Д. КОЛИСНИЧЕНКО

САМОУЧИТЕЛЬ системного администратора Linux

- Дистрибутивы Fedora 13, Mandriva 2010.1 Spring, openSUSE 11.3, Ubuntu 10
- Установка и настройка операционной системы
- Подробное рассмотрение файловой системы Linux
- Настройка сети и маршрутизации без конфигураторов
- Брандмауэры iptables и ebtables, chroot-окружение
- Настройка серверов: Web, FTP, DNS, DHCP, почтового и сервера баз данных
- Прокси-серверы Squid и SquidGuard
- Linux-сервер в Windows-сети: свой среди чужих
- Виртуальные частные сети (VPN)
- Создание LiveCD
- Сетевой сканер nmap
- Система управления доступом Tomoyo
- Защита и оптимизация Linux-сервера
- Автоматизация задач с помощью bash
- Программные RAID-массивы

СИСАДМИН
СИСТЕМНЫЙ АДМИНИСТРАТОР

Денис Колисниченко

САМОУЧИТЕЛЬ системного администратора Linux

Санкт-Петербург

«БХВ-Петербург»

2011

УДК 681.3.06
ББК 32.973.26-018.2
К60

Колисниченко Д. Н.

К60 Самоучитель системного администратора Linux. — СПб.: БХВ-Петербург, 2011. — 544 с.: ил. — (Системный администратор)

ISBN 978-5-9775-0639-7

Описаны основы сетевого взаимодействия, планирование и монтаж сети (Ethernet и Wi-Fi), настройка сети и маршрутизации без конфигураторов. Даны примеры настройки различных типов серверов: Web, FTP, DNS, DHCP, почтового сервера, сервера баз данных. Рассмотрены дистрибутивы Fedora 13, Mandriva 2010.1 Spring, openSUSE 11.3, Ubuntu 10, файловая система Linux, установка и базовая настройка Linux, а также связки Apache + MySQL + PHP. Особое внимание уделено защите сетевых сервисов и оптимизации работы сервера: использованию брандмауэров iptables и ebtables, прокси-серверов Squid и SquidGuard, созданию chroot-окружения, управлению доступом с помощью системы Tomoyo, настройке VPN-сервера, аудиту сети при помощи сетевого сканера nmap. Приведены практические рекомендации по стратегии администрирования и уходу за аппаратными средствами, работе Linux-сервера в Windows-сети, созданию LiveCD, автоматизации задач с помощью bash, использованию программных RAID-массивов.

Для администраторов Linux

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Григорий Добин*
Компьютерная верстка	*Натальи Караваевой*
Корректор	*Виктория Пиотровская*
Дизайн серии	*Инны Тачиной*
Оформление обложки	*Елены Беляевой*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 03.11.10.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 43,86.
Тираж 1800 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12.

ISBN 978-5-9775-0639-7

Оглавление

Введение

На этот раз введение не будет длинным. Тому есть две причины. Во-первых, не хочется занимать ваше драгоценное время. Во-вторых, как показывает практика, больше половины читателей считают введение чем-то скучным и вообще его не читают. Зачем же тратить время и бумагу?

Хочется сказать несколько слов об особенностях этой книги, которые выделяют ее среди других книг, посвященных системному администрированию Linux. Я старался написать ее так, чтобы она не стала "еще одной книгой по настройке Linux-сервера". В ней есть все, что нужно знать будущему системному администратору. Так, в первой части книги рассматриваются основы основ: принципы работы компьютерных сетей, адресация в сетях, монтаж сети. Все это подано настолько подробно, чтобы у будущего администратора не возникали вопросы типа "А что такое сетевая маска?" или "Как обжать витую пару?". Могу с уверенностью сказать, что благодаря этой информации вы не только настроите Linux-сервер, но сможете построить локальную сеть.

В остальных частях книги рассматривается настройка Linux (в том числе и настройка сети — как же без нее?), установка программного обеспечения, настройка сетевых служб (Apache, DNS, DHCP, ssh, Squid и т. д.). Особое внимание уделяется безопасности настраиваемого сервера — как локальной, так и безопасности в сети: подробно описываются создание шлюза (маршрутизатора), конфигурирование брандмауэра iptables, а также настройка виртуальной частной сети. Вопросам безопасности в книге действительно уделяется много внимания, а помимо всего прочего мы рассмотрим и сканер nmap — чтобы администратор мог сам просканировать свою сеть на предмет потенциальных уязвимостей.

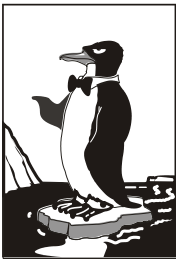
Материал книги основан на дистрибутивах Fedora 13, Mandriva 2010.1 Spring, openSUSE 11.3, Ubuntu 10.04, Debian 5 и Slackware 13. Учитывая столь обширный список дистрибутивов, могу с уверенностью сказать, что книга подойдет большинству администраторов.

Вот теперь можно с чистой совестью приступить к чтению книги. И даже если вы не новичок, а действующий администратор, рекомендую все-таки прочитать первую часть книги — в ней вы найдете для себя много полезного.

ЧАСТЬ I

Основы администрирования

Первая часть посвящена основам администрирования. Мы рассмотрим краткую историю Linux, основы сетевого взаимодействия, познакомимся с моделью OSI, адресацией в TCP/IP-сетях, с монтажом сетей Ethernet и Wi-Fi. А в следующей части книги поговорим об установке и настройке Linux.



Глава 1

Становимся администратором

1.1. Краткая история Linux

В далеком 1969 году сотрудники фирмы Bell Labs пытались возродить ОС Multics, но превзошли сами себя, и то, что получилось, уже никак не тянуло на обычный "апгрейд" для Multics — это была совершенно новая операционная система, которую назвали UNIX. Интересно, что поначалу UNIX называлась "UNICS", но позже американцы, как они это любят делать, немного упростили название системы.

В начале 70-х годов прошлого века ОС UNIX была существенно доработана. В ее ядро добавили много новых функций, а главное — она была переписана на языке C, что обеспечило легкость переноса этой ОС на другие аппаратные платформы (первоначально UNIX была написана на ассемблере и предназначалась для компьютера PDP-7).

Важно, что с самого рождения UNIX была многопользовательской и многозадачной. Таким образом, идеи, заложенные в представленную в 1995 году Windows 95, оказались, по сути, идеями 20-летней давности — в UNIX все это уже было реализовано 20 лет назад. Да, не было красивого "фантика" — графического интерфейса, — но ведь не это главное в операционной системе.

В начале 1980-х годов появились первые персональные компьютеры фирмы IBM. Однако мощности IBM PC никак не хватало для запуска UNIX. Поэтому в мире персональных компьютеров десять лет царствовала операционная система DOS компании Microsoft. Начиная с 1990-х все изменилось — мощность "персоналок" уже позволяла запускать UNIX. К этому времени (прошло более 20 лет с момента появления первой версии UNIX) разными фирмами, университетами и отдельными энтузиастами было создано много UNIX-подобных операционных систем (IRIX, XENIX, HP-UX, BSD, Minix и др.).

Огромное значение в развитии Linux сыграла одна из UNIX-подобных операционных систем — Minix, которая не была полноценной системой, а создавалась, чтобы демонстрировать основные принципы и устройство настоящих операционных систем. Да, она не была совершенной, но зато ее исходный код (всего 12 тысяч строк) был опубликован в книге А. Таненбаума "Операционные системы". Именно эту книгу и купил Линус Торвальдс (Linus Torvalds).

В 1991 году Линус Торвальдс установил на свой компьютер ОС Minix, но та не оправдала его ожиданий, поэтому он принял решение несколько ее переработать — ведь исходные коды вместе с комментариями были под рукой. Сначала Торвальдс

просто переписал программу эмуляции терминала, а затем фактически взялся за создание собственной операционной системы.

25 августа 1991 года ОС Linux (версия 0.01) была создана. Конечно, это была не та Linux, что есть сейчас, но она уже тогда была лучше Minix, поскольку в ней запускались командный интерпретатор bash и компилятор gcc. Сообщение о создании новой операционной системы было помещено в группу новостей comp.os.minix, там же предлагалось всем желающим протестировать ее.

С этого и началось интенсивное развитие Linux, а к ее разработке в помощь Торвальдсу подключились энтузиасты со всего мира, — ведь ничто так не сокращает расстояния, как Интернет. С момента появления версии 0.01, которой практически нельзя было пользоваться, до создания версии 1.0, пригодной для обычных пользователей, а не программистов, прошло почти три года (она появилась в апреле 1994 года). И уже эта первая версия обладала поддержкой сети (поддерживался протокол TCP/IP), а также графическим интерфейсом X Window, появившимся в Linux еще в 1992 году одновременно с поддержкой TCP/IP.

Первые версии Linux распространялись на обыкновенных дискетах. Комплект состоял из двух дискет: первая содержала ядро, а вторая — корневую файловую систему и необходимые программы. Установить подобную версию Linux на компьютер мог только специалист. Чуть позже появились первые дистрибутивы, которые, помимо того же ядра и корневой файловой системы, включали также программу для установки всего этого на компьютер. Программа установки поставлялась, как правило, на отдельной дискете.

Первые дистрибутивы появились в 1992 году — тогда отдельные энтузиасты или группы энтузиастов выпускали разные дистрибутивы (каждый, естественно, под своим именем). Фактически они отличались друг от друга лишь названием и программой установки. В дальнейшем различия между дистрибутивами стали более существенными.

Самый первый дистрибутив, созданный в Манчестерском компьютерном центре (Manchester Computing Centre, MCC), появился в начале 1992 года и назывался MCC Interim Linux. Чуть позже появился дистрибутив TAMU, разработанный в Техасском университете.

Настоящий прорыв произвел дистрибутив SLS, выпущенный в октябре 1992 года, поскольку именно он содержал поддержку TCP/IP и систему X Window. Впоследствии данный дистрибутив бурно развивался и постепенно трансформировался в один из самых популярных дистрибутивов — Slackware.

Со временем дистрибутивы разрослись до таких размеров, что распространять их на дискетах стало нельзя. Вы можете себе представить дистрибутив на 50 дискетах (дистрибутивы того времени занимали 50–70 Мбайт)? А что делать, если, скажем, дискета № 47 окажется бракованной? Как раз к тому времени лазерные компакт-диски и их приводы немного подешевели, и компания Red Hat стала одной из первых, выпустивших свою разработку на компакт-диске.

Кроме получения на дискетах или компакт-диске, дистрибутив того времени (как, впрочем, и сейчас) можно было бесплатно скачать из Интернета (если не считать стоимости самого Интернета). Но далеко не все могли себе позволить Интернет в online-режиме (тогда online-режимом считалась работа с WWW, а offline — с почтой и новостями Usenet). Да и привод CD-ROM (односкоростной) стоил около

100 долларов. Поэтому в начале 1990-х основными носителями для распространения Linux все же были дискеты. А вот начиная с середины 1990-х Linux постепенно почти полностью перекочевала на компакт-диски.

О дистрибутивах можно говорить еще очень долго. Важно запомнить следующее:

- основные дистрибутивы: Red Hat, Slackware и Debian, все остальные — это производные от них. Например, Mandrake произошел от Red Hat, ALT Linux потом взял за основу Mandrake, а ASPLinux — Red Hat. Потом на смену Red Hat пришел дистрибутив Fedora Core (сейчас просто Fedora), а на смену Mandrake — Mandriva;
- номер версии дистрибутива не совпадает с номером ядра — это принципиально разные вещи.

1.2. Почему именно Linux?

А почему именно Linux? Почему бы не использовать ту же FreeBSD, у которой родства с UNIX намного больше, чем у Linux? На базе FreeBSD, как и на базе Linux, можно построить стабильный сервер. Но у Linux есть одно неоспоримое преимущество — она популярнее. А это значит, что для нее больше русскоязычной документации, на Linux уже обращают внимание производители оборудования (вы без особых проблем найдете драйвер для вашего "железа"), да и Linux более дружественна к пользователю. Да, именно к пользователю. Конечно, для администратора сервера это не столь важно, но Linux более универсальна, что позволяет ее использовать как на сервере, так и на рабочих станциях. Получается, что можно установить одну и ту же операционную систему на всех компьютерах сети — следовательно, вам будет проще обслуживать эту сеть, чем "разношерстную" сеть, где компьютеры работают под всевозможными версиями Windows, Mac OS и Linux.

1.3. Основные задачи системного администратора

Сейчас мы рассмотрим основные задачи системного администратора. У нас ведь как бывает: сисадмин и монтажом сети занимается, и обучением пользователей (далеко не все умеют "на кнопки" нажимать). Поэтому сразу скажу: далее приведен список обязанностей администратора Linux-сервера, работающего в идеальных условиях.

- **Установка и настройка программного обеспечения** — после установки самой Linux вам нужно будет установить дополнительное программное обеспечение, например, Web-сервер, FTP-сервер, а затем настроить это программное обеспечение.
- **Управление пользователями** — в обязанности администратора также входит создание, модификация и удаление учетных записей пользователей сервера. Возможно, придется ограничить место на диске, предоставляемое каждому пользователю (эта операция называется *квотированием*).
- **Инсталляция и деинсталляция аппаратных средств** — кому как не администратору подключать новые жесткие диски и подготавливать их для использо-

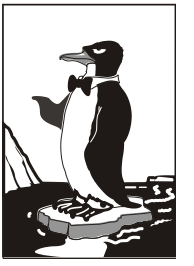
вания сервером. Причем часто бывает, что устанавливать "железо" (впрочем, как и "софт") придется не только на сервере, но и на рабочих станциях — такова уж судьба сисадмина…

- **Резервное копирование** — это одна из самых важных задач системного администратора. Часто резервное копирование, к сожалению, не выполняется или выполняется не так, как нужно. В результате — потеря данных. Да, это не интересно, да — это рутинно. Но выполнять эту задачу нужно.
- **Поиск неисправностей** — время от времени аппаратные средства выходят из строя. Иногда случаются "глюки" в программном обеспечении. Найти и устранить неисправность — задача системного администратора. Сразу предупреждаю: часто найти неисправность сложнее, чем ее устранить.
- **Защита сети.** Обеспечение безопасности сети и контроль защиты — очень важная задача, ведь вы же не хотите, чтобы ваш сервер взломали? Часто бывает, что "врага" нужно ожидать не извне, а изнутри — это могут быть любопытные либо недовольные пользователи, способные посягнуть на неприступность вашего сервера.
- **Мониторинг системы** — важно ежедневно просматривать журналы системы. В журналах можно найти много интересной и полезной информации: попытки взлома, ошибки в конфигурации системы и т. д.
- **Консультации и техническая поддержка пользователей** — чтобы не отвлекать вас от основных задач, желательно, чтобы эту задачу выполнял ваш помощник. Но если вы единственный администратор в компании, то этим придется заниматься лично вам.
- **Ведение локальной документации** — чтобы вам (или тому человеку, который, возможно, займет впоследствии ваше место) было проще в будущем, следует протоколировать все свои действия: разводку кабелей сети, инсталлируемые программные средства, изменения в конфигурации системы, изменения в схеме сети и т. д.

Вот теперь вы знаете, с чем вам придется столкнуться при выполнении своих обязанностей. Но это только базовый комплект — вполне возможно, что на практике появится еще несколько задач, которые вам придется решать.

Профессиональный праздник

Так уж получилось, что данная глава была написана в день системного администратора, поэтому не упомянуть об этом профессиональном празднике я просто не могу. День системного администратора отмечается в последнюю пятницу июля (в 2010 году это 30 июля). Основатель праздника — Тед Кекатос (Ted Kekatos), системный администратор из Чикаго. Именно он посчитал, что раз в году "бойцы невидимого фронта" должны чувствовать благодарность со стороны пользователей. Кстати, в США этот праздник называется День благодарности системному администратору (System Administrator Appreciation Day). Первый раз праздник был отмечен 28 июля 1999 года.



Глава 2

Классификация сетей

2.1. Краткая история сетей

С появлением первых электронно-вычислительных машин (не персональных компьютеров, а именно первых огромных вычислительных машин, которые занимали целые комнаты) возникла проблема переноса данных между ними. С того момента было создано много различных сетей. Сейчас мы вкратце рассмотрим историю сетей, чтобы вы знали, откуда они появились, а потом попробуем классифицировать все имеющиеся виды сетей.

2.1.1. 1941–1975 годы

Первый период развития вычислительных сетей начинается в 1941 году (тогда, если вы помните, появилась первая "большая" ЭВМ) и называется *лабораторным* — в то время сети, как впрочем и ЭВМ, не выходили за пределы лабораторий научных институтов. С самого начала ставилась задача объединения в сеть ЭВМ без привязки к конкретной аппаратуре.

Любопытно

Казалось бы, как давно это было! Но самое интересное, что мы до сих пор используем решения, разработанные в то время. Последовательный интерфейс RS-232C и параллельный интерфейс Centronics (да, тот, который служит для подключения принтеров) используются до сих пор. Интерфейс RS-232C постепенно вытесняется современными последовательными интерфейсами: USB и IEEE 1394 (FireWire), и на некоторых современных компьютерах его больше нет вообще. Однако интерфейс Centronics имеется на каждом современном стационарном компьютере, хотя большинство производителей принтеров уже практически перешло на USB. Наличие "старых" интерфейсов зависит только от производителя материнской платы — как он решит, так и будет. Мой компьютер, на котором я пишу эти строки, был куплен в феврале 2008 года. Тогда я не обратил внимания на наличие/отсутствие старых интерфейсов, но потом выяснилось, что на материнской плате отсутствует RS-232C, но имеется Centronics (LPT), а также USB, IEEE 1394, HDMI (правда, он не имеет никакого отношения к сетевым интерфейсам) и другие современные разъемы, которых не было на более старых компьютерах. С другой стороны, в продаже до сих пор имеются материнские платы с RS-232C, а также предлагаются отдельные PCI-контроллеры, добавляющие два порта RS-232C, если в них возникает острая необходимость.

Интерфейсы RS-232C и Centronics — это, в принципе, хорошо, но они годятся только для связи "точка–точка", то есть для непосредственной связи отправителя

и получателя данных. Понятно, что в сети может быть гораздо больше, чем две ЭВМ, поэтому разработчики сетей на этом не остановились, и в 1974 году компания IBM представила универсальную архитектуру вычислительных сетей: SNA (System Network Architecture). Эта архитектура, помимо всего прочего, поддерживала *адресацию узлов* сети, смысл которой в том, что каждому узлу сети присваивается уникальный адрес, по которому можно обратиться к этому узлу. Сейчас для адресации узлов преимущественно используются протоколы IPv4 и IPv6, о которых мы поговорим далее в этой главе.

2.1.2. 1976–1982 годы

Второй период развития сетей начался в 1976 году, когда сети вышли за пределы лабораторий и начали активно разрабатываться сетевые архитектуры и технологии передачи данных. Тогда и появилось семейство протоколов X.25 — протоколов передачи данных в системах с коммутацией пакетов. Разработка протоколов X.25 стала очень важным событием, поскольку до появления Интернета они были единственными протоколами, используемыми для создания глобальных сетей, — именно X.25-сети связывали тогда весь мир в единое целое. Затем на базе X.25 был создан протокол Frame Relay, а на его базе — технология ATM. Подробно рассматривать все производные протоколов X.25 мы не будем, поскольку нас сейчас интересуют только ключевые события в развитии сетей (описание истории появления каждого сетевого протокола займет целую книгу, прочитать которую у вас не хватит терпения). Отмечу только, что Frame Relay, как и ATM, здравствуют и по сей день.

