FreeBSD CDROM, пожалуйста свяжитесь с распространителем:

- Cylogistics  
 809B Cuesta Dr., #2149  
 Mountain View, CA 94040  
 США  
 Телефон: +1 650 694-4949  
 Факс: +1 650 694-4953  
 Email: <sales@cylogistics.com>  
 WWW: <http://www.cylogistics.com/>
- FreeBSD Services Ltd  
 11 Lapwing Close  
 Bicester  
 OX26 6XR  
 Великобритания  
 WWW: <http://www.freebsd-services.com/>
- Kudzu, LLC  
 7375 Washington Ave. S.  
 Edina, MN 55439  
 США  
 Телефон: +1 952 947-0822  
 Факс: +1 952 947-0876  
 Email: <sales@kudzuenterpises.com>
- LinuxCenter.Ru  
 ул. Галерная, 55  
 Санкт-Петербург  
 190000  
 Россия  
 Телефон: +7-812-3125208

Email: <info@linuxcenter.ru>

WWW: <http://linuxcenter.ru/freebsd>

- Navarre Corp  
7400 49th Ave South  
New Hope, MN 55428  
США  
Телефон: +1 763 535-8333  
Факс: +1 763 535-0341  
WWW: <http://www.navarre.com/>

## А.2. FTP сайты

Официальным источником FreeBSD являются анонимные FTP зеркала по всему миру. Сайт <ftp://ftp.FreeBSD.org/pub/FreeBSD/> имеет хорошее подключение и поддерживает большое количество одновременных соединений, но для вас возможно потребуется найти "ближайшее" зеркало (особенно если вы решили настроить у себя какой-то из видов зеркал).

База данных зеркал FreeBSD (<http://mirrorlist.FreeBSD.org/>) предпочтительнее по сравнению со списком зеркал в Руководстве, поскольку информация в базе собирается из DNS, а не из статического списка узлов.

Кроме того, FreeBSD доступна через анонимный FTP со следующих зеркал. Если вы выбрали получение FreeBSD через анонимный FTP, пожалуйста выберите ближайший к вам сайт. Зеркала из списка "Основных зеркал" обычно содержат полный архив FreeBSD (все доступные на данный момент версии для каждой архитектуры), скорость загрузки возможно будет больше с зеркала, расположенного в вашей стране или регионе. Сайты каждой страны содержат последнюю версию для наиболее популярных архитектур, но на них может не быть полного архива FreeBSD. Все сайты предоставляют доступ через анонимный FTP, а некоторые предоставляют доступ и другими методами. Для каждого сайта приведен список методов доступа в скобках после имени хоста.

Центральные серверы, Основные зеркала, Indonesia, Австралия, Австрия, Аргентина, Бразилия, Великобритания, Венгрия, Германия, Гонконг, Греция, Дания, Ирландия, Исландия, Испания, Италия, Канада, Китай, Корея, Литва, Нидерланды, Норвегия, Польша, Португалия, Россия, Румыния, США, Саудовская Аравия, Сингапур, Словацкая Республика, Словения, Тайвань, Турция, Украина, Финляндия, Франция, Хорватия, Чехия, Швейцария, Швеция, Эстония, Южная Африка, Япония.

(as of 2004/10/26 16:33:28 UTC)

Центральные серверы

- <ftp://ftp.FreeBSD.org/pub/FreeBSD/> (ftp)

Основные зеркала

В случае проблем пожалуйста свяжитесь с администратором <mirror-admin@FreeBSD.org> этого домена.

- <ftp://ftp1.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp3.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp4.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp5.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp6.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp7.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp8.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp9.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp10.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp11.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp12.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp13.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp14.FreeBSD.org/pub/FreeBSD/> (ftp)

#### Indonesia

В случае проблем пожалуйста свяжитесь с администратором `<hostmaster@id.FreeBSD.org>` этого домена.

- <ftp://ftp.id.FreeBSD.org/FreeBSD/> (ftp / [http \(http://ftp.id.FreeBSD.org/\) / rsync](http://ftp.id.FreeBSD.org/))

#### Австралия

В случае проблем пожалуйста свяжитесь с администратором `<hostmaster@au.FreeBSD.org>` этого домена.

- <ftp://ftp.au.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.au.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp3.au.FreeBSD.org/pub/FreeBSD/> (ftp)

#### Австрия

В случае проблем пожалуйста свяжитесь с администратором `<hostmaster@at.FreeBSD.org>` этого домена.

- <ftp://ftp.at.FreeBSD.org/pub/FreeBSD/> (ftp / [http \(http://ftp.at.FreeBSD.org/pub/FreeBSD/\)](http://ftp.at.FreeBSD.org/pub/FreeBSD/))
- <ftp://ftp2.at.FreeBSD.org/pub/FreeBSD/> (ftp / [http \(http://ftp2.at.freebsd.org/pub/FreeBSD/\) / rsync](http://ftp2.at.freebsd.org/pub/FreeBSD/))

## Аргентина

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@ar.FreeBSD.org> этого домена.

- <ftp://ftp.ar.FreeBSD.org/pub/FreeBSD/> (ftp)

## Бразилия

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@br.FreeBSD.org> этого домена.

- <ftp://ftp.br.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp.br.FreeBSD.org/>))
- <ftp://ftp2.br.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp3.br.FreeBSD.org/pub/FreeBSD/> (ftp / rsync)
- <ftp://ftp4.br.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp5.br.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp6.br.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp7.br.FreeBSD.org/pub/FreeBSD/> (ftp)

## Великобритания

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@uk.FreeBSD.org> этого домена.

- <ftp://ftp.uk.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.uk.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp2.uk.FreeBSD.org/>) / rsync)
- <ftp://ftp3.uk.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp4.uk.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp5.uk.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp6.uk.FreeBSD.org/pub/FreeBSD/> (ftp)

## Венгрия

В случае проблем пожалуйста свяжитесь с администратором <mohacsi@ik.bme.hu> этого домена.

- <ftp://ftp.hu.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp.hu.FreeBSD.org/>) / rsync)
- <ftp://ftp2.hu.FreeBSD.org/pub/FreeBSD/> (ftp)

## Германия

В случае проблем пожалуйста свяжитесь с администратором <de-bsd-hubs@de.FreeBSD.org > этого домена.

- <ftp://ftp.de.FreeBSD.org/pub/FreeBSD/> (ftp)

- <ftp://ftp2.de.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp2.de.FreeBSD.org/pub/FreeBSD/>) / rsync)
- <ftp://ftp3.de.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp4.de.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp4.de.FreeBSD.org/pub/FreeBSD/>) / rsync)
- <ftp://ftp5.de.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp6.de.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp7.de.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp7.de.FreeBSD.org/pub/FreeBSD/>))
- <ftp://ftp8.de.FreeBSD.org/pub/FreeBSD/> (ftp)

#### Гонконг

- <ftp://ftp.hk.FreeBSD.org/pub/FreeBSD/> (ftp)

#### Греция

В случае проблем пожалуйста свяжитесь с администратором [<hostmaster@gr.FreeBSD.org>](mailto:hostmaster@gr.FreeBSD.org) этого домена.

- <ftp://ftp.gr.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.gr.FreeBSD.org/pub/FreeBSD/> (ftp)

#### Дания

В случае проблем пожалуйста свяжитесь с администратором [<hostmaster@dk.FreeBSD.org>](mailto:hostmaster@dk.FreeBSD.org) этого домена.

- <ftp://ftp.dk.FreeBSD.org/pub/FreeBSD/> (ftp / ftpv6 / http (<http://ftp.dk.FreeBSD.org/pub/FreeBSD/>))
- <ftp://ftp2.dk.FreeBSD.org/pub/FreeBSD/> (ftp)

#### Ирландия

В случае проблем пожалуйста свяжитесь с администратором [<hostmaster@ie.FreeBSD.org>](mailto:hostmaster@ie.FreeBSD.org) этого домена.

- <ftp://ftp.ie.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.ie.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp2.ie.FreeBSD.org/pub/FreeBSD/>) / rsync)
- <ftp://ftp3.ie.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp3.ie.FreeBSD.org/pub/FreeBSD/>) / rsync)

#### Исландия

В случае проблем пожалуйста свяжитесь с администратором [<hostmaster@is.FreeBSD.org>](mailto:hostmaster@is.FreeBSD.org) этого домена.

- <ftp://ftp.is.FreeBSD.org/pub/FreeBSD/> (ftp / rsync)

## Испания

В случае проблем пожалуйста свяжитесь с администратором `<hostmaster@es.FreeBSD.org>` этого домена.

- <ftp://ftp.es.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.es.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp3.es.FreeBSD.org/pub/FreeBSD/> (ftp)

## Италия

В случае проблем пожалуйста свяжитесь с администратором `<hostmaster@it.FreeBSD.org>` этого домена.

- <ftp://ftp.it.FreeBSD.org/pub/FreeBSD/> (ftp)

## Канада

В случае проблем пожалуйста свяжитесь с администратором `<hostmaster@ca.FreeBSD.org>` этого домена.

- <ftp://ftp.ca.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.ca.FreeBSD.org/> (ftp)

## Китай

В случае проблем пожалуйста свяжитесь с администратором `<phj@cn.FreeBSD.org>` этого домена.

- <ftp://ftp.cn.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.cn.FreeBSD.org/pub/FreeBSD/> (ftp)

## Корея

В случае проблем пожалуйста свяжитесь с администратором `<hostmaster@kr.FreeBSD.org>` этого домена.

- <ftp://ftp.kr.FreeBSD.org/pub/FreeBSD/> (ftp / rsync)
- <ftp://ftp2.kr.FreeBSD.org/pub/FreeBSD/> (ftp)



## Литва

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@lt.FreeBSD.org> этого домена.

- <ftp://ftp.lt.FreeBSD.org/pub/FreeBSD/> (ftp)

## Нидерланды

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@nl.FreeBSD.org> этого домена.

- <ftp://ftp.nl.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp.nl.FreeBSD.org/os/FreeBSD/>) / rsync)
- <ftp://ftp2.nl.FreeBSD.org/pub/FreeBSD/> (ftp)

## Норвегия

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@no.FreeBSD.org> этого домена.

- <ftp://ftp.no.FreeBSD.org/pub/FreeBSD/> (ftp / rsync)
- <ftp://ftp3.no.FreeBSD.org/pub/FreeBSD/> (ftp)

## Польша

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@pl.FreeBSD.org> этого домена.

- <ftp://ftp.pl.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.pl.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp3.pl.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp5.pl.FreeBSD.org/pub/FreeBSD/> (ftp)

## Португалия

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@pt.FreeBSD.org> этого домена.

- <ftp://ftp.pt.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.pt.FreeBSD.org/pub/freebsd/> (ftp)
- <ftp://ftp4.pt.FreeBSD.org/pub/ISO/FreeBSD/> (ftp)

## Россия

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@ru.FreeBSD.org> этого домена.

- <ftp://ftp.ru.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.ru.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp2.ru.FreeBSD.org/pub/FreeBSD/>) / rsync)
- <ftp://ftp3.ru.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp4.ru.FreeBSD.org/pub/FreeBSD/> (ftp)

## Румыния

В случае проблем пожалуйста свяжитесь с администратором `<hostmaster@ro.FreeBSD.org>` этого домена.

- <ftp://ftp.ro.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp1.ro.FreeBSD.org/pub/FreeBSD/> (ftp / ftpv6)

## США

В случае проблем пожалуйста свяжитесь с администратором `<hostmaster@us.FreeBSD.org>` этого домена.

- <ftp://ftp1.us.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.us.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp3.us.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp4.us.FreeBSD.org/pub/FreeBSD/> (ftp / ftpv6)
- <ftp://ftp5.us.FreeBSD.org/pub/FreeBSD/> (ftp / rsync)
- <ftp://ftp6.us.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp6.us.FreeBSD.org/pub/FreeBSD/>))
- <ftp://ftp7.us.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp7.us.FreeBSD.org/pub/FreeBSD/>) / rsync)
- <ftp://ftp8.us.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp9.us.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp9.us.FreeBSD.org/pub/FreeBSD/>))
- <ftp://ftp10.us.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp11.us.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp12.us.FreeBSD.org/pub/FreeBSD/> (ftp / rsync)
- <ftp://ftp13.us.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp13.us.FreeBSD.org/pub/FreeBSD/>) / rsync)
- <ftp://ftp14.us.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp15.us.FreeBSD.org/pub/FreeBSD/> (ftp)

## Саудовская Аравия

В случае проблем пожалуйста свяжитесь с администратором `<ftpadmin@isu.net.sa>` этого домена.

- <ftp://ftp.isu.net.sa/pub/mirrors/ftp.freebsd.org/> (ftp)

## Сингапур

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@sg.FreeBSD.org> этого домена.

- <ftp://ftp.sg.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp.sg.FreeBSD.org/pub/FreeBSD/>) / rsync)

## Словацкая Республика

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@sk.FreeBSD.org> этого домена.

- <ftp://ftp.sk.FreeBSD.org/pub/FreeBSD/> (ftp)

## Словения

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@si.FreeBSD.org> этого домена.

- <ftp://ftp.si.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.si.FreeBSD.org/pub/FreeBSD/> (ftp)

## Тайвань

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@tw.FreeBSD.org> этого домена.

- <ftp://ftp.tw.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp2.tw.FreeBSD.org/pub/FreeBSD/>) / rsync)
- <ftp://ftp3.tw.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp4.tw.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp5.tw.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp6.tw.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp6.tw.FreeBSD.org/>))
- <ftp://ftp7.tw.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp8.tw.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp9.tw.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp10.tw.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp11.tw.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp11.tw.FreeBSD.org/FreeBSD/>))
- <ftp://ftp12.tw.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp13.tw.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp14.tw.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp15.tw.FreeBSD.org/pub/FreeBSD/> (ftp)

## Турция

- <ftp://ftp.tr.FreeBSD.org/pub/FreeBSD/> (ftp)

## Украина

- <ftp://ftp.ua.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.ua.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp3.ua.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp5.ua.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp6.ua.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp7.ua.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp8.ua.FreeBSD.org/pub/FreeBSD/> (ftp)

## Финляндия

В случае проблем пожалуйста свяжитесь с администратором `<hostmaster@fi.FreeBSD.org>` этого домена.

- <ftp://ftp.fi.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp3.fi.FreeBSD.org/pub/FreeBSD/> (ftp)

## Франция

В случае проблем пожалуйста свяжитесь с администратором `<hostmaster@fr.FreeBSD.org>` этого домена.

- <ftp://ftp.fr.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.fr.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp3.fr.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp5.fr.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp6.fr.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp8.fr.FreeBSD.org/pub/FreeBSD/> (ftp)

## Хорватия

В случае проблем пожалуйста свяжитесь с администратором `<hostmaster@hr.FreeBSD.org>` этого домена.

- <ftp://ftp.hr.FreeBSD.org/pub/FreeBSD/> (ftp)

## Чехия

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@cz.FreeBSD.org> этого домена.

- <ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp.cz.FreeBSD.org/pub/FreeBSD/>) / rsync)

## Швейцария

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@ch.FreeBSD.org> этого домена.

- <ftp://ftp.ch.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.ch.FreeBSD.org/pub/FreeBSD/> (ftp)

## Швеция

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@se.FreeBSD.org> этого домена.

- <ftp://ftp.se.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.se.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp3.se.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp5.se.FreeBSD.org/pub/FreeBSD/> (ftp / http (<http://ftp5.se.FreeBSD.org/>) / rsync)

## Эстония

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@ee.FreeBSD.org> этого домена.

- <ftp://ftp.ee.FreeBSD.org/pub/FreeBSD/> (ftp)

## Южная Африка

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@za.FreeBSD.org> этого домена.

- <ftp://ftp.za.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.za.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp3.za.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp4.za.FreeBSD.org/pub/FreeBSD/> (ftp)

## Япония

В случае проблем пожалуйста свяжитесь с администратором <hostmaster@jp.FreeBSD.org> этого домена.

- <ftp://ftp.jp.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp2.jp.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp3.jp.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp4.jp.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp5.jp.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp6.jp.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp7.jp.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp8.jp.FreeBSD.org/pub/FreeBSD/> (ftp)
- <ftp://ftp9.jp.FreeBSD.org/pub/FreeBSD/> (ftp)

## А.3. Анонимный CVS

### А.3.1. Введение

Анонимный CVS (известный также как, *anoncvs*) это возможность, предоставляемая утилитами CVS, поставляемыми с FreeBSD для синхронизации с удаленным CVS репозиторием. Помимо прочего, он позволяет пользователям FreeBSD без специальных привилегий (с правами только на чтение) выполнять CVS операции на одном из официальных *anoncvs* серверов проекта FreeBSD. Для использования этой возможности, просто установите переменную CVSROOT на соответствующий сервер *anoncvs*, введите общеизвестный пароль "anoncvs" в ответ на приглашение команды `cvs login`, а затем используйте `cvs(1)` для доступа к репозиторию как к обычному локальному CVS.

**Замечание:** Команда `cvs login` сохраняет пароли, использованные для авторизации на сервере CVS в файле с именем `.cvspass` в каталоге HOME. Если этот файл не существует, вы можете получить сообщение об ошибке при первом запуске `cvs login`. Просто создайте пустой файл `.cvspass` и попробуйте еще раз.

Хотя можно сказать, что сервисы CVSup и *anoncvs* выполняют в сущности похожие функции, есть несколько отличий, которые могут повлиять на выбор метода синхронизации. По своей сути, **CVSup** гораздо более эффективно использует сетевые ресурсы, и из двух этих программ он более интеллекттуален, но за его эффективность приходится платить. Во-первых, для того, чтобы появилась возможность хоть что-то получить этим методом, **CVSup** потребуется установить и настроить специальную программу, а во-вторых, этот **CVSup** позволяет синхронизацию только довольно больших частей исходных текстов, называемых *коллекциями*.

**Anoncvs** напротив может использоваться для получения как одного файла, так и всей программы (например `ls` или `grep`), с помощью ссылки на имя модуля CVS. Конечно, **anoncvs** хорошо подходит только при операциях на чтение CVS репозитория, поэтому если вы намереваетесь производить локальную разработку в одном из репозиториях проекта FreeBSD, вам подойдет только **CVSup**.

### А.3.2. Использование анонимного CVS

Настройка cvs(1) для использования анонимного CVS репозитория означает установку переменной окружения CVSROOT на один из *anoncvs* серверов проекта FreeBSD. На момент написания этого документа доступны следующие сервера:

- *Австрия*: `:pserver:anoncvs@anoncvs.at.FreeBSD.org:/home/ncvs` (Используйте `cvs login` и введите любой пароль в ответ на приглашение.)
- *Франция*: `:pserver:anoncvs@anoncvs.fr.FreeBSD.org:/home/ncvs` (`pserver` (пароль "anoncvs"), `ssh` (нет пароля))
- *Германия*: `:pserver:anoncvs@anoncvs.de.FreeBSD.org:/home/ncvs` (Используйте `cvs login` и введите пароль "anoncvs" в ответ на приглашение.)
- *Германия*: `:pserver:anoncvs@anoncvs2.de.FreeBSD.org:/home/ncvs` (`rsh`, `pserver`, `ssh`, `ssh/2022`)
- *Япония*: `:pserver:anoncvs@anoncvs.jp.FreeBSD.org:/home/ncvs` (Используйте `cvs login` и введите пароль "anoncvs" в ответ на приглашение.)
- *Швеция*: `freebsdanoncvs@anoncvs.se.FreeBSD.org:/home/ncvs` (только `ssh` - без пароля)
- *США*: `freebsdanoncvs@anoncvs.FreeBSD.org:/home/ncvs` (только `ssh` - без пароля)
- *США*: `anoncvs@anoncvs1.FreeBSD.org:/home/ncvs` (только `ssh` - без пароля)

Поскольку CVS позволяет выполнить "check out" практически любой версии исходных текстов FreeBSD, которые когда-либо существовали (или, в некоторых случаях, будут существовать), вам необходимо познакомиться с флагом cvs(1) для ревизий (`-r`) и с параметрами, которые могут быть использованы с этим флагом в репозитории проекта FreeBSD.

Есть два вида тегов, теги ревизий и теги ветвей. Теги ревизий отвечают за определенную ревизию. Их значение остается прежним изо дня в день. Теги ветвей сопоставляются последней ревизии определенной ветви разработки, в любой момент времени. Поскольку теги ветвей не сопоставлены какой-то определенной ревизии, завтра они могут означать совсем не то, что сегодня.

Разд. А.6 содержит теги ревизий, которые могут заинтересовать пользователей. Опять же, ни один из них не подходит для коллекции портов, поскольку в коллекции портов нет разнообразия ревизий.

Если вы указываете тег ветви, то обычно получаете последнюю версию файлов этой ветви разработки. Если вы хотите получить предыдущую версию, это можно сделать путем указания даты флагом `-D дата`. Обратитесь к странице справочника cvs(1) за более подробной информацией.

### А.3.3. Примеры

Хотя перед тем, как что-то делать, рекомендуется полностью прочесть страницу справочника cvs(1), вот несколько кратких примеров использования Anonymous CVS:

**Пример А-1. Извлечение кода из -CURRENT (ls(1)) с последующим удалением:**

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.FreeBSD.org:/home/ncvs
% cvs login
At the prompt, enter the password "anoncvs".
% cvs co ls
% cvs release -d ls
```

```
% cvs logout
```

#### Пример А-2. Использование SSH для извлечения дерева src/:

```
% cvs -d freebsdanoncvs@anoncvs.FreeBSD.org:/home/ncvs co src
The authenticity of host 'anoncvs.freebsd.org (128.46.156.46)' can't be established.
DSA key fingerprint is 52:02:38:1a:2f:a8:71:d3:f5:83:93:8d:aa:00:6f:65.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'anoncvs.freebsd.org' (DSA) to the list of known hosts.
```

#### Пример А-3. Извлечение ls(1) из ветви для 3.X-STABLE:

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.FreeBSD.org:/home/ncvs
% cvs login
At the prompt, enter the password "anoncvs".
% cvs co -rRELENG_3 ls
% cvs release -d ls
% cvs logout
```

#### Пример А-4. Создание списка изменений ls(1) (в виде unified diff)

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.FreeBSD.org:/home/ncvs
% cvs login
At the prompt, enter the password "anoncvs".
% cvs rdiff -u -rRELENG_3_0_0_RELEASE -rRELENG_3_4_0_RELEASE ls
% cvs logout
```

#### Пример А-5. Поиск доступных имен модулей:

```
% setenv CVSROOT :pserver:anoncvs@anoncvs.FreeBSD.org:/home/ncvs
% cvs login
At the prompt, enter the password "anoncvs".
% cvs co modules
% more modules/modules
% cvs release -d modules
% cvs logout
```

### А.3.4. Другие ресурсы

В изучении CVS могут помочь следующие дополнительные ресурсы:

- Учебник по CVS (<http://www.csc.calpoly.edu/~dbutler/tutorials/winter96/cvs/>) от Cal Poly.
- CVS Home (<http://www.cvshome.org/>), сообщество разработки и поддержки CVS.



- CVSweb (<http://www.FreeBSD.org/cgi/cvsweb.cgi>) это веб интерфейс к CVS проекта FreeBSD.

## А.4. Использование СТМ

**СТМ** это метод синхронизации удаленного дерева исходных текстов с центральным. Он был разработан для использования с деревом исходных текстов FreeBSD, хотя может быть полезен и для других целей. На данный момент существует очень немного (если она вообще существует) документации по процессу создания дельта-файлов (deltas, разница между имеющимися и актуальными исходными текстами), поэтому обратитесь в список рассылки `ctm-users` (<http://lists.FreeBSD.org/mailman/listinfo/ctm-users>) за дополнительной информацией, если захотите использовать **СТМ** для других целей.

### А.4.1. Зачем мне может понадобиться СТМ?

**СТМ** создаст для вас локальную копию исходных текстов FreeBSD. Существует множество разновидностей дерева исходных текстов. Захотите ли вы поддерживать все дерево CVS или только одну из ветвей, **СТМ** может предоставить вам все необходимое. Если вы разработчик FreeBSD, но ваше соединение по TCP/IP некачественное или отсутствует, или же вы просто хотите, чтобы изменения автоматически отправлялись вам, **СТМ** предназначен для вас. Вам потребуется получать до трех изменений в день для наиболее активных ветвей. Они будут отправляться вам по электронной почте. Размеры обновлений всегда поддерживаются настолько малыми, насколько это возможно. Они обычно меньше 5K, хотя случается (раз из десяти) достигают 10-50K и время от времени даже 100K или больше.

Вам также потребуется узнать о различных особенностях работы непосредственно с исходными текстами в разработке, вместо готовых релизов. Это в частности относится к выбору исходных текстов ветви `-CURRENT`. Рекомендуется прочесть материал На переднем крае разработок.

### А.4.2. Что потребуется для использования СТМ?

Вам потребуются две вещи: программа **СТМ** и исходные тексты, которые требуется обновить (до актуального состояния).

Программа **СТМ** была частью FreeBSD все время с момента выпуска релиза 2.0, она находится в каталоге `/usr/src/usr.sbin/ctm`, если у вас есть копия исходных текстов.

Если вы работаете с версией FreeBSD до 2.0, можете получить актуальные исходные тексты **СТМ** непосредственно с:

<http://www.FreeBSD.org/cgi/cvsweb.cgi/src/usr.sbin/ctm/>

”Дельта-файлы” **СТМ** могут приходить двумя путями, через FTP или email. Если у вас есть доступ к интернет по FTP, доступ к **СТМ** может быть получен через следующие FTP сайты:

<ftp://ftp.FreeBSD.org/pub/FreeBSD/CTM/>

или с сайтов из списка зеркал.

Зайдите по FTP в соответствующий каталог и прочтите файл README.

Если вы предпочитаете получать дельта-файлы по почте:

Подпишитесь на один из списков распространения **CTM**. `ctm-cvs-cur` (<http://lists.FreeBSD.org/mailman/listinfo/ctm-cvs-cur>) поддерживает все дерево CVS. `ctm-src-cur` (<http://lists.FreeBSD.org/mailman/listinfo/ctm-src-cur>) поддерживает главную ветвь разработки. `ctm-src-4` (<http://lists.FreeBSD.org/mailman/listinfo/ctm-src-4>) поддерживает ветвь 4.X, и т.д. Если вы не знаете, как подписаться на список, нажмите на ссылку выше или на ссылку <http://lists.FreeBSD.org/mailman/listinfo>, а затем на ссылку соответствующего списка. Страница списка должна содержать все необходимые инструкции по подписке.

Когда вы начнете получать обновления **CTM** по почте, используйте программу `ctm_rmail` для распаковки и применения обновлений. Вы можете использовать программу `ctm_rmail` непосредственно из `/etc/aliases`, если хотите полностью автоматизировать процесс. Прочтите страницу справочника `ctm_rmail` с более подробной информацией.

**Замечание:** Вне зависимости от способа получения дельта-файлов **CTM**, вам потребуется подписка на список рассылки `ctm-announce` (<http://lists.FreeBSD.org/mailman/listinfo/ctm-announce>). В будущем этот список станет единственным местом, где будут анонсироваться операции, относящиеся к системе **CTM**. Нажмите на ссылку выше и следуйте инструкциям для подписки на эту рассылку.

### А.4.3. Использование CTM в первый раз

Перед тем, как вы сможете начать использование дельта-файлов **CTM**, потребуется определить исходную точку для последующего их применения.

Сначала потребуется определить, что уже имеется. Каждый может начать с "пустого" (`empty`) каталога. Вы должны использовать дельта-файл `Empty` для создания поддерживаемого **CTM** дерева. Планируется распространение "стартовых" дельта-файлов на CD, но на данный момент это не делается.

Поскольку дерево исходных текстов может быть объемом в десятки мегабайт, предпочтительно начать с каких-то уже имеющихся текстов. Если у вас есть `-RELEASE CD`, распакуйте исходные тексты с него. Это снизит объем передаваемых данных.

Вы можете распознать "стартовый" дельта-файл по символу `x`, добавляемому к номеру (например, `src-cur.3210XEmpty.gz`). Обозначение перед `x` соответствует имеющимся исходным текстам. `Empty` это пустой каталог. Как правило файл с `Empty` создается через каждые 100 дельта-файлов. Между прочем, эти файлы большие! От 70 до ;) мегабайт упакованных `gzip` данных это обычный размер для `XEmpty`.

Как только вы получили основной дельта-файл, потребуются также все дельта-файлы с последующими номерами.

### А.4.4. Повседневное использование CTM

Для применения дельта-файлов, просто выполните:

```
# cd /where/ever/you/want/the/stuff
# ctm -v -v /where/you/store/your/deltas/src-xxx.*
```

**СТМ** работает с дельта-файлами, упакованными с помощью `gzip`, поэтому вам не требуется их распаковывать.

Хотя весь процесс довольно безопасен, **СТМ** не будет изменять дерево исходных текстов. Для проверки дельта-файла вы также можете использовать флаг `-c`, **СТМ** только проверит целостность дельта-файла а также его применимость к существующим исходным текстам.

Это все. Каждый раз после получения дельта-файла, запускайте **СТМ** для поддержания исходных текстов в актуальном состоянии.

Не удаляйте дельта-файлы, если их сложно загрузить еще раз. Вы возможно захотите сохранить их на всякий случай. Если вы можете сохранить их только на диске, используйте для создания копии `fdwrite`.

#### А.4.5. Сохранение локальных изменений

Как разработчик вы возможно захотите поэкспериментировать и изменить файлы в дереве исходных текстов. **СТМ** имеет ограниченную поддержку локальных изменений: перед проверкой наличия файла `foo`, сначала проверяется `foo.ctm`. Если он присутствует, **СТМ** будет работать с ним вместо `foo`.

Такое поведение обеспечивает простой путь поддержки локальных изменений: просто скопируйте файлы, которые вы хотите изменить, в файлы с соответствующими именами и суффиксом `.ctm`. Вы можете свободно изменять код, а **СТМ** будет поддерживать файлы `.ctm` в актуальном состоянии.

#### А.4.6. Другие интересные возможности СТМ

##### А.4.6.1. Определение файлов, которые будут затронуты обновлением

Вы можете определить список изменений, которые **СТМ** внесет в исходные тексты, используя параметр **СТМ** `-l`.

Это полезно, если вы хотите поддерживать лог изменений, выполнить предварительную или последующую обработку изменяемых файлов, или если вы просто немного параноидальны.

##### А.4.6.2. Создание резервных копий перед обновлением

Иногда вам может понадобиться сделать резервные копии всех файлов, которые будут изменены **СТМ**.

С параметром `-b backup-file` **СТМ** выполнит резервное копирование всех изменяемых дельта-файлом **СТМ** файлов в `backup-file`.

##### А.4.6.3. Ограничение обновлений для определенных файлов

Иногда необходимо ограничить набор файлов, обновляемых **СТМ**, или даже задать обновление нескольких определенных файлов.

Вы можете управлять списком файлов, с которыми будет работать **СТМ**, путем указания фильтрующих регулярных выражений с параметрами `-e` и `-x`.

Например, для извлечения свежей копии `lib/libc/Makefile` из коллекции сохраненных дельта-файлов **СТМ**, выполните команду:

```
# cd /where/ever/you/want/to/extract/it/
# ctm -e '^lib/libc/Makefile' ~ctm/src-xxx.*
```

Для каждого указанного **СТМ** дельта-файла, параметры `-e` и `-x` применяются в порядке их задания в командной строке. Файл обрабатывается **СТМ** только если он помечается как подходящий после обработки всех параметров `-e` и `-x`.

#### А.4.7. Дальнейшие планы для СТМ

Масса планов:

- Использовать какой-то из видов аутентификации в системе **СТМ**, позволяющий обнаружение поддельных дат **СТМ**.
- Доработать параметры **СТМ**, поскольку они могут ввести в заблуждение.

#### А.4.8. Разное

Существует набор дельта-файлов и для коллекции `ports`, но интерес к нему пока не так высок.

#### А.4.9. Зеркала СТМ

СТМ/FreeBSD доступен через анонимный FTP с нижеприведенных зеркал. Если вы выбрали получение **СТМ** через анонимный FTP, используйте ближайший к вам сервер.

В случае возникновения проблем обратитесь в список рассылки `ctm-users` (<http://lists.FreeBSD.org/mailman/listinfo/ctm-users>).

California, Bay Area, официальный источник

- <ftp://ftp.FreeBSD.org/pub/FreeBSD/development/CTM/>

South Africa, резервный сервер для старых дельта-файлов

- <ftp://ftp.za.FreeBSD.org/pub/FreeBSD/CTM/>

Taiwan/R.O.C.

- <ftp://ctm.tw.FreeBSD.org/pub/FreeBSD/development/CTM/>
- <ftp://ctm2.tw.FreeBSD.org/pub/FreeBSD/development/CTM/>
- <ftp://ctm3.tw.FreeBSD.org/pub/FreeBSD/development/CTM/>

Если вы не нашли ближайшего к вам зеркала, или зеркало неполно, попробуйте воспользоваться поиском, например alltheweb (<http://www.alltheweb.com/>).

## A.5. Использование CVSup

### A.5.1. Введение

**CVSup** это пакет программного обеспечения для распространения и обновления исходных текстов с основного репозитория CVS на удаленном сервере. Исходные тексты FreeBSD поддерживаются в репозитории CVS на центральной машине разработки в Калифорнии. С помощью **CVSup** пользователи FreeBSD легко могут поддерживать собственные исходные тексты в актуальном состоянии.

**CVSup** использует так называемую *опрашивающую* (pull) модель обновления. Работая по этой модели, каждый клиент запрашивает обновления с сервера, если и когда они нужны. Сервер пассивно ожидает запросы на обновление от своих клиентов. Таким образом, все обновления инициируются клиентами. Сервер никогда не высылает не запрошенные обновления. Пользователь для получения обновлений должен либо запустить **CVSup** клиента вручную, либо добавить задание в cron для настройки периодических обновлений в автоматическом режиме.

Термин **CVSup**, написанный с заглавными буквами, означает весь пакет программного обеспечения. Его основные компоненты это клиент cvsup, запускаемый на каждом пользовательском компьютере, и сервер cvsupd, работающий на каждом зеркале FreeBSD.

При чтении документации FreeBSD и списков рассылки вы могли встретить указания на **sup**. **Sup** был предшественником **CVSup**, и выполнял похожие функции. **CVSup** в основном используется тем же способом, что и sup, и, фактически, использует файлы настройки, обратно совместимые с файлами sup. **Sup** более не используется проектом FreeBSD, поскольку **CVSup** и быстрее и более гибок.

### A.5.2. Установка

Простейший способ установки **CVSup** это использование прекомпилированного пакета net/cvsup из коллекции пакетов FreeBSD. Если вы предпочтете собрать **CVSup** из исходных текстов, можете использовать вместо этого порт net/cvsup. Но имейте ввиду: порт net/cvsup зависит от системы Modula-3, которой потребуется существенный объем времени и пространства на диске для загрузки и установки.

**Замечание:** Если вы собираетесь использовать **CVSup** на компьютере, где нет установленного **XFree86** или **Xorg**, например на сервере, используйте порт, не включающий **CVSup** GUI, net/cvsup-without-gui.

### A.5.3. Настройка CVSup

Работа **CVSup** контролируется файлом настройки, называемым supfile. В каталоге /usr/share/examples/cvsup/ находится несколько примеров supfile.

Информация в `supfile` отвечает на следующие вопросы **CVSup**:

- Какие файлы вы хотите получить?
- Какие их версии вам нужны?
- Откуда вы хотите их получить?
- Где вы хотите разместить их на своем компьютере?
- Где вы хотите разместить файлы статуса?

В следующих разделах мы составим типичный `supfile` последовательным ответом на каждый из этих вопросов. Сначала опишем общую структуру `supfile`.

`supfile` это текстовый файл. Комментарии, начинающиеся с `#`, продолжаются до конца строки. Пустые строки и строки, содержащие только комментарии, игнорируются.

Каждая из оставшихся строк описывает набор файлов, получаемых пользователем. Строка начинается с имени "коллекции", логического объединения файлов, определяемых сервером. Имя коллекции говорит серверу о том, какие файлы вам нужны. После имени коллекции следуют одно или больше полей, разделенных пробелом. Эти поля отвечают на вопросы, заданные выше. Есть два типа полей: флаги и значения. Поле флага состоит из одного ключевого слова, например `delete` или `compress`. Поле значения также начинается с ключевого слова, но за ним без пробела следует = и второе слово. Например, `release=cvs` это поле значения.

`supfile` обычно задает получение более одной коллекции. Одним из способов построения `supfile` является указание всех соответствующих полей для каждой коллекции явно. Однако, это приводит к появлению слишком длинных строк в `supfile` и это неудобно, поскольку большинство полей одинаковы для всех коллекций `supfile`. **CVSup** предоставляет механизм задания значений по умолчанию для устранения этих проблем. Строки, начинающиеся со специального имени псевдо-коллекции `*default`, могут быть использованы для установки флагов и значений, которые послужат значениями по умолчанию для всех последующих коллекций `supfile`. Значение по умолчанию может быть перезаписано для каждой отдельной коллекции путем указания другого значения в ее собственных параметрах. Значения по умолчанию также могут быть изменены или расширены в любом месте `supfile` дополнительными строками `*default`.

Получив эту информацию, мы начнем строить `supfile` для получения и обновления главного дерева исходных текстов FreeBSD-CURRENT.

- Какие файлы вы хотите получить?

Файлы, доступные через **CVSup**, организованы в именованные группы, называемые "коллекциями". Доступные коллекции описаны в следующем разделе. В этом примере мы получим все дерево исходных текстов системы FreeBSD. Существует одиночная большая коллекция `src-all`, которая позволит нам сделать это. В качестве первого шага по созданию `supfile`, добавим список коллекций, по одной на строку (в нашем случае, только одну строку):

```
src-all
```

- Какие их версии вам нужны?

С **CVSup** вы в принципе можете получить любую версию исходных текстов, которая когда-либо существовала. Это возможно, поскольку **cvsupd** сервер работает непосредственно с репозиторием

CVS, который содержит все версии. Вы указываете ту, которая вам нужна, с помощью полей `tag=` и `date=`.

**Внимание:** Будьте очень осторожны при задании любых полей `tag=`. Некоторые теги существуют только в определенных коллекциях файлов. Если вы укажете некорректный тег или ошибетесь в его написании, **CVSup** удалит файлы, которые вы возможно не хотели удалять. В частности, используйте *только* `tag=` для коллекций `ports-*`.

Поля `tag=` означают символьные имена в репозитории. Существует два вида тегов, теги ревизий и теги ветвей. Теги ревизий означают определенную ревизию. Они не меняются со временем. Теги ветвей, с другой стороны, означают последнюю ревизию заданной линии разработки в любой момент времени. Поскольку тег ветви не относится к определенной ревизии, он может означать завтра что-то иное чем сегодня.

Разд. А.6 содержит теги ветвей, которые могут быть интересны пользователям. Тег, указанный в файле настройки **CVSup**, должен предваряться строкой `tag=` (`RELENG_4` превратится в `tag=RELENG_4`). Помните, что для коллекции портов подходит только `tag=.`

**Внимание:** Будьте очень осторожны при наборе имен тегов. **CVSup** не может отличить правильные и неправильные теги. Если вы неправильно наберете тег, **CVSup** поведет себя так, как если бы вы указали тег, не содержащий файлов. В этом случае он удалит существующие исходные тексты.

При указании тега ветви вы получите последние версии файлов этой ветви разработки. Если вы хотите получить какую-то из предыдущих версий, сделайте это указав дату в поле значения `date=`. Страница справочника `cvsup(1)` описывает как сделать это.

В нашем примере мы хотим получить FreeBSD-CURRENT. Мы добавим эту строку в начало `supfile`:

```
*default tag=.
```

Существует важный специальный случай, когда не задаются ни поле `tag=`, ни поле `date=`. В этом случае вы получите последние версии RCS файлов непосредственно из CVS репозитория сервера вместо получения определенной версии. Разработчики обычно предпочитают этот способ.

Поддерживая копию репозитория в своей системе, они получают возможность просмотра истории ревизий и проверки последних версий файлов. Однако это достигается ценой большего занимаемого дискового пространства.

- Откуда вы хотите их получить?

Мы используем поле `host=` для сообщения `cvsup` откуда забирать обновления. Подойдет любое из CVSup зеркал, хотя вы должны попробовать выбрать ближайшее к вам зеркало. В этом примере мы используем не существующий сервер, `cvsup99.FreeBSD.org`:

```
*default host=cvsup99.FreeBSD.org
```

Вам потребуется подставить адрес одного из существующих хостов перед запуском **CVSup**. При каждом запуске `cvsup`, вы можете перезаписать настройку хоста из командной строки, используя параметр `-h hostname`.

- Где вы хотите разместить их на своем компьютере?

Поле `prefix=` указывает `cvsup` где размещать получаемые файлы. В этом примере, мы поместим файлы непосредственно в главное дерево исходных текстов, `/usr/src`. Каталог `src` уже подразумевается в коллекциях, которые мы собираемся получить, поэтому корректное указание таково:

```
*default prefix=/usr
```

- Где вы хотите разместить файлы статуса `cvsup`?

Клиент **CVSup** поддерживает определенные файлы статуса в так называемом "base" каталоге. Эти файлы помогают **CVSup** работать более эффективно путем поддержки истории уже полученных обновлений. Мы будем использовать стандартный каталог `base`, `/var/db`:

```
*default base=/var/db
```

Если `base` каталог не существует, теперь подходящий момент для его создания. Клиент `cvsup` не запустится, если `base` каталог не существует.

- Различные настройки `supfile`:

Существует еще одна строка, которая обычно должна присутствовать в `supfile`:

```
*default release=cvs delete use-rel-suffix compress
```

`release=cvs` означает, что сервер должен получать информацию из главного репозитория FreeBSD CVS. Это обычно всегда так, но существуют несколько иных вариантов, изложение которых выходит за пределы этой главы.

`delete` дает **CVSup** возможность удалять файлы. Вы должны всегда указывать этот параметр, чтобы **CVSup** мог поддерживать дерево исходных текстов полностью актуальным. **CVSup** удалит только те файлы, за которые отвечает. Все другие файлы останутся нетронутыми.

`use-rel-suffix` это ... мистика. Если вы действительно хотите узнать о нем, обратитесь к странице справочника `cvsup(1)`. Иначе просто укажите это поле и не беспокойтесь о нем больше.

`compress` включает использование `gzip` сжатия при передаче данных. Если ваше сетевое подключение работает со скоростью T1 или быстрее, вам возможно не потребуется использование сжатия. Иначе оно обычно помогает.

- Все это вместе:

Вот полный `supfile` для нашего примера:

```
*default tag=.
*default host=cvsup99.FreeBSD.org
*default prefix=/usr
*default base=/var/db
*default release=cvs delete use-rel-suffix compress

src-all
```



### А.5.3.1. Файл `refuse`

Как упомянуто выше, **CVSup** использует *метод опроса*. В основном это означает, что вы подключаетесь к **CVSup** серверу, и он говорит "Вот то, что вы можете загрузить с меня...", и клиент отвечает "ОК, я возьму это, это и это." С настройкой по умолчанию **CVSup** клиент заберет каждый файл, связанный с коллекцией и тегом, выбранным в файле настройки. Однако, это не всегда то, что вам нужно, особенно если вы синхронизируете деревья `doc`, `ports`, или `www` — большинство людей не могут читать на пяти или четырех языках, и следовательно им не требуются локализованные файлы. Если вы обновляете коллекцию портов, это можно обойти путем индивидуального указания каждой коллекции (например, *ports-astrology*, *ports-biology* и т.д. вместо коллекции *ports-all*). Однако, поскольку для деревьев `doc` и `www` нет специфичных для языка коллекций, используйте одну из замечательных возможностей **CVSup**: файл `refuse`.

Файл `refuse` говорит **CVSup**, что он не должен забирать каждый файл из коллекции; другими словами, он говорит клиенту *отказаться* (`refuse`) от получения с сервера определенных файлов. Файл `refuse` можно найти (или, если у вас его еще нет, должен быть помещен) в `base/sup/`. `base` определен в файле `supfile`; наше стандартное расположение `base` это `/var/db`, что означает, что по умолчанию файл `refuse` это `/var/db/sup/refuse`.

Формат файла `refuse` очень прост; он содержит имена файлов или каталогов, которые вы не хотите загружать. Например, если вы не можете разговаривать ни на каком языке кроме английского и русского, и вы не будете читать документацию на этих языках, поместите следующие строки в файл `refuse`:

```
doc/da_*
doc/de_*
doc/el_*
doc/es_*
doc/fr_*
doc/it_*
doc/ja_*
doc/nl_*
doc/no_*
doc/pl_*
doc/pt_*
doc/sr_*
doc/zh_*
```

и так далее для других языков (полный список находится на странице FreeBSD CVS репозитория (<http://www.FreeBSD.org/cgi/cvsweb.cgi/>)).

С этой очень полезной возможностью пользователи с низкоскоростным подключением или с поминутной платой за подключение смогут сохранить время, поскольку им более не потребуется загружать файлы, которые они никогда не будут использовать. За более подробной информацией о файлах `refuse` и замечательных возможностях **CVSup**, обратитесь к его справочной странице.

### А.5.4. Запуск **CVSup**

Теперь вы готовы к тестированию обновления. Командная строка для этого очень проста:

```
# cvsup supfile
```

где *supfile* это конечно имя *supfile*, который только что был создан. При запуске под X11, *cvsup* отобразит GUI интерфейс с несколькими полезными кнопками. Нажмите кнопку **go** и смотрите за его работой.

Поскольку в этом примере обновляется существующее дерево */usr/src* вам потребуется, запуск программы из под *root*, чтобы у *cvsup* были права, необходимые для обновления файлов. Если файл настройки только что создан, и эта программа раньше никогда не использовалась, это может действовать вам на нервы. Есть простой способ для пробного запуска без затрагивания ваших драгоценных файлов. Просто создайте где-нибудь пустой каталог и поместите его в качестве дополнительного аргумента командной строки:

```
# mkdir /var/tmp/dest
# cvsup supfile /var/tmp/dest
```

Указанный каталог будет использоваться в качестве места назначения всех обновлений. **CVSup** будет работать с файлами из */usr/src*, но не станет изменять или удалять их. Вместо этого все обновления файлов будут помещены в */var/tmp/dest/usr/src*. При запуске таким способом **CVSup** оставит также неприкосновенным каталог *base*. Новые версии этих файлов будут записаны в указанный каталог. Если у вас есть права на чтение каталога */usr/src*, вам даже не потребуется работать под *root* для выполнения пробного обновления.

Если вы не работаете с X11 или просто не любите GUI, добавьте набор параметров командной строки при запуске *cvsup*:

```
# cvsup -g -L 2 supfile
```

Параметр *-g* указывает **CVSup** не использовать GUI. Он действует автоматически, если вы не работаете под X11, но иначе вам потребуется его указать.

Параметр *-L 2* указывает **CVSup** выводить информацию о каждом производимом обновлении. Есть три уровня протоколирования, от *-L 0* до *-L 2*. Уровень по умолчанию 0, что означает полное отсутствие сообщений, за исключением сообщений об ошибках.

Доступно множество других параметров. Для получения их краткого списка, наберите *cvsup -h*. За более подробным описанием обратитесь к странице справочника.

После проверки работоспособности обновлений вы можете организовать регулярные запуски **CVSup** с помощью *cron*(8). Очевидно, вы не должны позволять **CVSup** использовать GUI при запуске из *cron*(8).

## A.5.5. Коллекции файлов CVSup

Коллекции файлов, доступные через **CVSup**, организованы иерархически. Существует несколько больших коллекций, и они разделены на несколько меньших под-коллекций. Получение большой коллекции эквивалентно получению каждой из ее подколлекции. Иерархические отношения между коллекциями отражаются путем использования отступов в списке ниже.

Наиболее используемые коллекции это *src-all*, и *ports-all*. Другие коллекции используются небольшими группами людей для специальных целей и некоторые сайты зеркал могут не содержать их все.

```
cvs-all release=cvs
```

Главный FreeBSD CVS репозиторий, включающий криптографический код.

```
distrib release=cvs
```

Файлы, относящиеся к распространению и зеркалированию FreeBSD.

```
doc-all release=cvs
```

Исходные тексты Руководства FreeBSD и другой документации. Они не включают файлы веб-сайта FreeBSD.

```
ports-all release=cvs
```

Коллекция портов FreeBSD.

**Важно:** Если вы не хотите обновлять всю коллекцию `ports-all` (все дерево портов), а только одну из подколлекций, перечисленных ниже, убедитесь, что вы *каждый раз* обновляете подколлекцию `ports-base`! При любых изменениях в инфраструктуре сборки портов подколлекции `ports-base`, они могут быть использованы "настоящими" портами довольно скоро. Таким образом, если вы обновляете только "настоящие" порты и они используют некоторые новые возможности, есть большой шанс того, что их сборка прервется с непонятным сообщением об ошибке. *Самое первое*, что вы должны сделать, это убедиться, что подколлекция `ports-base` обновлена.

```
ports-archivers release=cvs
```

Архиваторы.

```
ports-astro release=cvs
```

Порты для астрономии.

```
ports-audio release=cvs
```

Поддержка звука.

```
ports-base release=cvs
```

Инфраструктура сборки портов - различные файлы, расположенные в подкаталогах `Mk/` и `Tools/` каталога `/usr/ports`.

**Замечание:** Пожалуйста, прочтите важное предупреждение выше: вы должны *всегда* обновлять эту подколлекцию, при каждом обновлении любой части коллекции портов FreeBSD!

```
ports-benchmarks release=cvs
```

Измерение производительности.

```
ports-biology release=cvs
```

Программы для биологии.

ports-cad release=cvcs

Инструменты САПР.

ports-chinese release=cvcs

Поддержка китайского языка.

ports-comms release=cvcs

Коммуникационные программы.

ports-converters release=cvcs

Преобразование кодировок.

ports-databases release=cvcs

Базы данных.

ports-deskutils release=cvcs

Вещи, использовавшиеся на рабочем столе до изобретения компьютеров.

ports-devel release=cvcs

Утилиты разработки.

ports-dns release=cvcs

Программы, имеющие отношение к DNS.

ports-editors release=cvcs

Редакторы.

ports-emulators release=cvcs

Эмуляторы других операционных систем.

ports-finance release=cvcs

Финансовые, расчетные и связанные с ними приложения.

ports-ftp release=cvcs

Клиентские и серверные утилиты FTP.

ports-games release=cvcs

Игры.

ports-german release=cvcs

Поддержка немецкого языка.

ports-graphics release=cvcs

Графические утилиты.

ports-hungarian release=cvs

Поддержка венгерского языка.

ports-irc release=cvs

IRC утилиты.

ports-japanese release=cvs

Поддержка японского языка.

ports-java release=cvs

Java утилиты.

ports-korean release=cvs

Поддержка корейского языка.

ports-lang release=cvs

Языки программирования.

ports-mail release=cvs

Почтовое программное обеспечение.

ports-math release=cvs

Математические программы.

ports-mbone release=cvs

Приложения MBone.

ports-misc release=cvs

Различные утилиты.

ports-multimedia release=cvs

Мультимедийное программное обеспечение.

ports-net release=cvs

Сетевое программное обеспечение.

ports-news release=cvs

Программное обеспечение новостей USENET.

ports-palm release=cvs

Программная поддержка для Palm™.

ports-polish release=cvs

Поддержка польского языка.

ports-portuguese release=cvs

Поддержка португальского языка.

ports-print release=cvs

Программы печати.

ports-russian release=cvs

Поддержка русского языка.

ports-security release=cvs

Утилиты безопасности.

ports-shells release=cvs

Оболочки командной строки.

ports-sysutils release=cvs

Системные утилиты.

ports-textproc release=cvs

Утилиты обработки текста (не включают настольные утилиты публикации).

ports-vietnamese release=cvs

Поддержка вьетнамского языка.

ports-www release=cvs

Программы, относящиеся к World Wide Web.

ports-x11 release=cvs

Порты с поддержкой X window system.

ports-x11-clocks release=cvs

Часы X11.

ports-x11-fm release=cvs

Файловые менеджеры X11.

ports-x11-fonts release=cvs

Шрифты и шрифтовые утилиты X11.

ports-x11-toolkits release=cvs

Пакеты разработки приложений для X11.

ports-x11-servers

Серверы X11.

ports-x11-wm

Оконные менеджеры X11.

src-all release=cvs

Основные исходные тексты FreeBSD, включая криптографический код.

src-base release=cvs

Различные файлы непосредственно из /usr/src.

src-bin release=cvs

Утилиты, которые могут потребоваться в однопользовательском режиме (/usr/src/bin).

src-contrib release=cvs

Утилиты и библиотеки, заимствованные проектом FreeBSD, используются почти без модификаций (/usr/src/contrib).

src-crypto release=cvs

Криптографические утилиты и библиотеки, заимствованные проектом FreeBSD, используются почти без модификаций (/usr/src/crypto).

src-eBones release=cvs

Kerberos и DES (/usr/src/eBones). Не используются в текущих релизах FreeBSD.

src-etc release=cvs

Файлы настройки системы (/usr/src/etc).

src-games release=cvs

Игры (/usr/src/games).

src-gnu release=cvs

Утилиты, попадающие под GNU Public License (/usr/src/gnu).

src-include release=cvs

Файлы заголовков (/usr/src/include).

src-kerberos5 release=cvs

Пакет безопасности Kerberos5 (/usr/src/kerberos5).

src-kerberosIV release=cvs

Пакет безопасности KerberosIV (/usr/src/kerberosIV).

src-lib release=cvs

Библиотеки (/usr/src/lib).

```
src-libexec release=cvs
```

Системные программы, обычно выполняемые другими программами (/usr/src/libexec).

```
src-release release=cvs
```

Файлы, необходимые для производства релизов FreeBSD (/usr/src/release).

```
src-sbin release=cvs
```

Системные утилиты для однопользовательского режима (/usr/src/sbin).

```
src-secure release=cvs
```

Криптографические библиотеки и команды (/usr/src/secure).

```
src-share release=cvs
```

Файлы, которые могут быть использованы несколькими системами (/usr/src/share).

```
src-sys release=cvs
```

Ядро (/usr/src/sys).

```
src-sys-crypto release=cvs
```

Криптографический код ядра (/usr/src/sys/crypto).

```
src-tools release=cvs
```

Различные инструменты для поддержки FreeBSD (/usr/src/tools).

```
src-usrbin release=cvs
```

Пользовательские утилиты (/usr/src/usr.bin).

```
src-usrsbin release=cvs
```

Системные утилиты (/usr/src/usr.sbin).

```
www release=cvs
```

Исходные тексты www сайта FreeBSD.

```
distrib release=self
```

Собственные файлы настройки сервера **CVSup**. Используются зеркалами **CVSup**.

```
gnats release=current
```

База данных отслеживания ошибок GNATS.