В 1979 году был создан первый модем для персональных (!) компьютеров. Я даже догадываюсь, о чем вы сейчас подумали: какие, мол, персональные компьютеры в 1979 году? Какой модем? Да, Personal Computer (PC) от IBM появился в 1981 году, но это не означает, что до этого не было *персональных компьютеров*. Для работы с первыми ЭВМ обычно требовался целый штат специалистов, а персональный компьютер — это компьютер, предназначенный только для одного человека, для одного пользователя. И настоящие персональные компьютеры, отвечающие этому определению, появились еще до 1980 года — это были компьютеры компании Apple. А словосочетание "Personal Computer" — всего лишь название, правда, весьма удачное, продукта компании IBM. IBM первая ввела термин PC, и с того времени все компьютеры со сходной архитектурой команд считаются PC-совместимыми.

А все современные модемы являются Hayes AT-совместимыми, то есть совместимыми с набором AT-команд управления модемом, разработанным компанией Hayes. Первый модем Micromodem II был выпущен этой компанией в 1979 году. Он развивал скорость в 300 бод (бит/с) и предназначался для компьютеров Apple.

Еще в лабораторном периоде были разработаны *системы с произвольным доступом*. Впервые они были использованы в начале 1970-х годов в сети Alohanet, объединяющей Гавайские острова. Сначала эти системы считались бесперспективными, но в мае 1973 года Боб Меткалф (Bob Metcalf) усовершенствовал метод CSMA, на котором они были основаны. Усовершенствованный метод назвали CSMA/CD (Carrier-Sense Multiple Access with Collision Detection, множественный доступ с контролем несущей и обнаружением коллизий). Боб Меткалф планировал

использовать этот метод для совместного доступа к сетевым принтерам, но он позже "перерос" в то, что сейчас называется Ethernet-сетью. Тогда сеть CSMA/CD передавала данные по коаксиальному кабелю (как первые Ethernet-сети) со скоростью 2,94 Мбит/с (для того времени это была значительная скорость), а максимальное расстояние передачи данных составляло 1,5 км. В 1978 году Меткалф зарегистрировал компанию 3Com Corporation (наверное, все мы слышали название этой компании), а в 1982 году выпустил первый в мире серийный Ethernet-адаптер для компьютера Apple.

В 1979 году произошло еще одно важное событие — был организован альянс DIX (DEC, Intel, Xerox), результатом деятельности которого стала в 1980 году разработка стандарта Ethernet.

В 1980 году была разработана *модель взаимосвязи открытых систем* (Open System Interconnect, OSI). Эта модель четко определяет семь уровней, которые обеспечивают передачу данных по сети. Модель OSI сугубо теоретическая, но она лежит в основе всех современных сетей. Мы подробно рассмотрим ее чуть позже в этой главе.

2.1.3. 1983–1989 годы

Начиная с 1983 года, в институтах и даже некоторых офисах стали появляться первые локальные сети, связывающие компьютеры толстым коаксиальным кабелем. В то время сетевой адаптер стоил очень дорого (например, для ЭВМ VAX стоимость сетевого адаптера превышала 3 тыс. долларов), поэтому локальную сеть могли себе позволить только самые крупные фирмы. Найти тогда "персоналку" с сетевым адаптером было сложно.

В 1985 году Институтом инженеров по электротехнике и электронике (IEEE) был прият стандарт IEEE 802.3 (10Base-5) — Ethernet-сеть на "толстом" коаксиальном кабеле. В 1989 году был принят стандарт IEEE 802.3a (10Base-2), предусматривающий передачу данных по "тонкому" коаксиальному кабелю. Подробно о стандартах Ethernet мы поговорим чуть позже в этой книге.

Понятно, что Ethernet-сети — не единственный вид локальной сети. В 1988 году IBM превзошла стандарт Ethernet, представив технологию Token Ring с максимальной скоростью передачи данных в 16 Мбит/с (Ethernet предусматривал передачу данных с максимальной скоростью в 10 Мбит/с).

В 1985 году компания StrataCom начала эксплуатацию первых линий T1 со скоростью передачи данных 1,54 Мбит/с. Чуть позже линии T1 стали доступны крупным компаниям и использовались в качестве магистралей для быстрой передачи данных на большие расстояния.

Индивидуальным пользователям в 1980-х годах сети "особо не светили", поскольку сетевое оборудование продолжало стоить весьма дорого. Так, в 1989 году компания Arc Electronics представила высокоскоростной модем (19,2 Кбит/с) стоимостью "всего" 3595 долларов. Интересно, что этот модем был относительно дешевле модемов других производителей, которые, к тому же, не обеспечивали заявленной ими скорости.

Кто мог позволить себе сети ISDN, радовался скорости передачи данных в 128 Кбит/с (сети ISDN BRI) или 1,54 Мбит/с (ISDN PRI). О цене говорить не будем — ISDN-сети стоили неприлично дорого.

Технологии — это, конечно, хорошо. Но сетевые адаптеры и прочее сетевое оборудование без программного обеспечения — просто железки. Чтобы компьютер мог работать в сети, нужна сетевая операционная система. В 1980-х годах сеть поддерживали следующие ОС: UNIX (и ее вариации), Novell Netware, Microsoft LAN Manager (оболочка для OS/2, появившаяся в 1987 году).

В 80-х годах прошлого века появились и первые сотовые сети — да, сотовая телефония! Первая система сотовой телефонной связи Nordic Mobile Telephone System (кто помнит — первые "мобилки", появившиеся у нас в 1990-х годах, поддерживали стандарт NMT) была запущена в Дании, Швеции, Финляндии и Норвегии в 1981 (!) году. В 1983 году заработали две сотовые сети в Северной Америке: AURORA-400 и AMPS.

2.1.4. 1990–1995 годы

В 1990 году произошел очередной "переворот" в Ethernet-сетях — был принят стандарт IEEE 802.3i (10Base-T), предусматривающий передачу данных по витой паре 3-й категории со скоростью 10 Мбит/с. Переворот заключался в том, что Ethernet-сети стали:

- *надежнее* — при использовании коаксиального кабеля все компьютеры подключались к общему кабелю, и если этот кабель обрывался, то вся сеть "падала". В случае с витой парой все компьютеры сети подключаются к центральному устройству сети — Ethernet-концентратору. Если происходит обрыв кабеля, ведущего к какому-нибудь узлу сети, этот узел исчезает из сети, но вся сеть продолжает работать;
- *проще в установке* — монтаж витой пары намного проще, чем коаксиального кабеля, особенно, если речь идет о "толстом" коаксиальном кабеле.

Позднее был принят стандарт IEEE 802.1D, в котором было определено понятие *моста* (bridge), и Ethernet-сети наконец-то стало можно делить на сегменты для локализации трафика. Сегментация сети особо важна для крупных сетей — ведь чем больше узлов, тем меньше производительность сети.

Через три года сети того времени стали напоминать современные — в них активно начали использоваться концентраторы и мосты, появились первые коммутаторы и двухуровневые сети. В двухуровневых сетях компьютеры одной рабочей группы (одного отдела компании) объединялись между собой концентратором, а сами рабочие группы (то есть концентраторы рабочих групп) подключались через мосты к общей корпоративной магистрали. В качестве магистрали обычно использовалось оптоволокно (стандарт 10Base-FL или IEEE 802.3j, принятый в 1993 году). С появлением 10Base-FL на оптоволокне Ethernet-сети выходят за пределы зданий и становятся средством для создания "кампусных" сетей. То есть если раньше Ethernet-сети использовались только для создания локальных сетей, то в 1994–1995 годах стандарт 10Base-FL применялся для связи локальных сетей, находящихся в разных зданиях.

Следующим шагом в создании корпоративных сетей стало изобретение многопортового устройства — центрального коммутатора, в котором были объединены все мосты сети. Такая конфигурация получила название collapsed-backbone ("маги-

страль в точке"). Примерно в это же время родилось понятие *структурированных кабельных сетей* (СКС).

Понятно, что сети росли, и скорости 10 Мбит/с для магистрали стало недостаточно. На тот момент существовала всего одна "быстрая" технология, обеспечивающая передачу данных по оптоволоконному кабелю со скоростью 100 Мбит/с — FDDI (Fiber Distributed Data Interface, распределенный волоконный интерфейс данных). Но в 1992 году компания Grand Junction начала разработку Ethernet-сети, работающей на скорости 100 Мбит/с, и она была стандартизирована в 1995 году (стандарт IEEE 802.3u, сети 100Base-TX, 100Base-T4 и 100Base-FX). В том же 1995 году компания Grand Junction была поглощена компанией Cisco Systems: закон выживания — выживают лишь сильнейшие. После принятия стандартов 100Base-* спрос на технологию FDDI резко пошел вниз, поскольку Ethernet-сети обеспечивали ту же скорость передачи данных, но стоили намного дешевле только за счет среды передачи данных — витая пара стоит намного дешевле, чем оптоволокно. А в 1998 году появились Ethernet-сети, передающие данные со скоростью 1 Гбит/с, но об этом позже.

А что же происходило в мире глобальных сетей? В 1990 году компания US Sprint начала предоставлять услуги объединения точек через Frame Relay по всей территории США. Тогда почти все высокоскоростные магистрали переводились на технологию ATM, но для подключения клиентов использовался Frame Relay. Однако в 1994 году компания Bell Atlantic начинает предлагать подключение клиентов по технологии ATM.

Не стоит забывать и об операционных системах. В 1993 году появилась первая действительно сетевая ОС от Microsoft — Windows NT, а в 1995 году — нашумевшая ОС Windows 95.

2.1.5. 1996–1999 годы

В эти годы ничего революционного в магистральных каналах связи не случилось, если не считать появления сервисов гарантирования качества обслуживания (QoS, Quality of Service). Но нас интересуют технологии, более близкие к пользователю. Можно сказать, что в эти годы (1995–1999) завершилась эра развития аналоговых модемов. В 1998 году был принят стандарт V.90, который используется и по сей день (если не считать его небольшого усовершенствования V.92, появившегося в 2000 году). Судя по всему, телефонные модемы отжили свое. Сегодня все больше и больше провайдеров предоставляют высокоскоростной доступ к Интернету, а обычные аналоговые модемы практически уже не используются.

Зарождение высокоскоростного доступа произошло как раз в 1995–1999 годах, когда появились первые кабельные и ADSL-модемы. Кабельные модемы (они передают данные по сетям операторов кабельного телевидения) преимущественно применялись в США. В Европе получили большее распространение ADSL-модемы, использующие для передачи данных обычный телефонный кабель. К сожалению, в те годы в России о таких модемах только слышали, но никто их практически не видел.

В мире локальных сетей в 1998 году появилась технология 1000Base-X, передающая данные со скоростью 1 Гбит/с по оптоволокну, а в 1999 году — технология 1000Base-T, передающая данные со скоростью 1 Гбит/с по витой паре.

2.1.6. 2000 — наше время

Понятно, что развитие сетей не останавливается, а только начинается. Все еще впереди. Лет через десять все современные технологии будут казаться нам такими же "древними", какими сейчас кажутся решения 20-летней давности.

Из интересного в мире Ethernet можно отметить появление в 2003 году технологий передачи данных со скоростью 10 Гбит/с (10GBase-SR, 10GBase-LR, 10GBase-ER, 10GBase-SW, 10GBase-LW, 10GBase-EW) и технологии PLC, обеспечивающей передачу данных по сети электропитания. В 2003 году это казалось странным, но сейчас — вполне нормально.

Если вы заметили, то в этой краткой истории практически ничего не было сказано о развитии беспроводных сетей. Это сделано умышлено. В *главе 6* мы поговорим о том, как данные передаются "по воздуху", рассмотрим краткую историю беспроводных сетей и существующие беспроводные стандарты.

2.2. Классификация сетей

Сети можно классифицировать по:

- занимаемой территории;
- топологии;
- ведомственной принадлежности;
- скорости передачи данных;
- типу среды передачи данных;
- организации взаимодействия компьютеров.

2.2.1. По занимаемой территории

По занимаемой территории сети могут быть локальными, региональными (они же муниципальные сети) и глобальными:

- *локальные* (LAN, Local Area Network) — сети, занимающие небольшую территорию, например, одну комнату или одно здание;
- *региональные* (MAN, Metropolitan Area Network) — сети, охватывающие город (отсюда другое название — муниципальные) или даже область;
- *глобальные* (WAN, Wide Area Network) — такие сети охватывают территории одного или нескольких государств или даже весь мир. Пример всемирной сети — Интернет.

С локальными и глобальными сетями все понятно, разберемся с сетями региональными. Сеть MAN, как правило, объединяет в единое целое несколько сетей — например, сети двух или более зданий. При этом среда передачи данных сети MAN может быть как проводной, так и беспроводной.

Беспроводная сеть обходится намного дешевле, чем сеть на базе оптоволокна, но она менее надежна и менее безопасна. Тем не менее, беспроводные технологии очень полезны для MAN — не всегда есть возможность проложить кабель. С другой стороны, MAN часто выступает в качестве магистральной сети, поэтому производительности беспроводной сети может оказаться недостаточно.

Сейчас особой необходимости в MAN-сетях нет, поскольку можно организовать *виртуальную частную сеть* (VPN, Virtual Private Network), использующую каналы Интернета для передачи данных. Представим следующую ситуацию: есть организация, главный офис которой находится в Москве, затем эта компания открыла свой филиал в Санкт-Петербурге. Как объединить сети офисов вместе? Вы только представьте себе, сколько кабеля для этого понадобится! Причем витой парой здесь не отделаешься, придется использовать дорогой оптоволоконный кабель — ведь расстояние-то большое. Беспроводные технологии тоже из-за расстояния отпадают. Остается только одно — использовать для передачи данных каналы Интернета. Сеть каждого офиса подключается к Интернету через каналы местного интернет-провайдера, и через Интернет создается виртуальная частная сеть. И дешево, и быстро — ведь высокоскоростное подключение к Интернету в настоящее время вполне доступно. Понятно, что данные будут передаваться по незащищенным каналам, поэтому в виртуальной частной сети используется шифрование всех передаваемых данных. Механизмы VPN позволяют не только объединить две разные сети в единое целое, но и обеспечить безопасность передаваемых данных.

2.2.2. По топологии

Существуют следующие топологии сети:

- *линейная* (рис. 2.1) — подключение по принципу гирлянды: каждый узел сети подключается к следующему узлу сети. В такой сети от узла с номером 1 до узла N будет всегда одинаковый маршрут: через узлы 2, 3, 4, …, $N - 1$. Понятно, в случае отказа одного из узлов сети, линейная сеть прекратит свое существование. В настоящее время линейные сети практически не используются (если не принимать во внимание нуль-модемное соединение);
- *кольцевая* (рис. 2.2) — каждый узел сети соединен с двумя соседними узлами, все узлы сети образуют кольцо. Кольцевая топология используется технологиями Token Ring, FDDI и некоторыми другими;
- *звездообразная* (рис. 2.3) — в такой сети есть один центральный узел, с которым связан каждый узел сети. Такие сети еще называются *централизованными*. "Падение" центрального узла означает "падение" всей сети. Обычно в качестве центрального узла используется концентратор (hub) или коммутатор (switch). Пример звездообразной сети — Ethernet на базе витой пары;
- *общая шина* (рис. 2.4) — все узлы сети подключаются к единой среде передачи данных, например, к коаксиальному кабелю. Слабое место такой сети — сама среда передачи данных: обрыв кабеля означает сбой всей сети. Пример сети на общей шине — Ethernet на базе коаксиала;
- *древовидная* (рис. 2.5) — топологию этой сети проще представить, чем описать или вникать в определение. В древовидной сети есть более двух конечных узлов и, по крайней мере, два промежуточных узла. В древовидной сети между двумя узлами есть только один путь. Чтобы вникнуть в правильное определение древовидной сети, нужно знать теорию графов, поскольку древовидная сеть — это неориентированный ациклический граф, не содержащий замкнутых путей и позволяющий соединить единственным образом пару узлов;

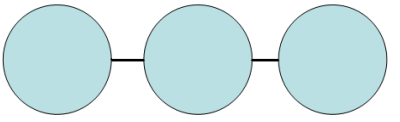


Рис. 2.1. Линейная топология

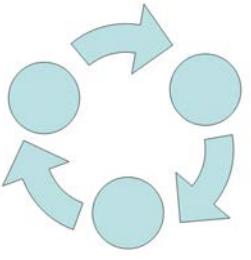


Рис. 2.2. Кольцевая топология

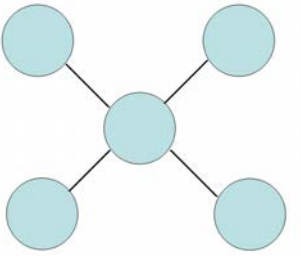


Рис. 2.3. Звезда

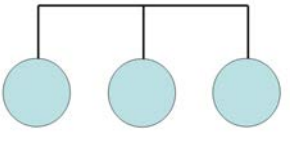


Рис. 2.4. Общая шина

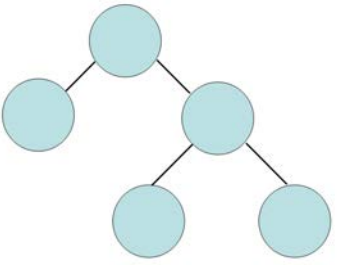


Рис. 2.5. Древовидная топология

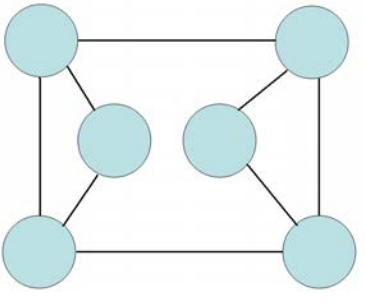


Рис. 21.6. Ячеистая топология

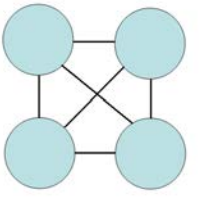


Рис. 2.7. Полносвязная топология

- *ячеистая* (рис. 2.6) — в такой сети есть, по крайней мере, два узла, имеющих два или более пути между ними;
- *полносвязная* (рис. 2.7) — сеть, в которой есть связь между любыми двумя узлами. Это самая надежная топология сети, но она практически никогда не используется, поскольку является самой дорогой и труднообслуживаемой.

2.2.3. По ведомственной принадлежности

По ведомственной принадлежности различают следующие виды сетей:

- *ведомственные* — принадлежат какой-то организации и находятся на ее территории;
- государственные — используются в госструктурах.

2.2.4. По скорости передачи данных

По скорости передачи данных сети делятся на низко-, средне- и высокоскоростные. Основной критерий разделения — скорость передачи данных. Понятно, что скорость передачи данных — понятие непостоянное. То, что сегодня считается среднескоростным соединением, завтра будет отнесено к низкоскоростным. Тем не менее, сегодня низкоскоростной сетью считается сеть со скоростью передачи информации до 10 Мбит/с. Среднескоростная сеть передает данные со скоростью до 100 Мбит/с, а высокоскоростные сети передают информацию со скоростью свыше 100 Мбит/с.

2.2.5. По типу среды передачи данных

Казалось бы, тут все просто: сети бывают проводными или беспроводными. Но очень часто "в природе" встречаются *смешанные*, или *гибридные* сети, сочетающие как проводные, так и беспроводные технологии. В этой книге больше внимания будет уделяться именно таким сетям. Представьте, что у вас дома есть компьютер и ADSL-модем. Вы купили ноутбук. Подключать ноутбук по Ethernet-кабелю не очень-то хочется — ноутбук по своей природе мобильное устройство, и хотелось бы использовать его по всей квартире (а если у вас свой дом, то и во дворе). Поэтому вам понадобится *точка доступа*, которую вы подключите к существующей сети. Точка доступа в данном случае будет выполнять функцию моста между беспроводной и проводной сетью. Именно с ее помощью ваш ноутбук сможет подключиться к Интернету. Поэтому у вас дома появится смешанная сеть, созданная своими руками.

2.2.6. По способу организации взаимодействия компьютеров

Сети бывают одноранговыми и типа клиент/сервер. В *одноранговой* сети нет выделенного сервера: каждый клиент сети может выступать как в роли сервера, то есть предоставлять услуги другим узлам сети, так и в роли клиента, то есть пользоваться услугами, которые предоставляют другие узлы сети.