```
mail-archive release=current
```

Архив списков рассылки FreeBSD.



www release=current

Файлы предобработки WWW сайта FreeBSD (не исходные файлы). Используются зеркалами WWW.

### A.5.6. Дальнейшая информация

**CVSup** FAQ и другая информация о **CVSup** находится на Домашней странице CVSup (<http://www.polstra.com/projects/freeware/CVSup/>).

Большая часть связанных с FreeBSD обсуждений **CVSup** проводилась в списке рассылки `freebsd-hackers` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers>). Новые версии программного обеспечения анонсируются здесь и в списке рассылки `freebsd-announce` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce>).

Вопросы и сообщения об ошибках адресуйте автору программы `<cvsup-bugs@polstra.com>`.

### A.5.7. Зеркала CVSup

CVSup серверы для FreeBSD работают на следующих сайтах:

Центральные серверы, Основные зеркала, Costa Rica, Indonesia, Kyrgyzstan, Thailand, Австралия, Австрия, Аргентина, Бразилия, Великобритания, Венгрия, Германия, Греция, Дания, Ирландия, Исландия, Испания, Италия, Канада, Китай, Корея, Кувейт, Латвия, Литва, Нидерланды, Новая Зеландия, Норвегия, Польша, Португалия, Россия, Румыния, США, Сан Марино, Сингапур, Словацкая Республика, Словения, Тайвань, Турция, Украина, Филиппины, Финляндия, Франция, Чехия, Швейцария, Швеция, Эстония, Южная Африка, Япония.

(as of 2004/10/26 16:33:28 UTC)

Центральные серверы

- `cvsup.FreeBSD.org`

Основные зеркала

- `cvsup1.FreeBSD.org`
- `cvsup2.FreeBSD.org`
- `cvsup3.FreeBSD.org`
- `cvsup4.FreeBSD.org`
- `cvsup5.FreeBSD.org`
- `cvsup6.FreeBSD.org`
- `cvsup7.FreeBSD.org`
- `cvsup8.FreeBSD.org`

- [cvsup9.FreeBSD.org](http://cvsup9.FreeBSD.org)
- [cvsup10.FreeBSD.org](http://cvsup10.FreeBSD.org)
- [cvsup11.FreeBSD.org](http://cvsup11.FreeBSD.org)
- [cvsup12.FreeBSD.org](http://cvsup12.FreeBSD.org)
- [cvsup13.FreeBSD.org](http://cvsup13.FreeBSD.org)
- [cvsup14.FreeBSD.org](http://cvsup14.FreeBSD.org)
- [cvsup15.FreeBSD.org](http://cvsup15.FreeBSD.org)
- [cvsup16.FreeBSD.org](http://cvsup16.FreeBSD.org)
- [cvsup18.FreeBSD.org](http://cvsup18.FreeBSD.org)

#### Costa Rica

- [cvsup1.cr.FreeBSD.org](http://cvsup1.cr.FreeBSD.org)

#### Indonesia

- [cvsup.id.FreeBSD.org](http://cvsup.id.FreeBSD.org)

#### Kyrgyzstan

- [cvsup.kg.FreeBSD.org](http://cvsup.kg.FreeBSD.org)

#### Thailand

- [cvsup.th.FreeBSD.org](http://cvsup.th.FreeBSD.org)

#### Австралия

- [cvsup.au.FreeBSD.org](http://cvsup.au.FreeBSD.org)
- [cvsup2.au.FreeBSD.org](http://cvsup2.au.FreeBSD.org)
- [cvsup3.au.FreeBSD.org](http://cvsup3.au.FreeBSD.org)
- [cvsup4.au.FreeBSD.org](http://cvsup4.au.FreeBSD.org)
- [cvsup5.au.FreeBSD.org](http://cvsup5.au.FreeBSD.org)

- [cvsup6.au.FreeBSD.org](http://cvsup6.au.FreeBSD.org)
- [cvsup7.au.FreeBSD.org](http://cvsup7.au.FreeBSD.org)

#### Австрия

- [cvsup.at.FreeBSD.org](http://cvsup.at.FreeBSD.org)
- [cvsup2.at.FreeBSD.org](http://cvsup2.at.FreeBSD.org)

#### Аргентина

- [cvsup.ar.FreeBSD.org](http://cvsup.ar.FreeBSD.org)

#### Бразилия

- [cvsup.br.FreeBSD.org](http://cvsup.br.FreeBSD.org)
- [cvsup2.br.FreeBSD.org](http://cvsup2.br.FreeBSD.org)
- [cvsup3.br.FreeBSD.org](http://cvsup3.br.FreeBSD.org)
- [cvsup4.br.FreeBSD.org](http://cvsup4.br.FreeBSD.org)
- [cvsup5.br.FreeBSD.org](http://cvsup5.br.FreeBSD.org)

#### Великобритания

- [cvsup.uk.FreeBSD.org](http://cvsup.uk.FreeBSD.org)
- [cvsup2.uk.FreeBSD.org](http://cvsup2.uk.FreeBSD.org)
- [cvsup3.uk.FreeBSD.org](http://cvsup3.uk.FreeBSD.org)
- [cvsup4.uk.FreeBSD.org](http://cvsup4.uk.FreeBSD.org)

#### Венгрия

- [cvsup.hu.FreeBSD.org](http://cvsup.hu.FreeBSD.org)

## Германия

- [cvsup.de.FreeBSD.org](http://cvsup.de.FreeBSD.org)
- [cvsup2.de.FreeBSD.org](http://cvsup2.de.FreeBSD.org)
- [cvsup3.de.FreeBSD.org](http://cvsup3.de.FreeBSD.org)
- [cvsup4.de.FreeBSD.org](http://cvsup4.de.FreeBSD.org)
- [cvsup5.de.FreeBSD.org](http://cvsup5.de.FreeBSD.org)
- [cvsup6.de.FreeBSD.org](http://cvsup6.de.FreeBSD.org)
- [cvsup7.de.FreeBSD.org](http://cvsup7.de.FreeBSD.org)
- [cvsup8.de.FreeBSD.org](http://cvsup8.de.FreeBSD.org)

## Греция

- [cvsup.gr.FreeBSD.org](http://cvsup.gr.FreeBSD.org)
- [cvsup2.gr.FreeBSD.org](http://cvsup2.gr.FreeBSD.org)

## Дания

- [cvsup.dk.FreeBSD.org](http://cvsup.dk.FreeBSD.org)
- [cvsup2.dk.FreeBSD.org](http://cvsup2.dk.FreeBSD.org)

## Ирландия

- [cvsup.ie.FreeBSD.org](http://cvsup.ie.FreeBSD.org)

## Исландия

- [cvsup.is.FreeBSD.org](http://cvsup.is.FreeBSD.org)

## Испания

- [cvsup.es.FreeBSD.org](http://cvsup.es.FreeBSD.org)
- [cvsup2.es.FreeBSD.org](http://cvsup2.es.FreeBSD.org)

- [cvsup3.es.FreeBSD.org](http://cvsup3.es.FreeBSD.org)

#### Италия

- [cvsup.it.FreeBSD.org](http://cvsup.it.FreeBSD.org)

#### Канада

- [cvsup.ca.FreeBSD.org](http://cvsup.ca.FreeBSD.org)

#### Китай

- [cvsup.cn.FreeBSD.org](http://cvsup.cn.FreeBSD.org)
- [cvsup2.cn.FreeBSD.org](http://cvsup2.cn.FreeBSD.org)
- [cvsup3.cn.FreeBSD.org](http://cvsup3.cn.FreeBSD.org)
- [cvsup4.cn.FreeBSD.org](http://cvsup4.cn.FreeBSD.org)
- [cvsup5.cn.FreeBSD.org](http://cvsup5.cn.FreeBSD.org)

#### Корея

- [cvsup.kr.FreeBSD.org](http://cvsup.kr.FreeBSD.org)
- [cvsup2.kr.FreeBSD.org](http://cvsup2.kr.FreeBSD.org)
- [cvsup3.kr.FreeBSD.org](http://cvsup3.kr.FreeBSD.org)

#### Кувейт

- [cvsup1.kw.FreeBSD.org](http://cvsup1.kw.FreeBSD.org)

#### Латвия

- [cvsup.lv.FreeBSD.org](http://cvsup.lv.FreeBSD.org)

## Литва

- [cvsup.lt.FreeBSD.org](http://cvsup.lt.FreeBSD.org)
- [cvsup2.lt.FreeBSD.org](http://cvsup2.lt.FreeBSD.org)
- [cvsup3.lt.FreeBSD.org](http://cvsup3.lt.FreeBSD.org)

## Нидерланды

- [cvsup.nl.FreeBSD.org](http://cvsup.nl.FreeBSD.org)
- [cvsup2.nl.FreeBSD.org](http://cvsup2.nl.FreeBSD.org)
- [cvsup3.nl.FreeBSD.org](http://cvsup3.nl.FreeBSD.org)
- [cvsup5.nl.FreeBSD.org](http://cvsup5.nl.FreeBSD.org)

## Новая Зеландия

- [cvsup.nz.FreeBSD.org](http://cvsup.nz.FreeBSD.org)

## Норвегия

- [cvsup.no.FreeBSD.org](http://cvsup.no.FreeBSD.org)

## Польша

- [cvsup.pl.FreeBSD.org](http://cvsup.pl.FreeBSD.org)
- [cvsup2.pl.FreeBSD.org](http://cvsup2.pl.FreeBSD.org)
- [cvsup3.pl.FreeBSD.org](http://cvsup3.pl.FreeBSD.org)

## Португалия

- [cvsup.pt.FreeBSD.org](http://cvsup.pt.FreeBSD.org)
- [cvsup2.pt.FreeBSD.org](http://cvsup2.pt.FreeBSD.org)
- [cvsup3.pt.FreeBSD.org](http://cvsup3.pt.FreeBSD.org)

## Россия

- [cvsup.ru.FreeBSD.org](http://cvsup.ru.FreeBSD.org)
- [cvsup2.ru.FreeBSD.org](http://cvsup2.ru.FreeBSD.org)
- [cvsup3.ru.FreeBSD.org](http://cvsup3.ru.FreeBSD.org)
- [cvsup4.ru.FreeBSD.org](http://cvsup4.ru.FreeBSD.org)
- [cvsup5.ru.FreeBSD.org](http://cvsup5.ru.FreeBSD.org)
- [cvsup6.ru.FreeBSD.org](http://cvsup6.ru.FreeBSD.org)

## Румыния

- [cvsup.ro.FreeBSD.org](http://cvsup.ro.FreeBSD.org)
- [cvsup1.ro.FreeBSD.org](http://cvsup1.ro.FreeBSD.org)
- [cvsup2.ro.FreeBSD.org](http://cvsup2.ro.FreeBSD.org)
- [cvsup3.ro.FreeBSD.org](http://cvsup3.ro.FreeBSD.org)

## США

- [cvsup1.us.FreeBSD.org](http://cvsup1.us.FreeBSD.org)
- [cvsup2.us.FreeBSD.org](http://cvsup2.us.FreeBSD.org)
- [cvsup3.us.FreeBSD.org](http://cvsup3.us.FreeBSD.org)
- [cvsup4.us.FreeBSD.org](http://cvsup4.us.FreeBSD.org)
- [cvsup5.us.FreeBSD.org](http://cvsup5.us.FreeBSD.org)
- [cvsup6.us.FreeBSD.org](http://cvsup6.us.FreeBSD.org)
- [cvsup7.us.FreeBSD.org](http://cvsup7.us.FreeBSD.org)
- [cvsup8.us.FreeBSD.org](http://cvsup8.us.FreeBSD.org)
- [cvsup9.us.FreeBSD.org](http://cvsup9.us.FreeBSD.org)
- [cvsup10.us.FreeBSD.org](http://cvsup10.us.FreeBSD.org)
- [cvsup11.us.FreeBSD.org](http://cvsup11.us.FreeBSD.org)
- [cvsup12.us.FreeBSD.org](http://cvsup12.us.FreeBSD.org)
- [cvsup13.us.FreeBSD.org](http://cvsup13.us.FreeBSD.org)
- [cvsup14.us.FreeBSD.org](http://cvsup14.us.FreeBSD.org)
- [cvsup15.us.FreeBSD.org](http://cvsup15.us.FreeBSD.org)
- [cvsup16.us.FreeBSD.org](http://cvsup16.us.FreeBSD.org)

- [cvsup18.us.FreeBSD.org](http://cvsup18.us.FreeBSD.org)

#### Сан Марино

- [cvsup.sm.FreeBSD.org](http://cvsup.sm.FreeBSD.org)

#### Сингапур

- [cvsup.sg.FreeBSD.org](http://cvsup.sg.FreeBSD.org)

#### Словацкая Республика

- [cvsup.sk.FreeBSD.org](http://cvsup.sk.FreeBSD.org)
- [cvsup2.sk.FreeBSD.org](http://cvsup2.sk.FreeBSD.org)

#### Словения

- [cvsup.si.FreeBSD.org](http://cvsup.si.FreeBSD.org)
- [cvsup2.si.FreeBSD.org](http://cvsup2.si.FreeBSD.org)

#### Тайвань

- [cvsup.tw.FreeBSD.org](http://cvsup.tw.FreeBSD.org)
- [cvsup3.tw.FreeBSD.org](http://cvsup3.tw.FreeBSD.org)
- [cvsup4.tw.FreeBSD.org](http://cvsup4.tw.FreeBSD.org)
- [cvsup5.tw.FreeBSD.org](http://cvsup5.tw.FreeBSD.org)
- [cvsup6.tw.FreeBSD.org](http://cvsup6.tw.FreeBSD.org)
- [cvsup7.tw.FreeBSD.org](http://cvsup7.tw.FreeBSD.org)
- [cvsup8.tw.FreeBSD.org](http://cvsup8.tw.FreeBSD.org)
- [cvsup9.tw.FreeBSD.org](http://cvsup9.tw.FreeBSD.org)
- [cvsup10.tw.FreeBSD.org](http://cvsup10.tw.FreeBSD.org)
- [cvsup11.tw.FreeBSD.org](http://cvsup11.tw.FreeBSD.org)
- [cvsup12.tw.FreeBSD.org](http://cvsup12.tw.FreeBSD.org)



- [cvsup13.tw.FreeBSD.org](http://cvsup13.tw.FreeBSD.org)

## Турция

- [cvsup.tr.FreeBSD.org](http://cvsup.tr.FreeBSD.org)

## Украина

- [cvsup2.ua.FreeBSD.org](http://cvsup2.ua.FreeBSD.org)
- [cvsup3.ua.FreeBSD.org](http://cvsup3.ua.FreeBSD.org)
- [cvsup4.ua.FreeBSD.org](http://cvsup4.ua.FreeBSD.org)
- [cvsup5.ua.FreeBSD.org](http://cvsup5.ua.FreeBSD.org)
- [cvsup6.ua.FreeBSD.org](http://cvsup6.ua.FreeBSD.org)
- [cvsup7.ua.FreeBSD.org](http://cvsup7.ua.FreeBSD.org)

## Филиппины

- [cvsup1.ph.FreeBSD.org](http://cvsup1.ph.FreeBSD.org)

## Финляндия

- [cvsup.fi.FreeBSD.org](http://cvsup.fi.FreeBSD.org)
- [cvsup2.fi.FreeBSD.org](http://cvsup2.fi.FreeBSD.org)

## Франция

- [cvsup.fr.FreeBSD.org](http://cvsup.fr.FreeBSD.org)
- [cvsup2.fr.FreeBSD.org](http://cvsup2.fr.FreeBSD.org)
- [cvsup3.fr.FreeBSD.org](http://cvsup3.fr.FreeBSD.org)
- [cvsup4.fr.FreeBSD.org](http://cvsup4.fr.FreeBSD.org)
- [cvsup5.fr.FreeBSD.org](http://cvsup5.fr.FreeBSD.org)
- [cvsup8.fr.FreeBSD.org](http://cvsup8.fr.FreeBSD.org)

## Чехия

- [cvsup.cz.FreeBSD.org](http://cvsup.cz.FreeBSD.org)

## Швейцария

- [cvsup.ch.FreeBSD.org](http://cvsup.ch.FreeBSD.org)

## Швеция

- [cvsup.se.FreeBSD.org](http://cvsup.se.FreeBSD.org)
- [cvsup3.se.FreeBSD.org](http://cvsup3.se.FreeBSD.org)

## Эстония

- [cvsup.ee.FreeBSD.org](http://cvsup.ee.FreeBSD.org)

## Южная Африка

- [cvsup.za.FreeBSD.org](http://cvsup.za.FreeBSD.org)
- [cvsup2.za.FreeBSD.org](http://cvsup2.za.FreeBSD.org)

## Япония

- [cvsup.jp.FreeBSD.org](http://cvsup.jp.FreeBSD.org)
- [cvsup2.jp.FreeBSD.org](http://cvsup2.jp.FreeBSD.org)
- [cvsup3.jp.FreeBSD.org](http://cvsup3.jp.FreeBSD.org)
- [cvsup4.jp.FreeBSD.org](http://cvsup4.jp.FreeBSD.org)
- [cvsup5.jp.FreeBSD.org](http://cvsup5.jp.FreeBSD.org)
- [cvsup6.jp.FreeBSD.org](http://cvsup6.jp.FreeBSD.org)

## А.6. Теги CVS

При получении или обновлении исходных текстов с использованием **cv**s или **CVSup** должен быть указан тег ревизии. Тег относится либо к определенной линии разработки FreeBSD либо к определенному моменту времени. Первый тип называется "тегом ветви", второй тип называется "тегом релиза".

### А.6.1. Теги ветвей

Все они, за исключением HEAD (который всегда работает), относятся только к дереву `src/`. Деревья `ports/`, `doc/` и `www/` не имеют тегов ветвей.

#### HEAD

Символическое имя для основной ветви разработки, FreeBSD-CURRENT. Используется по умолчанию, если тег не указан.

В **CVSup**, этот тег представлен символом `.` (не знак пунктуации, а символ `"."`).

**Замечание:** В CVS тег по умолчанию используется, если тег не указан вообще. Обычно не очень хорошей идеей является обновление до CURRENT на машине со STABLE, только если вы не стремитесь именно к этому.

#### RELENG\_5

Ветвь разработки FreeBSD-5.X, известная также как FreeBSD 5-STABLE.

#### RELENG\_5\_3

Ветвь релиза для FreeBSD-5.3, используемая только для исправлений безопасности и других критических исправлений.

#### RELENG\_5\_2

Ветвь релиза для FreeBSD-5.2 и FreeBSD-5.2.1, используемая только для исправлений безопасности и других критических исправлений.

#### RELENG\_5\_1

Ветвь релиза для FreeBSD-5.1, используемая только для исправлений безопасности и других критических исправлений.

#### RELENG\_5\_0

Ветвь релиза для FreeBSD-5.0, используемая только для исправлений безопасности и других критических исправлений.

#### RELENG\_4

Ветвь разработки FreeBSD-4.X, известная также как FreeBSD 4-STABLE.

#### RELENG\_4\_10

Ветвь релиза для FreeBSD-4.10, используемая только для исправлений безопасности и других критических исправлений.

#### RELENG\_4\_9

Ветвь релиза для FreeBSD-4.9, используемая только для исправлений безопасности и других критических исправлений.

#### RELENG\_4\_8

Ветвь релиза для FreeBSD-4.8, используемая только для исправлений безопасности и других критических исправлений.

#### RELENG\_4\_7

Ветвь релиза для FreeBSD-4.7, используемая только для исправлений безопасности и других критических исправлений.

#### RELENG\_4\_6

Ветвь релиза для FreeBSD-4.6 и FreeBSD-4.6.2, используемая только для исправлений безопасности и других критических исправлений.

#### RELENG\_4\_5

Ветвь релиза для FreeBSD-4.5, используемая только для исправлений безопасности и других критических исправлений.

#### RELENG\_4\_4

Ветвь релиза для FreeBSD-4.4, используемая только для исправлений безопасности и других критических исправлений.

#### RELENG\_4\_3

Ветвь релиза для FreeBSD-4.3, используемая только для исправлений безопасности и других критических исправлений.

#### RELENG\_3

Ветвь разработки для FreeBSD-3.X, известная также как 3.X-STABLE.

#### RELENG\_2\_2

Ветвь разработки для FreeBSD-2.2.X, известная также как 2.2-STABLE. Эта ветвь по большей части устарела.

### А.6.2. Теги релизов

Эти теги относятся к определенным моментам времени, когда были выпущены конкретные версии FreeBSD. Процесс выпуска релизов более детально документирован в информации о выпусках релизов (<http://www.FreeBSD.org/releng/>) и статье о процессе подготовки релизов ([http://www.FreeBSD.org/doc/ru\\_RU.KOI8-R/articles/releng/release-proc.html](http://www.FreeBSD.org/doc/ru_RU.KOI8-R/articles/releng/release-proc.html)). Дерево `src` использует имена

тегов, начинающиеся с RELENG\_. Деревья ports и doc используют теги, имена которых начинаются с RELEASE. Наконец, в дереве www нет специальных имен для релизов.

RELENG\_5\_3\_0\_RELEASE

FreeBSD 5.3

RELENG\_4\_10\_0\_RELEASE

FreeBSD 4.10

RELENG\_5\_2\_1\_RELEASE

FreeBSD 5.2.1

RELENG\_5\_2\_0\_RELEASE

FreeBSD 5.2

RELENG\_4\_9\_0\_RELEASE

FreeBSD 4.9

RELENG\_5\_1\_0\_RELEASE

FreeBSD 5.1

RELENG\_4\_8\_0\_RELEASE

FreeBSD 4.8

RELENG\_5\_0\_0\_RELEASE

FreeBSD 5.0

RELENG\_4\_7\_0\_RELEASE

FreeBSD 4.7

RELENG\_4\_6\_2\_RELEASE

FreeBSD 4.6.2

RELENG\_4\_6\_1\_RELEASE

FreeBSD 4.6.1

RELENG\_4\_6\_0\_RELEASE

FreeBSD 4.6

RELENG\_4\_5\_0\_RELEASE

FreeBSD 4.5

RELENG\_4\_4\_0\_RELEASE

FreeBSD 4.4

RELENG\_4\_3\_0\_RELEASE

FreeBSD 4.3

RELENG\_4\_2\_0\_RELEASE

FreeBSD 4.2

RELENG\_4\_1\_1\_RELEASE

FreeBSD 4.1.1

RELENG\_4\_1\_0\_RELEASE

FreeBSD 4.1

RELENG\_4\_0\_0\_RELEASE

FreeBSD 4.0

RELENG\_3\_5\_0\_RELEASE

FreeBSD-3.5

RELENG\_3\_4\_0\_RELEASE

FreeBSD-3.4

RELENG\_3\_3\_0\_RELEASE

FreeBSD-3.3

RELENG\_3\_2\_0\_RELEASE

FreeBSD-3.2

RELENG\_3\_1\_0\_RELEASE

FreeBSD-3.1

RELENG\_3\_0\_0\_RELEASE

FreeBSD-3.0

RELENG\_2\_2\_8\_RELEASE

FreeBSD-2.2.8

RELENG\_2\_2\_7\_RELEASE

FreeBSD-2.2.7

RELENG\_2\_2\_6\_RELEASE

FreeBSD-2.2.6

RELENG\_2\_2\_5\_RELEASE

FreeBSD-2.2.5

RELENG\_2\_2\_2\_RELEASE

FreeBSD-2.2.2

RELENG\_2\_2\_1\_RELEASE

FreeBSD-2.2.1

RELENG\_2\_2\_0\_RELEASE

FreeBSD-2.2.0

## A.7. AFS сайты

AFS серверы для FreeBSD работают на следующих сайтах:

Швеция

Часть файлов: /afs/stacken.kth.se/ftp/pub/FreeBSD/

stacken.kth.se	# Stacken Computer Club, KTH, Sweden
130.237.234.43	#hot.stacken.kth.se
130.237.237.230	#fishburger.stacken.kth.se
130.237.234.3	#milko.stacken.kth.se

Ответственный <ftp@stacken.kth.se>

## A.8. rsync сайты

Следующие сайты организуют доступ к FreeBSD через протокол rsync. Утилита **rsync** работает в основном тем же путем, что и команда rcp(1), но поддерживает больше параметров и использует протокол удаленного обновления rsync, который передает только разницу между двумя наборами файлов, что значительно повышает скорость синхронизации по сети. Это особенно полезно, если вы поддерживаете зеркало сервера FreeBSD FTP, или репозитория CVS. Пакет **rsync** доступен для многих операционных систем, в FreeBSD используйте порт net/rsync или пакет.

Чешская республика

rsync://ftp.cz.FreeBSD.org/

Доступные коллекции:

- ftp: Частичное зеркало FreeBSD FTP сервера.
- FreeBSD: Полное зеркало FreeBSD FTP сервера.

## Германия

`rsync://grappa.unix-ag.uni-kl.de/`

Доступные соединения:

- `freebsd-cvs`: Полный FreeBSD CVS репозиторий.

Этот компьютер помимо прочих также зеркалирует CVS репозитории проектов NetBSD и OpenBSD.

## Нидерланды

`rsync://ftp.nl.FreeBSD.org/`

Доступные коллекции:

- `vol/3/freebsd-core`: Полное зеркало FreeBSD FTP сервера.

## Великобритания

`rsync://rsync.mirror.ac.uk/`

Доступные коллекции:

- `ftp.FreeBSD.org`: Полное зеркало FreeBSD FTP сервера.

## Соединенные Штаты Америки

`rsync://ftp-master.FreeBSD.org/`

Этот сервер может использоваться только основными зеркалами FreeBSD.

Доступные коллекции:

- `FreeBSD`: Основной архив FreeBSD FTP сервера.
- `acl`: Основной ACL список FreeBSD.

`rsync://ftp13.FreeBSD.org/`

Доступные коллекции:

- `FreeBSD`: Полное зеркало FreeBSD FTP сервера.



# Приложение В. Библиография

Так как страницы Справочника FreeBSD предоставляют лишь описание отдельных частей операционной системы FreeBSD, они не очень удобны для иллюстрации объединения этих частей вместе для того, чтобы настроить ОС и сделать ее работу более гладкой. Для этого незаменимы хорошая книга по системному администрированию UNIX и хорошее руководство пользователя.

## В.1. Книги и журналы, специализирующиеся на FreeBSD

*Международные книги и журналы:*

- Using FreeBSD (<http://jdli.tw.FreeBSD.org/publication/book/freebsd2/index.html>) (на китайском).
- FreeBSD Unleashed (перевод на китайский), опубликовано China Machine Press (<http://www.hzbook.com/>). ISBN 7-111-10201-0.
- FreeBSD From Scratch First Edition (на китайском), опубликовано China Machine Press. ISBN 7-111-07482-3.
- FreeBSD From Scratch Second Edition (на китайском), опубликовано China Machine Press. ISBN 7-111-10286-X.
- FreeBSD Handbook (на китайском), опубликовано Posts & Telecom Press (<http://www.ptpress.com.cn/>). ISBN 7-115-10541-3.
- FreeBSD 3.x Internet (на китайском), опубликовано Tsinghua University Press (<http://www.tup.tsinghua.edu.cn/>). ISBN 7-900625-66-6.
- FreeBSD & Windows (на китайском), ISBN 7-113-03845-X
- FreeBSD Internet Services HOWTO (на китайском), ISBN 7-113-03423-3
- FreeBSD for PC 98'ers (на японском), выпущено SHUWA System Co, LTD. ISBN 4-87966-468-5 C3055 P2900E.
- FreeBSD (на японском), выпущено CUTT. ISBN 4-906391-22-2 C3055 P2400E.
- Complete Introduction to FreeBSD (<http://www.shoeisha.com/book/Detail.asp?bid=650>) (на японском), выпущено Shoeisha Co., Ltd (<http://www.shoeisha.co.jp/>). ISBN 4-88135-473-6 P3600E.
- Personal UNIX Starter Kit FreeBSD (<http://www.ascii.co.jp/pb/book1/shinkan/detail/1322785.html>) (на японском), выпущено ASCII (<http://www.ascii.co.jp/>). ISBN 4-7561-1733-3 P3000E.
- FreeBSD Handbook (японский перевод), выпущено ASCII (<http://www.ascii.co.jp/>). ISBN 4-7561-1580-2 P3800E.
- FreeBSD mit Methode (на немецком), выпущено Computer und Literatur Verlag (<http://www.cul.de/>) Vertrieb Hanser, 1998. ISBN 3-932311-31-0.
- FreeBSD 4 - Installieren, Konfigurieren, Administrieren (<http://www.cul.de/freebsd.html>) (на немецком), выпущено Computer und Literatur Verlag (<http://www.cul.de/>), 2001. ISBN 3-932311-88-4.
- FreeBSD 5 - Installieren, Konfigurieren, Administrieren (<http://www.cul.de/freebsd.html>) (на немецком), выпущено Computer und Literatur Verlag (<http://www.cul.de/>), 2003. ISBN 3-936546-06-1.

- FreeBSD de Luxe (<http://www.mitp.de/vmi/mitp/detail/pWert/1343/>) (на немецком), выпущено Verlag Modere Industrie (<http://www.mitp.de>), 2003. ISBN 3-8266-1343-0.
- FreeBSD Install and Utilization Manual (<http://www.pc.mycom.co.jp/FreeBSD/install-manual.html>) (на японском), выпущено Mainichi Communications Inc. (<http://www.pc.mycom.co.jp/>).
- Onno W Purbo, Dodi Maryanto, Syahril Hubbany, Widjil Widodo *Создание Интернет Сервера с использованием FreeBSD* (<http://maxwell.itb.ac.id/>) (на Индонезийском языке), выпущено Elex Media Komputindo (<http://www.elexmedia.co.id/>).

*Книги и журналы на английском языке:*

- Absolute BSD: The Ultimate Guide to FreeBSD (<http://www.AbsoluteBSD.com/>), выпущено No Starch Press (<http://www.nostarch.com/>), 2002. ISBN: 1886411743
- The Complete FreeBSD (<http://www.freebsdmail.com/cgi-bin/fm/bsdcomp>), выпущено O'Reilly (<http://www.oreilly.com/>), 2003. ISBN: 0596005164
- The FreeBSD Corporate Networker's Guide (<http://www.freebsd-corp-net-guide.com>), выпущено Addison-Wesley (<http://www.awl.com/awl/>), 2000. ISBN: 0201704811
- FreeBSD: An Open-Source Operating System for Your Personal Computer (<http://andrsn.stanford.edu/FreeBSD/introbook/>), выпущено The Bit Tree Press, 2001. ISBN: 0971204500
- Teach Yourself FreeBSD in 24 Hours, выпущено Sams (<http://www.sampublishing.com/>), 2002. ISBN: 0672324245
- FreeBSD unleashed, выпущено Sams (<http://www.sampublishing.com/>), 2002. ISBN: 0672324563
- FreeBSD: The Complete Reference, выпущено McGrawHill (<http://books.mcgraw-hill.com>), 2003. ISBN: 0072224096

## В.2. Руководства для пользователей

- Computer Systems Research Group, UC Berkeley. *4.4BSD User's Reference Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-075-9
- Computer Systems Research Group, UC Berkeley. *4.4BSD User's Supplementary Documents*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-076-7
- *UNIX in a Nutshell*. O'Reilly & Associates, Inc., 1990. ISBN 093717520X
- Mui, Linda. *What You Need To Know When You Can't Find Your UNIX System Administrator*. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-104-6
- Ohio State University (<http://www-wks.acs.ohio-state.edu/>) написал Ознакомительный Курс UNIX ([http://www-wks.acs.ohio-state.edu/unix\\_course/unix.html](http://www-wks.acs.ohio-state.edu/unix_course/unix.html)) который доступен в Online в HTML и PS форматах.

Итальянский перевод ([http://www.FreeBSD.org/doc/it\\_IT.ISO8859-15/books/unix-introduction/index.html](http://www.FreeBSD.org/doc/it_IT.ISO8859-15/books/unix-introduction/index.html)) этого документа доступен как часть FreeBSD Italian Documentation Project.

- Jpman Project, Japan FreeBSD Users Group (<http://www.jp.FreeBSD.org/>). FreeBSD User's Reference Manual (<http://www.pc.mycom.co.jp/FreeBSD/urm.html>) (Японский перевод). Mainichi Communications Inc. (<http://www.pc.mycom.co.jp/>), 1998. ISBN4-8399-0088-4 P3800E.
- Эдинбургский Университет (<http://www.ed.ac.uk/>) составил Online Путеводитель (<http://unixhelp.ed.ac.uk/>) для новичков в UNIX.

### В.3. Руководства для администраторов

- Albitz, Paul and Liu, Cricket. *DNS and BIND*, 4th Ed. O'Reilly & Associates, Inc., 2001. ISBN 1-59600-158-4
- Computer Systems Research Group, UC Berkeley. *4.4BSD System Manager's Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-080-5
- Costales, Brian, et al. *Sendmail*, 2nd Ed. O'Reilly & Associates, Inc., 1997. ISBN 1-56592-222-0
- Frisch, Aileen. *Essential System Administration*, 2nd Ed. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-127-5
- Hunt, Craig. *TCP/IP Network Administration*, 2nd Ed. O'Reilly & Associates, Inc., 1997. ISBN 1-56592-322-7
- Nemeth, Evi. *UNIX System Administration Handbook*. 2nd Ed. Prentice Hall, 2000. ISBN 0-13-020601-6
- Stern, Hal *Managing NFS and NIS* O'Reilly & Associates, Inc., 1991. ISBN 0-937175-75-7
- Jpman Project, Japan FreeBSD Users Group (<http://www.jp.FreeBSD.org/>). FreeBSD System Administrator's Manual (<http://www.pc.mycom.co.jp/FreeBSD/sam.html>) (Japanese translation). Mainichi Communications Inc. (<http://www.pc.mycom.co.jp/>), 1998. ISBN4-8399-0109-0 P3300E.
- Dreyfus, Emmanuel. *Cahiers de l'Admin: BSD* (<http://www.eyrolles.com/Informatique/Livre/9782212114638/>) 2nd Ed. (на французском), Eyrolles, 2004. ISBN 2-212-11463-X

### В.4. Руководства для программистов

- Asente, Paul, Converse, Diana, and Swick, Ralph. *X Window System Toolkit*. Digital Press, 1998. ISBN 1-55558-178-1
- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Reference Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-078-3
- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Supplementary Documents*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-079-1
- Harbison, Samuel P. and Steele, Guy L. Jr. *C: A Reference Manual*. 4th ed. Prentice Hall, 1995. ISBN 0-13-326224-3
- Kernighan, Brian and Dennis M. Ritchie. *The C Programming Language*. 2nd Ed. PTR Prentice Hall, 1988. ISBN 0-13-110362-8
- Lehey, Greg. *Porting UNIX Software*. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-126-7
- Plauger, P. J. *The Standard C Library*. Prentice Hall, 1992. ISBN 0-13-131509-9

- Spinellis, Diomidis. *Code Reading: The Open Source Perspective* (<http://www.spinellis.gr/codereading/>). Addison-Wesley, 2003. ISBN 0-201-79940-5
- Stevens, W. Richard. *Advanced Programming in the UNIX Environment*. Reading, Mass. : Addison-Wesley, 1992. ISBN 0-201-56317-7
- Stevens, W. Richard. *UNIX Network Programming*. 2nd Ed, PTR Prentice Hall, 1998. ISBN 0-13-490012-X
- Wells, Bill. "Writing Serial Drivers for UNIX". *Dr. Dobbs's Journal*. 19(15), December 1994. pp68-71, 97-99.

## **В.5. Внутренности операционной системы**

- Andleigh, Prabhat K. *UNIX System Architecture*. Prentice-Hall, Inc., 1990. ISBN 0-13-949843-5
- Jolitz, William. "Porting UNIX to the 386". *Dr. Dobbs's Journal*. January 1991-July 1992.
- Leffler, Samuel J., Marshall Kirk McKusick, Michael J Karels and John Quarterman *The Design and Implementation of the 4.3BSD UNIX Operating System*. Reading, Mass. : Addison-Wesley, 1989. ISBN 0-201-06196-1
- Leffler, Samuel J., Marshall Kirk McKusick, *The Design and Implementation of the 4.3BSD UNIX Operating System: Answer Book*. Reading, Mass. : Addison-Wesley, 1991. ISBN 0-201-54629-9
- McKusick, Marshall Kirk, Keith Bostic, Michael J Karels, and John Quarterman. *The Design and Implementation of the 4.4BSD Operating System*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-54979-4  
(глава 2 этой книги доступна онлайн ([http://www.FreeBSD.org/doc/en\\_US.ISO8859-1/books/design-44bsd/book.html](http://www.FreeBSD.org/doc/en_US.ISO8859-1/books/design-44bsd/book.html)) как часть Проекта документирования FreeBSD, и глава 9 доступна здесь ([http://www.netapp.com/tech\\_library/nfsbook.print](http://www.netapp.com/tech_library/nfsbook.print)).)
- Marshall Kirk McKusick, George V. Neville-Neil *The Design and Implementation of the FreeBSD Operating System*. Boston, Mass. : Addison-Wesley, 2004. ISBN 0-201-70245-2
- Stevens, W. Richard. *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63346-9
- Schimmel, Curt. *Unix Systems for Modern Architectures*. Reading, Mass. : Addison-Wesley, 1994. ISBN 0-201-63338-8
- Stevens, W. Richard. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP and the UNIX Domain Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63495-3
- Vahalia, Uresh. *UNIX Internals -- The New Frontiers*. Prentice Hall, 1996. ISBN 0-13-101908-2
- Wright, Gary R. and W. Richard Stevens. *TCP/IP Illustrated, Volume 2: The Implementation*. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63354-X

## В.6. Безопасность

- Cheswick, William R. and Steven M. Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63357-4
- Garfinkel, Simson and Gene Spafford. *Practical UNIX & Internet Security*. 2nd Ed. O'Reilly & Associates, Inc., 1996. ISBN 1-56592-148-8
- Garfinkel, Simson. *PGP Pretty Good Privacy* O'Reilly & Associates, Inc., 1995. ISBN 1-56592-098-8

## В.7. Оборудование

- Anderson, Don and Tom Shanley. *Pentium Processor System Architecture*. 2nd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40992-5
- Ferraro, Richard F. *Programmer's Guide to the EGA, VGA, and Super VGA Cards*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-62490-7
- Intel Corporation publishes documentation on their CPUs, chipsets and standards on their developer web site (<http://developer.intel.com/>), usually as PDF files.
- Shanley, Tom. *80486 System Architecture*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40994-1
- Shanley, Tom. *ISA System Architecture*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40996-8
- Shanley, Tom. *PCI System Architecture*. 4th ed. Reading, Mass. : Addison-Wesley, 1999. ISBN 0-201-30974-2
- Van Gillsuwe, Frank. *The Undocumented PC*, 2nd Ed. Reading, Mass: Addison-Wesley Pub. Co., 1996. ISBN 0-201-47950-8
- Messmer, Hans-Peter. *The Indispensable PC Hardware Book*, 4th Ed. Reading, Mass: Addison-Wesley Pub. Co., 2002. ISBN 0-201-59616-4

## В.8. История UNIX

- Lion, John *Lion's Commentary on UNIX, 6th Ed. With Source Code*. ITP Media Group, 1996. ISBN 1573980137
- Raymond, Eric S. *The New Hacker's Dictionary, 3rd edition*. MIT Press, 1996. ISBN 0-262-68092-0. Also known as the Jargon File (<http://www.catb.org/~esr/jargon/html/index.html>)
- Salus, Peter H. *A quarter century of UNIX*. Addison-Wesley Publishing Company, Inc., 1994. ISBN 0-201-54777-5
- Simon Garfinkel, Daniel Weise, Steven Strassmann. *The UNIX-HATERS Handbook*. IDG Books Worldwide, Inc., 1994. ISBN 1-56884-203-1. Не печаталась, доступна по адресу online (<http://research.microsoft.com/~daniel/unix-haters.html>).
- Don Libes, Sandy Ressler *Life with UNIX* — special edition. Prentice-Hall, Inc., 1989. ISBN 0-13-536657-7
- *The BSD family tree*. <http://www.FreeBSD.org/cgi/cvsweb.cgi/src/share/misc/bsd-family-tree> или `/usr/share/misc/bsd-family-tree` на компьютере FreeBSD.

- *The BSD Release Announcements collection*. 1997. <http://www.de.FreeBSD.org/de/ftp/releases/>
- *Networked Computer Science Technical Reports Library*. <http://www.ncstrl.org/>
- *Old BSD releases from the Computer Systems Research group (CSRG)*. <http://www.mckusick.com/csrg/>: В набор на 4CD включены все версии BSD от 1BSD до 4.4BSD и 4.4BSD-Lite2 (но к сожалению нет 2.11BSD). На последнем диске находится также последняя версия исходных текстов и файлы SCCS.

## В.9. Прочие издания

- *The C/C++ Users Journal*. R&D Publications Inc. ISSN 1075-2838
- *Sys Admin — The Journal for UNIX System Administrators* Miller Freeman, Inc., ISSN 1061-2688
- *freeX — Das Magazin für Linux - BSD - UNIX* (на немецком) Computer- und Literaturverlag GmbH, ISSN 1436-7033

# Приложение С. Ресурсы в интернет

Высокая скорость прогресса FreeBSD делает непрактичным использование печатных изданий для информирования о последних разработках. Электронные ресурсы это лучший, а зачастую и единственный способ информирования о новых возможностях. Поскольку FreeBSD основывается на усилиях добровольцев, сообщество пользователей само по себе зачастую выполняет роль "службы технической поддержки", а электронная почта и новости USENET это наиболее эффективный способ обращения к этому сообществу.

Наиболее важная контактная информация сообщества пользователей FreeBSD показана ниже. Если вам известно о других ресурсах, не упомянутых здесь, пожалуйста отправьте информацию о них в Список рассылки Проекта Документации FreeBSD (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc>), чтобы мы могли включить в этот документ.

## С.1. Списки рассылки

Хотя многие участники разработки FreeBSD читают USENET, мы не можем всегда гарантировать, что ответим на ваши вопросы в краткий срок (или вообще), если вы отправите их только в группы `comp.unix.bsd.freebsd.*`. Вопросы, отправленные в соответствующий список рассылки, достигнут и нас и обширной аудитории FreeBSD, что несомненно гарантирует лучшую (или как минимум более быструю) поддержку.

**Внимание:** Все сообщения в приведенные ниже списки рассылки должны быть составлены *только на английском языке*.

Описание каждой рассылки дано в конце этого документа. *Пожалуйста, прочтите описание перед подпиской или отправкой почты в любой из списков*. Большинство наших подписчиков получают многие сотни относящихся к FreeBSD сообщений каждый день, и определяя правила использования рассылок мы стремимся удержать высокое соотношение "сигнал к шуму". При меньшем соотношении списки рассылки перестанут быть эффективной средой общения участников проекта.

Архивы поддерживаются для всех списков рассылки и поиск по ним организован на WWW сервере FreeBSD (<http://www.FreeBSD.org/ru/search/index.html>). Поиск в архиве по ключевым словам дает отличный способ получения ответов на часто задаваемые вопросы и должен быть выполнен перед отправкой вопроса.

### С.1.1. Списки рассылки

*Общие списки:* Ниже представлены общие списки рассылки, к которым каждый может (и приглашается) присоединиться:

Список	Назначение
cvs-all ( <a href="http://lists.FreeBSD.org/mailman/listinfo/cvs-all">http://lists.FreeBSD.org/mailman/listinfo/cvs-all</a> )	Изменения, вносимые в дерево исходных текстов FreeBSD

Список	Назначение
freebsd-advocacy ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-advocacy">http://lists.FreeBSD.org/mailman/listinfo/freebsd-advocacy</a> )	В защиту FreeBSD
freebsd-announce ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce">http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce</a> )	Важные события и вехи проекта
freebsd-arch ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-arch">http://lists.FreeBSD.org/mailman/listinfo/freebsd-arch</a> )	Обсуждения архитектуры и дизайна системы
freebsd-bugbusters ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugbusters">http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugbusters</a> )	Обсуждения, относящиеся к поддержке базы данных сообщений о проблемах FreeBSD и соответствующим инструментам
freebsd-bugs ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugs">http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugs</a> )	Сообщения о проблемах
freebsd-chat ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-chat">http://lists.FreeBSD.org/mailman/listinfo/freebsd-chat</a> )	Не-технические темы, относящиеся к сообществу FreeBSD
freebsd-confi g ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-confi g">http://lists.FreeBSD.org/mailman/listinfo/freebsd-confi g</a> )	Разработка инструментов установки и настройки FreeBSD
freebsd-current ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-current">http://lists.FreeBSD.org/mailman/listinfo/freebsd-current</a> )	Обсуждения, относящиеся к использованию FreeBSD-CURRENT
freebsd-isp ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-isp">http://lists.FreeBSD.org/mailman/listinfo/freebsd-isp</a> )	Вопросы использования FreeBSD провайдерами
freebsd-jobs ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-jobs">http://lists.FreeBSD.org/mailman/listinfo/freebsd-jobs</a> )	Вакансии и резюме, относящиеся к FreeBSD, с полной и частичной занятостью
freebsd-newbies ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-newbies">http://lists.FreeBSD.org/mailman/listinfo/freebsd-newbies</a> )	Рассылка FreeBSD для новых пользователей
freebsd-policy ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-policy">http://lists.FreeBSD.org/mailman/listinfo/freebsd-policy</a> )	Публикация правил FreeBSD Core team. Только для чтения, малое количество сообщений
freebsd-questions ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions">http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions</a> )	Вопросы пользователей и техническая поддержка
freebsd-security-notifi cations ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-security-notifi cations">http://lists.FreeBSD.org/mailman/listinfo/freebsd-security-notifi cations</a> )	Уведомления безопасности
freebsd-stable ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable">http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable</a> )	Обсуждения, относящиеся к использованию FreeBSD-STABLE



Список	Назначение
freebsd-test ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-test">http://lists.FreeBSD.org/mailman/listinfo/freebsd-test</a> )	Рассылка для отправки тестовых сообщений (вместо обычных списков рассылки)
<i>Технические списки:</i> Следующие списки предназначены для технических обсуждений. Вам необходимо внимательно прочитать описание перед подпиской или отправкой почты в один из этих списков, поскольку они предназначены для использования внутри проекта.	
Список	Назначение
freebsd-acpi ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi">http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi</a> )	Разработка ACPI и системы управления энергопотреблением
freebsd-afs ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-afs">http://lists.FreeBSD.org/mailman/listinfo/freebsd-afs</a> )	Портирование AFS на FreeBSD
freebsd-aic7xxx ( <a href="http://lists.FreeBSD.org/mailman/listinfo/aic7xxx">http://lists.FreeBSD.org/mailman/listinfo/aic7xxx</a> )	Разработка драйверов для Adaptec AIC 7xxx
freebsd-alpha ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-alpha">http://lists.FreeBSD.org/mailman/listinfo/freebsd-alpha</a> )	Портирование FreeBSD на Alpha
freebsd-amd64 ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-amd64">http://lists.FreeBSD.org/mailman/listinfo/freebsd-amd64</a> )	Портирование FreeBSD на системы AMD64
freebsd-apache ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-apache">http://lists.FreeBSD.org/mailman/listinfo/freebsd-apache</a> )	Обсуждение портов, относящихся к <b>Apache</b>
freebsd-arm ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-arm">http://lists.FreeBSD.org/mailman/listinfo/freebsd-arm</a> )	Портирование FreeBSD на процессоры ARM®
freebsd-atm ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-atm">http://lists.FreeBSD.org/mailman/listinfo/freebsd-atm</a> )	Использование ATM сетей с FreeBSD
freebsd-audit ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-audit">http://lists.FreeBSD.org/mailman/listinfo/freebsd-audit</a> )	Проект аудита исходных текстов
freebsd-binup ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-binup">http://lists.FreeBSD.org/mailman/listinfo/freebsd-binup</a> )	Дизайн и разработка системы бинарных обновлений
freebsd-cluster ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-cluster">http://lists.FreeBSD.org/mailman/listinfo/freebsd-cluster</a> )	Использование FreeBSD в кластерах
freebsd-cvsweb ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-cvsweb">http://lists.FreeBSD.org/mailman/listinfo/freebsd-cvsweb</a> )	Поддержка CVSweb
freebsd-database ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-database">http://lists.FreeBSD.org/mailman/listinfo/freebsd-database</a> )	Обсуждение использования и разработки баз данных под FreeBSD
freebsd-doc ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc">http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc</a> )	Создание относящихся к FreeBSD документов

Список	Назначение
freebsd-emulation ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-emulation">http://lists.FreeBSD.org/mailman/listinfo/freebsd-emulation</a> )	Эмуляция других систем, таких как Linux/DOS/Windows
freebsd-fi rewire ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-fi-rewire">http://lists.FreeBSD.org/mailman/listinfo/freebsd-fi-rewire</a> )	Техническое обсуждение FreeBSD FireWire® (iLink, IEEE 1394)
freebsd-fs ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-fs">http://lists.FreeBSD.org/mailman/listinfo/freebsd-fs</a> )	Файловые системы
freebsd-geom ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-geom">http://lists.FreeBSD.org/mailman/listinfo/freebsd-geom</a> )	Относящиеся к GEOM обсуждения и реализации
freebsd-gnome ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-gnome">http://lists.FreeBSD.org/mailman/listinfo/freebsd-gnome</a> )	Портирование <b>GNOME</b> и приложений <b>GNOME</b>
freebsd-hackers ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers">http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers</a> )	Общее техническое обсуждение
freebsd-hardware ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-hardware">http://lists.FreeBSD.org/mailman/listinfo/freebsd-hardware</a> )	Общее обсуждение оборудования для FreeBSD
freebsd-i18n ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-i18n">http://lists.FreeBSD.org/mailman/listinfo/freebsd-i18n</a> )	Интернационализация FreeBSD
freebsd-ia32 ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-ia32">http://lists.FreeBSD.org/mailman/listinfo/freebsd-ia32</a> )	FreeBSD на платформе IA-32 (Intel x86)
freebsd-ia64 ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-ia64">http://lists.FreeBSD.org/mailman/listinfo/freebsd-ia64</a> )	Портирование FreeBSD на будущие системы Intel IA64
freebsd-ipfw ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-ipfw">http://lists.FreeBSD.org/mailman/listinfo/freebsd-ipfw</a> )	Технические обсуждения, относящиеся к переработке кода IP брандмауэра
freebsd-isdn ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-isdn">http://lists.FreeBSD.org/mailman/listinfo/freebsd-isdn</a> )	Разработчики ISDN
freebsd-java ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-java">http://lists.FreeBSD.org/mailman/listinfo/freebsd-java</a> )	Разработчики Java и люди, портирующие JDK на FreeBSD
freebsd-kde ( <a href="http://freebsd.kde.org/mailman/listinfo/kde-freebsd">http://freebsd.kde.org/mailman/listinfo/kde-freebsd</a> )	Портирование <b>KDE</b> и приложений <b>KDE</b>
freebsd-lfs ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-lfs">http://lists.FreeBSD.org/mailman/listinfo/freebsd-lfs</a> )	Портирование LFS на FreeBSD
freebsd-libh ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-libh">http://lists.FreeBSD.org/mailman/listinfo/freebsd-libh</a> )	Второе поколение программы установки системы и пакетов
freebsd-mips ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-mips">http://lists.FreeBSD.org/mailman/listinfo/freebsd-mips</a> )	Портирование FreeBSD на MIPS®

**Список**

freebsd-mobile  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-mobile>)

freebsd-mozilla  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-mozilla>)

freebsd-multimedia  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-multimedia>)

freebsd-new-bus  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-new-bus>)

freebsd-net  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-net>)

freebsd-openoffice  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-openoffice>)

freebsd-performance  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-performance>)

freebsd-perl  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-perl>)

freebsd-pf  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-pf>)

freebsd-platforms  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-platforms>)

freebsd-ports  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports>)

freebsd-ports-bugs  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports-bugs>)

freebsd-ppc  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-ppc>)

freebsd-qa  
(<http://lists.FreeBSD.org/mailman/listinfo/freebsd-qa>)

**Назначение**

Обсуждение портативных компьютеров

Портирование **Mozilla** на FreeBSD

Мультимедиа приложения

Технические обсуждения архитектуры шины

Обсуждения, относящиеся к сети и исходному тексту TCP/IP

Портирование **OpenOffice.org** и **StarOffice** на FreeBSD

Вопросы оптимизации производительности для быстрых/работающих под большой нагрузкой серверов

Поддержка различных относящихся к **perl** портов

Обсуждение брандмауэра на базе packet filter

Относится к портам для платформ не-Intel архитектуры

Обсуждения коллекции портов

Обсуждения относящихся к портам ошибок/PR

Портирование FreeBSD на PowerPC®

Обсуждение гарантий качества (Quality Assurance), обычно перед релизом

Список	Назначение
freebsd-realtime ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-realtime">http://lists.FreeBSD.org/mailman/listinfo/freebsd-realtime</a> )	Разработка расширений реального времени для FreeBSD
freebsd-scsi ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-scsi">http://lists.FreeBSD.org/mailman/listinfo/freebsd-scsi</a> )	Подсистема SCSI
freebsd-security ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-security">http://lists.FreeBSD.org/mailman/listinfo/freebsd-security</a> )	Сообщения безопасности, касающиеся FreeBSD
freebsd-small ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-small">http://lists.FreeBSD.org/mailman/listinfo/freebsd-small</a> )	Использование FreeBSD во встроенных приложениях
freebsd-smp ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-smp">http://lists.FreeBSD.org/mailman/listinfo/freebsd-smp</a> )	Обсуждение [не]симметричной мультипроцессорной архитектуры
freebsd-sparc64 ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-sparc64">http://lists.FreeBSD.org/mailman/listinfo/freebsd-sparc64</a> )	Портирование FreeBSD на системы, основанные на Sparc®
freebsd-standards ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-standards">http://lists.FreeBSD.org/mailman/listinfo/freebsd-standards</a> )	Соответствие FreeBSD стандартам C99 и POSIX
freebsd-threads ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-threads">http://lists.FreeBSD.org/mailman/listinfo/freebsd-threads</a> )	Потоки в FreeBSD
freebsd-testing ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-testing">http://lists.FreeBSD.org/mailman/listinfo/freebsd-testing</a> )	Тестирование производительности и стабильности FreeBSD
freebsd-tokenring ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-tokenring">http://lists.FreeBSD.org/mailman/listinfo/freebsd-tokenring</a> )	Поддержка Token Ring в FreeBSD
freebsd-usb ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-usb">http://lists.FreeBSD.org/mailman/listinfo/freebsd-usb</a> )	Обсуждение поддержки USB в FreeBSD
freebsd-vuxml ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-vuxml">http://lists.FreeBSD.org/mailman/listinfo/freebsd-vuxml</a> )	Обсуждение инфраструктуры VuXML
freebsd-x11 ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-x11">http://lists.FreeBSD.org/mailman/listinfo/freebsd-x11</a> )	Сопровождение и поддержка X11 в FreeBSD

*Ограниченные списки:* Следующие списки рассылки предназначены для более специализированной (и более официальной) аудитории и вероятно не могут заинтересовать широкую публику. Вероятно хорошей идеей будет сначала наладить общение в технических списках рассылки перед присоединением к ограниченным спискам, так вы сможете освоить этику общения.

Список	Назначение
freebsd-hubs ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-hubs">http://lists.FreeBSD.org/mailman/listinfo/freebsd-hubs</a> )	Люди, поддерживающие зеркала (поддержка инфраструктуры)
freebsd-user-groups ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-user-groups">http://lists.FreeBSD.org/mailman/listinfo/freebsd-user-groups</a> )	Координация групп пользователей
freebsd-vendors ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-vendors">http://lists.FreeBSD.org/mailman/listinfo/freebsd-vendors</a> )	Координация поставщиков перед релизом
freebsd-www ( <a href="http://lists.FreeBSD.org/mailman/listinfo/freebsd-www">http://lists.FreeBSD.org/mailman/listinfo/freebsd-www</a> )	Ответственные за <a href="http://www.FreeBSD.org">www.FreeBSD.org</a> ( <a href="http://www.FreeBSD.org/index.html">http://www.FreeBSD.org/index.html</a> )

*Дайджест рассылки:* Все вышеприведенные списки доступны в формате дайджеста. После подписки на рассылку, вы можете изменить параметры дайджеста в разделе настроек учетной записи.

*CVS рассылки:* Следующие рассылки предназначены для людей, заинтересованных в просмотре сообщений об изменении в различных областях дерева исходных текстов. Это списки *только для чтения* и вы не должны отправлять туда почту.

Рассылка	Область исходного текста	Описание области исходного текста
cvs-all ( <a href="http://lists.FreeBSD.org/mailman/listinfo/cvs-all">http://lists.FreeBSD.org/mailman/listinfo/cvs-all</a> )	/usr/ (CVSROOT   doc   ports   projects)	Все изменения в любой области дерева исходных текстов (надмножество других списков рассылки CVS)
cvs-doc ( <a href="http://lists.FreeBSD.org/mailman/listinfo/cvs-doc">http://lists.FreeBSD.org/mailman/listinfo/cvs-doc</a> )	/usr/ (doc   www)	Все изменения в дереве исходных текстов документации и веб сервера
cvs-ports ( <a href="http://lists.FreeBSD.org/mailman/listinfo/cvs-ports">http://lists.FreeBSD.org/mailman/listinfo/cvs-ports</a> )	/usr/ports	Все изменения в дереве портов
cvs-projects ( <a href="http://lists.FreeBSD.org/mailman/listinfo/cvs-projects">http://lists.FreeBSD.org/mailman/listinfo/cvs-projects</a> )	/usr/projects	Все изменения в дереве проектов
cvs-src ( <a href="http://lists.FreeBSD.org/mailman/listinfo/cvs-src">http://lists.FreeBSD.org/mailman/listinfo/cvs-src</a> )	/usr/src	Все изменения в дереве исходных текстов системы

## С.1.2. Как подписаться

Для подписки на рассылку, нажмите на название списка рассылки выше или воспользуйтесь ссылкой <http://lists.FreeBSD.org/mailman/listinfo> и нажмите на имя рассылки, которой вы заинтересовались.

Страница списка рассылки содержит все необходимые инструкции по подписке.

Для отправки сообщения в выбранный список рассылки, отправьте письмо в `<listname@FreeBSD.org>`. Это письмо будет разослано участникам рассылки по всему миру.

Для отписки от рассылки, нажмите на ссылку, находящуюся внизу каждого письма, отправляемого через список рассылки. Возможна также отписка путем отправки письма на `<listname-unsubscribe@FreeBSD.org>`.

Напоминаем, что обсуждение в технических списках рассылки должно оставаться в рамках технической темы. Если вас интересует только получение важных анонсов, мы предлагаем подписаться на рассылку с небольшим трафиком `freebsd-announce` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce>).

## С.1.3. Описание рассылок

Все списки рассылки FreeBSD имеют определенные основные правила, которых должен придерживаться каждый использующий их. Несоблюдение этих правил приведет к отправлению двух (2) предупреждений от FreeBSD Postmaster `<postmaster@FreeBSD.org>`, после которых, после третьего нарушения, подписчик будет удален из всех списков рассылки FreeBSD и дальнейшие его сообщения будут отфильтровываться. Мы сожалеем, что эти правила и меры вообще необходимы, но современный интернет это довольно суровая среда и многие его механизмы довольно слабы.

Основные правила:

- Тема любого сообщения должна соответствовать назначению списка рассылки, в который это сообщение отправляется. Например, если список рассылки посвящен техническим вопросам, сообщение должно быть техническим. Продолжающееся обсуждение вне темы, или флейм только понижают ценность рассылки для всех ее участников и поэтому не разрешаются. Для обсуждений вне какой-либо определенной темы необходимо использовать Список рассылки, посвященный неформальным беседам о FreeBSD (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-chat>), специально для этого предназначенный.
- Ни одно сообщение не должно отправляться более чем в 2 рассылки, отправка сообщения в 2 рассылки должна выполняться только при наличии простой и очевидной причины для дублирования сообщения. В большинстве рассылок подписчикам уже приходит много избыточного материала, и за исключением редких случаев (скажем, "-stable & -scsi"), на самом деле нет причины отправлять сообщение более чем в один список рассылки. Если сообщение отправлено вам так, что в поле Сс находятся несколько списков рассылки, необходимо урезать поле Сс перед отправкой ответа. *Именно вы отвечаете за собственные сообщения, независимо от того, кто был автором исходного письма.*
- Персональные нападки и профанация (в контексте аргументов) не разрешены, это относится и к пользователям, и к разработчикам. Грубые нарушения сетевой этики, такие как цитирование или пересылка личной переписки без специального на то разрешения, осуждаются но специальные меры в этом случае не принимаются. *Однако, существует несколько специальных случаев, когда такие*

письма не отвечают назначению списка рассылки и, следовательно, могут повлечь отправку предупреждения (или исключение из списка рассылки).

- Реклама не-FreeBSD продуктов или сервисов строго запрещена и исключение из списка рассылки последует незамедлительно, если станет очевидным, что это спам.

*Описания рассылок:*

freebsd-acpi (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-acpi>)

*Разработка ACPI и системы управления энергопотреблением*

freebsd-afs (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-afs>)

*Andrew File System*

Этот список предназначен для обсуждения портирования и использования AFS от CMU/Transarc

freebsd-announce (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-announce>)

*Важные события / вехи проекта*

Этот список рассылки предназначен для тех, кто интересуется только периодическими анонсами значительных событий FreeBSD. Сюда включаются анонсы снэпшотов и других релизов, а также новых возможностей FreeBSD. Рассылка может содержать призыв к добровольцам и т.п. Это строго модулируемый список рассылки с малым объемом трафика.

freebsd-arch (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-arch>)

*Обсуждение архитектуры и дизайна системы*

Эта рассылка предназначена для обсуждения архитектуры FreeBSD. Сообщения в основном строго технические. Примеры подходящих тем:

- Как изменить систему сборки для одновременной сборки нескольких по-разному настроенных систем.
- Что необходимо исправить в VFS для включения слоев Heidemann.
- Как необходимо изменить интерфейс драйверов устройств для использования одних и тех же драйверов на множестве шин и архитектур.
- Как написать сетевой драйвер.

freebsd-audit (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-audit>)

*Проект аудита исходных текстов*

Это список рассылки для проекта аудита исходных текстов FreeBSD. Хотя первоначально он предназначался для изменений, связанных с безопасностью, его назначение было расширено для пересмотра всех изменений кода.

В эту рассылку отправляется большой объем исправлений, и она вероятно не представляет интереса для обычного пользователя FreeBSD. Обсуждения безопасности, не относящиеся к определенному изменению в коде, ведутся в `freebsd-security`. Разработчикам предлагается отправлять изменения в этот список рассылки для просмотра, особенно если эти изменения затрагивают части кода, ошибки в которых могут повлечь нарушение целостности системы.

`freebsd-binup` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-binup>)

*Проект бинарного обновления FreeBSD*

Этот список предназначен для обсуждений системы бинарного обновления системы, или **binup**. В этой рассылке обсуждаются вопросы дизайна, детали реализации, исправления, сообщения об ошибках, сообщения о статусе, запросы на расширение функциональности, протоколы коммитов, и все, что относится к **binup**.

`freebsd-bugbusters` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugbusters>)

*Координация усилий по обработке сообщений о проблемах*

Назначение этой рассылки в координации и предоставлении места для обсуждения для лиц, обслуживающих базу данных сообщений о проблемах (`bugmeister`, `bugbusters`) и для всех сторон, интересующихся базой данных PR. Эта рассылка не предназначена для обсуждения отдельных проблем, исправлений или PR.

`freebsd-bugs` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-bugs>)

*Сообщения об ошибках*

Этот список рассылки предназначен для отправки сообщений об ошибках в FreeBSD. Когда это возможно, сообщения должны отправляться с использованием `send-pr(1)` или через WEB интерфейс (<http://www.FreeBSD.org/ru/send-pr.html>) к `send-pr`.

`freebsd-chat` (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-chat>)

*Не-технические темы, относящиеся к сообществу FreeBSD*

В эту рассылку входят все темы, не подходящие для других рассылок, с не-технической, социальной информацией. Она включает обсуждения на темы: кто пьет слишком много кофе, где варят лучшее пиво, кто варит пиво в своем подвале, и так далее. Нерегулярные анонсы важных событий (такие как будущие встречи, свадьбы, дни рождения, новая работа и т.д.) могут быть опубликованы в технических рассылках, но ответы должны отправляться в `-chat`.

`freebsd-core`

*Команда FreeBSD core*



Это внутренний список рассылки, используемый членами core. Сообщения в эту рассылку могут быть отправлены по серьезной, имеющей отношение к FreeBSD причине, которая требует рассмотрения на самом высоком уровне.

freebsd-current (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-current>)

*Обсуждения, касающиеся использования FreeBSD-CURRENT*

Это список рассылки для пользователей FreeBSD-CURRENT. Он включает предупреждения о новых возможностях, вносимых в -CURRENT, влияющих на пользователей, и инструкции относительно действий, которые должны быть предприняты для поддержки -CURRENT. Всякий, работающий с "CURRENT", должен подписаться на эту рассылку. Это технический список рассылки, все сообщения должны быть строго техническими.

freebsd-cvsweb (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-cvsweb>)

*FreeBSD CVSweb Project*

Технические обсуждения использования, разработки и поддержки FreeBSD-CVSweb.

freebsd-doc (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-doc>)

*Проект документирования*

Этот список рассылки предназначен для обсуждения вопросов и проектов, относящихся к созданию документации для FreeBSD. Члены этой рассылки все вместе обозначаются как "The FreeBSD Documentation Project". Это открытая рассылка; присоединяйтесь и участвуйте!

freebsd-fi rewire ([http://lists.FreeBSD.org/mailman/listinfo/freebsd-fi rewire](http://lists.FreeBSD.org/mailman/listinfo/freebsd-fi-rewire))

*FireWire (iLink, IEEE 1394)*

Это список рассылки, предназначенный для обсуждения дизайна и реализации подсистемы FireWire (также известной как IEEE 1394 или iLink) в FreeBSD. Соответствующие темы относятся к стандартам, устройствам шины и их протоколам, наборам плат/карт/чипов адаптера, а также архитектуре и реализации кода для их правильной поддержки.

freebsd-fs (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-fs>)

*Файловые системы*

Обсуждения, относящиеся к файловым системам FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-geom (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-geom>)

*GEOM*

Обсуждения, относящиеся к GEOM и связанным с GEOM реализациям. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-gnome (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-gnome>)

#### *GNOME*

Обсуждения, относящиеся к десктопу **GNOME** для системы FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-ipfw (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-ipfw>)

#### *IP брандмауэр*

Это форум для технических обсуждений, относящихся к редизайну кода IP брандмауэра в FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-ia64 (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-ia64>)

#### *Портирование FreeBSD на IA64*

Это технический список рассылки для тех, кто активно работает над портированием FreeBSD на платформу IA-64 от Intel, предназначенный для поднятия вопросов или обсуждения альтернативных решений. Те, кто интересуется обсуждаемыми проблемами, также приглашаются к участию в рассылке.

freebsd-isdn (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-isdn>)

#### *ISDN соединения*

Это список рассылки для обсуждения разработки поддержки ISDN для FreeBSD.

freebsd-java (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-java>)

#### *Разработка Java*

Этот список рассылки предназначен для обсуждения ключевых приложений Java для FreeBSD, а также портирования и поддержки JDK.

freebsd-jobs (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-jobs>)

#### *Предложение и поиск работы*

Это форум для публикации вакансий и резюме, относящихся к FreeBSD. Например, если вы ищете работу, относящуюся к FreeBSD, или у вас есть работа, связанная с FreeBSD, вы можете разместить соответствующую информацию именно здесь. Эта рассылка *не* предназначена для обсуждения

общих вопросов о приеме на работу, поскольку форумы на соответствующие темы уже существуют на других сайтах.

Имейте ввиду, что эта рассылка, как и другие рассылки на FreeBSD.org, распространяется по всему миру. Поэтому вам необходимо четко указать свое местоположение и область, с которой возможны телекоммуникации или помощь в перемещении.

Письма должны быть составлены только в открытых форматах — предпочтителен чистый текст, но Portable Document Format (PDF), HTML, и некоторые другие форматы могут быть прочитаны многими. Закрытые форматы, такие как Microsoft Word (.doc) будут отброшены сервером почтовой рассылки.

freebsd-kde (<http://freebsd.kde.org/mailman/listinfo/kde-freebsd>)

#### *KDE*

Обсуждения, относящиеся к **KDE** в системах FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-hackers (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-hackers>)

#### *Технические обсуждения*

Это форум для технических обсуждений, относящихся к FreeBSD. Это в основном технический список рассылки. Он предназначен для тех, кто активно работает над FreeBSD, и служит для поднятия вопросов или обсуждения альтернативных решений. Те, кто интересуется обсуждаемыми вопросами, также приглашаются к участию в обсуждении. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-hardware (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-hardware>)

#### *Общее обсуждение оборудования FreeBSD*

Общее обсуждение типов оборудования, на котором работает FreeBSD, различных проблем и предложений относительно того, какое оборудование можно покупать а какое нет.

freebsd-hubs (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-hubs>)

#### *Сайты зеркал*

Анонсы и обсуждения для поддерживающих зеркала FreeBSD.

freebsd-isp (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-isp>)

#### *Вопросы использования FreeBSD провайдерами*

Этот список рассылки предназначен для обсуждения тем, имеющих значение для провайдеров, использующих FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-newbies (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-newbies>)

*Обсуждение деятельности новых пользователей*

Эта рассылка охватывает всю деятельность новых пользователей, которая не обсуждается где-то еще. Сюда включаются: независимое обучение и техника решения проблем, поиск и использование ресурсов и запрос помощи где-то еще, как использовать списки рассылки и какие из них использовать, разговоры на общие темы, обсуждение ошибок, хвастовство, обмен идеями, истории, моральная (но не техническая) поддержка, активное участие в сообществе FreeBSD. Мы сообщаем о своих проблемах и отправляем запросы на поддержку в freebsd-questions, а freebsd-newbies используем для встречи с теми, кто делает то же, что и мы, будучи новыми пользователями.

freebsd-openoffice (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-openoffice>)

*OpenOffice.org*

Обсуждения, относящиеся к портированию и поддержке **OpenOffice.org** и **StarOffice**.

freebsd-performance (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-performance>)

*Обсуждения оптимизации или повышения скорости FreeBSD*

Этот список рассылки существует как место для обсуждения тем, имеющих отношение к производительности FreeBSD, хакерами, администраторами, и/или заинтересованными сторонами. Приемлемые темы включают обсуждения установок FreeBSD, которые находятся под высокой нагрузкой и сталкиваются с проблемами производительности, или преодоление ограничений FreeBSD. Заинтересованным сторонам, собирающимся работать над улучшением производительности FreeBSD, настоятельно рекомендуется подписаться на эту рассылку. Это техническая рассылка, идеально подходящая для пользователей, хакеров или администраторов, заинтересованных в скорости, стабильности и расширяемости FreeBSD. Это не рассылка вопросов-и-ответов, заменяющая чтение документации, а место, где можно внести свой вклад или получить информацию по еще незатронутой теме, связанной с производительностью.

freebsd-pf (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-pf>)

*Обсуждение брандмауэра на базе packet filter*

Обсуждения, касающиеся работы пакетного фильтра pf под FreeBSD. Допускаются как вопросы пользователей, так и технические дискуссии. Помимо этого, в данном списке уместно обсуждать инфраструктуру ALTQ QoS.

freebsd-platforms (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-platforms>)

*Портирование на не-Intel платформы*

Кросс-платформенные вопросы FreeBSD, общее обсуждение и предложения для не-Intel портов FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-policy (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-policy>)

*Правила core team*

Это рассылка с малым количеством сообщений, только для чтения, предназначенная для публикации решений FreeBSD Core Team.

freebsd-ports (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports>)

*Обсуждения "ports"*

Обсуждения, относящиеся к "коллекции портов" FreeBSD, ( /usr/ports), инфраструктуры портов и общих усилий по координации портов. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-ports-bugs (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-ports-bugs>)

*Обсуждение проблем в "ports"*

Обсуждения, относящиеся к сообщениям о проблемах для "коллекции портов" FreeBSD (/usr/ports), предлагаемых портов, или изменений к портам. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-questions (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-questions>)

*Вопросы пользователей*

Это список рассылки по вопросам о FreeBSD. Вы не должны отправлять вопросы "как сделать" в технические рассылки, если только не уверены, что ваш вопрос чисто технический.

freebsd-scsi (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-scsi>)

*Подсистема SCSI*

Это список рассылки для тех, кто работает над подсистемой SCSI для FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-security (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-security>)

*Вопросы безопасности*

Вопросы безопасности FreeBSD (DES, Kerberos, известные проблемы безопасности и исправления, и т.п.). Это технический список рассылки, предназначенный только для технических обсуждений. Обратите внимание, что это не рассылка вопросов-и-ответов, но дополнения в FAQ (И вопрос И ответ) приветствуются.

freebsd-security-notifications (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-security-notifications>)

*Уведомления безопасности*

Уведомления о проблемах безопасности FreeBSD и исправления. Эта рассылка не предназначена для обсуждений. Для обсуждения предназначена рассылка FreeBSD-security.

freebsd-small (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-small>)

*Использование FreeBSD во встроенных приложениях*

В этой рассылке обсуждаются темы, связанные с необычно малыми и встроенными установками FreeBSD. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-stable (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-stable>)

*Обсуждения, касающиеся использования FreeBSD-STABLE*

Этот список рассылки предназначен для пользователей FreeBSD-STABLE. Он включает предупреждения о новых возможностях, добавляемых в -STABLE, и влияющих на пользователей, и инструкции по действиям, которые необходимы для поддержки системы в состоянии -STABLE. Всякий, использующий "STABLE", должен подписаться на эту рассылку. Это технический список рассылки, предназначенный только для технических обсуждений.

freebsd-standards (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-standards>)

*Соответствие C99 и POSIX*

Это форум для технических обсуждений, относящихся к соответствию FreeBSD стандартам C99 и POSIX.

freebsd-usb (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-usb>)

*Обсуждение поддержки USB в FreeBSD*

Это форум для технических обсуждений, относящихся к поддержке в FreeBSD устройств с интерфейсом USB.

freebsd-user-groups (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-user-groups>)

*Список координации групп пользователей*

Этот список рассылки предназначен для обсуждения вопросов координаторами каждой группы пользователей и назначенным членом Core Team. Обсуждения в этой рассылке ограничены темой встреч и координацией проектов, относящихся к группам пользователей.

freebsd-vendors (<http://lists.FreeBSD.org/mailman/listinfo/freebsd-vendors>)

### *Поставщики*

Обсуждения, относящиеся к координации между FreeBSD Project и поставщиками программного и аппаратного обеспечения для FreeBSD.

## **С.1.4. Фильтрация списков рассылки**

Списки рассылки FreeBSD фильтруются различными способами для предотвращения распространения спама, вирусов, и другой нежелательной почты. Действия по фильтрации, описанные в этом разделе, не включают всех используемых для фильтрации списков рассылки проекта действий.

Только определенные типы вложений разрешены в списках рассылки. Все вложения с типами MIME содержимого, не входящие в список ниже, будут вырезаться перед тем, как письмо будет отправлено в список рассылки.

- application/octet-stream
- application/pdf
- application/pgp-signature
- application/x-pkcs7-signature
- message/rfc822
- multipart/alternative
- multipart/related
- multipart/signed
- text/html
- text/plain
- text/x-diff
- text/x-patch

**Замечание:** Некоторые из списков рассылки могут пропускать вложения других типов MIME, но список выше применим к большинству рассылок.

Если письмо содержит как HTML, так и только текстовую версию, версия HTML будет удалена. Если письмо содержит только HTML версию, она будет конвертирована в простой текст.

## **С.2. Новостные группы Usenet**

В дополнение к двум относящимся к FreeBSD группам новостей, существуют множество других, где обсуждается FreeBSD или куда помещается другая информация, относящаяся к пользователям

FreeBSD. Архивы с поиском по ключевому слову ([http://minnie.tuhs.org/BSD-info/bsdnews\\_search.html](http://minnie.tuhs.org/BSD-info/bsdnews_search.html)) доступны для некоторых из этих новостных групп благодаря Warren Toomey <wkt@cs.adfa.edu.au>.

### C.2.1. Относящиеся к **BSD** новостные группы

- comp.unix.bsd.freebsd.announce (news:comp.unix.bsd.freebsd.announce)
- comp.unix.bsd.freebsd.misc (news:comp.unix.bsd.freebsd.misc)
- de.comp.os.unix.bsd (news:de.comp.os.unix.bsd) (German)
- fr.comp.os.bsd (news:fr.comp.os.bsd) (French)
- it.comp.os.freebsd (news:it.comp.os.freebsd) (Italian)

### C.2.2. Другие интересные **UNIX** новостные группы

- comp.unix (news:comp.unix)
- comp.unix.questions (news:comp.unix.questions)
- comp.unix.admin (news:comp.unix.admin)
- comp.unix.programmer (news:comp.unix.programmer)
- comp.unix.shell (news:comp.unix.shell)
- comp.unix.user-friendly (news:comp.unix.user-friendly)
- comp.security.unix (news:comp.security.unix)
- comp.sources.unix (news:comp.sources.unix)
- comp.unix.advocacy (news:comp.unix.advocacy)
- comp.unix.misc (news:comp.unix.misc)
- comp.bugs.4bsd (news:comp.bugs.4bsd)
- comp.bugs.4bsd.ucb-fi xes (news:comp.bugs.4bsd.ucb-fi xes)
- comp.unix.bsd (news:comp.unix.bsd)

### C.2.3. X Window System

- comp.windows.x.i386unix (news:comp.windows.x.i386unix)
- comp.windows.x (news:comp.windows.x)
- comp.windows.x.apps (news:comp.windows.x.apps)
- comp.windows.x.announce (news:comp.windows.x.announce)
- comp.windows.x.intrinsics (news:comp.windows.x.intrinsics)
- comp.windows.x.motif (news:comp.windows.x.motif)



- comp.windows.x.pex (news:comp.windows.x.pex)
- comp.emulators.ms-windows.wine (news:comp.emulators.ms-windows.wine)

### С.3. Серверы World Wide Web

Центральные серверы, Costa Rica, Indonesia, Kyrgyzstan, Thailand, Австралия, Австрия, Аргентина, Бельгия, Бразилия, Великобритания, Венгрия, Германия, Гонконг, Греция, Дания, Ирландия, Исландия, Испания, Италия, Канада, Китай, Корея, Кувейт, Латвия, Литва, Нидерланды, Новая Зеландия, Норвегия, Польша, Португалия, Россия, Румыния, США, Сан Марино, Сингапур, Словацкая Республика, Словения, Тайвань, Турция, Украина, Филиппины, Финляндия, Франция, Чехия, Швейцария, Швеция, Эстония, Южная Африка, Япония.

(as of 2004/10/26 16:33:28 UTC)

- 

Центральные серверы

- <http://www.FreeBSD.org/>

- 

Costa Rica

- <http://www1.cr.FreeBSD.org/>

- 

Indonesia

- <http://www.id.FreeBSD.org/>

- 

Kyrgyzstan

- <http://www.kg.FreeBSD.org/>

- 

Thailand

- <http://www.th.FreeBSD.org/>

-

Австралия

- <http://www.au.FreeBSD.org/>
- <http://www2.au.FreeBSD.org/>

•

Австрия

- <http://www.at.FreeBSD.org/>
- <http://www2.at.FreeBSD.org/> (IPv6)

•

Аргентина

- <http://www.ar.FreeBSD.org/>

•

Бельгия

- <http://freebsd.unixtech.be/>

•

Бразилия

- <http://www.br.FreeBSD.org/>
- <http://www2.br.FreeBSD.org/www.freebsd.org/>
- <http://www3.br.FreeBSD.org/>

•

Великобритания

- <http://www.uk.FreeBSD.org/>
- <http://www2.uk.FreeBSD.org/>
- <http://www3.uk.FreeBSD.org/>
- <http://www4.uk.FreeBSD.org/>
- <http://www1.uk.FreeBSD.org/> (IPv6)

- 

#### Венгрия

- <http://www.hu.FreeBSD.org/>
- <http://www2.hu.FreeBSD.org/>

- 

#### Германия

- <http://www.de.FreeBSD.org/>
- <http://www1.de.FreeBSD.org/>
- <http://www2.de.FreeBSD.org/> (IPv6)

- 

#### Гонконг

- <http://www.hk.FreeBSD.org/>

- 

#### Греция

- <http://www.gr.FreeBSD.org/>
- <http://www.FreeBSD.gr/>

- 

#### Дания

- <http://www.dk.FreeBSD.org/> (IPv6)
- <http://www3.dk.FreeBSD.org/>

- 

#### Ирландия

- <http://www.ie.FreeBSD.org/>
- <http://www2.ie.FreeBSD.org/>

-

Исландия

- <http://www.is.FreeBSD.org/>

•

Испания

- <http://www.es.FreeBSD.org/>
- <http://www2.es.FreeBSD.org/>
- <http://www3.es.FreeBSD.org/>

•

Италия

- <http://www.it.FreeBSD.org/>
- <http://www.gufi.org/mirrors/www.freebsd.org/data/>

•

Канада

- <http://www.ca.FreeBSD.org/>
- <http://www2.ca.FreeBSD.org/>

•

Китай

- <http://www.cn.FreeBSD.org/>

•

Корея

- <http://www.kr.FreeBSD.org/>
- <http://www2.kr.FreeBSD.org/>

•

Кувейт

- <http://www.kw.FreeBSD.org/>

- Латвия
  - <http://www.lv.FreeBSD.org/>
- Литва
  - <http://www.lt.FreeBSD.org/>
- Нидерланды
  - <http://www.nl.FreeBSD.org/>
  - <http://www2.nl.FreeBSD.org/>
- Новая Зеландия
  - <http://www.nz.FreeBSD.org/>
- Норвегия
  - <http://www.no.FreeBSD.org/>
  - <http://www2.no.FreeBSD.org/> (IPv6)
- Польша
  - <http://www.pl.FreeBSD.org/>
  - <http://www2.pl.FreeBSD.org/>
- Португалия
  - <http://www.pt.FreeBSD.org/>

- <http://www1.pt.FreeBSD.org/>
- <http://www4.pt.FreeBSD.org/>
- <http://www5.pt.FreeBSD.org/>

•

#### Россия

- <http://www.ru.FreeBSD.org/>
- <http://www2.ru.FreeBSD.org/>
- <http://www3.ru.FreeBSD.org/>
- <http://www4.ru.FreeBSD.org/>

•

#### Румыния

- <http://www.ro.FreeBSD.org/>
- <http://www1.ro.FreeBSD.org/>
- <http://www2.ro.FreeBSD.org/>
- <http://www3.ro.FreeBSD.org/>

•

#### США

- <http://www2.us.FreeBSD.org/>
- <http://www4.us.FreeBSD.org/> (IPv6)
- <http://www5.us.FreeBSD.org/> (IPv6)

•

#### Сан Марино

- <http://www.sm.FreeBSD.org/>

•

#### Сингапур

- <http://www2.sg.FreeBSD.org/>

•

Словацкая Республика

- <http://www.sk.FreeBSD.org/>
- <http://www2.sk.FreeBSD.org/>

•

Словения

- <http://www.si.FreeBSD.org/>
- <http://www2.si.FreeBSD.org/>

•

Тайвань

- <http://www.tw.FreeBSD.org/>
- <http://www2.tw.FreeBSD.org/>
- <http://www3.tw.FreeBSD.org/>
- <http://www4.tw.FreeBSD.org/>

•

Турция

- <http://www.tr.FreeBSD.org/>
- <http://www2.tr.FreeBSD.org/>
- <http://www.enderunix.org/freebsd/>

•

Украина

- <http://www.ua.FreeBSD.org/>
- <http://www2.ua.FreeBSD.org/>
- <http://www5.ua.FreeBSD.org/>
- <http://www4.ua.FreeBSD.org/>

•

Филиппины

- <http://www.FreeBSD.org.ph/>

•

Финляндия

- <http://www.fi.FreeBSD.org/>
- <http://www2.fi.FreeBSD.org/>

•

Франция

- <http://www.fr.FreeBSD.org/>

•

Чехия

- <http://www.cz.FreeBSD.org/>

•

Швейцария

- <http://www.ch.FreeBSD.org/>
- <http://www2.ch.FreeBSD.org/>

•

Швеция

- <http://www.se.FreeBSD.org/>
- <http://www2.se.FreeBSD.org/>

•

Эстония

- <http://www.ee.FreeBSD.org/>

•

Южная Африка



- <http://www.za.FreeBSD.org/>
- <http://www2.za.FreeBSD.org/>

- 

Япония

- <http://www.jp.FreeBSD.org/www.FreeBSD.org/> (IPv6)

## С.4. Адреса Email

Следующие группы пользователей предоставляют для своих участников почтовые адреса. Приведенные в списке администраторы оставляют за собой право удалить адреса при любом злоупотреблении.

Домен	Возможности	Группа пользователей	Администратор
ukug.uk.FreeBSD.org	Только пересылка	<freebsd-users@uk.FreeBSD.org>	Lee Johnston <lee@uk.FreeBSD.org>

## С.5. Shell доступ

Следующие группы пользователей предоставляют shell доступ для тех, кто активно поддерживает проект FreeBSD. Приведенные в списке администраторы оставляют за собой право закрыть учетную запись при любом злоупотреблении.

Хост	Доступ	Возможности	Администратор
storm.uk.FreeBSD.org	Только SSH	CVS только для чтения, место для личной странички, почта	Brian Somers <brian@FreeBSD.org>
dogma.freebsd-uk.eu.org	Telnet/FTP/SSH	Email, место для сайта, анонимный FTP	Lee Johnston <lee@uk.FreeBSD.org>

# Приложение D. Ключи PGP

В случае если Вам необходимо проверить подпись или послать зашифрованное электронное письмо одному из офицеров безопасности или члену core группы FreeBSD ряд PGP ключей представлены здесь для Вашего удобства. Полный список ключей пользователей FreeBSD.org доступен для скачивания с <http://www.FreeBSD.org/doc/pgpkeyring.txt>.

## D.1. Офицеры

### D.1.1. Группа Офицеров Безопасности <security-officer@FreeBSD.org>