В сети *клиент/сервер* есть выделенный сервер, предоставляющий определенные сетевые услуги (какие именно, зависит от самой сети).

2.3. Способы передачи данных в сетях

Любая сеть данных должна использовать какой-нибудь метод коммутации своих абонентов, то есть сеть должна знать, как отправить данные тому или иному ком-

пьютеру. В современных сетях распространены три основных метода коммутации: коммутация каналов, коммутация сообщений и коммутация пакетов.

Коммутация каналов используется в аналоговых (нецифровых) телефонных сетях. Для передачи компьютерных данных используется *коммутация пакетов*. Разница между этими методами просто огромна. В первом случае (коммутация каналов) для передачи данных нужен физический канал между двумя узлами. Понятно, что прокладывать кабель между каждой парой узлов сети (между каждой парой телефонов) экономически нецелесообразно, поэтому были созданы *коммутаторы* (сейчас мы говорим о телефонных коммутаторах), соединяющие между собой двух разных абонентов сети по их вызову.

В компьютерных сетях такой способ коммутации совершенно не годится, поскольку канал большую часть времени будет простаивать и без пользы занимать ресурсы коммутатора. Кроме того, при отправке большого количества информации по такой сети на коммутатор ляжет огромная нагрузка, поскольку данные будут переданы за один раз.

Метод коммутации пакетов заключается в том, что передаваемые данные разбиваются на части — *пакеты*. Каждый пакет отправляется отдельно, и, что интересно, два разных пакета, отправленные одним отправителем, могут дойти к получателю разными маршрутами. Например, вы отправляете пакеты компьютеру, не принадлежащему вашей сети. Сначала пакеты отправятся провайдеру, затем — какому-нибудь маршрутизатору Интернета, но если этот маршрутизатор окажется недоступен (мало ли чего может случиться), автоматически будет задействован резервный канал, отправляющий данные через другой маршрутизатор. В итоге получится, что первый пакет будет доставлен одним маршрутом, а второй — другим. Однако оба пакета будут доставлены получателю.

К тому же метод коммутации пакетов позволяет использовать физически одну и ту же среду передачи данных (читайте — один и тот же кабель) для (почти) одновременной отправки данных несколькими компьютерами. Рассмотрим ситуацию: у вас в квартире установлено два параллельных телефонных аппарата, и вы разговариваете по одному из них. Если кто-то поднимет трубку второго телефона, то не сможет набрать номер, поскольку среда передачи информации (телефонный кабель) занята.

В случае с коммутацией пакетов такого нет — в сетях с архитектурой "общая шина" (Ethernet) данные отправляются почти одновременно. Например, компьютерам А и Б нужно отправить данные. Допустим, первым получил доступ к общей среде компьютер А. Он отправляет пакет фиксированного размера. Пока компьютер А отправляет пакет, компьютер Б ожидает доступ к среде. После отправки пакета компьютером А компьютер Б сможет получить доступ к общей среде и отправить свой пакет. Компьютер А в это время делает небольшую паузу. Потом компьютер Б должен сделать паузу, за время которой компьютер А успеет передать следующий пакет. Сами понимаете, время ожидания настолько мизерно, что пользователям компьютеров А и Б кажется, что компьютеры отправляют данные одновременно. Если бы в компьютерной сети использовался метод коммутации каналов, то компьютер Б должен был ждать, пока компьютер А не передаст все данные.

Метод *коммутации сообщений* в чистом виде практически нигде не используется, но он послужил прототипом для метода коммутации пакетов.

2.4. Модель OSI

В 80-х годах прошлого века появилась необходимость стандартизировать различные сетевые технологии. Ведь без стандартизации в мире компьютерных сетей воцарился бы хаос: оборудование различных производителей не смогло бы работать вместе. Поэтому международная организация по стандартизации (International Organization for Standardization, IOS) разработала *модель взаимодействия открытых систем* (Open System Interconnection, OSI). Вы также можете встретить другие названия этой модели: *семиуровневая модель OSI*, или просто *модель OSI*. Эта модель предусматривает семь уровней взаимодействия систем:

- ❒ Физический.
- ❒ Канальный.
- ❒ Сетевой.
- ❒ Транспортный.
- ❒ Сеансовый.
- ❒ Представительный.
- ❒ Прикладной.

Зачем нужна такая система? Предположим, что нам необходимо заставить вместе работать две сети, использующие разную среду передачи данных, — например, витую пару и радиоволны (беспроводную сеть). Если бы не было модели OSI, то для каждой сети пришлось бы разрабатывать модель взаимодействия, а потом придумывать способ, позволяющий заставить две разные по своей архитектуре сети работать вместе. В случае с моделью OSI не нужно "изобретать велосипед" заново. Следует взять за основу уже имеющуюся сеть (в данном случае Ethernet) и переписать физический уровень. В итоге нам не придется разрабатывать браузеры, почтовые клиенты и другие сетевые приложения для каждой сети — браузеру все равно, какая среда передачи данных используется. Как видите, модель OSI хоть и теоретическая, зато очень полезная. Рассмотрим ее уровни:

- ❒ на **физическом уровне** определяются характеристики электрических сигналов, то есть описывается физическая среда данных. На этом уровне и происходит физическая передача данных по кабелю или радиоволнам (в зависимости от типа сети). Пример протокола физического уровня: 1000Base-T — спецификация Ethernet, передающая данные по витой паре 5-й категории (о категориях витой пары мы поговорим позднее) со скоростью 1000 Мбит/с;
- ❒ **канальный уровень** используется для передачи данных между компьютерами (и другими устройствами, например, сетевыми принтерами) одной сети. Пример протокола канального уровня: PPP (Point-to-Point Protocol). Топология сети (шина, звезда и т. д.) определяется как раз на канальном уровне (ранее мы подробно рассмотрели все используемые топологии сетей). На канальном уровне все передаваемые данные разбиваются на части, называемые *кадрами* (frames). Канальный уровень передает кадры физическому уровню, а тот, в свою очередь, отправляет их в сеть.

  На канальном уровне вводится понятие MAC-адреса. *MAC-адрес* — это уникальный аппаратный адрес сетевого устройства (например, сетевого адаптера, точки доступа). Каждому производителю сетевых устройств выделяется свой диапазон MAC-адресов. В мире нет двух сетевых устройств с одинаковыми MAC-адресами;

- теперь рассмотрим **сетевой уровень**. Он используется для объединения нескольких сетей, то есть для организации межсетевого взаимодействия, — ведь протоколы канального уровня могут работать только в пределах одной сети. Канальный уровень не может передать кадр компьютеру, который находится в другой сети. Понятно, что если бы у нас был только канальный уровень и не было сетевого, мы не смогли бы передавать данные между двумя сетями, например, между локальной сетью и Интернетом. Пример протокола сетевого уровня: IP (Internet Protocol). Конечно, IP — это не единственный протокол сетевого уровня, но в этой книге мы будем рассматривать только TCP/IP-сети, поэтому нет смысла упоминать другие протоколы.

  При всем своем желании мы не можем построить огромную сеть, охватывающую весь мир (даже если бы это и удалось, не думаю, что такая сеть работала бы быстро). Поэтому Интернет состоит из совокупности различных сетей, которые объединяются в одно целое маршрутизаторами. Расстояние между сетями исчисляется в количестве переходов пакетов (на сетевом уровне передаваемые данные разбиваются именно на пакеты) через маршрутизаторы. Единица такого перехода называется *хопом* (от англ. hop). Количество хопов равно количеству маршрутизаторов между двумя сетями. Например, от моего узла до узла **volia.net** 6 хопов (шесть переходов), что показано на рис. 2.8;

```
denis@denis-desktop:~$ traceroute volia.net
traceroute to volia.net (82.144.192.47), 30 hops max, 40 byte packets
 1  router.shtorm.net (195.62.14.2)  0.330 ms  0.306 ms  0.295 ms
 2  border.shtorm.com (195.62.14.7)  0.523 ms  0.515 ms  0.504 ms
 3  194.44.13.13 (194.44.13.13)  9.435 ms  9.426 ms  9.415 ms
 4  volia-10G-gw.ix.net.ua (195.35.65.221)  9.618 ms  9.613 ms  9.603 ms
 5  v109.TenGig3-8.diamond.volia.net (82.144.193.192)  9.358 ms  9.562 ms  9.554
 ms
 6  tower.volia.net (82.144.192.47)  9.538 ms  9.251 ms  9.240 ms
denis@denis-desktop:~$
```

Рис. 2.8. Количество переходов от моего узла до узла **volia.net**

- **транспортный уровень** отвечает за доставку пакетов получателю. Не секрет, что при передаче по каналам связи данные могут быть повреждены или вовсе потеряны. Гарантирует доставку пакета в первозданном виде именно транспортный уровень. На этом уровне осуществляются обнаружение и исправление ошибок, восстановление прерванной связи и некоторые дополнительные сервисы вроде срочной доставки (приоритет пакета) и мультиплексирование нескольких соединений. Самым известным и распространенным протоколом транспортного уровня является TCP (Transport Control Protocol);
- **сеансовый уровень** отвечает за установку и за разрыв соединения между компьютерами. На этом уровне также предоставляются средства синхронизации. Сеанс сетевого уровня заключается в установке соединения (при установке стороны, между которыми будут передаваться данные, могут договариваться о дополнительных параметрах связи, например, обмениваться ключами), передаче информации и разрыве соединения.

Многие часто путают сеансы сетевого уровня и сеанс связи. Вы можете установить сеанс связи (например, подключиться к Интернету), но не устанавливать логического соединения, то есть не запустить браузер для соединения с удаленным узлом;

- как было отмечено чуть ранее, на сеансовом уровне стороны могут договориться о дополнительных параметрах, были также упомянуты *ключи*. Однако само шифрование и дешифрование данных осуществляется **представительным уровнем**. Пример протокола этого уровня — SSL (Secure Socket Layer);
- последний (самый высокий) уровень — **прикладной**. На этом уровне работает множество разных протоколов, например, HTTP (Hyper Text Transfer Protocol), FTP (File Transfer Protocol), SMTP (Simple Mail Transfer Protocol) и т. д.

2.5. Что такое протокол?

В этой главе довольно часто упоминалось слово "протокол". *Протокол* — это правила, определяющие взаимодействие компьютеров в вычислительной сети. Рассмотрим несколько самых важных протоколов.

В основе Интернета лежит протокол TCP/IP (Transmission Control Protocol/ Internet Protocol). Чтобы система смогла работать в Интернете, она должна поддерживать протокол TCP/IP. Вообще говоря, TCP/IP — это совокупность двух протоколов. Протокол TCP, как уже было отмечено ранее, отвечает за корректность передачи данных по Интернету (точнее, по любой сети, использующий этот протокол), то есть гарантирует доставку данных по сети. Протокол IP используется для адресации компьютеров сети. Дело здесь в том, что у каждого компьютера сети имеется свой уникальный адрес (IP-адрес) и, чтобы передать данные компьютеру, нужно его IP-адрес знать. Чуть позже мы поговорим о системе доменных имен (DNS, Domain Name System).

Кроме протоколов TCP и IP посетители Интернета работают с теми или иными серверами, использующими следующие протоколы:

- HTTP (Hyper Text Transfer Protocol) — протокол передачи гипертекста. Все Web-серверы Интернета используют протокол HTTP или его безопасную версию HTTPS (HTTP Secure);
- FTP (File Transfer Protocol) — протокол передачи файлов. Используется для обмена файлами между компьютерами. Вы можете подключиться к FTP-серверу и скачать необходимые вам файлы или же, наоборот, закачать свои файлы на сервер, если вы обладаете надлежащими правами доступа к серверу. В Интернете много публичных FTP-серверов, к которым разрешен анонимный доступ. Как правило, с таких серверов скачивать файлы разрешено всем желающим. Иногда, но очень редко, разрешается и запись файлов на публичный сервер. В состав любой операционной системы входит FTP-клиент — программа ftp. К тому же многие браузеры можно использовать в качестве FTP-клиента;
- SMTP (Simple Mail Transfer Protocol) — простой протокол передачи почты. Используется для отправки почты (e-mail);

- POP (Post Office Protocol) — протокол, используемый для получения сообщений электронной почты;
- IMAP (Internet Message Access Protocol) — еще один протокол для получения почты, но, в отличие от протокола POP, этот протокол позволяет читать почту без ее загрузки на компьютер пользователя. Протокол IMAP, по сути, намного удобнее, чем POP. Ведь почта хранится на сервере, и вы можете получить доступ к ней с любой точки земного шара, используя любой клиент. К тому же IMAP поддерживает поиск писем на сервере, что позволяет найти нужное письмо без загрузки всех писем на свой компьютер. Однако у IMAP есть один существенный недостаток, благодаря чему до сих пор распространен протокол POP — IMAP требует постоянного соединения с сервером. Если нет соединения с сервером, то вы не прочитаете не только новые сообщения, но и те, которые были получены ранее, поскольку все они хранятся на сервере. Так что в случае с IMAP об автономной работе (без подключения к Интернету) можно забыть.

Примечание

На днях полностью "перекочевал" в Linux (после очередного "падения" Windows). Так вот, до сих пор не знаю, как перенести в Linux 20 тыс. сообщений (разбитых на множество папок со своими правилами сортировки) общим объемом около 4 Гбайт. Почта хранилась в программе The Bat!, а я сейчас использую почтовый клиент Mozilla Thunderbird. Программы для конвертации почтового формата The Bat! в формат Thunderbird пока не разработано… Если бы я использовал протокол IMAP, проблема отпала бы автоматически.

2.6. Адресация компьютеров

Для идентификации узлов Интернета используются IP-адреса. IP-адрес представляет собой четыре числа, разделенные точками (или одно 32-разрядное число, которое записывается в виде четырех восьмиразрядных чисел, разделенных точками, — как кому больше нравится). Нужно сразу отметить, что такая идентификация неоднозначная, поскольку IP-адреса могут быть статическими (постоянными) и динамическими. *Постоянные* (статические) IP-адреса обычно назначаются серверам, а *динамические* — обычным пользователям. Так что сегодня определенный динамический IP-адрес может быть назначен одному пользователю, а завтра — другому. Поэтому если в случае с аппаратными MAC-адресами еще можно говорить о какой-то однозначности (и то существуют способы подделки MAC-адресов), то IP-адреса по определению однозначными не являются.

Рассмотрим примеры IP-адресов: 127.0.0.1, 192.168.1.79, 111.33.12.99. Как было сказано ранее, IP-адрес — это одно 32-разрядное число или четыре 8-разрядных числа. Возведем 2 в восьмую степень и получим максимальное значение для каждого из четырех восьмиразрядных чисел — 256. Таким образом, учитывая, что некоторые IP-адреса зарезервированы для служебного использования, протокол IP может адресовать примерно 4,3 млрд узлов. Однако с каждым годом количество узлов во Всемирной паутине увеличивается, поэтому была разработана шестая версия протокола IP — IPv6 (если упоминается просто протокол IP, то, как правило, имеется в виду четвертая версия протокола — IPv4). Новый протокол использует 128-битные адреса (вместо 32-битных), что позволяет увеличить число узлов

до 10^{12} и количество сетей до 10^{9}. IPv6-адреса отображаются как 8 групп шестнадцатеричных цифр, разделенных двоеточиями. Вот пример адреса нового поколения: 1628:0d48:12a3:19d7:1f35:5a61:17a0:765d.

> ***Примечание***
>
> Впрочем, массовый переход на IPv6 (который еще называют IPng — IP Next Generation) пока так и не состоялся, хотя его используют несколько сотен сетей по всему миру. В этой книге мы будем рассматривать только протокол IPv4, поскольку, судя по всему, Интернет не перейдет на IPv6 в ближайшие несколько лет. Интересующиеся могут прочитать об IPv6 по адресу: **http://ru.wikipedia.org/wiki/IPv6**.

IP-адреса выделяются *сетевым информационным центром* (NIC, Network Information Center). Чтобы получить набор IP-адресов для своей сети, вам надо обратиться в этот центр. Но, оказывается, это приходится делать далеко не всем. Существуют специальные IP-адреса, зарезервированные для использования в локальных сетях. Ни один узел глобальной сети (Интернета) не может обладать таким "локальным" адресом. Вот пример локального IP-адреса — 192.168.1.1. В своей локальной сети вы можете использовать любые локальные IP-адреса без согласования с кем бы то ни было. Когда же вы надумаете подключить свою локальную сеть к Интернету, вам понадобится всего один "реальный" IP-адрес — он будет использоваться на маршрутизаторе (шлюзе) доступа к Интернету.

Чтобы узлы локальной сети (которым назначены локальные IP-адреса) смогли "общаться" с узлами Интернета, используется специальная технология *трансляции сетевого адреса* (NAT, Network Address Translation). Маршрутизатор получает пакет от локального узла, адресованный интернет-узлу, и преобразует IP-адрес отправителя, заменяя его своим IP-адресом. При получении ответа от интернет-узла маршрутизатор выполняет обратное преобразование, поэтому нашему локальному узлу "кажется", что он общается непосредственно с интернет-узлом. Если бы маршрутизатор отправил пакет как есть, то есть без преобразования, то его отверг бы любой маршрутизатор Интернета, и пакет так и не был бы доставлен к получателю.

Наверное, вам не терпится узнать, какие IP-адреса можно использовать без согласования с NIC? Об этом говорить пока рано — ведь мы еще ничего не знаем о *классах* сетей. IP-адреса используются не только для адресации отдельных компьютеров, но и целых сетей. Вот, например, IP-адрес сети — 192.168.1.0. Отличительная черта адреса сети — 0 в последнем октете.

Сети поделены на классы в зависимости от их размеров:

- класс A — огромные сети, которые могут содержать 16777216 адресов, IP-адреса сетей лежат в пределах 1.0.0.0 — 126.0.0.0;
- класс B — средние сети, содержат до 65536 адресов. Диапазон адресов — от 128.0.0.0 до 191.255.0.0;
- класс C — маленькие сети, каждая сеть содержит до 256 адресов.

Существуют еще и классы D и E, но класс E не используется, а зарезервирован на будущее (хотя будущее — это IPv6), а класс D зарезервирован для служебного использования (широковещательных рассылок).

Представим ситуацию. Вы хотите стать интернет-провайдером. Тогда вам нужно обратиться в NIC для выделения диапазона IP-адресов по вашу сеть. Скажем, вы планируете сеть в 1000 адресов. Понятно, что сети класса C вам будет недостаточно.

Поэтому можно или арендовать четыре сети класса C, или одну класса B. Но, с другой стороны, 65536 адресов для вас — много, и если выделить вам всю сеть класса B, то это приведет к нерациональному использованию адресов. Так что самое время поговорить о *маске сети*. Маска сети определяет, сколько адресов будет использоваться сетью, фактически — маска задает размер сети. Маски полноразмерных сетей классов A, B и C представлены в табл. 2.1.

Таблица 2.1. *Маски сетей классов A, B и C*

Класс сети	**Маска сети**
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

Маска 255.255.255.0 вмещает 256 адресов (в последнем октете IP-адреса могут быть цифры от 0 до 255). Например, если адрес сети 192.168.1.0, а маска 255.255.255.0, то в сети могут быть IP-адреса от 192.168.1.0 до 192.158.1.255. Первый адрес (192.168.1.0) называется IP-адресом сети, последний — зарезервирован для широковещательных рассылок. Следовательно, для узлов сети остаются 254 адреса — от 192.168.1.1 до 192.168.1.254.

А вот пример маски сети на 32 адреса: 255.255.255.224 (255 – 224 = 31 + "нулевой" IP-адрес, итого 32).