```
pub 1024D/CA6CDFB2 2002-08-27 FreeBSD Security Officer <security-officer@FreeBSD.org>
   Key fingerprint = C374 0FC5 69A6 FBB1 4AED B131 15D6 8804 CA6C DFB2
sub 2048g/A3071809 2002-08-27
```

### D.1.2. Секретарь CORE <core-secretary@FreeBSD.org>

```
pub 1024R/FF8AE305 2002-01-08 core-secretary@FreeBSD.org
   Key fingerprint = CE EF 8A 48 70 00 B5 A9 55 69 DE 87 E3 9A E1 CD
```

## D.2. Члены Core

### D.2.1. Warner Losh <imp@FreeBSD.org>

```
pub 1024R/F6A8F561 1996-01-10 M. Warner Losh <imp@village.org>
   Key fingerprint = D4 31 FD B9 F7 90 17 E8 37 C5 E7 7F CF A6 C1 B9
```

### D.2.2. John Baldwin <jhb@FreeBSD.org>

```
pub 1024R/C10A874D 1999-01-13 John Baldwin <jbaldwin@weather.com>
   Key fingerprint = 43 33 1D 37 72 B1 EF 5B 9B 5F 39 F8 BD C1 7C B5
uid                               John Baldwin <john@baldwin.cx>
uid                               John Baldwin <jhb@FreeBSD.org>
uid                               John Baldwin <jobaldwi@vt.edu>
```

### D.2.3. Jun Kuriyama <kuriyama@FreeBSD.org>

```
pub 1024D/FE3B59CD 1998-11-23 Jun Kuriyama <kuriyama@imgsrc.co.jp>
   Key fingerprint = 5219 55CE AC84 C296 3A3B B076 EE3C 4DBB FE3B 59CD
uid                               Jun Kuriyama <kuriyama@FreeBSD.org>
```

```
uid                               Jun Kuriyama <kuriyama@jp.FreeBSD.org>
sub  2048g/1CF20D27 1998-11-23
```

#### D.2.4. Murray Stokely <murray@FreeBSD.org>

```
pub  1024D/0E451F7D 2001-02-12 Murray Stokely <murray@freebsd.org>
    Key fingerprint = E2CA 411D DD44 53FD BB4B  3CB5 B4D7 10A2 0E45 1F7D
sub  1024g/965A770C 2001-02-12
```

#### D.2.5. Peter Wemm <peter@FreeBSD.org>

```
pub  1024D/7277717F 2003-12-14 Peter Wemm <peter@wemm.org>
    Key fingerprint = 622B 2282 E92B 3BAB 57D1  A417 1512 AE52 7277 717F
uid                               Peter Wemm <peter@FreeBSD.ORG>
sub  1024g/8B40D9D1 2003-12-14
pub  1024R/D89CE319 1995-04-02 Peter Wemm <peter@netplex.com.au>
    Key fingerprint = 47 05 04 CA 4C EE F8 93  F6 DB 02 92 6D F5 58 8A
uid                               Peter Wemm <peter@perth.dialix.oz.au>
uid                               Peter Wemm <peter@haywire.dialix.com>
```

#### D.2.6. Wes Peters <wes@FreeBSD.org>

```
pub  1024D/AD10C6C4 2000-10-19 Wes Peters <wes@freebsd.org>
    Key fingerprint = 76C4 753C 83FF D982 C57D  3A2A 7387 E292 AD10 C6C4
uid                               Wes Peters <wes@softweyr.com>
sub  1024g/237A5EF9 2000-10-19
```

### D.3. Разработчики

#### D.3.1. Will Andrews <will@FreeBSD.org>

```
pub  1024D/F81672C5 2000-05-22 Will Andrews (Key for official matters) <will@FreeBSD.org>
    Key fingerprint = 661F BBF7 9F5D 3D02 C862  5F6C 178E E274 F816 72C5
uid                               Will Andrews <will@physics.purdue.edu>
uid                               Will Andrews <will@puck.firepipe.net>
uid                               Will Andrews <will@c-60.org>
uid                               Will Andrews <will@csociety.org>
uid                               Will Andrews <will@csociety.ecn.purdue.edu>
uid                               Will Andrews <will@telperion.openpackages.org>
sub  1024g/55472804 2000-05-22
```

**D.3.2. Mathieu Arnold <mat@FreeBSD.org>**

```

pub 1024D/6CFFBE67 2003-07-21 Mathieu Arnold <mat@FreeBSD.org>
   Key fingerprint = 1A5F 2DBB 56F5 817E 8AFF 51B2 A986 29CC 6CFF BE67
uid                               Mathieu Arnold <m@absolight.net>
uid                               Mathieu Arnold <arn-mat@micronet.fr>
uid                               Mathieu Arnold <mat@mat.cc>
uid                               Mathieu Arnold <arn_mat@club-internet.fr>
uid                               Mathieu Arnold <mat@cpan.org>
uid                               Mathieu Arnold <mathieu.arnold@t-online.fr>
sub 1024g/830CAC4B 2003-07-21 [expires: 2005-01-21]
sub 2048R/A498469E 2004-01-22
sub 2048R/09EB7735 2004-01-22

```

**D.3.3. Satoshi Asami <asami@FreeBSD.org>**

```

pub 1024R/1E08D889 1997-07-23 Satoshi Asami <asami@cs.berkeley.edu>
   Key fingerprint = EB 3C 68 9E FB 6C EB 3F DB 2E 0F 10 8F CE 79 CA
uid                               Satoshi Asami <asami@FreeBSD.ORG>

```

**D.3.4. Doug Barton <DougB@FreeBSD.org>**

```

pub 1024D/D5B2F0FB 2003-01-16 Doug Barton <DougB@DougBarton.net>
   Key fingerprint = 9DD1 E44C 8660 ADA6 580F 83B6 C886 A42B D5B2 F0FB
uid                               Doug Barton <DougB@FreeBSD.org>
sub 4096g/2DBB3F89 2003-01-16

```

**D.3.5. Anton Berezin <tobez@FreeBSD.org>**

```

pub 1024D/7A7BA3C0 2000-05-25 Anton Berezin <tobez@catpipe.net>
   Key fingerprint = CDD8 560C 174B D8E5 0323 83CE 22CA 584C 7A7B A3C0
uid                               Anton Berezin <tobez@tobez.org>
uid                               Anton Berezin <tobez@FreeBSD.org>
sub 1024g/ADC71E87 2000-05-25

```

**D.3.6. Martin Blapp <mbr@FreeBSD.org>**

```

pub 1024D/D300551E 2001-12-20 Martin Blapp <mb@imp.ch>
   Key fingerprint = B434 53FC C87C FE7B 0A18 B84C 8686 EF22 D300 551E
sub 1024g/998281C8 2001-12-20

```

**D.3.7. Hartmut Brandt <harti@FreeBSD.org>**

```
pub 1024D/5920099F 2003-01-29 Hartmut Brandt <brandt@fokus.fraunhofer.de>
   Key fingerprint = F60D 09A0 76B7 31EE 794B BB91 082F 291D 5920 099F
uid                               Hartmut Brandt <harti@freebsd.org>
sub 1024g/21D30205 2003-01-29
```

**D.3.8. Oliver Braun <obraun@FreeBSD.org>**

```
pub 1024D/EF25B1BA 2001-05-06 Oliver Braun <obraun@unsane.org>
   Key fingerprint = 6A3B 042A 732E 17E4 B6E7 3EAF C0B1 6B7D EF25 B1BA
uid                               Oliver Braun <obraun@obraun.net>
uid                               Oliver Braun <obraun@freebsd.org>
uid                               Oliver Braun <obraun@haskell.org>
sub 1024g/09D28582 2001-05-06
```

**D.3.9. Jonathan M. Bresler <jmb@FreeBSD.org>**

```
pub 1024R/97E638DD 1996-06-05 Jonathan M. Bresler <jmb@Bresler.org>
   Key fingerprint = 31 57 41 56 06 C1 40 13 C5 1C E3 E5 DC 62 0E FB
uid                               Jonathan M. Bresler <jmb@FreeBSD.ORG>
uid                               Jonathan M. Bresler
uid                               Jonathan M. Bresler <Jonathan.Bresler@USi.net>
uid                               Jonathan M. Bresler <jmb@Frb.GOV>
```

**D.3.10. Christian Brueffer <brueffer@FreeBSD.org>**

```
pub 1024D/A0ED982D 2002-10-14 Christian Brueffer <chris@unixpages.org>
   Key fingerprint = A5C8 2099 19FF AACA F41B B29B 6C76 178C A0ED 982D
uid                               Christian Brueffer <brueffer@hitnet.rwth-aachen.de>
uid                               Christian Brueffer <brueffer@FreeBSD.org>
sub 4096g/1DCC100F 2002-10-14
```

**D.3.11. Markus Brueffer <markus@FreeBSD.org>**

```
pub 1024D/78F8A8D4 2002-10-21 Markus Brueffer <brueffer@phoenix-systems.de>
   Key fingerprint = 3F9B EBE8 F290 E5CC 1447 8760 D48D 1072 78F8 A8D4
uid                               Markus Brueffer <buff@hitnet.rwth-aachen.de>
uid                               Markus Brueffer <mbrueffer@mi.rwth-aachen.de>
uid                               Markus Brueffer <markus@FreeBSD.org>
uid                               Markus Brueffer <markus@brueffer.de>
sub 4096g/B7E5C7B6 2002-10-21
```

**D.3.12. Wilko Bulte <wilko@FreeBSD.org>**

```
pub 1024R/EADEE409 1997-06-16 Wilko Bulte <wilko@tcja.nl>
    Key fingerprint = CF F5 30 59 20 AE 17 69 98 F6 58 F0 4F F0 EB 3E
uid                               Wilko Bulte <wilko@freebsd.org>
uid                               "Wilko Bulte <wilko@freebie.xs4all.nl>"
```

**D.3.13. Jonathan Chen <jon@FreeBSD.org>**

```
pub 1024D/2539468B 1999-10-11 Jonathan Chen <jon@spock.org>
    Key fingerprint = EE31 CDA1 A105 C8C9 5365 3DB5 C2FC 86AA 2539 468B
uid                               Jonathan Chen <jon@freebsd.org>
uid                               Jonathan Chen <chenj@rpi.edu>
uid                               Jonathan Chen <spock@acm.rpi.edu>
uid                               Jonathan Chen <jon@cs.rpi.edu>
sub 3072g/B81EF1DB 1999-10-11
```

**D.3.14. Luoqi Chen <luoqi@FreeBSD.org>**

```
pub 1024D/2926F3BE 2002-02-22 Luoqi Chen <luoqi@FreeBSD.org>
    Key fingerprint = B470 A815 5917 D9F4 37F3 CE2A 4D75 3BD1 2926 F3BE
uid                               Luoqi Chen <luoqi@bricore.com>
uid                               Luoqi Chen <lchen@onetta.com>
sub 1024g/5446EB72 2002-02-22
```

**D.3.15. Andrey A. Chernov <ache@FreeBSD.org>**

```
pub 1024R/D99D08EB 1992-08-20 Andrey A. Chernov <ache@FreeBSD.org>
    Key fingerprint = 33 03 9F 48 33 7B 4A 15 63 48 88 0A C4 97 FD 49
uid                               Andrey A. Chernov <ache@nagual.pp.ru>
```

**D.3.16. Sean Chittenden <seanc@FreeBSD.org>**

```
pub 1024D/EE278A28 2004-02-08 Sean Chittenden <sean@chittenden.org>
    Key fingerprint = E41F F441 7E91 6CBA 1844 65CF B939 3C78 EE27 8A28
sub 2048g/55321853 2004-02-08
```

**D.3.17. Junho CHOI <cjh@FreeBSD.org>**

```
pub 1024D/E60260F5 2002-10-14 CHOI Junho (Work) <cjh@wdb.co.kr>
    Key fingerprint = 1369 7374 A45F F41A F3C0 07E3 4A01 C020 E602 60F5
uid                               CHOI Junho (Personal) <cjh@kr.FreeBSD.org>
uid                               CHOI Junho (FreeBSD) <cjh@FreeBSD.org>
sub 1024g/04A4FDD8 2002-10-14
```

**D.3.18. Crist J. Clark <cjc@FreeBSD.org>**

```
pub 1024D/FE886AD3 2002-01-25 Crist J. Clark <cjclark@jhu.edu>
   Key fingerprint = F04E CCD7 3834 72C2 707F 0A8F 259F 8F4B FE88 6AD3
uid                               Crist J. Clark <cjclark@alum.mit.edu>
uid                               Crist J. Clark <cjc@freebsd.org>
sub 1024g/9B6BAB99 2002-01-25
```

**D.3.19. Joe Marcus Clarke <marcus@FreeBSD.org>**

```
pub 1024D/FE14CF87 2002-03-04 Joe Marcus Clarke (FreeBSD committer address) <marcus@FreeBSD.org>
   Key fingerprint = CC89 6407 73CC 0286 28E4 AFB9 6F68 8F8A FE14 CF87
uid                               Joe Marcus Clarke <marcus@marcuscom.com>
sub 1024g/B9ACE4D2 2002-03-04
```

**D.3.20. Nik Clayton <nik@FreeBSD.org>**

```
pub 1024D/2C37E375 2000-11-09 Nik Clayton <nik@freebsd.org>
   Key fingerprint = 15B8 3FFC DDB4 34B0 AA5F 94B7 93A8 0764 2C37 E375
uid                               Nik Clayton <nik@slashdot.org>
uid                               Nik Clayton <nik@crf-consulting.co.uk>
uid                               Nik Clayton <nik@ngo.org.uk>
uid                               Nik Clayton <nik@bsd.i.com>
sub 1024g/769E298A 2000-11-09
```

**D.3.21. Ceri Davies <ceri@FreeBSD.org>**

```
pub 1024D/34B7245F 2002-03-08 Ceri Davies <ceri@submonkey.net>
   Key fingerprint = 9C88 EB05 A908 1058 A4AE 9959 A1C7 DCC1 34B7 245F
uid                               Ceri Davies <setantae@submonkey.net>
uid                               Ceri Davies <ceri@FreeBSD.org>
sub 1024g/0C482CBC 2002-03-08
```

**D.3.22. Brooks Davis <brooks@FreeBSD.org>**

```
pub 1024D/F2381AD4 2001-02-10 Brooks Davis (The Aerospace Corporation) <brooks@aero.org>
   Key fingerprint = 655D 519C 26A7 82E7 2529 9BF0 5D8E 8BE9 F238 1AD4
uid                               Brooks Davis <brooks@one-eyed-alien.net>
uid                               Brooks Davis <brooks@FreeBSD.org>
uid                               Brooks Davis <brooks@aero.org>
sub 2048g/CFDACA7A 2003-01-25 [expires: 2008-01-24]
sub 1024g/42921194 2001-02-10 [expires: 2009-02-08]
```

**D.3.23. Pawel Jakub Dawidek <pjd@FreeBSD.org>**

```

pub 1024D/B1293F34 2004-02-02 Pawel Jakub Dawidek <Pawel@Dawidek.net>
    Key fingerprint = A3A3 5B4D 9CF9 2312 0783 1B1D 168A EF5D B129 3F34
uid                                     Pawel Jakub Dawidek <pjd@FreeBSD.org>
uid                                     Pawel Jakub Dawidek <pjd@FreeBSD.pl>
sub 2048g/3EEC50A7 2004-02-02 [expires: 2006-02-01]

```

**D.3.24. Brian S. Dean <bsd@FreeBSD.org>**

```

pub 1024D/723BDEE9 2002-01-23 Brian S. Dean <bsd@FreeBSD.org>
    Key fingerprint = EF49 7ABE 47ED 91B3 FC3D 7EA5 4D90 2FF7 723B DEE9
sub 1024g/4B02F876 2002-01-23

```

**D.3.25. Dima Dorfman <dd@FreeBSD.org>**

```

pub 1024D/69FAE582 2001-09-04 Dima Dorfman <dima@trit.org>
    Key fingerprint = B340 8338 7DA3 4D61 7632 098E 0730 055B 69FA E582
uid                                     Dima Dorfman <dima@unixfreak.org>
uid                                     Dima Dorfman <dd@freebsd.org>
sub 1024g/A51DD1C0 2001-09-04 [expires: 2003-09-04]
sub 2048g/65AF3B89 2003-08-19 [expires: 2005-08-18]

```

**D.3.26. Alex Dupre <ale@FreeBSD.org>**

```

pub 1024D/CE5F554D 1999-06-27 Alex Dupre <sysadmin@alexdupre.com>
    Key fingerprint = DE23 02EA 5927 D5A9 D793 2BA2 8115 E9D8 CE5F 554D
uid                                     Alex Dupre <ale@FreeBSD.org>
uid                                     [jpeg image of size 5544]
uid                                     Alex Dupre <ICQ:5431856>
sub 2048g/FD5E2D21 1999-06-27

```

**D.3.27. Peter Edwards <peadar@FreeBSD.org>**

```

pub 1024D/D80B4B3F 2004-03-01 Peter Edwards <peadar@FreeBSD.org>
    Key fingerprint = 7A8A 9756 903E BEF2 4D9E 3C94 EE52 52F7 D80B 4B3F
uid                                     Peter Edwards <pmedwards@eircom.net>

```

**D.3.28. Josef El-Rayes <josef@FreeBSD.org>**

```

pub 2048R/A79DB53C 2004-01-04 Josef El-Rayes <josef@FreeBSD.org>
    Key fingerprint = 58EB F5B7 2AB9 37FE 33C8 716B 59C5 22D9 A79D B53C
uid                                     Josef El-Rayes <josef@daemon.li>

```



**D.3.29. Udo Erdelhoff <ue@FreeBSD.org>**

```
pub 1024R/E74FA871 1994-07-19 Udo Erdelhoff <uer@de.uu.net>
    Key fingerprint = 8C B1 80 CA 2C 52 73 81 FB A7 B4 03 C5 32 C8 67
uid                                Udo Erdelhoff <ue@nathan.ruhr.de>
uid                                Udo Erdelhoff <ue@freebsd.org>
uid                                Udo Erdelhoff <uerdelho@eu.uu.net>
uid                                Udo Erdelhoff <uerdelho@uu.net>
```

**D.3.30. Ruslan Ermilov <ru@FreeBSD.org>**

```
pub 1024D/996E145E 2004-06-02 Ruslan Ermilov (FreeBSD) <ru@FreeBSD.org>
    Key fingerprint = 274E D201 71ED 11F6 9CCB 0194 A917 E9CC 996E 145E
uid                                Ruslan Ermilov (FreeBSD Ukraine) <ru@FreeBSD.org.ua>
uid                                Ruslan Ermilov (IPNet) <ru@ip.net.ua>
sub 1024g/557E3390 2004-06-02 [expires: 2007-06-02]
```

**D.3.31. Lukas Ertl <le@FreeBSD.org>**

```
pub 1024D/F10D06CB 2000-11-23 Lukas Ertl <le@FreeBSD.org>
    Key fingerprint = 20CD C5B3 3A1D 974E 065A B524 5588 79A9 F10D 06CB
uid                                Lukas Ertl <a9404849@unet.univie.ac.at>
uid                                Lukas Ertl <l.ertl@univie.ac.at>
uid                                Lukas Ertl <le@univie.ac.at>
sub 1024g/5960CE8E 2000-11-23
```

**D.3.32. Stefan Farfeleder <stefanf@FreeBSD.org>**

```
pub 1024D/8BEFD15F 2004-03-14 Stefan Farfeleder <stefanf@fafoe.narf.at>
    Key fingerprint = 4220 FE60 A4A1 A490 5213 27A6 319F 8B28 8BEF D15F
uid                                Stefan Farfeleder <stefanf@complang.tuwien.ac.at>
uid                                Stefan Farfeleder <stefanf@FreeBSD.org>
uid                                Stefan Farfeleder <stefanf@ten15.org>
sub 2048g/418753E9 2004-03-14 [expires: 2007-03-14]
```

**D.3.33. Chris D. Faulhaber <jedgar@FreeBSD.org>**

```
pub 1024D/FE817A50 2000-12-20 Chris D. Faulhaber <jedgar@FreeBSD.org>
    Key fingerprint = A47D A838 9216 F921 A456 54FF 39B6 86E0 FE81 7A50
uid                                Chris D. Faulhaber <jedgar@fxp.org>
sub 2048g/93452698 2000-12-20
```

**D.3.34. Brian F. Feldman <green@FreeBSD.org>**

```
pub 1024D/41C13DE3 2000-01-11 Brian Fundakowski Feldman <green@FreeBSD.org>
   Key fingerprint = 6A32 733A 1BF6 E07B 5B8D AE14 CC9D DCA2 41C1 3DE3
sub 1024g/A98B9FCC 2000-01-11 [expires: 2001-01-10]

pub 1024D/773905D6 2000-09-02 Brian Fundakowski Feldman <green@FreeBSD.org>
   Key fingerprint = FE23 7481 91EA 5E58 45EA 6A01 B552 B043 7739 05D6
sub 2048g/D2009B98 2000-09-02
```

**D.3.35. Mário Sérgio Fujikawa Ferreira <lioux@FreeBSD.org>**

```
pub 1024D/A6381352 2003-06-01 Mario Sergio Fujikawa Ferreira (Key For Official Matters) <lioux@FreeBSD.org>
   Key fingerprint = 9CB8 0712 2E58 8B14 7C00 B2FB F26B 6E64 A638 1352
uid                               Mario Sergio Fujikawa Ferreira <lioux@brturbo.com>
uid                               Mario Sergio Fujikawa Ferreira <lioux@uol.com.br>
sub 4096g/F28E9D8C 2003-06-01 [expires: 2006-05-31]
sub 1024g/40575E32 2003-06-01 [expires: 2004-05-31]
sub 1024D/4764FFBA 2003-06-01 [expires: 2004-05-31]
```

**D.3.36. Tony Finch <fanf@FreeBSD.org>**

```
pub 1024D/84C71B6E 2002-05-03 Tony Finch <dot@dotat.at>
   Key fingerprint = 199C F25B 2679 6D04 63C5 2159 FFC0 F14C 84C7 1B6E
uid                               Tony Finch <fanf@FreeBSD.org>
uid                               Tony Finch <fanf@apache.org>
uid                               Tony Finch <fanf2@cam.ac.uk>
sub 2048g/FD101E8B 2002-05-03
```

**D.3.37. Marc Fonvieille <blackend@FreeBSD.org>**

```
pub 1024D/6C6481CA 2002-04-20 Marc Fonvieille <blackend@FreeBSD.org>
   Empreinte de la clé = BA56 0F99 83D0 2601 F9F0 2ADA F354 F531 6C64 81CA
uid                               Marc Fonvieille <marc@blackend.org>
uid                               Marc Fonvieille <marc@freebsd-fr.org>
sub 1024g/D187C27A 2002-04-20
```

**D.3.38. Pete Fritchman <petef@FreeBSD.org>**

```
pub 1024D/74B91CFD 2001-01-30 Pete Fritchman <petef@FreeBSD.org>
   Key fingerprint = 9A9F 8A13 DB0D 7777 8D8E 1CB2 C5C9 A08F 74B9 1CFD
uid                               Pete Fritchman <petef@databits.net>
uid                               Pete Fritchman <petef@csh.rit.edu>
sub 1024g/0C02AF0C 2001-01-30
```

**D.3.39. Bill Fumerola <billf@FreeBSD.org>**

```
pub 1024D/7F868268 2000-12-07 Bill Fumerola (FreeBSD Developer) <billf@FreeBSD.org>
   Key fingerprint = 5B2D 908E 4C2B F253 DAEB FC01 8436 B70B 7F86 8268
uid                               Bill Fumerola (Security Yahoo) <fumerola@yahoo-inc.com>
sub 1024g/43980DA9 2000-12-07
```

**D.3.40. Patrick S. Gardella <patrick@FreeBSD.org>**

```
pub 1024R/527BC62B 1997-06-23 Patrick Gardella <patrick@windministries.org>
   Key fingerprint = 54 7C D6 81 B1 63 D2 5E BD DD 58 4D A4 0E E4 E1
uid                               Patrick Gardella <patrick@freebsd.org>
uid                               patrick@freebsd.org
uid                               Patrick Gardella <patrick@intothewind.cx>

pub 1024D/C6AF4835 2000-01-19 Patrick Gardella <patrick@freebsd.org>
   Key fingerprint = 4B2D 3CFA 963B E310 847E 7D6C C072 B447 C6AF 4835
uid                               Patrick Gardella <patrick@windministries.org>
uid                               Patrick Gardella <patrick@intothewind.cx>
sub 2048g/DB3B83E7 2000-01-19

pub 1024R/EE2D47A9 1996-08-17 Patrick Gardella <patrick@freebsd.org>
   Key fingerprint = A6 DE 3C C0 33 CF 36 E6 D1 F5 BB E8 3C C4 67 50
```

**D.3.41. Sebastien Gioria <gioria@FreeBSD.org>**

```
pub 1024D/7C8DA4F4 2002-02-09 Sebastien Gioria <eagle@freebsd-fr.org>
   Key fingerprint = 41F4 4885 7C23 6ED3 CC24 97AA 6DDD B426 7C8D A4F4
uid                               Sebastien Gioria <gioria@FreeBSD.ORG>
uid                               Sebastien Gioria <gioria@Francenet.fr>
uid                               Sebastien Gioria <gioria@fluxus.net>
sub 4096g/F147E4D3 2002-02-09
```

**D.3.42. John-Mark Gurney <jmg@FreeBSD.org>**

```
pub 1024R/3F9951F5 1997-02-11 John-Mark Gurney <johnmark@gladstone.uoregon.edu>
   Key fingerprint = B7 EC EF F8 AE ED A7 31 96 7A 22 B3 D8 56 36 F4
uid                               John-Mark Gurney <gurney_j@efn.org>
uid                               John-Mark Gurney <jmg@cs.uoregon.edu>
uid                               John-Mark Gurney <gurney_j@resnet.uoregon.edu>
```

**D.3.43. Daniel Harris <dannyboy@FreeBSD.org>**

```
pub 1024D/84D0D7E7 2001-01-15 Daniel Harris <dannyboy@worksforfood.com>
   Key fingerprint = 3C61 B8A1 3F09 D194 3259 7173 6C63 DA04 84D0 D7E7
uid                               Daniel Harris <dannyboy@freebsd.org>
```

```
uid          Daniel Harris <dh@askdh.com>
uid          Daniel Harris <dh@wordassault.com>
sub 1024g/9DF0231A 2001-01-15
```

#### D.3.44. Daniel Hartmeier <dhartmei@FreeBSD.org>

```
pub 1024R/6A3A7409 1994-08-15 Daniel Hartmeier <dhartmei@freebsd.org>
Key fingerprint = 13 7E 9A F3 36 82 09 FE FD 57 B8 5C 2B 81 7E 1F
```

#### D.3.45. John Hay <jhay@FreeBSD.org>

```
pub 2048R/A9275B93 2000-05-10 John Hay <jhay@icomtek.csir.co.za>
Key fingerprint = E7 95 F4 B9 D4 A7 49 6A 83 B9 77 49 28 9E 37 70
uid          John Hay <jhay@mikom.csir.co.za>
uid          Thawte Freemail Member <jhay@mikom.csir.co.za>
uid          John Hay <jhay@csir.co.za>
uid          John Hay <jhay@FreeBSD.ORG>
```

#### D.3.46. Sheldon Hearn <sheldonh@FreeBSD.org>

```
pub 1024D/74A06ACD 2002-06-20 Sheldon Hearn <sheldonh@starjuice.net>
Key fingerprint = 01A3 EF91 9C5A 3633 4E01 8085 A462 57F1 74A0 6ACD
sub 1536g/C42F8AC8 2002-06-20
```

#### D.3.47. Mike Heffner <mikeh@FreeBSD.org>

```
pub 1024D/CDECBF99 2001-02-02 Michael Heffner <mheffner@novacoxmail.com>
Key fingerprint = AFAB CCEB 68C7 573F 5110 9285 1689 1942 CDEC BF99
uid          Michael Heffner <mheffner@vt.edu>
uid          Michael Heffner <mikeh@FreeBSD.org>
uid          Michael Heffner <spock@techfour.net>
uid          Michael Heffner (ACM sysadmin) <mheffner@acm.vt.edu>
sub 1024g/3FE83FB5 2001-02-02
```

#### D.3.48. Guy Helmer <ghelmer@FreeBSD.org>

```
pub 1024R/35F4ED2D 1997-01-26 Guy G. Helmer <ghelmer@freebsd.org>
Key fingerprint = A2 59 4B 92 02 5B 9E B1 B9 4E 2E 03 29 D5 DC 3A
uid          Guy G. Helmer <ghelmer@cs.iastate.edu>
uid          Guy G. Helmer <ghelmer@palisadesys.com>
```

**D.3.49. Maxime Henrion <mux@FreeBSD.org>**

```
pub 1024D/881D4806 2003-01-09 Maxime Henrion <mux@FreeBSD.org>
   Key fingerprint = 81F1 BE2D 12F1 184A 77E4 ACD0 5563 7614 881D 4806
sub 2048g/D0B510C0 2003-01-09
```

**D.3.50. Michael L. Hostbaek <mich@FreeBSD.org>**

```
pub 1024D/0F55F6BE 2001-08-07 Michael L. Hostbaek <mich@freebsdcluster.org>
   Key fingerprint = 4D62 9396 B19F 38D3 5C99 1663 7B0A 5212 0F55 F6BE
uid                               Michael L. Hostbaek <mich@freebsdcluster.dk>
uid                               Michael L. Hostbaek <mich@icommerce-france.com>
uid                               Micahel L. Hostbaek <mich@freebsd.dk>
uid                               Michael L. Hostbaek <mich@the-lab.org>
uid                               Michael L. Hostbaek <mich@freebsd.org>
sub 1024g/8BE4E30F 2001-08-07
```

**D.3.51. Howard F. Hu <foxfair@FreeBSD.org>**

```
pub 1024D/4E9BCA59 2003-09-01 Foxfair Hu <foxfair@FreeBSD.org>
   Key fingerprint = 280C A846 CA1B CAC9 DDCF F4CB D553 4BD5 4E9B CA59
sub 1024g/3356D8C1 2003-09-01 [expires: 2004-02-28]
```

**D.3.52. Jordan K. Hubbard <jkh@FreeBSD.org>**

```
pub 1024R/8E542D5D 1996-04-04 Jordan K. Hubbard <jkh@FreeBSD.org>
   Key fingerprint = 3C F2 27 7E 4A 6C 09 0A 4B C9 47 CD 4F 4D 0B 20
```

**D.3.53. Trevor Johnson <trevor@FreeBSD.org>**

```
pub 1024D/3A3EA137 2000-04-20 Trevor Johnson <trevor@jppj.net>
   Key fingerprint = 7ED1 5A92 76C1 FFCB E5E3 A998 F037 5A0B 3A3E A137
sub 1024g/46C24F1E 2000-04-20
```

**D.3.54. Poul-Henning Kamp <phk@FreeBSD.org>**

```
pub 1024R/0358FCBD 1995-08-01 Poul-Henning Kamp <phk@FreeBSD.org>
   Key fingerprint = A3 F3 88 28 2F 9B 99 A2 49 F4 E2 FA 5A 78 8B 3E
```

**D.3.55. Josef Karthausen <joe@FreeBSD.org>**

```
pub 1024D/E6B15016 2000-10-19 Josef Karthausen <joe@FreeBSD.org>
    Key fingerprint = 7266 8EAF 82C2 D439 5642 AC26 5D52 1C8C E6B1 5016
uid                               Josef Karthausen <joe@tao.org.uk>
uid                               Josef Karthausen <joe@uk.FreeBSD.org>
uid                               [revoked] Josef Karthausen <josef@bsd1.com>
uid                               [revoked] Josef Karthausen <joe@pavilion.net>
sub 2048g/1178B692 2000-10-19
```

**D.3.56. Vinod Kashyap <vkashyap@FreeBSD.org>**

```
pub 1024R/04FCCDD3 2004-02-19 Vinod Kashyap (gnupg key) <vkashyap@freebsd.org>
    Key fingerprint = 9B83 0B55 604F E491 B7D2 759D DF92 DAA0 04FC CDD3
```

**D.3.57. Kris Kennaway <kris@FreeBSD.org>**

```
pub 1024D/68E840A5 2000-01-14 Kris Kennaway <kris@citusc.usc.edu>
    Key fingerprint = E65D 0E7D 7E16 B212 1BD6 39EE 5ABC B405 68E8 40A5
uid                               Kris Kennaway <kris@FreeBSD.org>
uid                               Kris Kennaway <kris@obsecurity.org>
sub 2048g/03A41C45 2000-01-14 [expires: 2006-01-14]
```

**D.3.58. Giorgos Keramidas <keramida@FreeBSD.org>**

```
pub 1024D/318603B6 2001-09-21 Giorgos Keramidas <keramida@FreeBSD.org>
    Key fingerprint = C1EB 0653 DB8B A557 3829 00F9 D60F 941A 3186 03B6
uid                               Giorgos Keramidas <charon@labs.gr>
uid                               Giorgos Keramidas <keramida@ceid.upatras.gr>
uid                               Giorgos Keramidas <keramida@hellug.gr>
uid                               Giorgos Keramidas <keramida@linux.gr>
sub 1024g/50FDBAD1 2001-09-21
```

**D.3.59. Max Khon <fjoe@FreeBSD.org>**

```
pub 1024D/414420F4 2003-04-29 Max Khon <fjoe@freebsd.org>
    Key fingerprint = CE1F 29CA A6BF 2F26 13E8 1B61 62AE 6B8F 4144 20F4
uid                               Max Khon <fjoe@iclub.nsu.ru>
sub 1024g/6585039B 2003-04-29
```

**D.3.60. Andreas Klemm <andreas@FreeBSD.org>**

```

pub 1024D/6C6F6CBA 2001-01-06 Andreas Klemm <andreas.klemm@eu.didata.com>
    Key fingerprint = F028 D51A 0D42 DD67 4109 19A3 777A 3E94 6C6F 6CBA
uid                                Andreas Klemm <andreas@klemm.gtn.com>
uid                                Andreas Klemm <andreas@FreeBSD.org>
uid                                Andreas Klemm <andreas@apsfilter.org>
sub 2048g/FE23F866 2001-01-06

```

**D.3.61. Sergei Kolobov <sergei@FreeBSD.org>**

```

pub 1024D/3BA53401 2003-10-10 Sergei Kolobov <sergei@FreeBSD.org>
    Key fingerprint = A2F4 5F34 0586 CC9C 493A 347C 14EC 6E69 3BA5 3401
uid                                Sergei Kolobov <sergei@kolobov.com>
sub 2048g/F8243671 2003-10-10

```

**D.3.62. Maxim Konovalov <maxim@FreeBSD.org>**

```

pub 1024D/2C172083 2002-05-21 Maxim Konovalov <maxim@FreeBSD.org>
    Key fingerprint = 6550 6C02 EFC2 50F1 B7A3 D694 ECF0 E90B 2C17 2083
uid                                Maxim Konovalov <maxim@macomnet.ru>
sub 1024g/F305DDCA 2002-05-21

```

**D.3.63. Joseph Koshy <jkoshy@FreeBSD.org>**

```

pub 1024D/D93798B6 2001-12-21 Joseph Koshy (FreeBSD) <jkoshy@freebsd.org>
    Key fingerprint = 0DE3 62F3 EF24 939F 62AA 2E3D ABB8 6ED3 D937 98B6
sub 1024g/43FD68E9 2001-12-21

```

**D.3.64. Roman Kurakin <rik@FreeBSD.org>**

```

pub 1024R/79512C58 2003-12-09 Roman Kurakin <rik@cronyx.ru>
    Key fingerprint = 8CB8 1060 6EA8 DDEC 2D49 1E07 7061 C8A6 7951 2C58

```

**D.3.65. Hideyuki KURASHINA <rushani@FreeBSD.org>**

```

pub 1024D/439ADC57 2002-03-22 Hideyuki KURASHINA <rushani@bl.mmtr.or.jp>
    Key fingerprint = A052 6F98 6146 6FE3 91E2 DA6B F2FA 2088 439A DC57
uid                                Hideyuki KURASHINA <rushani@FreeBSD.org>
uid                                Hideyuki KURASHINA <rushani@jp.FreeBSD.org>
sub 1024g/64764D16 2002-03-22

```

**D.3.66. Clement Laforet <clement@FreeBSD.org>**

```
pub 1024D/0723BA1D 2003-12-13 Clement Laforet (FreeBSD committer address) <clement@FreeBSD.org>
    Key fingerprint = 3638 4B14 8463 A67B DC7E 641C B118 5F8F 0723 BA1D
uid                                Clement Laforet <sheepkiller@cultdeadsheep.org>
uid                                Clement Laforet <clement.laforet@cotds.org>
sub 2048g/23D57658 2003-12-13
```

**D.3.67. Max Laier <mllaier@FreeBSD.org>**

```
pub 1024D/3EB6046D 2004-02-09 Max Laier <mllaier@freebsd.org>
    Key fingerprint = 917E 7F25 E90F 77A4 F746 2E8D 5F2C 84A1 3EB6 046D
uid                                Max Laier <max@love2party.net>
```

**D.3.68. Alexander Langer <alex@FreeBSD.org>**

```
pub 1024D/8E2523FD 2003-01-28 Alexander Langer <alex@FreeBSD.org>
    Key fingerprint = B757 5ABF 691F 4011 F565 3765 BFBB 3506 8E25 23FD
sub 1024g/16C3B12D 2003-01-28
```

**D.3.69. Erwin Lansing <erwin@FreeBSD.org>**

```
pub 1024D/15256990 1998-07-03 Erwin Lansing <erwin@lansing.dk>
    Key fingerprint = FB58 9797 299A F18E 2D3E 73D6 AB2F 5A5B 1525 6990
uid                                Erwin Lansing <erwin@FreeBSD.org>
uid                                Erwin Lansing <erwin@droso.dk>
uid                                Erwin Lansing <erwin@droso.org>
sub 2048g/7C64013D 1998-07-03
```

**D.3.70. Yen-Ming Lee <leeym@FreeBSD.org>**

```
pub 1024R/5EB52E51 1999-03-03 Yen-Ming Lee <leeym@FreeBSD.org>
    Key fingerprint = FE 7E 04 56 FC D7 24 21 B9 3B 53 06 E8 93 B5 81
uid                                Yen-Ming Lee <leeym@civil.ncku.edu.tw>
uid                                Yen-Ming Lee <leeym@cae.ce.ntu.edu.tw>
uid                                Yen-Ming Lee <leeym@leeym.com>
```

**D.3.71. Alexander Leidinger <netchild@FreeBSD.org>**

```
pub 1024D/B0063FE7 1999-11-11 Alexander Leidinger (Privat) <Alexander@Leidinger.net>
    Key fingerprint = C518 BC70 E67F 143F BE91 3365 79E2 9C60 B006 3FE7
sub 1024g/263CAC76 1999-11-11

pub 1024D/72077137 2002-01-31 Alexander Leidinger <netchild@FreeBSD.org>
```



```

Key fingerprint = AA3A 8F69 B214 6BBD 5E73 C9A0 C604 3C56 7207 7137
sub 2048g/8C9828D3 2002-01-31

```

### D.3.72. Ying-Chieh Liao <ijliao@FreeBSD.org>

```

pub 1024D/11C02382 2001-01-09 Ying-Chieh Liao <ijliao@CCCA.NCTU.edu.tw>
Key fingerprint = 4E98 55CC 2866 7A90 EFD7 9DA5 ACC6 0165 11C0 2382
uid Ying-Chieh Liao <ijliao@FreeBSD.org>
uid Ying-Chieh Liao <ijliao@csie.nctu.edu.tw>
uid Ying-Chieh Liao <ijliao@dragon2.net>
uid Ying-Chieh Liao <ijliao@tw.FreeBSD.org>
sub 4096g/C1E16E89 2001-01-09

```

### D.3.73. Clive Lin <clive@FreeBSD.org>

```

pub 1024D/A008C03E 2001-07-30 Clive Lin <clive@tongi.org>
Key fingerprint = FA3F 20B6 A77A 6CEC 1856 09B0 7455 2805 A008 C03E
uid Clive Lin <clive@CirX.ORG>
uid Clive Lin <clive@FreeBSD.org>
sub 1024g/03C2DC87 2001-07-30 [expires: 2005-08-25]

```

### D.3.74. Tilman Linneweh <arved@FreeBSD.org>

```

pub 1024D/807AC53A 2002-06-03 Tilman Linneweh <e0025974@student.tuwien.ac.at>
Key fingerprint = A92F 344F 31A8 B8DE DDFA 7FB4 7C22 C39F 807A C53A
uid Tilman Linneweh <arved@arved.at>
uid Tilman Linneweh <arved@FreeBSD.org>
uid Tilman Linneweh <linneweh@zid.tuwien.ac.at>
sub 1024g/FA351986 2002-06-03 [expires: 2006-07-19]

```

### D.3.75. Scott Long <scottl@FreeBSD.org>

```

pub 1024D/017C5EBF 2003-01-18 Scott A. Long (This is my official FreeBSD key) <scottl@freebsd.org>
Key fingerprint = 34EA BD06 44F7 F8C3 22BC B52C 1D3A F6D1 017C 5EBF
sub 1024g/F61C8F91 2003-01-18

```

### D.3.76. Pav Lucistnik <pav@FreeBSD.org>

```

pub 1024D/C14EB282 2003-08-25 Pav Lucistnik <pav@FreeBSD.org>
Key fingerprint = 2622 B7E3 7DA5 5C53 2079 855B 9ED7 583F C14E B282
uid Pav Lucistnik <pav@oook.cz>
sub 1024g/7287A947 2003-08-25

```

**D.3.77. Bruce A. Mah <bmah@FreeBSD.org>**

```

pub 1024D/5BA052C3 1997-12-08 Bruce A. Mah <bmah@acm.org>
    Key fingerprint = F829 B805 207D 14C7 7197 7832 D8CA 3171 5BA0 52C3
uid                                     Bruce A. Mah <bmah@ca.sandia.gov>
uid                                     Bruce A. Mah <bmah@ieee.org>
uid                                     Bruce A. Mah <bmah@cisco.com>
uid                                     Bruce A. Mah <bmah@employees.org>
uid                                     Bruce A. Mah <bmah@freebsd.org>
uid                                     Bruce A. Mah <bmah@packetdesign.com>
uid                                     Bruce A. Mah <bmah@kitchenlab.org>
sub 2048g/B4E60EA1 1997-12-08

```

**D.3.78. Mike Makonnen <mtm@FreeBSD.org>**