Предположим, у нас есть IP-адрес произвольной сети, например, 192.168.1.0. Как узнать, к какому классу она принадлежит? Для этого нужно преобразовать первый октет адреса в двоичное представление. Число 192 в двоичной системе будет выглядеть так: **110**00000. Проанализируем первые биты первого октета. Если первые биты содержат двоичные цифры 110, то перед нами сеть класса C. Теперь проделаем то же самое с сетью 10.0.0.0. Первый октет равен 10, и в двоичной системе он будет выглядеть так: 00001010. Первый бит — 0, поэтому сеть относится к классу A. Опознать класс сети по первым битам первого октета поможет табл. 2.2.

Таблица 2.2. *Опознание класса сети*

Класс сети	**Первые биты**
A	0
B	10
C	110
D	1110
E	11110

Теперь поговорим о специальных зарезервированных адресах. Адрес 255.255.255.255 является *широковещательным*. Если пакет отправляется по этому адресу, то он будет доставлен всем компьютерам, находящимся с отправителем в одной сети. Можно уточнить сеть, компьютеры которой должны получить широковещатель-

ную рассылку, например, таким образом: 192.168.5.255. Этот адрес означает, что пакет получат все компьютеры сети 192.168.5.0.

Вам также следует знать адрес 127.0.0.1. Этот адрес зарезервирован для обозначения локального компьютера и называется *адресом обратной петли*. Если отправить пакет по этому адресу, то его получит ваш же компьютер, то есть получатель является отправителем, и наоборот. Данный адрес обычно используется для тестирования поддержки сети. Более того, к локальному компьютеру относится любой адрес из сети класса A с адресом 127.0.0.0. Поэтому при реальной настройке сети нельзя использовать IP-адреса, начинающиеся со 127.

А теперь можно рассмотреть IP-адреса сетей, зарезервированные для локального использования. В локальных сетях вы можете использовать следующие адреса сетей:

- 192.168.0.0 — 192.168.255.0 — сети класса C (всего 256 сетей, маска 255.255.255.0);
- 172.16.0.0 — 172.31.0.0 — сети класса B (всего 16 сетей, маска 255.255.0.0);
- 10.0.0.0 — сеть класса A (одна сеть, маска 255.0.0.0).

Обычно в небольших домашних и офисных сетях используются IP-адреса из сети класса C, то есть из диапазона 192.168.0.0 — 192.168.255.0. Но поскольку назначение адресов контролируется только вами, вы можете назначить в своей локальной сети любые адреса, например, адреса из сети 10.0.0.0, даже если у вас в сети всего 5 компьютеров. Так что выбор сети — это дело вкуса. Можете себя почувствовать администратором огромной сети и использовать адреса 10.0.0.0.

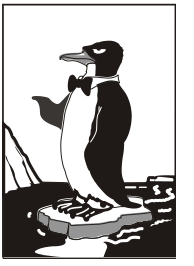
2.7. Система DNS

Узлов в Интернете достаточно много, поэтому ни один человек не способен запомнить IP-адреса всех необходимых ему узлов. Да и гораздо легче запомнить символьный адрес, скажем, **www.bhv.ru** или **www.dkws.org.ua**, чем их IP-адреса. Тем более, относительно недавно появилась возможность регистрации доменных имен на русском языке (точнее — на кириллице). Не знаю, приживутся ли такие доменные имена, но то, что они существуют, — это факт.

За преобразование IP-адресов в доменные имена и обратно отвечает *система доменных имен* (DNS, Domain Name System). Когда вы вводите доменное имя в строке браузера, система сначала разрешает это имя в IP-адрес (путем обращения к DNS-серверу), а потом подключается к узлу по полученному IP-адресу.

Не нужно думать, что система DNS появилась недавно. Она более "древняя", чем вы можете предположить. Впервые DNS была представлена в 1984 году. Правда, тогда далеко не все сети перешли на использование DNS-серверов. До этого доменные имена разрешались в IP-адреса с помощью файла hosts, в котором содержалась таблица соответствия доменных имен IP-адресам. Понятно, что такой файл нужно постоянно поддерживать в актуальном состоянии. Когда количество узлов увеличилось и поддержка этого файла стала проблемой для администратора сети, вот тогда и началась эра DNS. Кстати, файлом hosts можно пользоваться до сих пор. Для обеспечения совместимости его можно использовать даже в самых современных ОС (как в UNIX/Linux, так и в Windows), но, сами понимаете, происходит это очень редко.

Система DNS более подробно будет рассмотрена в *главе 27*. Мы даже настроим собственный DNS-сервер.



Глава 3

Основные сетевые устройства

3.1. Активное и пассивное сетевое оборудование

Для построения компьютерной сети, то есть для организации передачи информации между компьютерами, используется сетевое оборудование. Сетевое оборудование бывает активным и пассивным. *Активным* называется оборудование, обладающее неким "интеллектом" — например, коммутатор (switch), маршрутизатор (router). *Пассивное* сетевое оборудование "интеллектом" не наделено. К пассивному оборудованию относят кабели (например, коаксиальный или витая пара), розетки (RJ45, RG58 и др.), повторитель (repeater), концентратор (hub) и т. д.

Стоп! Если вы хоть немного знакомы с Ethernet-сетями, вы можете запутаться. Ведь концентратор, как и коммутатор, можно использовать в качестве центрального сетевого устройства в Ethernet-сети, почему тогда концентратор — это пассивное устройство, а коммутатор — активное? Дело в том, что концентратор не проявляет никакой интеллектуальной деятельности — он просто получает сигналы и копирует (повторяет) их на все свои порты, равно как и повторитель. Повторитель получает сигнал, усиливает его и повторяет на другой порт. Повторители обычно используются для увеличения дальности передаваемого сигнала. Коммутатор же "знает", к какому порту подключен какой компьютер, поэтому передает полученный сигнал не на все порты, а только на определенный порт, к которому подключен компьютер-назначение.

Различного сетевого оборудования очень много. Мы не будем пытаться объять необъятное, поэтому в этой книге рассмотрим только оборудование, необходимое для построения проводных Ethernet-сетей и беспроводных сетей Wi-Fi.

3.2. Оборудование, необходимое для построения Ethernet-сети

Для организации современной Ethernet-сети (имеются в виду спецификации Fast Ethernet и Gigabit Ethernet) необходим всего один коммутатор (switch). Конечно, если сеть большая, то понадобится несколько коммутаторов, общее количество портов которых сможет обеспечить подключение всех узлов сети. На рис. 3.1 изображен так называемый *промышленный* коммутатор от Linksys.

Рис. 3.1. 16-портовый коммутатор от Linksys

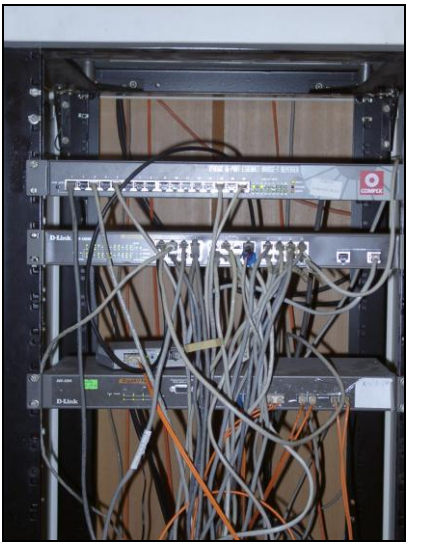


Рис. 3.2. Стойка с коммутаторами

Рис. 3.3. Шкаф с сетевым оборудованием

Рис. 3.4. 8-портовый гигабитный коммутатор от D-Link

Дизайн корпуса промышленного коммутатора обычно не очень эффектен, но сделано это умышленно — чтобы коммутатор можно было поместить в стойку сетевого оборудования. Ведь в больших корпоративных сетях обычно несколько коммутаторов, которые помещаются в специальную стойку (или в специальный шкаф сетевого оборудования, который можно закрыть и тем самым ограничить физический доступ к нему). На рис. 3.2 изображена типичная стойка с коммутаторами.

А на рис. 3.3 показан шкаф с коммутаторами. Такой шкаф может быть большего размера и содержать другое оборудование (например, серверы сети), но главное отличие шкафа от стойки — наличие двери, которая ограничивает доступ к сетевому оборудованию.

Если вы хотите построить небольшую домашнюю или офисную сеть, то можете выбрать коммутатор с более интересным дизайном, который лучше впишется в ваш интерьер. На рис. 3.4 представлен 8-портовый гигабитный коммутатор от D-Link. Вид у него более "дружелюбный", но в стойку его уже не поместишь (впрочем, при организации домашней сети никакой стойки у вас и не будет).

Давайте теперь уточним, почему в современных сетях не стоит использовать концентраторы (hub). Представим, что у нас есть сеть на четыре компьютера. Назовем их А, Б, В и Г. Пусть компьютер А отправляет данные компьютеру Г. Концентратор отправит полученный от компьютера А сигнал на все свои порты — то есть сигнал, отправленный компьютером А, получат все компьютеры сети. Затем каждый компьютер анализирует заголовки пакета, в которых указан компьютер-получатель. Если адрес компьютера совпадает с адресом получателя, компьютер принимает пакет, в противном случае — игнорирует его. Таким образом, использование концентратора приводит к "брожению" по сети паразитного трафика. По сути, концентратор — это обычный многопортовый повторитель (усилитель) сигналов. И чем больше сеть, тем медленнее она работает в случае использования концентратора, поскольку "брожение" паразитного трафика носит лавинообразный характер. Вы только представьте, что в сети не четыре компьютера, а несколько десятков… Поэтому в больших сетях концентраторы существенно снижают производительность сети.

Коммутатор же, в отличие от концентратора, строит специальную таблицу соответствия, позволяющую однозначно узнать, к какому порту какой компьютер подключен (см., например, табл. 3.1).

Таблица 3.1. *Таблица соответствия портов коммутатора и адресов компьютеров*

Номер порта	Адрес компьютера
1	Б
2	А
3	Г
4	В

Когда компьютер А, подключенный ко второму порту коммутатора, отправляет пакет компьютеру Г, коммутатор знает, что компьютер Г подключен к третьему порту, и отправляет пакет только на третий порт. При этом снижается нагрузка на сеть, потому что компьютеры не получают "лишних" пакетов.

Кроме того, поскольку концентратор отправляет данные каждому компьютеру сети, становится очень простым перехват данных. Существуют специальные программы, переводящие сетевой адаптер компьютера в режим мониторинга, в котором он осуществляет принятие всех данных, даже тех, которые не адресованы этому компьютеру. Поэтому, если в сети используется концентратор, все передаваемые данные становятся общим достоянием — их может перехватить любой компьютер, подключенный к концентратору.

Итак, использование коммутатора позволяет повысить производительность сети и повысить ее безопасность. Ранее сети в основном строились на базе концентрато-

ров, поскольку их стоимость была существенно ниже стоимости коммутаторов. Со снижением цен на коммутаторы концентраторы практически исчезли с магазинных полок. Однако в некоторых старых сетях они еще используются. Если вам придется обслуживать такую сеть, первым делом замените концентратор на коммутатор — вы сразу почувствуете разницу.

Какой коммутатор применить: Fast Ethernet (100Base-T) или Gigabit Ethernet (1000Base-T)? В первом случае максимальная (теоретическая) скорость передачи данных составляет 100 Мбит/с, во втором случае — 1000 Мбит/с. Коммутаторы Gigabit Ethernet стоят немного дороже (цены приводить не буду, поскольку через год они станут еще доступнее, а через два — о Fast Ethernet забудут, как в свое время забыли о коаксиале и концентраторах).

Учитывая, что сеть строится не на день и не на два, лучше выбрать Gigabit Ethernet. С точки зрения монтажа сети ничего не изменится — даже если вы сейчас установите коммутатор Fast Ethernet, то завтра без проблем сможете заменить его на Gigabit Ethernet. Но нужно помнить следующее: чтобы сеть работала в режиме 1000Base-T, необходимо, чтобы 1000Base-T поддерживали сетевые адаптеры компьютеров. Практически на всех современных материнских платах встроенные сетевые адаптеры уже поддерживают 1000Base-T, но если в вашей сети есть компьютеры, которым 2–3 года, скорее всего, вам придется докупать для них сетевые адаптеры с поддержкой 1000Base-T.

Идем дальше — количество портов. Обычно в продаже есть коммутаторы на 5, 8, 16, 24 порта. Промышленные коммутаторы могут иметь большее число портов, например 32 или 48. Может быть, в скором времени появятся коммутаторы с еще большим числом узлов, но я сомневаюсь. Поскольку обычно один коммутатор обслуживает одну подсеть, я не думаю, что в одной подсети будет больше 48 компьютеров. А если это случится, такую подсеть желательно (из соображений локализации трафика) разделить на несколько подсетей с меньшим числом компьютеров.

Так что для домашней сети покупайте коммутатор, способный подключить все имеющиеся дома компьютеры, — большой запас портов вам вряд ли понадобится. Обычно в домашней сети 2–4 компьютера. В этом случае вам будет достаточно 5-портового коммутатора — 5-й порт пригодится для подключения этого коммутатора к другому коммутатору сети. В коммутаторах с большим числом портов для подключения к другому коммутатору обычно используется один из имеющихся портов (например, порт 1).

Промышленные коммутаторы иногда имеют так называемый *магистральный* порт. Например, 16 портов, работающих в режиме 100Base-T, и один порт, работающий в режиме 1000Base-T, — для подключения к магистрали сети, работающей со скоростью 1000 Мбит/с. Иногда вместо порта 1000Base-T оборудуется оптоволоконный порт, например, 100Base-FB. В этом случае скорость магистрали такая же, как и скорость сети, но расстояние передачи сигнала намного выше (более 2 км), что позволяет использовать оптоволоконный кабель для соединения сетей двух (или более) зданий в одну большую сеть.

В случае с офисной сетью количество портов коммутатора должно в два раза превышать количество компьютеров сети. Например, если в вашей сети четыре компьютера, то нужен 8-портовый коммутатор. Дополнительные четыре порта

могут понадобиться, если придется подключить дополнительные компьютеры, например, ноутбуки ваших клиентов, если у вас пока еще нет для них точки доступа Wi-Fi.

По большому счету, для организации сети больше ничего и не нужно (разумеется, кроме кабеля и коннекторов RJ45, но это уже детали, о которых мы поговорим в *третьей части* книги).

3.3. Оборудование, необходимое для построения сети Wi-Fi

Как и в случае с Ethernet-сетью, нам понадобятся сетевые адаптеры и центральное устройство сети. Только сетевые адаптеры нужны не обычные, а беспроводные. А роль центрального устройства сети будет играть *точка доступа* (access point).

Все современные модели ноутбуков по умолчанию оснащены адаптером Wi-Fi, а вот стационарные (настольные) компьютеры придется дооснастить беспроводными сетевыми адаптерами. Проще всего купить беспроводной адаптер, подключающийся к компьютеру по USB. Есть также адаптеры, выполненные в виде PCI-карты, устанавливаемой в свободный PCI-слот компьютера. Такие адаптеры используются редко, поскольку их установка требует вскрытия корпуса компьютера, что несколько неудобно (особенно, если компьютер еще на гарантии — тогда придется нести его в сервисный центр, а что делать, если таких компьютеров много?).

USB-адаптеры могут быть выполнены в разных корпусах. На рис. 3.5 изображен небольшой беспроводной адаптер, напоминающий по своим размерам флешку. У такого адаптера антенна встроенная, поэтому его можно использовать только, если компьютер находится в зоне уверенного приема. Если же компьютер установлен ближе к "мертвой" зоне, лучше выбрать адаптер, выполненный в виде отдельного устройства (рис. 3.6). Такой адаптер обычно имеет небольшой размер и подключается к компьютеру USB-кабелем (питание адаптер получает тоже по USB). Преимущество этого адаптера заключается в следующем — его можно легко передвинуть в пределах длины USB-кабеля, чтобы попасть в зону уверенного приема сети. Ноутбук можно легко переместить в эту зону — просто взяли и перенесли. Со стационарным компьютером такого не сделаешь — у каждого стационарного компьютера есть свое место. А что делать, если в том месте, где установлен компьютер, не обеспечивается уверенный прием беспроводных сигналов? Не переносить же компьютер? В этой ситуации поможет адаптер, изображенный на рис. 3.6. Иногда перемещение адаптера всего на несколько сантиметров дает весьма ощутимые результаты. Да и антенна у такого адаптера обладает большей чувствительностью, чем встроенная антенна адаптера, изображенного на рис. 3.5. К тому же к подобным адаптерам (с внешней антенной) обычно можно подключить дополнительную антенну с еще большей чувствительностью. Обо всем этом мы поговорим, когда будем строить свою собственную беспроводную сеть. А сейчас перейдем лучше к точке доступа.

Примечание

При выборе Wi-Fi-адаптера учитывайте наличие драйверов — особенно, если вы планируете использовать его в Linux. Чтобы не получилось так, что Linux не поддержит купленный Wi-Fi-адаптер.

Рис. 3.5. USB Wi-Fi-адаптер со встроенной антенной

Рис. 3.6. USB Wi-Fi-адаптер с внешней антенной

Рис. 3.7. Точка доступа от D-Link с тремя антеннами

Точка доступа (рис. 3.7) выполняет в беспроводной сети роль центрального устройства. Казалось было, все здесь просто: устанавливаем Wi-Fi-адаптеры, подсоединяем точку доступа, и беспроводная сеть готова — беспроводные клиенты могут обмениваться данными. Однако, если вы планируете купить точку доступа прямо сейчас, не следует покупать первую попавшуюся. Сначала желательно определиться, какие функции точки доступа вам нужны, затем "вычислить" модели точек доступа, обеспечивающие необходимые вам функции, и просмотреть в Интернете отзывы об этих моделях. Только так можно выбрать лучшую точку доступа.

Точка доступа может предоставлять дополнительные функции — например, функции *маршрутизатора*. Предположим, у вас дома есть несколько ноутбуков. К одному ноутбуку подключен ADSL-модем. Как организовать общий доступ к Интернету? Покупается точка доступа, к которой этот ADSL-модем и подключается. Ноутбуки (беспроводные клиенты) будут подключаться к Интернету по Wi-Fi, а точка доступа выступит в роли маршрутизатора.

3.4. Дополнительные сетевые устройства

Представим, что у нас есть два (или более) обычных (настольных) компьютера и одно ADSL-соединение. И нужно обеспечить общий доступ к Интернету. Это можно сделать средствами Windows. Тогда в один компьютер надо будет установить дополнительный сетевой адаптер. Первый сетевой адаптер будет использоваться

для подключения к Интернету, а второй — для подключения к локальной сети (для связи с остальными компьютерами сети). Компьютер с двумя сетевыми адаптерами для остальных компьютеров сети будет выполнять роль *шлюза* (gateway). Преимущество такого решения — дешевизна: ведь мы обеспечили общий доступ к Интернету практически без дополнительных устройств. Недостаток заключается в том, что компьютер-шлюз должен быть постоянно включен, иначе остальные компьютеры не смогут подключиться к Интернету.

Решить эту проблему можно, купив отдельное устройство, называемое *маршрутизатором* (при рассмотрении выбора точки доступа мы это устройство уже упоминали). Маршрутизатор обеспечивает передачу пакетов по заданному маршруту. В нашем случае — от локальных компьютеров к интернет-провайдеру. Таким образом, все компьютеры сети будут подключаться к центральному коммутатору, а он, в свою очередь, — к маршрутизатору. Также к маршрутизатору будет подключен и ADSL-модем.

Маршрутизаторы бывают разные. Некоторые могут выполнять роль коммутатора. Купив такой маршрутизатор, вы сократите количество активного сетевого оборудования (а значит, сэкономите деньги) до двух единиц — маршрутизатора и ADSL-модема. Если же у вас в сети компьютеров немного (2–4), можно подыскать ADSL-модем с функциями маршрутизатора. В этом случае у вас будет всего одна "коробочка" — все компьютеры сети будут подключены к этому устройству, которое, в свою очередь, будет подключено к телефонной сети. Этим вы сэкономите еще больше средств. Поэтому очень важно перед построением сети спланировать сей процесс. Хорошее планирование не только позволяет сэкономить деньги, но и время, впоследствии потраченное на дальнейшую модернизацию сети.

А теперь представим, что в нашей сети есть два (или больше, количество — не принципиально) стационарных компьютера и несколько ноутбуков. Ноутбуки было бы хорошо подключать по Wi-Fi. Стационарные компьютеры принято подключать по Ethernet (хотя бы потому, что не хочется покупать для них беспроводные адаптеры). Так вот, можно купить устройство, которое одновременно является ADSL-модемом, беспроводной точкой доступа и коммутатором. Одним из таких устройств является DSL-2640U от D-Link (далее мы рассмотрим процесс настройки этого устройства). Это устройство (рис. 3.8) позволяет объединить в сеть несколько беспроводных клиентов (это наши ноутбуки) и четыре проводных клиента. Все клиенты (как проводные, так и беспроводные) автоматически настраиваются на доступ к Интернету по совместно используемому ADSL-каналу. Кроме того, это

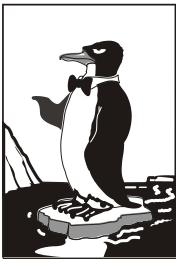
Рис. 3.8. ADSL-модем, маршрутизатор, коммутатор и беспроводная точка доступа D-Link DSL-2640U

устройство обладает встроенным брандмауэром, что позволяет защитить вашу сеть от вторжения извне.

Простота настройки сети с помощью такого устройства просто поражает. Все, что вам нужно — это включить устройство, подключить к нему клиентов, запустить программу настройки (как это сделать, написано в руководстве по устройству) и установить базовые параметры сети, а именно: имя пользователя и пароль для ADSL-соединения, идентификатор беспроводной сети (SSID) и выбрать тип шифрования беспроводных соединений. Вот и все — сеть будет работать. Клиентов можно вовсе не настраивать — они будут автоматически настроены по протоколу DHCP (Dynamic Host Configuration Protocol, протокол динамической настройки узла).

Впрочем, у всех комбинированных устройств есть один недостаток — плохая масштабируемость. Если ваша сеть будет расти, добавить новых клиентов в нее будет сложно, а в некоторых случаях вовсе невозможно. Тогда придется покупать отдельные устройства. Например, коммутатор, к которому будут подключаться до 48 проводных клиентов, и точку доступа для подключения беспроводных клиентов. В свою очередь, точка доступа и коммутатор будут подключаться к ADSL-маршрутизатору. Хотя в сложных случаях целесообразнее использовать программный (не аппаратный) маршрутизатор — компьютер под управлением UNIX/Linux. Такой компьютер можно использовать в роли маршрутизатора и на нем запустить брандмауэр, DNS-, WWW-, FTP- и почтовый серверы.

Итак, в этой главе мы ознакомились с основными сетевыми устройствами. Следующая глава будет сугубо теоретической. Мы поговорим о том, что должен знать каждый администратор и опытный пользователь любой сети, рассмотрим модель OSI и адресацию в TCP/IP-сети.



Глава 4

Планирование сети

4.1. Важность планирования

Вспомним старую русскую пословицу "семь раз отмерь, а один раз отрежь". Она очень точно подходит к нашему случаю. Конечно, бытует мнение, что пока семь раз будете мерить, кто-то уже отрежет. Согласен, но не сейчас. Сейчас вы планируете сеть, вы — главный, и вам никто не мешает. Очень важно продумать все нюансы, связанные с построением сети. Ведь корпоративная сеть — это очень сложная система, состоящая из тысяч различных компонентов. Это в маленькой домашней сети могут быть два-три компьютера, коммутатор, модем и принтер, подключенный к одному из компьютеров (не думаю, что в домашней сети кто-то организует принт-сервер). А в корпоративной сети могут быть самые разнообразные устройства, которые некоторые домашние пользователи даже ни разу в жизни и не видели. Скажем, кто из обычных домашних пользователей видел настоящий *мейнфрейм*, *кластер* или хотя бы обычный *терминал*, подключаемый к мейнфрейму?

Очень важно ориентироваться во всем этом оборудовании. Ведь жизнь не стоит не месте — все развивается с очень большой скоростью, особенно информационные технологии. Модель маршрутизатора, которая была популярна в прошлом году, уже давно такой не является — на ее место пришла новая, с более совершенными функциями, позволяющими эффективнее использовать всю систему в целом. Поэтому прежде чем закупать оборудование для сети, нужно ознакомиться с возможностями самых последних моделей устройств, а также сравнить устройства других производителей. Вот пример: всю жизнь вы считали, что устройства фирмы AAA (не хочется делать никому никакой рекламы — ни хорошей, ни плохой) — лучшие, но вот всего полгода назад на рынке появилась компания BBB, которая начала производство устройств, которые по всем своим характеристикам превосходят устройства компании AAA. Вы привыкли к компании AAA, поэтому всеми правдами и неправдами (мол, устройство от BBB еще не проверены временем и т. д.) будете уговаривать себя остановить свой выбор на устройстве от AAA, хотя прекрасно знаете, что устройство от BBB явно превосходит его характеристиками. С одной стороны, вы правы — проверенные временем, надежные устройства обеспечивают безотказную работу сети. А с другой стороны — нет, ведь уже через полгода все будут пользоваться принципиально новыми устройствами BBB, а вы построили свою сеть на устаревшем оборудовании от AAA.

Интернет внес огромные изменения в корпоративную сеть. Сейчас по каналам Интернета можно передать любую информацию: если раньше преимущественно передавался текст, графика и иногда звук, то сейчас видеоконференции он-лайн — это норма. Кроме того, Интернет можно использовать как компонент корпоративной сети — для передачи корпоративной информации по каналам Интернета: это существенно дешевле, чем прокладывать свои линии связи.

4.1.1. Планирование как основа безопасности

При планировании нужно учитывать еще и *безопасность* сети. Да, это нужно делать именно при планировании, а не после того, как сеть уже построена. Поэтому о безопасности — отдельный разговор.

Небольшой пример уже был приведен ранее: использовать проверенные временем решения или применить новые? Это касается не только оборудования, но и программного обеспечения, которое также является компонентом корпоративной сети, причем очень важным компонентом. Весьма желательно найти золотую середину между проверенными временем решениями и новыми разработками.

В предыдущем разделе мы начали говорить о влиянии Интернета на корпоративную сеть. По данным ISC (**www.isc.org**) в январе 2010 года (более новых данных пока нет, поэтому будем считать, что это последние данные) в Сети насчитывалось более 732 миллионов (!) узлов (рис. 4.1). Сейчас их еще больше. Посмотреть отчет ISC можно по адресу: **https://www.isc.org/solutions/survey**.

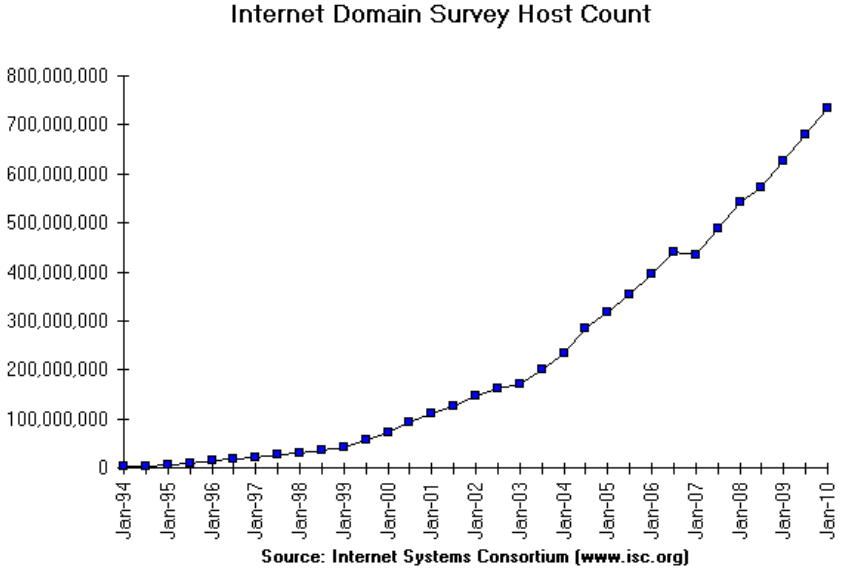


Рис. 4.1. Динамика роста количества узлов (по данным ISC)

Для сравнения: в январе 2009 года было 625 миллионов узлов. За год (с 1 января 2009 года по 1 января 2010 года) добавилось 107 миллионов узлов. Это целая армия

потенциальных клиентов вашей компании. И вам жизненно необходимо предоставить им всевозможную рекламную (и не только) информацию о своем бизнесе. Сейчас практически у каждой компании есть собственный Web-сайт. На сайте можно не только ознакомиться с предлагаемым товаром, но и приобрести его. Понятно, что в данном случае общение с клиентом происходит с помощью Интернета: по электронной почте, ICQ, Skype и т. п. Помимо общения с клиентами, общение с партнерами также происходит по Интернету.

Вот теперь мы подошли к самому главному. Ранее считалось, что 80% трафика корпоративной сети будет внутренним, а только 20% — внешним. Сейчас ситуация в корне изменилась. Точные цифры назвать нельзя, потому что они зависят от специфики работы компании: у кого-то соотношение будет 50/50, а у кого-то 20/80, но точно не 80/20. Поэтому возрастает нагрузка на пограничные маршрутизаторы и брандмауэры, которые уже не так эффективно справляются с поставленными задачами. Так что, приобретая оборудование или программное обеспечение, нужно думать о будущем и покупать с некоторым запасом, чтобы потом, уже через год-два, не перестраивать всю сеть заново.

Развитие Интернета несет в себе потенциальную угрозу для корпоративной сети — чем больше пользователей, тем потенциально больше желающих взломать вашу сеть. Нужно также подумать и о защите информации, передаваемой по каналам Интернета. На этапе планирования важно решить, какое программное обеспечение будет использоваться в сети. Особое внимание следует уделить программам, с помощью которых пользователи вашей сети будут "общаться" с внешним миром. Оно должно поддерживать шифрование, как средство безопасной передачи данных. Нужно регулярно следить за обновлением программного обеспечения — ведь в новых версиях не только появляются новые функции, но и устраняются ошибки старых версий.

Как показывает практика, в среднем срок устаревания IT-продуктов (аппаратного и программного обеспечения) не превышает 3–5 лет. Неужели каждые три года нужно перестраивать всю сеть? Понятно, что так работать нельзя, поскольку на это потребуются большие деньги. А сеть тем временем будет устаревать. Поэтому необходимо следить за основными тенденциями развития мира информационных технологий и постепенно на регулярной основе вносить изменения в существующую сеть — так всегда ваша сеть будет идти в ногу со временем.

4.1.2. Построение транспортной системы корпоративной сети

Всем нужно, чтобы сеть работала быстро. Понятно, что никому не хочется ждать: ни пользователю сети, ни клиенту компании, который обращается к ее Web-серверу. Чтобы сеть работала быстро, нужно продумать ее *транспортную систему*.

Транспортная инфраструктура

Все чаще и чаще возникает необходимость в повышении пропускной способности каналов между клиентами сети и серверами. Причины, думаю, ясны: "средняя" современная рабочая станция по производительности мощнее "среднего" 5-летнего

сервера. Мощные компьютеры позволяют эффективно работать с мультимедиа и передавать мультимедиаинформацию по сети и Интернету. Пять лет назад далеко не каждый обычный пользователь мог выкачать из Интернета фильм или посмотреть видео в режиме реального времени. А сейчас Интернет стал дешевле, качественнее, быстрее. Если сеть была построена раньше, понятно, что ее транспортная инфраструктура не выдерживает возросших нагрузок — отсюда и "эффект торможения" сети.

Построение транспортной системы — задача не простая. Сложность заключается в том, что требования к пропускной способности неоднородны для различных подсетей крупной корпоративной сети. Сомневаюсь, что все клиенты будут с одинаковой интенсивностью обмениваться данными с внешними и внутренними серверами, поэтому часть сегментов будет загружена больше, а часть — меньше. Понятно, что с экономической точки зрения нужно предоставлять подсетям ту пропускную способность, которая им требуется.

Магистраль для корпоративной сети

Магистраль — один из самых главных и, следовательно, самых дорогих компонентов сети. Через магистраль проходит большая часть трафика сети, поэтому она влияет на работу всей сети в целом. Решение о выборе магистрали является одним из самых важных при планировании сети.

Кроме протокола, который будет использоваться магистралью, нужно еще учесть структуру самой магистрали — эта структура потом будет положена в основу кабельной системы. В некоторых сетях стоимость кабельной системы доходит до 20% от стоимости всей сети — сами понимаете, во что может вылиться неправильное решение. Выбор магистрали — это всегда компромисс между скоростью и стоимостью. Можно выбрать все самое быстродействующее и дорогое, но в некоторых случаях применение такого оборудования не оправдает себя. Все зависит от потребностей пользователей (также не забываем и о завтрашнем дне): может, самое дорогое и не нужно? На структуру влияет и выбранная технология, поскольку она и только она определяет максимальную длину сегмента, расстояния между рабочими станциями, возможность использования резервных кабелей, конечно, типы самого кабеля и т. п.

Итак, какую технологию выбрать в качестве магистральной? Иногда в качестве магистрали вполне будет достаточно сети Gigabit Ethernet 1000Base-T на базе витой пары, а остальные подсети будут работать со скоростью 100Мбит/с (Fast Ethernet, 100Base-T) — недорого и сердито.

Но что если нам нужны более высокие скорости? Тогда можно использовать стандарт 10GBASE (скорость передачи 10 Гбит/с), но этот стандарт пока достаточно молод, а оборудование для него — весьма дорогое. С другой стороны, этот стандарт идеально подходит для интеграции с сетями Gigabit Ethernet. А другие технологии, например, ATM, будут еще дороже. Зато ATM обеспечивает максимальную скорость передачи данных в 40 Гбит/с, что идеально подходит для работы с потоковым видео. Повторюсь, выбор магистральной технологии зависит от конкретных задач, которые ставятся перед вашей сетью.

Быстрый и экономичный доступ удаленных пользователей к сети компании

Иногда нужно обеспечить доступ удаленных пользователей к сети компании. Ранее для доступа к корпоративной сети использовались так называемые *dial-in-серверы* (серверы входящих звонков). Сейчас такое решение нельзя назвать эффективным. Во-первых, для организации сервера входящих звонков нужна многоканальная телефонная линия и модемный пул, а все это стоит недешево. Во-вторых, сервер входящих звонков рационально использовать, если удаленный абонент находится в пределах города, — междугородние и международные звонки весьма дорогие. Поэтому нужно использовать что-то более доступное.

Намного дешевле, да и проще использовать для доступа к сети возможности Интернета. В этом случае нам поможет *виртуальная частная сеть* (Virtual Private Network, VPN). Администратор настраивает VPN-сервер, который будет предоставлять доступ пользователям к ресурсам корпоративной сети. В качестве "среды" передачи данных используется Интернет. Найти интернет-кафе с беспроводным доступом (Wi-Fi) проще, чем просить кого-то разрешить подключиться к его телефонной линии. Даже если рядом нет точки беспроводного доступа к Интернету, можно подключиться к Интернету с помощью мобильного телефона — сейчас многие операторы сотовой связи существенно снизили тарифы на доступ к Интернету. Для компании же организация VPN-сервера обойдется намного дешевле организации сервера входящих звонков: не нужна ни многоканальная телефонная линия, ни модемный пул.

Помните о Wi-Fi

Если вы планируете использовать Wi-Fi, то о беспроводной сети нужно помнить с самого начала планирования сети. Очень часто о беспроводной сети забывают, а потом пытаются добавить беспроводную часть в уже существующую сеть, что не всегда эффективно. Более эффективно будет заранее спланировать сочетание Wi-Fi-сети с проводной сетью.

4.2. Обеспечение безопасности сети

О безопасности нужно думать еще на этапе планирования сети, особенно, если вы планируете использовать для построения сети только аппаратные решения. Например, для связи удаленных офисов лучше бы сразу приобрести маршрутизаторы с поддержкой VPN, чтобы потом не пришлось "изобретать велосипед" заново.

4.2.1. Защита данных, передаваемых по публичным каналам связи

Сначала нужно определиться, какие данные будут передаваться по Интернету, определить степень их секретности и уже после этого выбрать способ их защиты. В нашем случае нужно защитить: данные, передаваемые удаленными пользователями, электронную почту, Web-трафик и административный трафик.

Доступ удаленных клиентов будет защищен самим VPN-каналом — при передаче данных по виртуальному каналу используется шифрование. Электронную почту, не содержащую коммерческих тайн, можно не шифровать, а вот при обмене информацией с партнерами желательно использовать шифрование PGP. Коммерческий Web-трафик (номера кредитных карточек, например) целесообразно передавать с использованием защищенного протокола HTTPS, а не обычного HTTP. Административный трафик (например, когда администратор сети из дома получает доступ к серверу) тоже нужно шифровать. Тут все зависит от типа доступа: если по VPN, то все и так уже зашифровано, а если VPN не используется, нужно использовать SSH, но не telnet.

4.2.2. Выдача IP-адресов по рабочим местам

В больших компаниях принято назначать IP-адреса по рабочим местам пользователей. Вот одна из схем:

```
10.<этаж>.<кабинет>.<номер_компьютера>
```

Например, IP-адрес 10.2.207.3 принадлежит компьютеру с номером 3, который находится на втором этаже в комнате 207. Можно придумать и свою схему. К безопасности особого отношения это не имеет, но вам будет удобнее управлять сетью, если IP-адреса назначены не хаотично, а упорядоченно.

4.2.3. Привязка IP-адресов к MAC-адресам

Представим, что в целях экономии трафика вы ограничили определенным пользователям доступ к Интернету. Зачем, например, он бухгалтерам? Поэтому вы решили закрыть доступ к Интернету всему третьему этажу, то есть всем адресам 10.3.*.*. Но среди бухгалтеров нашелся один "продвинутый" пользователь, который додумался изменить свой IP-адрес. В результате он получит доступ к Интернету. Чтобы такого не произошло, нужно выполнить привязку IP-адресов к MAC-адресам сетевых адаптеров. MAC-адрес уникален — в мире нет двух сетевых устройств с одинаковыми MAC-адресами. Если сервер сети обнаружит, что MAC-адрес не соответствует IP-адресу, доступ к сети предоставлен не будет.

Конечно, вам предстоит огромная работа — ведь нужно переписать MAC-адреса всех компьютеров сети. В этом вам поможет программа, позволяющая просканировать сеть и вывести MAC-адреса всех сетевых адаптеров сети[1].

Правда, для квалифицированного пользователя не составит особого труда подменить и MAC-адрес (далее в этой книге будет показано, как это сделать). Но в любом случае дополнительная защита в виде привязки к MAC-адресам — неплохое решение.

4.2.4. Антивирусные серверные решения

Вирусы чаще всего попадают в сеть из Интернета, поэтому необходимо обеспечить контроль интернет-трафика (как WWW, так и почтового). Ранее контролировать весь трафик было накладно — уж очень сильно это замедляло работу всей сети, проверялся лишь почтовый трафик собственного SMTP-сервера. Сейчас это возможно.

---

[1] Могу назвать одну из таких программ: TCPNetView. Скачать ее можно по адресу: **http://gorlach.etype.net/netview/download.html**.

Также желательно настроить прокси-сервер корпоративной сети так, чтобы он запрещал доступ пользователей к сомнительным ресурсам, которые потенциально могут содержать вирусы. Прокси-сервер Squid в паре со SquidGuard *(см. главу 42)* вполне в состоянии справиться с этой задачей.