```

pub 1024D/7CD41F55 2004-02-06 Michael Telahun Makonnen <mtm@FreeBSD.Org>
    Key fingerprint = AC7B 5672 2D11 F4D0 EBF8 5279 5359 2B82 7CD4 1F55
uid                                     Michael Telahun Makonnen <mtm@tmsa-inc.com>
uid                                     Mike Makonnen <mtm@identd.net>
uid                                     Michael Telahun Makonnen <mtm@acs-et.com>
sub 2048g/E7DC936B 2004-02-06

```

**D.3.79. David Malone <dwmalone@FreeBSD.org>**

```

pub 512/40378991 1994/04/21 David Malone <dwmalone@maths.tcd.ie>
    Key fingerprint = 86 A7 F4 86 39 2C 47 2C C1 C2 35 78 8E 2F B8 F5

```

**D.3.80. Makoto Matsushita <matusita@FreeBSD.org>**

```

pub 1024D/20544576 1999-04-18 Makoto Matsushita <matusita@FreeBSD.org>
    Key fingerprint = 71B6 13BF B262 2DD8 2B7C 6CD0 EB2D 4147 2054 4576
uid                                     Makoto Matsushita <matusita@matatabi.or.jp>
uid                                     Makoto Matsushita <matusita@jp.FreeBSD.ORG>
uid                                     Makoto Matsushita <matusita@ics.es.osaka-u.ac.jp>
sub 1024g/F1F3C94D 1999-04-18

```

**D.3.81. Kenneth D. Merry <ken@FreeBSD.org>**

```

pub 1024D/54C745B5 2000-05-15 Kenneth D. Merry <ken@FreeBSD.org>
    Key fingerprint = D25E EBC5 F17A 9E52 84B4 BF14 9248 F0DA 54C7 45B5
uid                                     Kenneth D. Merry <ken@kdm.org>
sub 2048g/89D0F797 2000-05-15

pub 1024R/2FA0A505 1995-10-30 Kenneth D. Merry <ken@plutotech.com>
    Key fingerprint = FD FA 85 85 95 C4 8E E8 98 1A CA 18 56 F0 00 1F

```

**D.3.82. Dirk Meyer <dinoex@FreeBSD.org>**

```
pub 1024R/331CDA5D 1995-06-04 Dirk Meyer <dinoex@FreeBSD.org>
    Key fingerprint = 44 16 EC 0A D3 3A 4F 28 8A 8A 47 93 F1 CF 2F 12
uid                                     Dirk Meyer <dirk.meyer@dinoex.sub.org>
uid                                     Dirk Meyer <dirk.meyer@guug.de>
```

**D.3.83. Yoshiro Sanpei MIHIRA <sanpei@FreeBSD.org>**

```
pub 1024R/391C5D69 1996-11-21 sanpei@SEAPLE.ICC.NE.JP
    Key fingerprint = EC 04 30 24 B0 6C 1E 63 5F 5D 25 59 3E 83 64 51
uid                                     MIHIRA Yoshiro <sanpei@sanpei.org>
uid                                     Yoshiro MIHIRA <sanpei@FreeBSD.org>
uid                                     MIHIRA Yoshiro <sanpei@yy.cs.keio.ac.jp>
uid                                     MIHIRA Yoshiro <sanpei@cc.keio.ac.jp>
uid                                     MIHIRA Yoshiro <sanpei@educ.cc.keio.ac.jp>
uid                                     MIHIRA Yoshiro <sanpei@st.keio.ac.jp>
```

**D.3.84. Jim Mock <jim@FreeBSD.org>**

```
pub 1024D/3AA4FEE0 2002-01-10 Jim Mock <jim@FreeBSD.org>
    Key fingerprint = AE EC 99 8D 78 28 D3 06 AFF 7 06 CA D8 FE 72 85 3AA 4 FEE 0
uid                                     Jim Mock <mij@soupnazi.org>
sub 1024g/3780652B 2002-01-10
```

**D.3.85. Marcel Moolenaar <marcel@FreeBSD.org>**

```
pub 1024D/61EE89F6 2002-02-09 Marcel Moolenaar <marcel@xc11nt.net>
    Key fingerprint = 68 BB E2 B7 49 AA FF 69 CA 3A DF 71 A6 05 A5 2D 61 EE 89 F6
sub 1024g/6EAB456 2002-02-09
```

**D.3.86. Thomas Möstl <tmm@FreeBSD.org>**

```
pub 1024D/419C776C 2000-11-28 Thomas Moestl <tmm@FreeBSD.org>
    Key fingerprint = 1C 97 A6 04 2B D0 E4 92 51 D0 9C 0F 1F E6 4F 1D 41 9C 77 6C
uid                                     Thomas Moestl <tmoestl@gmx.net>
uid                                     Thomas Moestl <t.moestl@tu-bs.de>
sub 2048g/ECE63CE6 2000-11-28
```

**D.3.87. Rich Murphey <rich@FreeBSD.org>**

```
pub 1024R/583443A9 1995-03-31 Rich Murphey <rich@lamprey.utmb.edu>
    Key fingerprint = AF A0 60 C4 84 D6 0C 73 D1 EF C0 E9 9D 21 DB E4
```

**D.3.88. Akinori MUSHА <knu@FreeBSD.org>**

```
pub 1024D/9FD9E1EE 2000-03-21 Akinori MUSHА <knu@and.or.jp>
    Key fingerprint = 081D 099C 1705 861D 4B70 B04A 920B EFC7 9FD9 E1EE
uid                               Akinori MUSHА <knu@FreeBSD.org>
uid                               Akinori MUSHА <knu@idaemons.org>
uid                               Akinori MUSHА <knu@ruby-lang.org>
sub 1024g/71BA9D45 2000-03-21
```

**D.3.89. Masafumi NAKANE <max@FreeBSD.org>**

```
pub 1024D/CE356B59 2000-02-19 Masafumi NAKANE <max@wide.ad.jp>
    Key fingerprint = EB40 BCAB 4CE5 0764 9942 378C 9596 159E CE35 6B59
uid                               Masafumi NAKANE <max@FreeBSD.org>
uid                               Masafumi NAKANE <max@accessibility.org>
uid                               Masafumi NAKANE <kd5pdi@qsl.net>
sub 1024g/FA9BD48B 2000-02-19
```

**D.3.90. Yoichi NAKAYAMA <yoichi@FreeBSD.org>**

```
pub 1024D/E0788E46 2000-12-28 Yoichi NAKAYAMA <yoichi@assist.media.nagoya-u.ac.jp>
    Key fingerprint = 1550 2662 46B3 096C 0460 BC03 800D 0C8A E078 8E46
uid                               Yoichi NAKAYAMA <yoichi@eken.phys.nagoya-u.ac.jp>
uid                               Yoichi NAKAYAMA <yoichi@FreeBSD.org>
sub 1024g/B987A394 2000-12-28
```

**D.3.91. Alexander Nedotsukov <bland@FreeBSD.org>**

```
pub 1024D/D004116C 2003-08-14 Alexander Nedotsukov <bland@FreeBSD.org>
    Key fingerprint = 35E2 5020 55FC 2071 4ADD 1A4A 86B6 8A5D D004 116C
sub 1024g/1CCA8D46 2003-08-14
```

**D.3.92. Simon L. Nielsen <simon@FreeBSD.org>**

```
pub 1024D/27359A51 2003-07-22 Simon L. Nielsen <simon@nitro.dk>
    Key fingerprint = 912B F341 7E90 2049 05B4 3800 87DA 5C0D 2735 9A51
uid                               Simon L. Nielsen <simon@FreeBSD.org>
sub 2048g/C11D9097 2003-07-22 [expires: 2006-07-21]
```

**D.3.93. Anders Nordby <anders@FreeBSD.org>**

```
pub 1024D/00835956 2000-08-13 Anders Nordby <anders@fix.no>
    Key fingerprint = 1E0F C53C D8DF 6A8F EAAD 19C5 D12A BC9F 0083 5956
uid                               Anders Nordby <anders@FreeBSD.org>
```

sub 2048g/4B160901 2000-08-13

### D.3.94. David O'Brien <obrien@FreeBSD.org>

```
pub 1024R/34F9F9D5 1995-04-23 David E. O'Brien <defunct - obrien@Sea.Legent.com>
Key fingerprint = B7 4D 3E E9 11 39 5F A3 90 76 5D 69 58 D9 98 7A
uid David E. O'Brien <obrien@Nuxi.com>
uid deobrien@ucdavis.edu
uid David E. O'Brien <whois Do38>
uid David E. O'Brien <obrien@FreeBSD.org>
uid David E. O'Brien <dobrien@seas.gwu.edu>
uid David E. O'Brien <obrien@cs.ucdavis.edu>
uid David E. O'Brien <defunct - obrien@media.sra.com>
uid David E. O'Brien <obrien@elsewhere.roanoke.va.us>
uid David E. O'Brien <obrien@Nuxi.com>

pub 1024D/7F9A9BA2 1998-06-10 "David E. O'Brien" <obrien@cs.ucdavis.edu>
Key fingerprint = 02FD 495F D03C 9AF2 5DB7 F496 6FC8 DABD 7F9A 9BA2
uid "David E. O'Brien" <obrien@Nuxi.com>
uid "David E. O'Brien" <obrien@FreeBSD.org>
sub 3072g/BA32C20D 1998-06-10
```

### D.3.95. Philip Paeps <philip@FreeBSD.org>

```
pub 1024D/049DB879 2001-01-04 Philip Paeps <philip@paeps.cx>
Key fingerprint = FA74 3C27 91A6 79D5 F6D3 FC53 BF4B D0E6 049D B879
uid Philip Paeps <philip@pub.telenet.be>
uid Philip Paeps <philip.paeps@pandora.be>
uid Philip Paeps <philip@nixsys.be>
uid Philip Paeps <philip@freebsd.org>
sub 2048g/476FFC94 2001-01-04
```

### D.3.96. Hiten Pandya <hmp@FreeBSD.org>

```
pub 1024D/938CACA8 2004-02-13 Hiten Pandya (FreeBSD) <hmp@FreeBSD.org>
Key fingerprint = 84EB C75E C75A 50ED 304E E446 D974 7842 938C ACA8
uid Hiten Pandya <hmp@backplane.com>
sub 2048g/783874B5 2004-02-13
```

### D.3.97. Mark Peek <mp@FreeBSD.org>

```
pub 1024D/330D4D01 2002-01-27 Mark Peek <mp@FreeBSD.org>
Key fingerprint = 510C 96EE B4FB 1B0A 2CF8 A0AF 74B0 0B0E 330D 4D01
sub 1024g/9C6CAC09 2002-01-27
```

**D.3.98. Peter Pentchev <roam@FreeBSD.org>**

```

pub 1024D/16194553 2002-02-01 Peter Pentchev <roam@ringlet.net>
   Key fingerprint = FDA8 FD79 C26F 3C51 C95E DF9E ED18 B68D 1619 4553
uid      Peter Pentchev <roam@sbnd.net>
uid      Peter Pentchev <roam@FreeBSD.org>
uid      Peter Pentchev <roam@online.bg>
uid      Peter Pentchev <roam@orbitel.bg>
uid      Peter Pentchev <roam@techlab.office1.bg>
uid      Peter Pentchev <roam@cnsys.bg>
sub 1024g/7074473C 2002-02-01

```

**D.3.99. Denis Peplin <den@FreeBSD.org>**

```

pub 1024D/485DDDF5 2003-09-11 Denis Peplin <den@FreeBSD.org>
   Key fingerprint = 495D 158C 8EC9 C2C1 80F5 EA96 6F72 7C1C 485D DDF5
sub 1024g/E70BA158 2003-09-11

```

**D.3.100. Jim Pirzyk <pirzyk@FreeBSD.org>**

```

pub 1024D/4E23DACA 2001-03-02 Jim Pirzyk <pirzyk@freebsd.org>
   Key fingerprint = 07EE A1BD 32E5 C402 59B6 22D5 D846 31D1 4E23 DACA
uid      Jim Pirzyk <Jim.Pirzyk@disney.com>
sub 1024g/F38895F7 2001-03-02

```

**D.3.101. John Polstra <jdp@FreeBSD.org>**

```

pub 1024R/BFBCF449 1997-02-14 John D. Polstra <jdp@polstra.com>
   Key fingerprint = 54 3A 90 59 6B A4 9D 61 BF 1D 03 09 35 8D F6 0D

```

**D.3.102. Kirill Ponomarew <krion@FreeBSD.org>**

```

pub 1024D/AEB426E5 2002-04-07 Kirill Ponomarew <ponomarew@oberon.net>
   Key fingerprint = 58E7 B953 57A2 D9DD 4960 2A2D 402D 46E9 AEB4 26E5
uid      Kirill Ponomarew <krion@netic.de>
uid      Kirill Ponomarew <krion@guug.de>
uid      Kirill Ponomarew <krion@FreeBSD.org>
sub 1024g/0CEE1A9B 2002-04-07 [expires: 2005-07-03]
sub 1024D/8081D57A 2003-01-29 [expires: 2003-04-29]

```

**D.3.103. Mark Pulford <markp@FreeBSD.org>**

```
pub 1024D/182C368F 2000-05-10 Mark Pulford <markp@FreeBSD.org>
   Key fingerprint = 58C9 C9BF C758 D8D4 7022 8EF5 559F 7F7B 182C 368F
uid                               Mark Pulford <mark@kyne.com.au>
sub 2048g/380573E8 2000-05-10
```

**D.3.104. Thomas Quinot <thomas@FreeBSD.org>**

```
pub 1024D/393D2469 1999-09-23 Thomas Quinot <thomas@cuivre.fr.eu.org>
   Empreinte de la clé = 4737 A0AD E596 6D30 4356 29B8 004D 54B8 393D 2469
uid                               Thomas Quinot <thomas@debian.org>
uid                               Thomas Quinot <thomas@FreeBSD.org>
sub 1024g/8DE13BB2 1999-09-23
```

**D.3.105. Doug Rabson <dfr@FreeBSD.org>**

```
pub 1024R/95C11771 2000-02-27 Doug Rabson <dfr@freebsd.org>
   Key fingerprint = 20 BB E4 38 5D 89 D2 D4 68 A6 2F DC 0A DE 10 3C
```

**D.3.106. Tom Rhodes <trhodes@FreeBSD.org>**

```
pub 1024D/1B9EBD43 2002-01-15 Tom Rhodes <trhodes@FreeBSD.org>
   Key fingerprint = 2D4D 1D31 A5C6 08AA 1075 C963 C0F9 A089 1B9E BD43
uid                               Tom Rhodes <darkklogik@pittgoth.com>
sub 3072g/F7A606E4 2002-01-15
```

**D.3.107. Benno Rice <benno@FreeBSD.org>**

```
pub 1024D/87C59909 2002-01-16 Benno Rice <benno@FreeBSD.org>
   Key fingerprint = CE27 DADA 08E3 FAA3 88F1 5B31 5E34 705A 87C5 9909
uid                               Benno Rice <benno@jeamland.net>
sub 1024g/4F7C2BAD 2002-01-16 [expires: 2007-01-15]
```

**D.3.108. Paul Richards <paul@FreeBSD.org>**

```
pub 2048R/BD42E037 2000-02-21 Paul Richards <paul@originative.co.uk>
   Key fingerprint = 38 4F 33 80 53 92 01 46 ED 4B A4 91 3E 1E 6B FE
uid                               Dr David Paul Richards <paul@originative.com>
uid                               Paul Richards <paul@originative.com>
uid                               Paul Richards <paul@FreeBSD.org>
uid                               Paul Richards <paul@apache.org>
```

**D.3.109. Ollivier Robert <roberto@FreeBSD.org>**

```
pub 1024D/7DCAE9D3 1997-08-21 Ollivier Robert <roberto@FreeBSD.org>
    Key fingerprint = 2945 61E7 D4E5 1D32 C100 DBEC A04F FB1B 7DCA E9D3
uid          Ollivier Robert <roberto@keltia.freenix.fr>
uid          Ollivier Robert <roberto@keltia.net>
uid          Ollivier Robert <roberto@eurocontrol.fr>
sub 2048g/C267084D 1997-08-21
```

**D.3.110. Guido van Rooij <guido@FreeBSD.org>**

```
pub 1024R/599F323D 1996-05-18 Guido van Rooij <guido@gvr.org>
    Key fingerprint = 16 79 09 F3 C0 E4 28 A7 32 62 FA F6 60 31 C0 ED
uid          Guido van Rooij <guido@gvr.win.tue.nl>

pub 1024D/A95102C1 2000-10-25 Guido van Rooij <guido@madison-gurkha.nl>
    Key fingerprint = 5B3E 51B7 0E7A D170 0574 1E51 2471 117F A951 02C1
uid          Guido van Rooij <guido@madison-gurkha.com>
sub 1024g/A5F20553 2000-10-25
```

**D.3.111. Niklas Saers <niklas@FreeBSD.org>**

```
pub 1024D/C822A476 2004-03-09 Niklas Saers <niklas@saers.com>
    Key fingerprint = C41E F734 AF0E 3D21 7499 9EB1 9A31 2E7E C822 A476
sub 1024g/81E2FF36 2004-03-09
```

**D.3.112. Mark Santcroos <marks@FreeBSD.org>**

```
pub 1024D/DE4546FF 2003-11-12 Mark Santcroos <marks@ripe.net>
    Key fingerprint = FDCC 0477 D8B4 62CD 6F05 6A73 003E CD35 DE45 46FF
uid          Mark Santcroos <marks@FreeBSD.org>
sub 1024g/3C702FF6 2003-11-12
```

**D.3.113. Hiroki Sato <hrs@FreeBSD.org>**

```
pub 1024D/2793CF2D 2001-06-12 Hiroki Sato <hrs@allbsd.org>
    Key fingerprint = BDB3 443F A5DD B3D0 A530 FFD7 4F2C D3D8 2793 CF2D
uid          Hiroki Sato <hrs@eos.ocn.ne.jp>
uid          Hiroki Sato <hrs@ring.gr.jp>
uid          Hiroki Sato <hrs@FreeBSD.org>
uid          Hiroki Sato <hrs@jp.FreeBSD.org>
uid          Hiroki Sato <hrs@vlsi.ee.noda.tus.ac.jp>
uid          Hiroki Sato <hrs@jp.NetBSD.org>
uid          Hiroki Sato <hrs@NetBSD.org>
sub 1024g/8CD251FF 2001-06-12
```



**D.3.114. Wolfram Schneider <wosch@FreeBSD.org>**

```
Type Bits/KeyID      Date      User ID
pub  1024/2B7181AD 1997/08/09 Wolfram Schneider <wosch@FreeBSD.org>
      Key fingerprint = CA 16 91 D9 75 33 F1 07 1B F0 B4 9F 3E 95 B6 09
```

**D.3.115. David Schultz <das@FreeBSD.org>**

```
pub  1024D/BE848B57 2001-07-19 David Schultz <das@FreeBSD.ORG>
      Key fingerprint = 0C12 797B A9CB 19D9 FDAF 2A39 2D76 A2DB BE84 8B57
uid  David Schultz <dschultz@uclink.Berkeley.EDU>
uid  David Schultz <das@FreeBSD.ORG>
sub  2048g/69206E8E 2001-07-19
```

**D.3.116. Jens Schweikhardt <schweikh@FreeBSD.org>**

```
pub  1024D/0FF231FD 2002-01-27 Jens Schweikhardt <schweikh@FreeBSD.org>
      Key fingerprint = 3F35 E705 F02F 35A1 A23E 330E 16FE EA33 0FF2 31FD
uid  Jens Schweikhardt <schweikh@schweikhardt.net>
sub  1024g/6E93CACC 2002-01-27 [expires: 2005-01-26]
```

**D.3.117. Gregory Neil Shapiro <gshapiro@FreeBSD.org>**

```
pub  1024R/4FBE2ADD 2000-10-13 Gregory Neil Shapiro <gshapiro@gshapiro.net>
      Key fingerprint = 56 D5 FF A7 A6 54 A6 B5 59 10 00 B9 5F 5F 20 09
uid  Gregory Neil Shapiro <gshapiro@FreeBSD.org>

pub  1024D/F76A9BF5 2001-11-14 Gregory Neil Shapiro <gshapiro@FreeBSD.org>
      Key fingerprint = 3B5E DAF1 4B04 97BA EE20 F841 21F9 C5BC F76A 9BF5
uid  Gregory Neil Shapiro <gshapiro@gshapiro.net>
sub  2048g/935657DC 2001-11-14

pub  1024D/FCE56561 2000-10-14 Gregory Neil Shapiro <gshapiro@FreeBSD.org>
      Key fingerprint = 42C4 A87A FD85 C34F E77F 5EA1 88E1 7B1D FCE5 6561
uid  Gregory Neil Shapiro <gshapiro@gshapiro.net>
sub  1024g/285DC8A0 2000-10-14 [expires: 2001-10-14]
```

**D.3.118. Arun Sharma <arun@FreeBSD.org>**

```
pub  1024D/7D112181 2003-03-06 Arun Sharma <arun@sharma-home.net>
      Key fingerprint = A074 41D6 8537 C7D5 070E 0F78 0247 1AE2 7D11 2181
uid  Arun Sharma <arun@freebsd.org>
uid  Arun Sharma <arun.sharma@intel.com>
sub  1024g/ACAD98DA 2003-03-06 [expires: 2005-03-05]
```

**D.3.119. Vanilla I. Shu <vanilla@FreeBSD.org>**

```
pub 1024D/ACE75853 2001-11-20 Vanilla I. Shu <vanilla@FreeBSD.org>
   Key fingerprint = 290F 9DB8 42A3 6257 5D9A 5585 B25A 909E ACE7 5853
sub 1024g/CE695D0E 2001-11-20
```

**D.3.120. Christopher Shumway <cshumway@FreeBSD.org>**

```
pub 1024D/3219F982 2001-05-17 Christopher Shumway <cshumway@freebsd.org>
   Key fingerprint = 45F5 931B 0646 BF84 E78E E274 6C29 340E 3219 F982
uid                               Christopher Shumway <cshumway@titan-project.org>
sub 1024g/D215EFED 2001-05-17
```

**D.3.121. Dmitry Sivachenko <demon@FreeBSD.org>**

```
pub 1024D/13D5DF80 2002-03-18 Dmitry Sivachenko <mitya@cavia.pp.ru>
   Key fingerprint = 72A9 12C9 BB02 46D4 4B13 E5FE 1194 9963 13D5 DF80
uid                               Dmitry S. Sivachenko <demon@FreeBSD.org>
sub 1024g/060F6DBD 2002-03-18
```

**D.3.122. Jesper Skriver <jesper@FreeBSD.org>**

```
pub 1024D/F9561C31 2001-03-09 Jesper Skriver <jesper@FreeBSD.org>
   Key fingerprint = 6B88 9CE8 66E9 E631 C9C5 5EB4 22AB F0EC F956 1C31
uid                               Jesper Skriver <jesper@skriver.dk>
uid                               Jesper Skriver <jesper@wheel.dk>
sub 1024g/777C378C 2001-03-09
```

**D.3.123. Ville Skyttä <scop@FreeBSD.org>**

```
pub 1024D/BCD241CB 2002-04-07 Ville Skyttä <ville.skytta@iki.fi>
   Key fingerprint = 4E0D EBAB 3106 F1FA 3FA9 B875 D98C D635 BCD2 41CB
uid                               Ville Skyttä <ville.skytta@xemacs.org>
uid                               Ville Skyttä <scop@FreeBSD.org>
sub 2048g/9426F4D1 2002-04-07
```

**D.3.124. Ken Smith <kensmith@FreeBSD.org>**

```
pub 1024D/29AEA7F6 2003-12-02 Ken Smith <kensmith@cse.buffalo.edu>
   Key fingerprint = 4AB7 D302 0753 8215 31E7 F1AD FC6D 7855 29AE A7F6
uid                               Ken Smith <kensmith@freebsd.org>
sub 1024g/0D509C6C 2003-12-02
```

**D.3.125. Ben Smithurst <ben@FreeBSD.org>**

```
pub 1024D/2CEF442C 2001-07-11 Ben Smithurst <ben@LSRfm.com>
Key fingerprint = 355D 0FFF B83A 90A9 D648 E409 6CFC C9FB 2CEF 442C
uid Ben Smithurst <ben@vinosystems.com>
uid Ben Smithurst <ben@smithurst.org>
uid Ben Smithurst <ben@FreeBSD.org>
uid Ben Smithurst <csxbscs@comp.leeds.ac.uk>
uid Ben Smithurst <ben@scientia.demon.co.uk>
sub 1024g/347071FF 2001-07-11
```

**D.3.126. Dag-Erling C. Smørgrav <des@FreeBSD.org>**

```
pub 1024D/A0D39168 2004-08-21 Dag-Erling Smørgrav <des@des.no>
Key fingerprint = 82B6 AED6 2F8B 1591 4A92 EE36 3FA0 DE08 A0D3 9168
uid Dag-Erling Smørgrav <des@teleplan.no>
uid Dag-Erling Smørgrav <des@freebsd.org>
uid [jpeg image of size 3868]
sub 1024g/E7611058 2004-08-21 [expires: 2006-08-21]
```

**D.3.127. Maxim Sobolev <sobomax@FreeBSD.org>**

```
pub 1024D/888205AF 2001-11-21 Maxim Sobolev <sobomax@FreeBSD.org>
Key fingerprint = 85C9 DCB0 6828 087C C977 3034 A0DB B9B7 8882 05AF
uid Maxim Sobolev <sobomax@mail.ru>
uid Maxim Sobolev <sobomax@altavista.net>
uid Maxim Sobolev <vegacap@i.com.ua>

pub 1024D/468EE6D8 2003-03-21 Maxim Sobolev <sobomax@portaone.com>
Key fingerprint = 711B D315 3360 A58F 9A0E 89DB 6D40 2558 468E E6D8
uid Maxim Sobolev <sobomax@FreeBSD.org>
uid Maxim Sobolev <sobomax@mail.ru>
uid Maxim Sobolev <vegacap@i.com.ua>

pub 1024D/6BEC980A 2004-02-13 Maxim Sobolev <sobomax@portaone.com>
Key fingerprint = 09D5 47B4 8D23 626F B643 76EB DFEE 3794 6BEC 980A
uid Maxim Sobolev <sobomax@FreeBSD.org>
uid Maksym Sobolyev (It's how they call me in official documents. Pretty I
uid Maksym Sobolyev (It's how they call me in official documents. Pretty I
sub 2048g/16D049AB 2004-02-13 [expires: 2005-02-12]
```

**D.3.128. Daniel C. Sobral <dcs@FreeBSD.org>**

```
pub 1024R/488A2DD5 2000-06-07 Daniel C. Sobral <dcs@newsguy.com>
Key fingerprint = AF 90 A6 A2 B5 8D 6C 28 37 F3 F4 47 8B 31 47 DF
uid Daniel C. Sobral <dcs@freebsd.org>
```

**D.3.129. Brian Somers <brian@FreeBSD.org>**

```
pub 1024R/666A7421 1997-04-30 Brian Somers <brian@freebsd-services.com>
    Key fingerprint = 2D 91 BD C2 94 2C 46 8F 8F 09 C4 FC AD 12 3B 21
uid          Brian Somers <brian@awfulhak.org>
uid          Brian Somers <brian@FreeBSD.org>
uid          Brian Somers <brian@OpenBSD.org>
uid          Brian Somers <brian@uk.FreeBSD.org>
uid          Brian Somers <brian@uk.OpenBSD.org>
```

**D.3.130. Nicolas Souchu <nsouch@FreeBSD.org>**

```
pub 1024D/C744F18B 2002-02-13 Nicholas Souchu <nsouch@freebsd.org>
    Key fingerprint = 992A 144F AC0F 40BA 55AE DE6D 752D 0A6C C744 F18B
sub 1024g/90BD3231 2002-02-13
```

**D.3.131. Volker Stolz <vs@FreeBSD.org>**

```
pub 1024R/3FD1B6B5 1998-06-16 Volker Stolz <vs@freebsd.org>
    Key fingerprint = 69 6F BD A0 2E FE 19 66 CF B9 68 6E 41 7D F9 B9
uid          Volker Stolz <stolz@i2.informatik.rwth-aachen.de> (LSK)
uid          Volker Stolz <vs@foldr.org>
```

**D.3.132. Gregory Sutter <gsutter@FreeBSD.org>**

```
pub 1024D/845DFEDD 2000-10-10 Gregory S. Sutter <gsutter@zer0.org>
    Key fingerprint = D161 E4EA 4BFA 2427 F3F9 5B1F 2015 31D5 845D FEDD
uid          Gregory S. Sutter <gsutter@freebsd.org>
uid          Gregory S. Sutter <gsutter@daemonnews.org>
uid          Gregory S. Sutter <gsutter@pobox.com>
sub 2048g/0A37BBCE 2000-10-10
```

**D.3.133. Yoshihiro Takahashi <nyan@FreeBSD.org>**

```
pub 1024D/8394B81F 2001-10-15 Yoshihiro TAKAHASHI <nyan@jp.FreeBSD.org>
    Key fingerprint = D4FA D8CA 2AED FCF4 90A3 3569 8666 0500 8394 B81F
uid          Yoshihiro TAKAHASHI <nyan@furiru.org>
uid          Yoshihiro TAKAHASHI <nyan@FreeBSD.org>
sub 1024g/B796F020 2001-10-15
```

**D.3.134. Mikhail Teterin <mi@FreeBSD.org>**

```
pub 1024R/3FC71479 1995-09-08 Mikhail Teterin <mi@aldan.star89.galstar.com>
Key fingerprint = 5F 15 EA 78 A5 40 6A 0F 14 D7 D9 EA 6E 2B DA A4
```

**D.3.135. Gordon Tetlow <gordon@FreeBSD.org>**

```
pub 1024D/357D65FB 2002-05-14 Gordon Tetlow <gordont@gnf.org>
Key fingerprint = 34EF AD12 10AF 560E C3AE CE55 46ED ADF4 357D 65FB
uid Gordon Tetlow <gordon@FreeBSD.org>
sub 1024g/243694AB 2002-05-14
```

**D.3.136. Thierry Thomas <thierry@FreeBSD.org>**