4.2.5. Антивирусные клиентские решения

Но антивирус на сервере — это еще не панацея, ведь вирус может проникнуть в компьютер пользователя со сменных носителей. Кто-то может специально или непреднамеренно инфицировать компьютер, открыв зараженный файл с дискеты или компакт-диска. Поэтому не нужно забывать и о клиентских антивирусных решениях. Таких довольно много, поэтому я думаю, что вы уже сделали выбор. Кроме антивируса я бы порекомендовал еще установить на каждую рабочую станцию брандмауэр и средство поиска spyware (шпионских программ).

4.2.6. Необходим ли дежурный администратор?

Практически все компьютеры (кроме серверов) выключены, все пользователи разошлись по домам. В здании осталась только охрана. Спрашивается, зачем нужен дежурный администратор? Оказывается, нужен... Представим, что ночью кто-то решил взломать сеть предприятия. Если дежурного администратора нет, факт взлома будет замечен только утром, а тогда может быть уже поздно. Поэтому на дежурном администраторе экономить не нужно. Конечно, можно выключить серверы на ночь, но это тоже не выход — как, например, пользователи из других стран, где не ночь, а день, получат доступ к вашему Web-серверу? Правильно, никак. Не солидно как-то…

4.3. Человеческий фактор

Человеческий фактор оказывает огромное влияние на безопасность сети. Как говорил один мой знакомый системный администратор: "Даже самая безопасная система не в силах устоять против несанкционированного доступа, если пароль пользователя написан на желтой бумажке, приклеенной к монитору".

4.3.1. Ограничение доступа

Понятно, что на пароль особо надеяться не нужно, даже если вы обойдете все комнаты и лично убедитесь, что пароль не написан маркером на мониторе или клавиатуре. Желательно, кроме назначения пароля, выполнять проверку и IP-адреса (который, в свою очередь, будет привязан к MAC-адресу), то есть разрешать доступ к тому или иному ресурсу корпоративной сети только по IP-адресу.

4.3.2. Как быть с обиженными или уволенными сотрудниками?

Все мы знаем, что такое месть. В ней нет ничего странного — так уж устроен человек... Просто кто-то может перебороть это чувство, а кто-то — нет. Существуют два типа недовольных сотрудников: просто обиженный и обиженный и уволен-

ный. Более опасен первый тип, поскольку второму можно закрыть доступ в сеть по причине его увольнения. Что же делать с первым? Все зависит от его прав доступа. Если администратор правильно распределил права доступа, то этот пользователь сможет повредить только свои данные, за которые он отвечает.

Намного сложнее ситуация, если увольняют одного из администраторов. Администратор — это человек, который знает о сети все, и даже если он не оставил "черный ход" в корпоративную сеть, то его знания могут быть использованы в не очень хороших целях. И даже не им самим, а конкурентами компании. Ведь они только и ждут, когда руководство что-то не поделит с администратором. А потом администратору последует очень интересное предложение, сами знаете какое. Поэтому в данном случае администратором должны заниматься уже не IT-специалисты компании, а служба безопасности.

4.3.3. Принцип "правая рука не знает, что делает левая"

Все мы знакомы с этим принципом. Давайте взглянем на него применительно к предприятию. Предположим, есть отдел. Пусть это будет отдел IT. Есть задача, которую нужно выполнить. Над ней должны работать, скажем, 3 человека. Но работать они должны не вместе, а по отдельности, — то есть цель у всех общая, но каждый должен дойти к ней своим путем. Получается, что в результате будет разработано не одно, а три решения задачи, — вам останется взять оптимальное. Такой способ очень эффективен: ведь если бы эти три человека работали вместе, то появилось бы всего одно решение. Администратору же нужно организовать такой режим доступа, чтобы эти три человека не могли получить доступ к файлам друг друга.

4.3.4. Планирование безопасности серверной комнаты/этажа

Серверное помещение — важнейшее помещение корпоративной сети. Для его защиты желательно установить электронные замки (доступ по паролю или чип-карте), а также систему видеонаблюдения, которая поможет определить время "миграции" сотрудников (ведь кто-то может выйти, а кто-то зайти — все это будет запечатлено с помощью видеокамеры). Иногда не будет лишней и охрана — все зависит от важности охраняемых данных. Помню, на одном из предприятий IT-отдел очень тесно работал со службой безопасности: на территорию предприятия нельзя было занести какой-либо носитель данных, не говоря о том, чтобы его вынести. IT-отдел проверял такие носители в случае, если кто-то пытался принести или вынести какую-либо информацию. А как же ноутбуки? Их вообще запрещалось там использовать. Да, похоже на паранойю, но в том случае безопасность была превыше всего, и введенные меры себя оправдывали. Что побудило предприятие пойти на такие меры? Кража информации, в результате которой предприятию был нанесен огромный экономический ущерб.

Также не следует забывать об элементарных правилах пожарной безопасности — все-таки компьютеров много, поэтому нужно приобрести пару огнетушителей.

4.4. Отдел системного администрирования и безопасности

А теперь опять рассмотрим человеческий фактор, но уже с другой стороны — с точки зрения подбора персонала.

4.4.1. Подбор персонала

Подбор персонала должен выполняться IT-специалистом, а не кадровиком предпенсионного возраста, который ничего не понимает в компьютерах. Это первый аспект. Второй заключается в том, чтобы подбор персонала выполнялся не только субъективно, то есть "по знакомству". Одно дело, если "по знакомству" находят действительно хорошего специалиста, но совсем другое, когда место программиста занимает "специалист", не имеющий элементарных представлений о IT.

В идеале должна существовать определенная система тестирования, разработанная IT-специалистом (желательно сторонним, чтобы исключить субъективный фактор). Для компьютера не существует "своих" и "чужих", для него все равны (при условии, что программа написана не с учетом "знакомых"), поэтому тестирование будет выполняться объективно. Кандидат, набравший большее количество баллов, будет принят на работу. Если времени заниматься разработкой такой системы нет, то можно пригласить для подбора персонала стороннего IT-специалиста.

Дипломы и сертификаты — конечно, хорошо. Хотя бывает так, что у человека нет даже высшего образования, а он является лучшим специалистом, чем дипломированный. Вот для этого и нужна система тестирования при приеме на работу — особенно с учетом особенностей нашей системы образования, когда за определенную сумму диплом приносят чуть ли не на дом. С сертификатами дело обстоит чуть иначе. Онлайн-сертификатам (которые можно получить любому желающему в Интернете) я бы не очень доверял — за "специалиста" пройти тест может кто угодно. А вот сертификаты, выданные крупными компаниями, например, Microsoft, порою говорят даже о большем, нежели дипломы о высшем образовании.

4.4.2. Инструктаж отдела IT

Желательно, чтобы при приеме на работу сотрудник знал не только название своей должности, но и свои права и обязанности. Чтобы каждый раз не рассказывать сотруднику, что он должен делать и какие результаты от него ожидаются, все это оформляется в служебную инструкцию, которую должен изучить сотрудник в первые дни работы. Вся его дальнейшая деятельность должна проходить в рамках инструкции. Что должно быть в инструкции? Прежде всего, общие положения, ожидаемые результаты деятельности данного сотрудника, права и обязанности сотрудника, правила взаимодействия с другими службами предприятия, критерии оценки результатов, ответственность и квалификационные требования. Инструкция должна разрабатываться опытным IT-специалистом, который мог бы не только создать инструкцию, но и обосновать необходимость того или иного пункта инструкции (ведь можно ее и из Интернета позаимствовать, а потом смотреть на нее большими от удивления глазами).

4.4.3. Распределение задач и сфер ответственности

Задачи IT-сотрудников и сферы их ответственности должны быть четко распределены. Кстати, для этого и пишется инструкция, в которой четко должно быть сказано, кто что должен делать и кто за что отвечает.

4.4.4. Контроль работы и иерархия

Обычно иерархия того или иного подразделения изображается в виде организационной или пирамидальной диаграммы. Лично мне больше нравится последняя (рис. 4.2).

Рис. 4.2. Пирамидальная диаграмма иерархии IT-отдела

Итак, на пирамиде мы видим: руководство IT-отдела, среднее звено и специалисты. К руководству относят следующие должности: директор департамента информационных технологий (попросту говоря, начальник IT-отдела) и руководитель проекта. Руководитель проекта — должность необязательная. Как правило, такая должность имеется на предприятиях, занимающихся разработкой IT-продуктов.

Директор IT-департамента занимается разработкой и внедрением информационных стратегий и созданием единой информационной структуры, подчиняется он только генеральному директору компании. Руководитель проекта подчиняется директору IT-департамента, но задача у него своя — он руководит отделом, который входит в структуру компании. В больших IT-компаниях может быть несколько IT-отделов, каждый из которых работает в своем направлении, и у каждого IT-отдела есть собственный руководитель проекта.

Теперь переходим к среднему звену. К нему могут относиться следующие должности: IT-менеджер, менеджер автоматизации, менеджер по работе с клиентами. Первый отвечает за бесперебойную работу всех информационных систем компании, второй занимается автоматизацией деятельности компании и ее филиалов, третий, как понятно из названия, работает с клиентами.

К специалистам относят следующие должности: системный администратор, главный программист, программист. В больших компаниях роли системного администратора и IT-менеджера четко разделены: системный администратор подчиняется IT-менеджеру и выполняет исполнительские обязанности. А в не очень больших компаниях системный администратор частенько выполняет функции IT-менеджера. Главный программист руководит программистами и отвечает за своевременное выполнение проекта.

4.5. Программы для планирования сети

В Интернете можно скачать различные программы для планирования вашей сети. Конечно, они не учитывают всех приведенных в этой главе аспектов, но с их помощью вы хотя бы набросаете схему сети, которая поможет потом при развертывании сети. Могу назвать одну из таких программ: LanFlow (рис. 4.3). Скачать ознакомительную версию программы можно на сайте **www.pacestar.com/lanflow**.

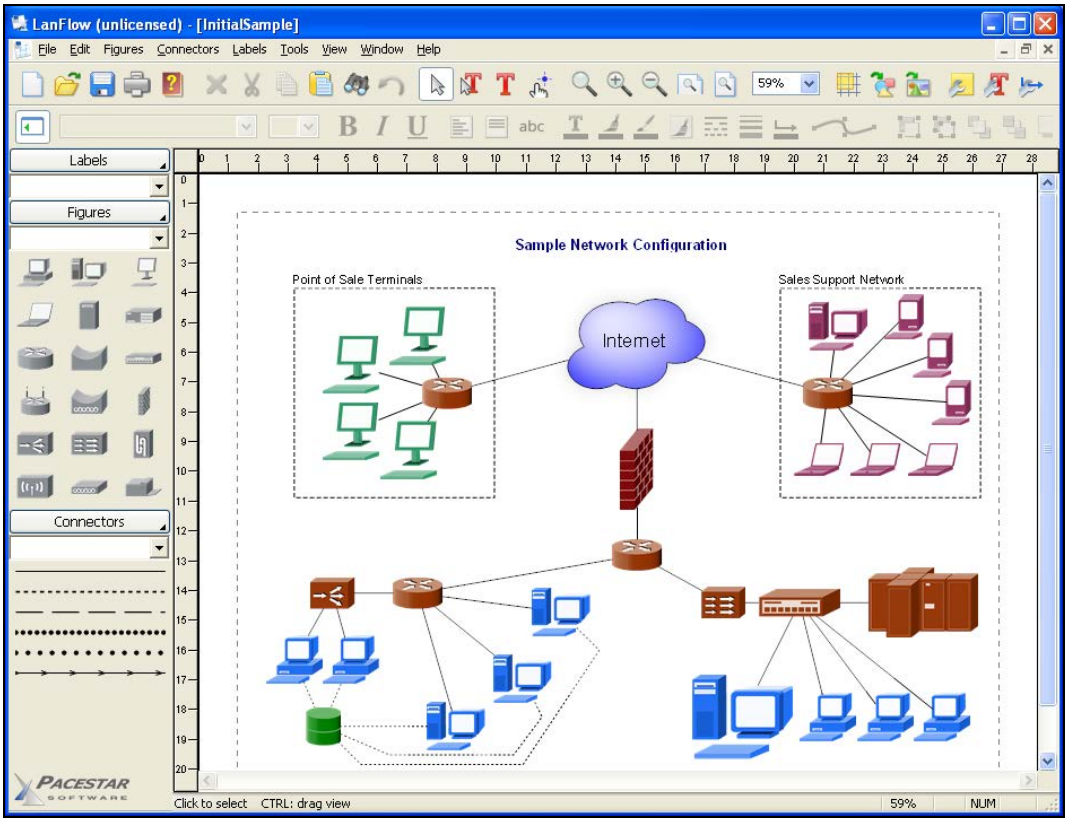
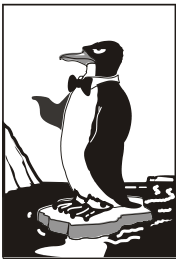


Рис. 4.3. Программа LanFlow



Глава 5

Монтаж Ethernet-сети

5.1. Развитие стандарта Ethernet

В *главе 2*, когда мы знакомились с краткой историей сетей, были рассмотрены основные этапы развития сетей Ethernet, сейчас же настало время поговорить о стандарте Ethernet подробнее.

Технология Ethernet намного "древнее", чем это можно себе представить. Хотя первая Ethernet-сеть была создана в 1975 году, она использует (как на момент создания, так и сейчас) метод доступа CSMA/CD (мы его рассмотрим позже), который был разработан во второй половине 60-х годов прошлого века.

Создателем Ethernet-сети является компания Xerox. В 1980 году эта компания вместе с компаниями DEC и Intel разработали вторую версию стандарта Ethernet, которая называлась DIX (DEC, Intel, Xerox) или Ethernet-II. В то время Ethernet-сети использовали в качестве среды передачи данных только коаксиальный кабель.

Чуть позже был разработан стандарт IEEE 802.3, который практически полностью повторял стандарт Ethernet II. У нового стандарта были лишь небольшие отличия в формате кадров, в нем не было протокола тестирования конфигурации, который существовал в DIX, и новый стандарт выделял два уровня MAC и LLC, которые в DIX были одним целым.

Среди ранних модификаций стандарта Ethernet можно выделить следующие:

- Xerox Ethernet — оригинальный стандарт, предусматривающий скорость передачи данных 3 Мбит/с. Как уже было отмечено, существовали две версии этого стандарта: 1 и 2 (DIX);
- 10BROAD36 — позволял передавать данные на большие расстояния, для чего использовал технологию широкополосной модуляции. Средой передачи данных служил коаксиальный кабель. Стандарт не получил широкого распространения;
- 1BASE5 (второе название StarLAN) — дальний предок стандарта 10Base-T. Скорость передачи данных — 1 Мбит/с. В качестве среды передачи данных использовал витую пару. Не получил большого распространения.

5.1.1. Модификации стандарта Ethernet

После принятия стандарта IEEE 802.3 технология Ethernet продолжала бурно развиваться — скорость передачи данных повысилась до 10 Мбит/с (это еще в стандарте 802.3), а в качестве среды передачи данных использовался не только коаксиальный

кабель, но и витая пара, и оптоволоконный кабель. Модификации стандарта Ethernet представлены в табл. 5.1.

Таблица 5.1. *Модификации стандарта Ethernet*

Стандарт	Описание
10Base-5 (IEEE 802.3)	Использовался "толстый" коаксиальный кабель RG-8, именно поэтому данный стандарт иногда называют "толстый Ethernet". Максимальная длина сегмента — 500 метров
10Base-2 (IEEE 802.3a)	Сеть на базе "тонкого коаксиала" (кабель RG-58). Поскольку тонкий коаксиал более гибкий, такую сеть было проще монтировать, чем сеть 10Base-5. Однако максимальная длина сегмента не превышает 200 метров. Этот стандарт прожил долгую жизнь. Помню, встречал сеть на его базе даже в 2002 году (правда, после уже не видел, но уверен, что где-то они до сих пор есть)
StarLAN 10	Первый стандарт, работающий на скорости 10 Мбит/с и использующий витую пару. Послужил прототипом стандарта 10Base-T
10Base-T (IEEE 802.3i)	Среда передачи данных: витая пара 3-й или 5-й категорий. Максимальная длина сегмента такой сети — 100 метров
FOIRL (Fiber-optic inter-repeater link)	Стандарт, использующий для передачи данных оптоволоконный кабель. Послужил прототипом для 10Base-F. Максимальное расстояние передачи данных (без повторителя) — 1 км
10Base-F (IEEE 802.3j)	Основной стандарт, послуживший базой для стандартов 10Base-FL, 10Base-FB, 10Base-FP. Все стандарты используют в качестве среды передачи данных оптоволоконный кабель. Расстояние — до 2 км, скорость передачи данных — 10 Мбит/с
10Base-FL (Fiber Link)	Практически то же самое, что и FOIRL, но длина передачи данных увеличена до 2 км
10Base-FB (Fiber Backbone)	Предназначался для объединения повторителей в магистраль
10Base-FP (Fiber Passive)	Топология "пассивная звезда", которая не предусматривает повторителей. Этот стандарт остался на бумаге — он не был реализован

5.1.2. Стандарты Fast Ethernet (100 Мбит/с)

Настоящий прорыв в развитии Ethernet произошел в 1995 году, когда появилась технология Fast Ethernet, позволяющая передавать данные со скоростью в 10 раз выше обычного Ethernet'a — 100 Мбит/с. С появлением Fast Ethernet коаксиальный кабель ушел в прошлое, новая технология в качестве среды передачи данных использовала только витую пару 5-й категории и оптоволоконный кабель.

Стандарты Fast Ethernet представлены в табл. 5.2.

Таблица 5.2. *Модификации стандарта Fast Ethernet*

Стандарт	Описание
100Base-T	Общее название стандарта для модификаций 100Base-TX, 100Base-T4 и 100Base-T, которые описаны далее. Все эти стандарты используют витую пару, а максимальная длина сегмента составляет 100 метров
100Base-TX (IEEE 802.3u)	Дальнейшее развитие стандарта 10Base-T. Как и в 10Base-T, используется топология "звезда"
100Base-T4	Создан из соображений обратной совместимости. Данный стандарт использует витую пару категории 3. Это значительно упрощает модернизацию сетей 10Base-T, где из соображений экономии использовалась витая пара 3-й категории. Нужно отметить, что этот стандарт сейчас практически не используется
100Base-T2	Еще один вариант на витой паре 3-й категории, сейчас практически не применяется. Отличительная особенность: использует полный дуплекс (то есть один и тот же кабель может *одновременно* использоваться как для приема, так и для передачи данных). Скорость приема/передачи данных (в одном направлении) — 50 Мбит/с
100Base-FX	Использует многомодовое оптоволокно, максимальная длина сегмента в полудуплексе — 400 метров, в полном дуплексе — 2 км
100Base-LX	Используется одномодовое волокно (оборудование на базе одномодового кабеля стоит дороже). Обеспечивает передачу данных на расстояние 15 км в режиме полного дуплекса, длина волны 1310 нм
100Base-LX WDM	То же, что и 100Base-LX, но допускается использование длин волны 1310 нм и 1550 нм. При этом с одной стороны используется передатчик с длиной волны 1310 нм, а с другой — 1550 нм

Что такое дуплекс

В приведенной здесь таблице встретился, возможно, незнакомый вам термин: *дуплекс*. Существуют два режима передачи данных по одному и тому же кабелю: полудуплекс (Half Duplex) и полный дуплекс (Full Duplex). Рассмотрим оба режима на примере обычного телефона. Телефонная связь работает в *полнодуплексном* режиме, поскольку вы можете и говорить, и одновременно слышать своего собеседника, то есть вы можете говорить с ним одновременно. Если бы телефон работал в *полудуплексном* режиме, то когда бы вы говорили, но не слышали бы своего собеседника, поскольку передача идет в одном направлении — отправки. Вам нужно было бы сказать фразу, нажать какую-то кнопку, переключающую аппарат в режим приема информации, и тогда вы бы смогли услышать ответ своего собеседника.

Типы оптоволоконных кабелей

Многомодовый кабель — это кабель, где есть несколько пространственных мод, *одномодовый* — где имеется только одна мода. *Мода* — это тип электромагнитной волны в оптическом кабеле. Оптоволоконные кабели и сети на их основе из-за их дороговизны и сложности монтажа мы не рассматриваем. Интересующиеся могут прочитать о различных типах кабелей и их внутреннем устройстве по адресу: **http://kgg.moldline.net/teaching/cable/cable_media.htm**.

5.1.3. Gigabit Ethernet (1000 Мбит/с)

В 1998 году появилась новая технология Gigabit Ethernet. Прорывом или революцией ее не назовешь. По сути, это количественное улучшение, а не качественное. Ничего нового создано не было: та же среда передачи данных, тот же метод разделения этой самой среды — CSMA/CD. Зато очень легко модернизировать сеть Fast Ethernet в Gigabit Ethernet: достаточно заменить сетевые адаптеры и коммутаторы, кабели трогать не нужно — они останутся прежними (нужно будет только переобжать концевые коннекторы, но об этом — позже). Модификации Gigabit Ethernet представлены в табл. 5.3.

Таблица 5.3. *Модификации стандарта Gigabit Ethernet*

Стандарт	Описание
1000Base-T (IEEE 802.3ab)	Использует витую пару категорий 5e или 6. В отличие от стандарта 100Base-TX, где используются только 2 пары кабеля (то есть 4 жилы), этот стандарт использует все 4 пары (8 жил), благодаря чему увеличивается скорость передачи данных
1000Base-TX	Разработан Ассоциацией телекоммуникационной промышленности (Telecommunications Industry Association, TIA) в 2001 году. Работает в полном дуплексе, скорость передачи данных в обоих направлениях — 500 Мбит/с. Использует 2 пары (4 жилы) для передачи данных и 2 — для приема, что упрощает конструкцию приемопередающих устройств, но требует витую пару более высокой категории — 6-й. Зато этот стандарт предполагает более простое оборудование, которое стоит дешевле, чем оборудование для 1000Base-T
1000Base-SX (IEEE 802.z)	Использует многомодовое оптоволокно, длина сегмента — 550 метров
1000Base-LX (IEEE 802.3z)	Дальность передачи данных при использовании многомодового оптоволокна — 550 м, а при использовании одномодового — до 40 км (без повторителей)
1000Base-CX	Подходит для передачи данных на небольшие расстояния (до 25 м) и использует экранированную витую пару (STP). Сейчас этот стандарт не применяется и заменен стандартом 1000Base-T
1000Base-LH (Long Haul)	Обеспечивает передачу данных на расстояние до 100 км без повторителей

5.1.4. Наше будущее — 10 Gigabit Ethernet

Относительно недавно был разработан новый стандарт, способный передавать данные со скоростью 10 Гбит/с — 10 Gigabit Ethernet. Этот стандарт пока еще очень молод, и понадобятся несколько лет, чтобы понять, какие его спецификации будут востребованы, а какие исчезнут с рынка.

Пока доступны следующие спецификации:

- 10GBase-CX4 — используется для передачи данных на короткие расстояния (до 15 м), применяются медный кабель CX4 и коннекторы InfiniBand. Мне кажется, что этот стандарт не получит особого распространения (как и 1000Base-CX), но поживем — увидим;
- 10GBase-SR — пригоден для передачи данных на небольшие расстояния (от 26 до 82 метров в зависимости от типа кабеля), использует многомодовое оптоволокно;
- 10GBase-LX4 — расстояние передачи данных от 240 до 300 метров по многомодовому оптоволокну или до 10 км по одномодовому оптоволокну;
- 10GBase-LR и 10GBase-ER — используются для передачи данных на расстояния до 10 и 40 км соответственно;
- 10GBase-T — принят в 2006 году (самый молодой стандарт из этого семейства), использует экранированную витую пару, длина сегмента (расстояние передачи данных) — 100 метров.

5.2. Несколько слов о коллизиях…

Чтобы иметь представление о Ethernet-сетях, вам нужно знать, что такое метод доступа CSMA/CD (Carrier-Sense-Multiply-Access with Collision Detection) — метод коллективного доступа с обнаружением несущей (carrier) и коллизий. Этот метод используется во всех сетях с логической топологией "общая шина". Да, с появлением коммутаторов Ethernet-сети преобразились, но метод CSMA/CD служит до сих пор.

Представим себе общую шину — общую среду передачи информации. Ее можно сравнить с гирляндами лампочек на елке — все они подключены к одному проводу. Поскольку кабель общий, одновременно обмениваться информацией могут всего два компьютера. Спрашивается, какая же будет эффективность такой сети, если вся сеть должна ждать, пока два компьютера обмениваются информацией? Однако все происходит иначе. Все мы помним, что перед передачей данные разбиваются на части — пакеты. Общий алгоритм передачи данных таков:

- Компьютер разбивает информацию на пакеты.
- Затем он проверяет, не занята ли среда передачи данных.
- Если среда свободна, компьютер передает один пакет.
- После передачи пакета компьютер должен подождать 9,6 мкс, а потом начать процесс передачи следующего пакета.

Иногда случается *коллизия* — ситуация, когда два или больше компьютеров пытаются одновременно передать данные. Почему происходят коллизии, ведь компьютер перед отправкой данных проверяет, свободна ли среда передачи данных? Сначала разберемся, как он это делает. Компьютер прослушивает *несущую частоту* (carrier sense — вот откуда взялись две начальные буквы CS в названии метода!). Если несущей частоты (5–10 МГц) нет, то среда свободна. А теперь представим, что компьютер А только начал передавать кадр, а компьютер Б, который находится где-то очень далеко, одновременно начал проверять занятость среды передачи данных. Понятно,

что несущая частота еще не "дошла" до компьютера Б, поэтому он тоже начал передачу данных. В результате передаваемые данные смешались — вот вам и коллизия…

Суть метода CSMA/CA в том и заключается, что когда два или большее количество узлов пытаются одновременно передать данные, CSMA/CA "просит" все узлы, кроме одного, прекратить передачу данных. "Счастливчик", которому разрешено передать данные, выбирается случайным образом. Но CSMA/CA может также предоставить приоритет узлу, который пытается передать данные, критические к времени (видео и/или голос).

В современных сетях на базе коммутаторов коллизии возникать, в общем-то, не должны — поскольку к каждому порту коммутатора подключено по одному компьютеру, и коммутатор передает пакет не всем компьютерам, а только тому, кому пакет адресован. Однако и в таких сетях коллизии порой возникают — например, когда сетевой адаптер и порт коммутатора одновременно начинают передавать кадры, решив, что кабель не занят. Правда, такая ситуация может сложиться только в полудуплексном режиме. В полнодуплексном режиме, как мы знаем, разрешена одновременная передача данных в обоих направлениях (прием и передача), поэтому в сети на базе коммутаторов, работающей в полнодуплексном режиме, коллизии не возникают.

5.3. Монтаж сети

5.3.1. Основные компоненты Ethernet-сети

Итак, давайте вспомним основные компоненты сети:

- **сетевые адаптеры** — с этим проблем сейчас нет, поскольку сетевые адаптеры встроены в материнские платы всех настольных компьютеров и ноутбуков;
- **вилки** (концевые коннекторы, разъемы) RJ-45 — ими обжимается витая пара, после чего обжатым кабелем можно соединить компьютер с коммутатором. Поскольку этими вилками кабель обжимается с обеих сторон, то количество вилок должно в два раза превышать количество компьютеров. Разъемы желательно покупать с запасом (они стоят копейки), поскольку во время обжима разъем легко повредить. На рис. 5.1 приведена схема нумерации контактов вилки RJ-45;
- **кабель "витая пара"** — о выборе витой пары мы поговорим чуть позже;
- **коммутатор** — перед покупкой коммутатора убедитесь, что он содержит достаточное количество портов, необходимое для подключения всех компьютеров вашей сети. Если у вас много компьютеров, скажем, больше 24, то имеет смысл купить два коммутатора по 24 порта, чем один на 48 — для локализации трафика и уменьшения нагрузки на коммутатор;
- **инструмент для обжима витой пары** — именно этим инструментом вы будете обжимать витую пару (рис. 5.2).

> ***Примечание***
>
> Интересно, что то, что практически все называют разъемом RJ-45, на самом деле называется вилкой 8P8C. Подробно об этом можно прочитать в Википедии: **http://ru.wikipedia.org/wiki/8P8C**.

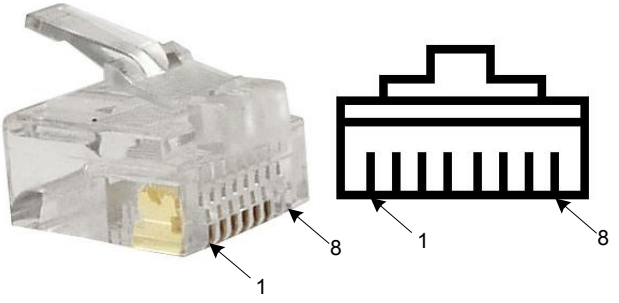


Рис. 5.1. Нумерация контактов вилки RJ-45

Рис. 5.2. Инструмент для обжима витой пары

Совет

Не покупайте дешевый инструмент для обжима витой пары! Помните пословицу о дешевой рыбке и соответствующей ей юшке? Когда-то я решил купить дешевый инструмент — он стоил в три раза дешевле, чем обычно. Да, на вид он был такой же. И вроде RG-45 обжимал. Как оказалось, именно "вроде". Сколько я разъемов испортил! Оказывается, этот инструмент не до конца обжимал разъем — обжимал не все его контакты. А чтобы обжать все контакты, приходилось сжимать разъем сильнее. Чуть-чуть перестарался, и все — разъем треснул. В итоге этот инструмент я выбросил и купил нормальный. Скупой платит дважды.

Если вам нужно построить небольшую сеть (скажем, до 10 компьютеров), тогда можно пойти путем наименьшего сопротивления и купить в компьютерном магазине готовые кабели. Они уже будут обжаты, и вам останется только соединить ими компьютеры с коммутатором. Обжатый кабель стоит немного дороже, чем комплект из витой пары необходимой длины и двух разъемов, но, учитывая, что компьютеров будет немного, а профессионально заниматься монтажом сети вы не собираетесь, то вы даже сэкономите. Хороший инструмент стоит дороже, чем 5–7 уже обжатых кабелей (в зависимости от длины). Правда, в случае с обжатыми кабелями есть один недостаток: они продаются только фиксированной длины — обычно по 5 или 10 метров, что не всегда удобно.

5.3.2. Подробнее о витой паре

Категории витой пары

Витая пара витой паре — рознь. Существуют различные категории витой пары. Тут все просто: чем выше категория, тем кабель более качественный и дорогой. Витая пара 1–4 категорий уже не используется для построения сетей (хотя для построения сетей и ранее использовалась витая пара только 3 и 4-й категорий, а кабели 1 и 2-й категорий применялись в телефонных сетях).

Для построения современных Ethernet-сетей используется витая пара 5 и 6-й категорий. Пятая категория (CAT5) представляет собой 4-парный кабель (4 пары жил) и служит для построения сетей 100Base-TX. В этих сетях задействованы всего 2 пары (4 провода), при этом достигается скорость передачи данных до 100 Мбит/с.

Более современная модификация пятой категории называется 5E. Сейчас практически вся продаваемая витая пара относится к этой категории. При покупке кабеля обратите внимание на надписи на его внешней оболочке — лучше приобрести витую пару именно CAT5E, чем просто CAT5, поскольку модернизированная версия может использоваться для построения сетей Gigabit Ethernet. Для передачи данных со скоростью 1000 Мбит/с используются все 4 пары.

В июне 2002 года появилась витая пара шестой категории — CAT6. Она стоит дороже, чем витая пара CAT5E. Если вы планируете использовать Gigabit Ethernet, то следует применить витую пару шестой категории — она лучше подходит для передачи данных со скоростью 1000 Мбит/с. К тому же, для скоростей 10 Гбит/с CAT5E не годится вовсе, нужна CAT6. Поэтому если вы в будущем собираетесь модернизировать свою сеть — ваш выбор CAT6.

Кроме 5-й и 6-й есть еще и седьмая категория. Кабель седьмой категории имеет общий экран и экран вокруг каждой пары. По сути, кабель CAT7 является S/FTP (Screened Fully Shielded Twisted Pair) кабелем (см. далее). Седьмая категория утверждена международным стандартом ISO11801 и поддерживает скорость передачи данных до 10 Гбит/с. Частота пропускаемого сигнала составляет 600–800 МГц.

Классификация витой пары по типу защиты

Существуют следующие виды витой пары:

- UTP (Unshielded twisted pair) — защита и экранирование отсутствуют, это самый дешевый вид витой пары и предназначен для использования внутри помещений (конечно, такой кабель можно использовать и снаружи, но в силу своей незащищенности долго он не прослужит);
- FTP (Foiled twisted pair) — есть один общий экран (для всех пар) из фольги. Тип более защищенный, чем UTP;
- STP (Shielded twisted pair) — экранированная витая пара, присутствует один экран для каждой пары;
- S/FTP (Shielded Foiled twisted pair) — почти то же, что и FTP, но присутствует дополнительный внешний экран из медной оплетки;
- S/STP (Screened shielded twisted pair) — похож на STP, но присутствует дополнительный общий внешний экран.

Типы витой пары приведены по мере возрастания защиты и стоимости. Для наружного использования рекомендуется STP. Хотя если позволяют средства, то можно купить и S/STP.

5.3.3. Обжим витой пары

Обжать витую пару — это значит поместить отдельные жилы витой пары в определенной последовательности в вилку RJ-45 и закрепить их в этой вилке с помощью инструмента. А вот теперь начинается самое интересное — ключевая фраза здесь "в определенной последовательности". Последовательность расположения жил зависит от:

- стандарта кабеля;
- от того, кабель какого стандарта вы пытаетесь получить.

Существуют два стандарта витой пары: 568A и 568B (маркировка стандарта нанесена на оболочку кабеля). Если не вникать в подробности этих стандартов, то они для нас отличаются порядком обжима отдельных жил витой пары. И вся беда в том, что вам нужно помнить (или держать под рукой) обе схемы обжима. Ведь сегодня вы построили сеть на базе кабеля 568A, а завтра, когда пришлось подключить дополнительный компьютер, в продаже был только 568B. Мы рассмотрим обе схемы обжима, но чуть позже.

Схема обжима может зависеть не только от стандарта кабеля, но еще и от того, что вы хотите соединить. Ethernet-кабель бывает *прямым*, а бывает — *перекрестным* (его еще называют кроссовер, от англ. cross-over).

- Прямой кабель используется для соединения компьютера с коммутатором и коммутатора с другим коммутатором. Схема обжима прямого кабеля с обеих сторон одинаковая.
- Перекрестный кабель служит для соединения двух компьютеров напрямую, без коммутатора (для организации сети из двух компьютеров) или для соединения некоторых старых коммутаторов/концентраторов, у которых имеется порт Uplink. Одна вилка перекрестного кабеля обжимается как и у прямого кабеля, а вторая — перекрестно (с другим порядком расположения жил — чуть позже я поясню, как именно). Поскольку у кроссовера порядок обжима витой пары изменен, его нельзя использовать для подключения компьютера к коммутатору.

Итак, когда мы знаем о стандартах 568A и 568B и о прямом и перекрестном обжиме, самое время приступить к обжиму. Напомню, что схема нумерации контактов вилки RJ-45 приведена на рис. 5.1.

Прямой кабель, Fast/Gigabit Ethernet

В табл. 5.4 приведена схема обжима прямого Ethernet-кабеля стандартов 568A и 568B для сети Fast/Gigabit Ethernet (100/1000 Мбит/с). Напомню, что прямой кабель обжимается одинаково с обеих сторон.

Таблица 5.4. *Прямой Ethernet-кабель (100/1000 Мбит/с)*

Номер контакта	Цвет жилы (для 568A)	Цвет жилы (для 568B)
1	Зелено-белый	Оранжево-белый
2	Зеленый	Оранжевый
3	Оранжево-белый	Зелено-белый
4	Синий	Синий
5	Сине-белый	Сине-белый
6	Оранжевый	Зеленый
7	Коричнево-белый	Коричнево-белый
8	Коричневый	Коричневый

Перекрестный кабель (кроссовер) для соединения 100 Мбит/с

В табл. 5.5 приводится номер контакта и схема обжима для первой стороны и для второй стороны кабеля. Обратите внимание: схема действительна только для кабеля 586B.

Таблица 5.5. *Кроссовер 100 Мбит/с*

Номер контакта	Сторона 1	Сторона 2
1	Оранжево-белый	Зелено-белый
2	Оранжевый	Зеленый
3	Зелено-белый	Оранжево-белый
4	Синий	Синий
5	Сине-белый	Сине-белый
6	Зеленый	Оранжевый
7	Коричнево-белый	Коричнево-белый
8	Коричневый	Коричневый

Перекрестный кабель (кроссовер) для соединения 1000 Мбит/с

В табл. 5.6 приводится схема обжима кроссовера для соединения со скоростью 1000 Мбит/с (Gigabit Ethernet).

Таблица 5.6. *Кроссовер 1000 Мбит/с*

Номер контакта	Сторона 1	Сторона 2
1	Оранжево-белый	Зелено-белый
2	Оранжевый	Зеленый
3	Зелено-белый	Оранжево-белый
4	Синий	Коричнево-белый
5	Сине-белый	Коричневый
6	Зеленый	Оранжевый
7	Коричнево-белый	Синий
8	Коричневый	Сине-белый

Проверка правильности обжима кабеля

Обжимать кабель нужно тщательно, но стараясь не поломать коннекторы. Проверить, правильно ли вы обжали кабель, можно с помощью самого же коммутатора. Подключите один конец кабеля к компьютеру, а другой к коммутатору (коммутатор и компьютер должны быть включены).

У каждого порта коммутатора есть минимум два индикатора: первый (обычно маркируется "Link/ACT") показывает, есть или нет связь, а второй (может маркироваться "Speed", или "100", или "1000" — в зависимости от устройства) — скорость работы порта. Технология Fast Ethernet подразумевает передачу данных со скоростью 100 Мбит/с, но поддерживает и старые устройства, которые могут работать только на 10 Мбит/с. Так вот, второй индикатор загорается только в том случае, если обеспечивается скорость 100 Мбит/с. Если же индикатор не загорается, давайте подумаем, в чем причина. Сетевой адаптер и коммутатор точно поддерживают скорость передачи данных 100 Мбит/с — старые устройства уже давно сняли с продажи, а новые помимо скорости 100 Мбит/с могут даже поддерживать скорость 1000 Мбит/с. Следовательно, дело в кабеле — вы неправильно его обжали (повреждение самого кабеля я исключаю), возможно, плохо обжался какой-нибудь контакт. Попробуйте, не снимая вилку, еще раз обжать ее. Если опять не получится, нужно срезать вилку и обжать кабель заново.

В случае с Gigabit Ethernet все аналогично: если не загорается индикатор "Speed", то кабель обжат неправильно. Если оба индикатора не горят, нужно обжать кабель заново — и так до тех пор, пока не обожмете правильно.

5.4. Ограничения при построении сети

Правильно обжать вилки RJ-45 — это еще не все. Нужно помнить о минимальной и максимальной длине кабеля. Минимальная длина — 1 метр, меньше никак. Максимальная — 100 метров. Что делать, если 100 метров мало? В этом случае нужно использовать несколько коммутаторов: к одному коммутатору вы подключаете близлежащие компьютеры и второй коммутатор. Ко второму коммутатору подключите остальные компьютеры. В итоге максимальное расстояние между двумя максимально удаленными компьютерами получится 210 метров (см. рис. 5.3).