```
pub 1024D/C71405A2 1997-10-11 Thierry Thomas <thierry@pompo.net>
Key fingerprint = 3BB8 F358 C2F1 776C 65C9 AE51 73DE 698C C714 05A2
uid Thierry Thomas <thierry@thomas.as>
uid Thierry Thomas <tthomas@mail.dotcom.fr>
uid Thierry Thomas (See <http://horde.org>.) <thierry@horde.org>
uid Thierry Thomas (FreeBSD committer) <thierry@FreeBSD.org>
sub 2048g/277D65DB 1997-10-11
sub 1024G/8866DD0F 2002-01-26
sub 1024g/36DA7AF9 2003-11-26 [expires: 2003-11-27]
sub 1024R/C5529925 2003-11-26
```

**D.3.137. Vincent Tougait <viny@FreeBSD.org>**

```
pub 1024D/A0D22385 2003-12-27 Vincent Tougait (viny) <viny@FreeBSD.org>
Key fingerprint = 2126 0A3B 54E9 CA39 8069 6D80 39CC B9D2 A0D2 2385
sub 2048g/8C47ED80 2003-12-27
```

**D.3.138. Jacques Vidrine <nectar@FreeBSD.org>**

```
pub 2048R/33C1627B 2001-07-05 Jacques A. Vidrine <nectar@celabo.org>
Key fingerprint = CB CE 7D A0 6E 01 DC 61 E5 91 0A BE 79 17 D3 82
uid Jacques A. Vidrine <jvidrine@verio.net>
uid Jacques A. Vidrine <n@nectar.com>
uid Jacques A. Vidrine <jacques@vidrine.cc>
uid Jacques A. Vidrine <nectar@FreeBSD.org>
uid Jacques A. Vidrine <n@nectar.cc>

pub 1024D/1606DB95 2001-07-05 Jacques A. Vidrine <nectar@celabo.org>
Key fingerprint = 46BC EA5B F70A CC81 5332 0832 8C32 8CFF 1606 DB95
uid Jacques A. Vidrine <jvidrine@verio.net>
uid Jacques A. Vidrine <n@nectar.com>
uid Jacques A. Vidrine <jacques@vidrine.cc>
```

```
uid          Jacques A. Vidrine <nectar@FreeBSD.org>
uid          Jacques A. Vidrine <n@nectar.cc>
sub 2048g/57EDEA6F 2001-07-05
```

### D.3.139. Adam Weinberger <adamw@FreeBSD.org>

```
pub 1024D/42C743FD 2002-10-12 Adam Weinberger <adam@vectors.cx>
   Key fingerprint = A980 3F2E 80A8 9619 9D1C 82E8 A3C2 8CD9 42C7 43FD
sub 1024g/15D67628 2002-10-12
```

### D.3.140. Nate Williams <nate@FreeBSD.org>

```
pub 1024D/C2AC6BA4 2002-01-28 Nate Williams (FreeBSD) <nate@FreeBSD.org>
   Key fingerprint = 8EE8 5E72 8A94 51FA EA68 E001 FFF9 8AA9 C2AC 6BA4
sub 1024g/03EE46D2 2002-01-28
```

### D.3.141. Garrett Wollman <wollman@FreeBSD.org>

```
pub 1024D/0B92FAEA 2000-01-20 Garrett Wollman <wollman@FreeBSD.org>
   Key fingerprint = 4627 19AF 4649 31BF DE2E 3C66 3ECF 741B 0B92 FAEA
sub 1024g/90D5EBC2 2000-01-20
```

### D.3.142. Jörg Wunsch <joerg@FreeBSD.org>

```
pub 1024D/69A85873 2001-12-11 Joerg Wunsch <j@uriah.heep.sax.de>
   Key fingerprint = 5E84 F980 C3CA FD4B B584 1070 F48C A81B 69A8 5873
uid          Joerg Wunsch <joerg_wunsch@interface-systems.de>
uid          Joerg Wunsch <joerg@FreeBSD.org>
uid          Joerg Wunsch <j@ida.interface-business.de>
sub 1024g/21DC9924 2001-12-11
```

### D.3.143. Alexey Zelkin <phantom@FreeBSD.org>

```
pub 1024D/9196B7D9 2002-01-28 Alexey Zelkin <phantom@FreeBSD.org>
   Key fingerprint = 4465 F2A4 28C1 C2E4 BB95 1EA0 C70D 4964 9196 B7D9
sub 1024g/E590ABA4 2002-01-28
```

# Глоссарий FreeBSD

Этот глоссарий содержит специфичные для FreeBSD термины и сокращения.

## В

### Berkeley Software Distribution

Это имя, которое в Computer Systems Research Group из Berkeley дали улучшениям и изменениям в AT&T's 32V UNIX.

### Bikeshed Building

Феномен, заключающийся в том, что многие активно участвуют в обсуждении простой темы, в то время как сложная тема привлекает мало внимания или вообще остается незамеченной. Информация о происхождении этого термина находится в FAQ ([../faq/misc.html#BIKESHED-PAINTING](http://faq/misc.html#BIKESHED-PAINTING)).

## BSD

*См.*: Berkeley Software Distribution

## К

### Kernel Scheduler Entities

Поддерживаемая ядром система потоков. Обратитесь к домашней странице проекта (<http://www.FreeBSD.org/kse>) за более детальной информацией.

## KSE

*См.*: Kernel Scheduler Entities

## М

### Merge From Current

Означает добавление функциональности или исправлений из ветви -CURRENT в другую ветвь, чаще всего в -STABLE.

## Merge From Stable

При нормальном ходе разработки FreeBSD, изменение вносится для тестирования в ветвь -CURRENT перед внесением в -STABLE. В редких случаях изменение сначала вносится в -STABLE, а затем переносится в -CURRENT.

Этот термин также используется, когда исправление переносится из -STABLE в ветвь исправлений безопасности.

*См. также:* Merge From Current.

## MFC

*См.:* Merge From Current

## MFS

*См.:* Merge From Stable

# P

## Principle Of Least Astonishment

При развитии FreeBSD, видимые пользователю изменения должны быть как можно менее неожиданны. Например, произвольное перемещение переменных в `/etc/defaults/rc.conf` нарушает POLA. Разработчики учитывают POLA при внесении видимых пользователю изменений.

## POLA

*См.:* Principle Of Least Astonishment



# Предметный указатель

## Symbols

-CURRENT, 564

использование, 565

компиляция, 566

Синхронизация при помощи CTM, 566

Синхронизация при помощи CVSup , 566

-STABLE, 564, 567

использование, 567

компиляция, 568

синхронизация при помощи CTM, 568

Синхронизация при помощи CVSup , 568

.rhosts, 508

/boot/kernel.old, 206

/etc, 325

/etc/gettytab, 600

/etc/groups, 380

/etc/login.conf, 377

/etc/mail/access, 660

/etc/mail/aliases, 660

/etc/mail/local-host-names, 660

/etc/mail/mailer.conf, 660

/etc/mail/mailertable, 660

/etc/mail/sendmail.cf, 660

/etc/mail/virtusertable, 660

/etc/remote, 606

/etc/ttys, 601

/usr, 325

/usr/bin/login, 599

/usr/local/etc, 328

/usr/share/skel, 372

/var, 325

10 base 2, 783

10 base T, 783

386BSD, 4, 8

386BSD Patchkit, 4

4.3BSD-Lite, 5

4.4BSD-Lite, 1, 2

802.11

(См. беспроводные сети)

анонимный CVS, 206, 569

аварийные дискеты, 510

баннерные страницы

(См. начальные страницы)

беспроводные сети, 752

точка доступа, 753

шифрование, 757

бездисковые конфигурации

/usr только для чтения, 780

настройка ядра, 776

безопасность, 383

backdoors, 385

DoS атаки

(См. отказ в обслуживании (Denial of Service, DoS))

OpenSSH, 442

OpenSSL, 428

межсетевые экраны, 418

одноразовые пароли, 394

взлом учетных записей, 384

защита FreeBSD, 385

шифрование, 393

бит в секунду, 235

бит-в-секунду, 588

брандмауэр, 769, 770

демон автоматического монтирования, 695

дискеты, 506

дискеты с резервными копиями, 506

диски

добавление, 480

файловая система в памяти (5.X), 518

файловые системы в памяти (4.X), 517

хранимые в файле (4.X), 515

отключение диска, отображаемого в память, 518

отображаемые в файлы (5.X), 516

память, 515

виртуальные, 515

шифрование, 523

дисковые квоты, 377, 520

ограничения, 521

проверка, 520, 523

двоичная совместимость

BSD/OS, 2

Linux, 2, 286

NetBSD, 2

SCO, 2

SVR4, 2

файл сервер

unix клиенты, 691

Windows клиенты, 737

- файловые системы
  - HFS, 491
  - ISO 9660, 490, 490
  - Joliet, 491
  - мгновенные копии, 519
- файлы gc, 368
  - gc.serial, 602
- файлы устройств, 183, 222
- фильтрация пакетов, 3
- графический чипсет Intel i810, 152
- группы, 380
- хосты с двойным подключением, 748
- иерархия каталогов, 105
- имя домена, 648
- интернационализация, 552
- исходный текст, 2
- исходящие соединения, 606
- использование fetchmail, 683
- использование prosmail, 684
- кабели RS-232C, 588, 589
- каталог шаблонов, 372
- каталоги, 104
- ключи rgr, 880
- кодировки, 553
- коллекция портов, 287
- командная строка, 121
- коммиттеры, 7
- компиляторы
  - C, 2
  - C++, 2
  - FORTRAN, 2
- конфигурация загрузчика, 363
- консоль, 100, 367
- корневая файловая система, 114
  - бездисковые конфигурации, 777
- корневая зона, 720
- корневой раздел, 514
- криптография, 737
- квоты, 377
- лог файлы, 341
  - DNS, 729
  - FTP, 741
  - named, 730
- логин, 619
- локализация, 552
  - корейский, 562
  - немецкий, 562
  - японский, 562
- традиционный китайский, 562
- локаль, 308
- маршрут по умолчанию, 648, 746, 747
- маршрутизация, 745
- маршрутизатор, 3, 749, 769
- многопользовательская доступность, 1
- многопользовательский режим, 368, 572
- модем, 598, 632, 650, 781
- мягкое ограничение, 522
- настройка
  - с помощью sysctl, 343
- настройка ядра, 297, 308
- настройка сетевой карты, 334
  - настройка, 335
  - поиск драйвера, 335
  - решение проблем, 338
  - тестирование карты, 337
- настройка системы, 325
- начальная загрузка, 360
- начальные страницы, 236, 255
- носители на магнитной ленте, 503
  - AIT, 505
  - DLT, 504
  - QIC-150, 504
  - магнитные ленты DDS (4мм), 503
  - магнитные ленты Exabyte (8мм), 503
  - магнитные ленты QIC, 503
- нуль-модемный кабель, 93, 230, 589, 610
- оболочка, 121
- оболочки Bourne, 122
- оболочки PPP, 625, 625
- оборудование, 90
- обратный DNS, 720
- однопользовательский режим, 365, 367, 572, 576
- одноразовые пароли, 394
- офисный пакет
  - KOffice, 172
  - OpenOffice.org, 174
- ограничение пользователей, 377
  - coredumpsize, 378
  - cputime, 378
  - fi lesize, 379
  - maxproc, 379
  - memorylocked, 379
  - memoryuse, 379
  - openfi les, 379
  - sbsize, 380
  - stacksize, 380

- квоты, 377
- оптимизация
  - параметры ядра, 348
- оптимизация системы, 325
- основы, 100
- пакеты, 129
  - удаление, 134
  - управление, 134
  - установка, 132
- параметры ядра
  - cpu, 208
  - cpu type, 208
  - ident, 209
  - LINUX, 287
  - machine, 208
  - maxusers, 209
  - MSDOSFS, 211
  - NFS, 211
  - NFS\_ROOT, 211
  - options BRIDGE, 770
  - options CPU\_ENABLE\_SSE, 188
  - options USER\_LDT, 188
- пароль, 619
- переменные окружения, 121
- перенаправление адреса, 787
- период отсрочки, 523
- печать, 226, 273
  - фильтры, 240, 242, 244
    - apsfilter, 253
  - начальных страниц, 237
- печать по сети, 261, 262
- подсеть, 745, 746
- подсеть IP, 768
- порт ввода-вывода, 183
- порты, 129
  - дисковое пространство, 141
  - обновление, 142
  - удаление, 141
  - установка, 136
  - установка с CD-ROM, 137
- последовательная консоль, 93, 609
- последовательные соединения, 588
- последовательный порт, 235
- почтовые демоны
  - exim, 658
  - postfix, 658
  - qmail, 658
  - sendmail, 658
- почтовые программы пользователей, 676
- почтовый хост, 660
- права доступа, 103
- права на файлы, 103
- приложения
  - Maple, 292
  - Mathematica, 290
  - MATLAB, 294
  - Oracle, 297
  - SAP R/3, 300
- принт сервер
  - Windows клиенты, 737
- принтеры, 560
  - USB, 229
    - характеристики, 236
    - использование, 273
    - ограничение доступа, 265
    - параллельные, 228, 234
    - последовательные, 228, 235, 239, 246
    - сетевые, 261, 262
- приоритетная многозадачность, 1
- программы резервного копирования
  - amanda, 509
  - cpio, 509
  - rar, 509
  - tag, 508
  - резервное копирование / восстановление, 507
- протокол управления передачей, 230
- ядро, 361
  - LINT, 207
  - NOTES, 207
  - флаги загрузки, 365
  - компиляция, 575
  - конфигурационный файл, 207
  - конфигурация, 181
  - настройка, 651, 785
  - взаимодействия во время загрузки, 365
- работа без диска, 771
- распространение маршрутов, 751
- раздел подкачки, 326
- разделы, 112, 480
- разделы BSD, 481
- разделы диска, 325
- размер раздела подкачки, 326
- редакторы, 123
- резолвер, 720
- сервер имен, 619, 648
- сервисы, 328

сетевой мост, 768  
сетевые группы, 707  
сеть TCP/IP, 650, 653  
сигнал 11, 582  
симметричная многопроцессорность (SMP), 2  
символические ссылки, 288  
синонимы ip, 339  
система спулинга LPD, 226  
сканеры, 198  
скорость передачи, 230  
слайсы, 112, 480  
сокрытие IP, 784  
совместно используемые библиотеки, 288  
совместное использование доступа в Интернет, 784  
список рассылки, 569  
стартовые скрипты, 101  
статические маршруты, 655  
статический IP адрес, 619  
страницы справочника, 126  
сжатие, 507  
текстовые редакторы, 123  
терминалы, 100, 593  
трансляция сетевых адресов, 769  
установка, 10  
    без монитора (последовательная консоль), 93  
    дискеты, 96  
    решение проблем, 90  
    с MS-DOS, 96  
    с ленты QIC/SCSI, 97  
    сеть  
        Ethernet, 97  
        FTP, 95  
        NFS, 98  
        параллельная (PLIP), 97  
        последовательная (SLIP или PPP), 97  
устройство loopback, 746  
устройство записи CD  
    ATAPI, 490  
    драйвер ATAPI/CAM, 490, 495  
учёт  
    дисковое пространство, 520  
участники, 7  
учет  
    использования принтера, 255, 269  
учет страниц, 255  
учетные записи  
    daemon, 371  
    nobody, 371  
    operator, 371  
добавление, 372  
группы, 380  
изменение, 372  
изменение пароля, 376  
ограничение, 377  
пользователей, 372  
системные, 371  
суперпользователь (root), 371  
удаление, 374  
жёсткое ограничение, 522  
веб сервер  
    настройка, 734  
    защита, 737  
веб-браузеры  
    Mozilla  
        (См. Mozilla)  
входящие соединения, 598  
видео пакеты, 191  
видео порты, 191  
виртуальная память, 2  
виртуальные диски, 515  
виртуальные консоли, 100  
виртуальные сервера, 339  
время ожидания, 622  
задания печати, 227, 238, 245, 267, 274  
    управления, 267  
загрузка, 360  
загрузчик, 363, 363  
загрузчик выполняемых классов, 321  
защита памяти, 1  
зоны  
    примеры, 720  
звуковые карты, 181  
шифрование, 393  
шлюз, 745  
шрифты, 291  
    TrueType, 154  
    антиалиасинг, 155  
    межсимвольное расстояние, 156  
шрифты с антиалиасингом, 155  
электронная почта, 3, 657  
электронная таблица  
    Abacus, 178  
    Gnumeric, 177  
частота горизонтальной развертки, 150  
частота вертикальной развертки, 150  
четность, 230, 235

- Отказ в обслуживании (Denial of Service, DoS), 384
- Отказ в обслуживании (DoS), 390
- Пересборка world, 569
  - затраченное время, 575
- Политика многоуровневой безопасности MAC, 466
- Пользователи
  - Крупные сайты, работающие под FreeBSD, 4
- Права доступа
  - символические, 105
- Принудительный контроль доступа, 452
- Проект FreeBSD
  - цели, 6
  - история, 4
  - модель разработки, 6
- Расширения физического адресного пространства (Physical Address Extensions (PAE))
  - Конфигурации с большим количеством оперативной памяти, 221
- Сообщения безопасности FreeBSD, 449
- Тонкая настройка X11, 150
- ТВ тюнеры, 196
- Шрифты
  - LCD-дисплей, 157
- Шрифты TrueType, 154

## A

- Abacus, 178
- AbiWord, 173
- ACL, 447
- Acrobat Reader, 175
- adduser, 372, 556
- AIX, 697
- Alpha, 13, 18, 89, 92
- Alpha BIOS, 92
- amanda, 509
- amd, 695
- Apache, 4, 734
  - файл настройки, 735
  - модули, 736
  - запуск или остановка, 735
- apsfilter, 253
- ARC, 92
- ASCII, 282, 553
- AT&T, 5
- AutoPPP, 627

## B

- BGP, 749
- BIND, 666, 719
  - конфигурационные файлы, 721
  - кэширующий сервер имён, 728
  - работа в песочнице, 728
  - запуск, 721
- bind9
  - установка, 732
- BIOS, 31
  - Basic Input/Output System, 360
- bits-per-second, 588
- Blue Mountain Arts, 4
- Bluetooth, 760
- BOOTP
  - бездисковые конфигурации, 774
- BSD Copyright, 6

## C

- CDROM
  - создание, 490
  - создание загрузочного, 491
  - запись, 491, 492
- centronics
  - (См. параллельные принтеры)
- CHAP, 619, 622, 628
- chpass, 375
- chroot, 728
- Cisco, 632
- Coda, 515
- Computer Systems Research Group (CSRG), 2, 8
- comsat, 387
- Concurrent Versions System
  - (См. CVS)
- core team, 7
- coredumpsize, 378
- country codes, 553
- cpio, 509
- cputime, 378
- cron, 566, 568
  - настройка, 331
- CTM, 206, 569
- cu, 93
- cuaa, 592
- CUPS, 281

## CVS

анонимный, 206  
репозиторий, 7  
cvsup, 206, 566, 568

## D

dangerously dedicated, 112  
DCE, 588  
DES, 393  
DEVFS, 580  
device.hints, 366  
DGA, 189  
DHCP  
    dhcpd.conf, 717  
    бездисковые конфигурации, 773  
    конфигурационные файлы, 716  
    конфигурационный файлы, 718  
    сервер, 716  
    требования, 715  
    установка, 717  
disk concatenation, 530  
disk mirroring, 532  
disk striping, 531  
disklabel, 510, 514  
DNS, 340, 627, 658, 670, 719  
    записи, 725  
DNS сервер, 3  
DOS, 16, 31, 91, 559  
DoS атаки  
    (См. отказ в обслуживании (Denial of Service, DoS))  
DSL, 769  
DSP, 183  
DTE, 588  
dump, 507  
DVD  
    DVD+RW, 498  
    DVD-RW, 499  
    DVD-Video, 498  
    запись, 496  
Dynamic Host Configuration Protocol  
    (См. DHCP)

## E

editors  
    emacs, 123  
    vi, 123  
ee, 123  
ELF, 322  
    branding, 322  
emacs, 123  
email, 657  
    настройка, 669  
    получение, 659  
    устранение неисправностей, 666  
    замена mta, 663  
Etherboot, 775  
Ethernet, 653  
    MAC адрес, 654, 746  
    MAC-адрес, 290

## F

fdisk, 480  
fi lesize, 379  
fi lesystems  
    mounted with fstab, 115  
    mounting, 116  
    unmounting, 117  
fi nger, 387  
Firefox, 171  
fi rewall, 418  
FORTRAN, 249  
Free Software Foundation, 5, 8  
FreshMeat, 131  
FreshPorts, 131  
FTP  
    через HTTP прокси, 47  
    Пассивный режим, 46  
FTP сервер, 740  
FTP серверы, 3

## G

gated, 655  
getty, 599  
Ghostscript, 247  
GIMP, 173  
GNOME, 161

- антиалиасинг шрифтов, 162

GNU General Public License (GPL), 6  
GNU Lesser General Public License (LGPL), 6  
GNU toolchain, 289  
GNU набор для разработки, 3  
GnuCash, 177  
Gnumeric, 177  
GQview, 176  
Greenman, David, 4  
Grimes, Rod, 4  
gv, 176  
gzip, 507

## H

HCI, 761  
hostname, 340  
hosts, 340  
HP-UX, 697  
Hubbard, Jordan, 4  
hw.ata.wc, 345

## I

ICMP\_BANDLIM, 391  
IEEE, 509  
ifconfi g, 758  
IMAP, 658, 659  
init, 361, 367  
installation

- network
  - FTP, 46

Internet Software Consortium (ISC), 714  
IP маскарадинг, 3  
IP по параллельному порту, 788  
IPSP, 623  
ipfw, 419

- настройка, 421
- включение, 419

IPX/SPX, 784

IRQ, 183  
ISA, 181  
ISDN, 769, 780

- адаптеры, 780
- отдельно стоящие мосты/маршрутизаторы, 782

ISO 9660, 490  
ISP, 619, 623

## J

JMA Wired, 4  
Jolitz, Bill, 4

## K

KDE, 162

- антиалиасинг шрифтов, 165
- менеджер дисплеев, 163

Kerberos5

- история, 410
- настройка центра распространения ключей, 410
- настройка клиента, 414
- ограничения и недостатки, 416
- пользовательские файлы настройки, 414
- решение проблем, 415
- включение сервисов, 412
- внешние ресурсы, 417

KerberosIV, 387, 392

- первый запуск, 404
- установка, 402

kermit, 632  
kern.ipc.somaxconn, 349  
kern.maxfiles, 348  
kernel

- пример конфигурационного файла, 207
- сборка / установка, 204
- сборка собственного ядра, 203

kernel options

- SMP, 213

kernel.old, 365  
keymap, 557  
KLD (загружаемый объект ядра), 287  
KOffice, 172  
Konqueror, 172

## L

- L2CAP, 762
- language codes, 553
- LCD-дисплей, 157
- LCP, 627
- LDAP, 739
- LINT, 207
- Linux, 697
  - двоичные файлы ELF, 289
  - файловая система /proc, 286
  - установка библиотек Linux, 287
- Linux - бинарная совместимость, 286
- LISA, 510
- locale, 553, 554, 556
- localization
  - Russian, 560
- login class, 554, 556
- LPRng, 281
- ls, 104

## M

- MAC, 452
  - File System Firewall Policy, 461
- MAC -- пример реализации, 470
- MAC Biba Integrity Policy, 468
- MAC Configuration Testing, 472
- MAC Example Implementation, 470
- MAC Interface Silencing Policy, 462
- MAC LOMAC, 469
- MAC Port Access Control List Policy, 463
- MAC Process Partition Policy, 465
- MAC See Other UIDs Policy, 460
- MAC Troubleshooting, 476
- MacOS, 397
- make, 574
- make.conf, 571
- MAKEDEV, 222, 580, 593
- Mandatory Access Control, 452
- Master Boot Record (MBR), 361
- maxproc, 379
- MD5, 393
- memorylocked, 379
- memoryuse, 379
- mencoder, 194
- mergemaster, 577
- mgetty, 627

- Microsoft Windows, 31, 737
- MIME, 554, 556
- MIT, 402
- mount, 93, 514
- mountd, 691
- moused, 557
- Mozilla, 157, 169
- MPlayer
  - использование, 193
  - сборка, 192
- MS-DOS, 282, 397
- multicast
  - options MROUTING, 752
- MX record, 658, 666, 669, 727
- MySQL, 559

## N

- natd, 784
- net.inet.ip.portrange.\*, 349
- Net/2, 5
- NetBIOS, 627, 738
- NetBSD, 8, 697
- Netcraft, 4
- Netscape, 171
- newfs, 514
- newsyslog.conf, 342
- NFS, 515, 523, 691
  - installing multiple machines, 585
  - бездисковые конфигурации, 775
  - использование, 694
  - монтирование, 694
  - настройка, 692
  - примеры экспортирования, 692
  - сервер, 691
- nfsd, 691
- NIS, 697
  - домены, 697
  - форматы паролей, 713
  - главный сервер, 699
  - имя домена, 700
  - карты, 701
  - клиент, 699
  - настройка клиента, 705
  - настройка сервера, 701
  - вторичный сервер, 699, 703
- NIS+, 739



NOTES, 207

Novell, 5

ntalk, 387

NTP, 741

- ntp.conf, 742

- ntpd, 741

- настройка, 742

- выбор серверов, 741

ntpdate, 742

## O

OBEX, 767

OpenBSD, 8, 697

openfiles, 379

OpenOffice.org, 174

OpenSSH, 442

- безопасное копирование, 443

- клиент, 443

- настройка, 444

- тунелирование, 445

- включение, 443

OpenSSL

- генерирование сертификатов, 429

Opera, 171

OS/2, 282, 482

OSPF, 749

## P

Pair Networks, 4

pairing, 764

PAP, 619, 622, 628

passwd, 376

password, 628

paх, 509

PCI, 181

PCL, 234, 283

PDF

- просмотр, 175, 176, 176

Perl, 737

PHP, 737

pkg\_add, 132

pkg\_delete, 134

pkg\_info, 134

pkg\_version, 134

PLIP, 788

POP, 658, 659

portmap, 691, 698

portupgrade, 142

POSIX, 509, 554

PostScript, 229, 233

- просмотр, 176

- эмуляция, 247

PPP, 618, 619, 666, 781

- configuration, 629

- kernel PPP, 618, 631

- NAT, 629

- over Ethernet, 618

- PPP уровня пользователя, 621

- user PPP, 618

- клиент, 632

- настройка, 620

- прием входящих соединений, 624

- решение проблем, 639

- с динамическими IP адресами, 623

- сервер, 631

- со статическими IP адресами, 620

- создание файлов устройств, 619

- через ATM, 644

- через Ethernet, 642

- Расширения Microsoft, 627

PPPoA

- (См. PPP, через ATM)

PPPoE

- (См. PPP, через Ethernet)

printer spool, 238

pw, 377, 556

## R

RAID, ??

- CCD, 483

- Software, 529

- Vinum, 486

- программный, 483, 486

- Оборудование, 486

RAID-1, 532

RAID-5, 532

rc files

- rc.serial, 593

rc файлы

- rc.conf, 327

rcNG, 333

resolv.conf, 340  
restore, 507  
RFCOMM, 763  
RIP, 656, 749  
rlogind, 387  
rmuser, 374  
routed, 630  
RPM, 304  
rshd, 387

## S

Samba сервер, 737  
sandboxes, 387  
sbsize, 380  
scp, 443  
screenmap, 557  
SCSI, 31  
SCSI\_DELAY  
    kern.cam.scsi\_delay, 346  
SDL, 189  
SDP, 765  
sendmail, 387, 630, 660  
shutdown, 368  
skeleton directory, 372  
SLIP, 618, 647, 652, 653  
    клиент, 647  
    маршрутизация, 655  
    сервер, 650  
    соединение, 648  
SMTP, 630, 669  
snapshot, 564  
Soft Updates, 346  
    детали, 346  
Solaris, 322, 697  
Sony Japan, 4  
Sophos Anti-Virus, 4  
SQL база данных, 739  
SRM, 92  
ssh, 392  
sshd, 387  
SSL, 737  
stacksize, 380  
su, 385, 481  
SunOS, 205, 700  
Supervalu, 4  
sysctl, 343, 343, 388

sysctl.conf, 343  
sysinstall, 557, 715  
    добавление дисков, 481  
syslog, 729, 741  
syslog.conf, 341  
sysutils/cdrtools, 490

## T

tar, 507, 508  
TCP Bandwidth Delay Product Limiting  
    net.inet.tcp.inflight\_enable, 350  
TCP Wrappers, 399  
TCP/IP стек, 1  
tcpwrapper, 706  
TELEHOUSE America, 4  
telnetd, 387  
TeX, 227, 276  
    печать файлов dvi, 248  
TFTP  
    бездисковые конфигурации, 775  
traceroute, 752  
Traditional Chinese  
    BIG-5 encoding, 555  
troff, 244  
Tru64 UNIX, 702  
ttyd, 592  
tunefs, 346

## U

U.C. Berkeley, 2, 5, 8  
UDP, 715  
Unicode, 559  
UNIX, 103, 619  
USB  
    диски, 488  
USENET, 3  
UUCP, 667

## V

vfs.hirunningspace, 345  
vfs.vmodirenable, 344  
vfs.write\_behind, 345  
vi, 123  
Vinum, 529

- striping (перемежение), 531
- сцепленные диски (concatenation), 530
- зеркалирование (mirroring), 532

vipw, 556  
vm.swap\_idle\_enabled, 345

## W

Walnut Creek CDROM, 4  
Weathernews, 4  
web серверы, 3  
WER, 757  
wheel, 386  
Williams, Nate, 4  
Windows, 397  
Windows NT, 697

## X

X Display Manager, 158  
X Window System, 2

- (См. также XFree86)
- Accelerated-X, 3
- XFree86, 3

X11, 149  
X11 Input Method (XIM), 559  
X11 True Type font server, 559  
XF86Config, 151  
XFree86, 149  
XFree86&#160;4.X, 149  
XML, 155  
Xorg, 149  
xorg.conf, 151  
Xpdf, 176  
XVideo, 189

## Y

Yahoo!, 4  
yellow pages  
(См. NIS)

## Z

zip drive, 218

# Colophon

Эта книга представляет собой результат совместной работы сотен людей в рамках "Проекта Документирования FreeBSD". Текст написан в SGML формате в соответствии с DocBook DTD и сформатирован из SGML во множество различных представительских форматов с помощью приложения **Jade**, открытой DSSSL системы. DSSSL стили от Нормы Волша (Norm Walsh) были использованы совместно с дополнительными стилями разработанными в рамках проекта для создания презентационных инструкций для **Jade**. Создание печатной версии этого документа было бы невозможно без языка T<sub>E</sub>X Дэвида Кнута, L<sup>A</sup>T<sub>E</sub>X Лисли Лэмпорта, и макро пакета **JadeT<sub>E</sub>X** Себастьяна Ратца.