Хочу заметить, что максимальная длина сегмента 100 метров — это только по стандарту, на практике можно "выжать" значительно больше. В зависимости от сетевого адаптера и коммутатора возможна максимальная длина сегмента до 150 метров. Только сами понимаете, никто не гарантирует, что:

- такой сегмент вообще будет работать (в некоторых условиях будет работать, а в некоторых — нет);
- будет достигнута максимальная скорость. Скорее всего, максимум, что получится выжать из такого длинного сегмента, — 10 Мбит/с. При превышении максимального расстояния дальнейшее развитие событий зависит от коммутатора и сетевого адаптера. Как минимум, вы получите потери сигнала в 40% (следовательно, и потерю скорости).

Однако вы должны знать, что такой вариант возможен. Иногда расстояние от одного из компьютеров до коммутатора составляет 105–110 метров. Ради одного компьютера и десяти лишних метров покупать еще один коммутатор не хочется, поэтому можно попробовать работать с превышением максимальной длины. Может и получится, а может — нет...

Если нужно еще большее расстояние, то лучше использовать оптоволоконный кабель — в этом случае максимальное расстояние достигнет 2000 м. Но в этой книге

мы не рассматриваем сети на базе оптоволоконного кабеля. Как правило, в домашних и офисных сетях среднего размера вполне можно обойтись витой парой.

В табл. 5.7 вы найдете ограничения для сетей Fast и Gigabit Ethernet.

Рис. 5.3. Схема простой, но "длинной" сети

Таблица 5.7. *Сводная таблица ограничений*

Характеристика	**Fast Ethernet**	**Gigabit Ethernet**
Минимальная длина кабеля	1 м	1 м
Максимальное расстояние между компьютером и коммутатором	100 м	100 м для 1000Base-T 25 м для 1000Base-CX
Максимальное расстояние между коммутаторами	90 м	90 м
Максимальный диаметр сети (расстояние между максимально удаленными друг от друга элементами)	210 (250) м	210 (250) м
Максимальное число компьютеров в сети	1024	1024
Витая пара (категория)	5, 5E, 6	5E, 6

А теперь предположим, что нам нужно построить сеть, подобную изображенной на рис. 5.3. То есть у нас есть два сегмента и два коммутатора. Какое может быть максимальное расстояние между коммутаторами? Из табл. 5.7 следует, что 90 метров. Считаем: если максимальное расстояние от компьютера 1 до коммутатора 1 — 100 метров и от компьютера 2 до коммутатора 2 — тоже 100 метров, а максимальный диаметр сети равен 250 метрам, то максимальное расстояние между коммутаторами может быть только 50 метров, а не 90, как указано в таблице. Обратите внимание, что и 250 метров — это теоретическое значение, на практике лучше ориентироваться на 210 метров (с запасом).

Иногда нужно увеличить расстояние между коммутаторами. Например, есть два здания, находящихся на небольшом расстоянии друг от друга. В каждом здании имеется коммутатор, к которому подключаются компьютеры, находящиеся в этом здании. Если длина кабеля между коммутаторами равна 90 метрам (теоретический предел), то максимальное расстояние от каждого коммутатора до конечного компьютера должно быть не более 80 метров: (250 – 90) / 2, см. рис. 5.4. Да и это довольно-таки теоретические построения. Лучше, как было сказано, ориентироваться на максимальную длину сети 210 метров.

Что же касается максимального числа узлов, то его с лихвой хватит для построения любой домашней и офисной сети среднего размера. Если узлов много, можно использовать несколько 48-портовых коммутаторов, объединенных в стек.

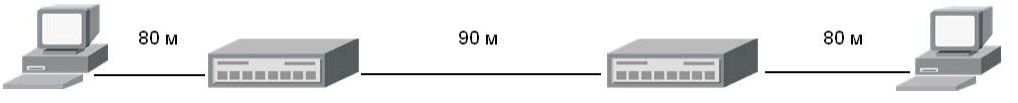
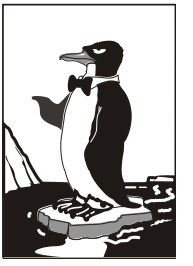


Рис. 5.4. Еще одна конфигурация сети



Глава 6

Основы беспроводной сети. Монтаж беспроводной сети

6.1. Преимущества и недостатки беспроводной сети

Итак, ваш офис нуждается в сети (пока ни слова о беспроводных сетях). Вы сразу сталкиваетесь с проблемой монтажа сети. Скорее всего, компьютеры вашей сети будут находиться в разных комнатах. Тогда вы предстанете перед выбором: или тянуть витую пару "в обход", или же воспользоваться перфоратором, чтобы сэкономить витую пару и избавиться от проводов на стенах. Даже на монтаж небольшой сети вы можете потратить от нескольких часов до целого дня — все зависит от вашего опыта в этом деле.

Рано или поздно ваша сеть заработает. Но в один прекрасный день к вам в офис придет клиент, которому нужно сбросить некоторые файлы на его ноутбук. Болванок, как обычно, нет, флешка почему-то "глючит" (уж очень много на рынке дешевых флешек, которые выходят из строя спустя несколько месяцев эксплуатации). Вы решаете подключить ноутбук клиента к сети. А у вас есть заранее обжатый Ethernet-кабель? Нет? А есть ли свободные порты на коммутаторе? В общем, опять проблемы.

А теперь представим, что офис — арендуемый (как оно обычно бывает), и нужно переезжать на новое место. Да, придется сворачивать сеть и разворачивать ее в другом помещении. Возможно, придется заново обжать некоторые кабели, а некоторые подготовить заранее, — но не всегда можно угадать с длиной.

Беспроводная сеть — это решение для современного офиса. Ее монтаж осуществляется быстро и без перфоратора, клещей для обжима витой пары и прочих инструментов. Не потребуется тянуть витую пару и делать дырки в стенах. Все, что вам нужно, — это правильно выбрать место для точки доступа (позже из этой книги вы узнаете, как это сделать), затем обойти с ноутбуком всю вашу территорию и убедиться, что везде, где нужно, обеспечивается хорошее качество сигнала. Ваши клиенты без особых проблем смогут подключиться к сети — не нужно думать ни о наличии свободных портов, ни о запасном Ethernet-кабеле. А демонтаж беспроводной сети заключается в демонтаже только точки доступа. Все просто и относительно дешево.

Однако "чисто" беспроводная сеть — это что-то из области фантастики. Полностью отказаться от кабелей все же не получится. Во-первых, скорость передачи данных "по воздуху" все еще намного ниже, чем по кабелю: 54 Мбит/с против

1000 Мбит/с (Gigabit Ethernet). Во-вторых, перехватить данные, передаваемые радиоканалом, проще — злоумышленник может находиться даже не внутри вашего здания (если радиоволны проникают наружу), в случае же с кабельной сетью злоумышленник должен находиться в здании, а это сильно усложняет его задачу. Поэтому серверы и другие компьютеры, которым нужен высокоскоростной и безопасный доступ к сети, лучше подключать по кабелю, а все остальные, например, ноутбуки мобильных пользователей — по беспроводной сети.

У беспроводной сети есть и недостатки. О них вы должны узнать прежде, чем начнете эксплуатировать собственную беспроводную сеть. Вот они:

- набор частот и каналов для разных стран может отличаться. Например, во многих европейских странах можно задействовать два дополнительных канала (12 и 13), использование которых запрещено в США. В Японии есть дополнительный — 14-й канал, а во Франции можно использовать всего 4 канала. На территории бывшего СНГ все проще: Wi-Fi-сети пока не требуют регистрации, и можно использовать все 13 каналов;
- довольно высокое энергопотребление — у вашего ноутбука быстрее сядет аккумулятор при работе в беспроводной сети, чем при использовании GPRS-соединения;
- при построении публичной сети вам придется использовать стандарт шифрования WEP (поскольку стандарты шифрования WPA и WPA2 не поддерживаются "древними" адаптерами), а его можно относительно просто взломать даже при правильной настройке сети. Причем для этого не требуется обладать какой-либо квалификацией — программы для взлома WEP легко найти в Интернете;
- радиус действия Wi-Fi-сети ограничен: 30–50 метров в помещении и около 100 метров снаружи. Ослабить уровень сигнала могут стены (все зависит от материала, из которого они построены), зеркала, микроволновки и соседние беспроводные сети;
- в многоквартирных домах, где жильцы разворачивают свои беспроводные сети, может возникнуть проблема интерференции (наложения сигнала);
- неполная совместимость устройств разных производителей или неполное их соответствие стандартам может привести к снижению производительности сети;
- производительность сети может уменьшиться даже во время дождя (для наружных сетей).

Далее мы рассмотрим основные теоретические принципы беспроводной сети.

6.2. Основные принципы работы беспроводной сети

Человечество научилось использовать радиоволны для передачи информации еще в 1920-х годах. Конечно, до современных беспроводных сетей тогда еще было далеко, но прорыв был сделан.

Примечание

На самом деле радио было изобретено в 90-х годах позапрошлого века. В разных странах радио возникло в разные годы, причем разработки велись параллельно, поэтому

нельзя сказать, что кто-то является единственным отцом радио. В России радио впервые изобрел А. Попов в 1895 году. В Италии разработчиком радио считается Гульельмо Маркони (1896), в США — Никола Тесла (но тут целая история: он первым запатентовал радио, а кто его первым создал — остается загадкой). Однако первооткрывателем способов передачи и приема электромагнитных волн является немецкий ученый Генрих Герц — он разработал эти способы еще в 1888 году.

Если вы в школе физику изучали, а не "проходили" мимо (в свое время я как раз "прошел" ее мимо, а наверстывать пришлось в институте, но это другая история, которая не имеет к книге никакого отношения), то наверняка помните уравнения Максвелла. Эти уравнения показывают изменение магнитного поля под воздействием электрического, и наоборот — как изменяется электрическое поле под действием магнитного. Когда ток перемещается по проводнику, освобождается часть энергии, которая трансформируется в магнитное поле. В свою очередь, это магнитное поле создает переменное электрическое поле, которое опять создает магнитное поле и т. д., пока самый первый поток не будет прерван.

Так вот, при переходе энергии из электрической в магнитную выделяется электромагнитная энергия, то есть собственно радиоволна. Устройство, которое порождает радиоволны, называется *радиопередатчиком*, а устройство, принимающее их, — *радиоприемником*.

Чтобы радиоприемник смог получить радиоволны от радиопередатчика, приемник и передатчик должны работать на одной частоте. Что это за частота? Это частота, на которой переменное электромагнитное поле перемещается из передатчика в пространство. Вот почему радиоволны от тех же FM-станций не смешиваются между собой — потому что каждая FM-станция работает на своей частоте.

Радиочастоты, как и другие частоты, выражаются в герцах (Гц). Радиосигналы передаются на частотах, измеряемых обычно в кило-, мега- и гигагерцах (соответственно, кГц, МГц, ГГц).

Для передачи по радио звука (например, музыки или речи) передатчик смешивает аудиосигнал с несущей волной (это пример амплитудной модуляции — AM) или модулирует аудиосигналом частоту в диапазоне низких частот (это частотная модуляция — FM, frequency modulation). Приемник (AM или FM) определяет несущую волну и отделяет аудиосигнал.

Понятно, что если два передатчика будут передавать сигналы на одной частоте, то сигналы перемешаются. Поэтому в каждой стране есть специальные комитеты связи, регулирующие использование радиочастот. Каждая радиостанция должна получить лицензию на вещание на определенной частоте. Однако есть некоторые частоты, зарезервированные для нелицензионного использования, — лицензия для работы на таких частотах не нужна. Беспроводная компьютерная сеть работает как раз на нелицензированной частоте — вы только представьте, что бы было, если бы каждому пользователю пришлось выдавать лицензию на использование беспроводной точки доступа! Большинство беспроводных сетей Wi-Fi работают на частоте 2,4 ГГц, некоторые отдельные варианты беспроводных сетей используют другой набор частот — 5 ГГц.

С одной стороны, использование нелицензированных частот — это хорошо, поскольку вы можете начать эксплуатировать свою сеть без всяких разрешений контролирующих органов. С другой стороны, массовость превращает достоинства

в недостаток. Если вы собираетесь организовать беспроводную сеть на необитаемом острове, то никаких осложнений не заметите. Но в современных офисных зданиях беспроводные сети могут быть расположены в каждом офисе, что приводит к интерференции сигналов, поскольку радиосигналы с легкостью проникают через стены офисов. Радиус действия беспроводной сети внутри помещения — примерно 35 метров. Но не забывайте, что радиосигналы распространяются сферически. Допустим, ваш офис находится на третьем этаже шестиэтажного здания. Тогда радиосигналы вашей сети будут доступны не только на третьем этаже, но также на первом, втором, четвертом, пятом и, возможно, шестом. По большому счету, одна беспроводная точка доступа может с легкостью охватить одно относительно небольшое здание (в среднем, один этаж занимает 3 метра в высоту, так что девятиэтажка по высоте — примерно 30 метров). Понятно, если еще кто-то в здании развернет беспроводную сеть (совсем не обязательно, что это будет ваш непосредственный сосед, — другая беспроводная сеть может находиться совсем на другом этаже), радиосигналы могут пересекаться. Чтобы все беспроводные сети работали нормально, администраторам этих сетей нужно собраться и скоординировать используемые радиоканалы. О том, как это сделать, будет сказано чуть позже.

Если скоординировать совместное использование сетей не получается или же источником интерференции сигналов является не другая сеть, а некий иной объект, избавиться от которого нельзя, следует понизить мощность радиопередатчика, что снизит и эффект наложения сигналов. Но в этом случае вы можете не охватить всю необходимую территорию. Кстати, по поводу территории. Помните, что радиоволны могут распространяться далеко за пределы вашего здания, и злоумышленнику, чтобы проникнуть в вашу беспроводную сеть, совсем не обязательно находиться на вашей территории — он может сидеть в машине, припаркованной неподалеку. Вот так...

6.3. Расширение спектра

Расширение спектра позволяет повысить эффективность передачи информации через канал с сильными линейными искажениями с помощью модулированных сигналов. Благодаря расширению спектра можно добиться увеличения базы сигнала.

В настоящее время используются три метода расширения спектра:

- FHSS (Frequency Hopping Spread Spectrum, псевдослучайная перестройка рабочей частоты) — несущая частота скачкообразно изменяется по некоторому алгоритму, который известен только приемнику и передатчику. Метод очень просто реализовать, но он не весьма эффективен. Такой метод используется технологией Bluetooth;
- DSSS (Direct Sequence Spread Spectrum, расширение спектра методом прямой последовательности) — более эффективен, чем FHSS, но и более сложен в реализации. Повышает тактовую частоту модуляции, каждому байту передаваемого сообщения ставится в соответствие некоторая достаточно длинная псевдослучайная последовательность;
- OFDM (Orthogonal Frequency Division Multiplexing, мультиплексирование с разделением по ортогональным частотам) — поток данных разбивается на 52 па-

раллельных потока, каждый из которых использует собственную радиочастоту и называется *поднесущей*. Четыре поднесущих содержат данные об остальных 48 потоках. Поскольку сами данные передаются по 48 потокам, а 4 потока используются для передачи служебной информации, реальная максимальная скорость чуть ниже заявленной. Метод OFDM используется в беспроводных сетях;

- CSS (Chirp Spread Spectrum, расширение спектра методом прямой последовательности) — несущая частота перестраивается по линейному закону. Данный метод используется преимущественно в радиолокации.

Современные беспроводные Wi-Fi-стандарты IEEE 802.11a и 802.11g используют метод OFDM. Разница между ними в том, что 802.11a работает на частоте 5 ГГц, а 802.11g — 2,4 ГГц. Есть еще и стандарт 802.11b, работающий на частоте 2,4 ГГц, но использующий метод DSSS.

6.4. Wi-Fi

Институтом инженеров электротехники и электроники (IEEE) разработан набор стандартов для беспроводных сетей — IEEE 802.11. Вот основные стандарты:

- IEEE 802.11 — разработан в 1997 году, охватывает два вида радиопередачи и сети на базе инфракрасных сигналов;
- IEEE 802.11a — охватывает высокоскоростные беспроводные сети;
- IEEE 802.11b — описывает дополнительные спецификации;
- IEEE 802.11g — на этом стандарте основаны практически все современные беспроводные сети.

Скоро будет окончательно утвержден новый стандарт — IEEE 802.11n. Он будет поддерживать скорость передачи данных 480 Мбит/с (текущий стандарт 802.11g поддерживает всего 54 Мбит/с). Поскольку этот стандарт еще не принят, то мы рассматривать его в книге не будем, а кому интересно, тот всегда сможет прочитать о нем по адресу: **http://ru.wikipedia.org/wiki/IEEE_802.11n**.

> ***Примечание***
>
> Некоторые производители оборудования уже начали выпускать оборудование, поддерживающее предварительную версию стандарта IEEE 802.11n, — так называемые pre-802.11n-устройства. Не советую покупать такие устройства, потому что нет никакой гарантии, что такие "предварительные" устройства будут новый стандарт полностью поддерживать, когда его окончательно утвердят.

Сегодня используется в основном стандарт 802.11g, стандарты 802.11a и 802.11b считаются устаревшими. Правда, устаревшие сетевые адаптеры стандарта 802.11b все еще можно использовать в сетях 802.11g, но из-за одного такого адаптера вся сеть будет вынуждена снизить скорость в лучшем случае до 11 Мбит/с, поэтому рекомендуется использовать оборудование одного стандарта. Так что если вы строите публичную сеть (например, сеть для публичной библиотеки, отеля или беспроводную сеть интернет-зала), где будут самые "разношерстные" клиенты, то выбирайте точки доступа стандарта 802.11g, которые будут поддерживать как клиентов 802.11g, так и клиентов с устаревшими адаптерами стандарта 802.11b.

Стандарты 802.11a, b и g — далеко не единственные стандарты семейства IEEE 802.11, остальные стандарты приведены в табл. 6.1.

Таблица 6.1. *Семейство стандартов IEEE 802.11*

Стандарт	**Описание**
IEEE 802.11	Поддерживались скорости 1 и 2 Мбит/с, частота 2,4 ГГц и сети на инфракрасных сигналах
IEEE 802.11a	Скорость передачи данных — 54 Мбит/с, частота 5 ГГц. Стандарт был утвержден в 1999 году, но первые продукты появились в 2001 г.
IEEE 802.11b	Скорости передачи данных 11 Мбит/с и 5,5 Мбит/с (1999 год)
IEEE 802.11c	Описывает операции с мостами
IEEE 802.11d	Поддерживает международные роуминговые расширения (2001 год)
IEEE 802.11e	Обеспечивает поддержку QoS (качество обслуживания)
IEEE 802.11F	Протокол Inter-Access Point Protocol (2003 год)
IEEE 802.11g	Скорость передачи данных — 54 Мбит/с, частота 2,4 ГГц. Стандарт обратно совместим с 802.11b. Дата утверждения стандарта — 2003 год
IEEE 802.11h	Распределение по спектру 802.11a (5 ГГц) для лучшей совместимости в Европе (2004 год)
IEEE 802.11i	Дополнения, касающиеся безопасности
IEEE 802.11j	Специальные расширения для Японии (2004 год)
IEEE 802.11k	Различные незначительные изменения
IEEE 802.11l	Не используется, зарезервирован
IEEE 802.11m	Различные незначительные изменения
IEEE 802.11n	Планируется скорость передачи данных до 480 Мбит/с, частота 2,4–2,5 или 5 ГГц. Обратная совместимость с 802.11a/b/g. Пока не утвержден
IEEE 802.11o	Не используется, зарезервирован
IEEE 802.11p	Беспроводной доступ для транспортных средств, например, для машин скорой помощи
IEEE 802.11q	Не используется, зарезервирован
IEEE 802.11r	Быстрый роуминг

Таблица 6.1 *(окончание)*

Стандарт	Описание
IEEE 802.11s	Расширенный набор сервисов (ESS) Mesh Networking
IEEE 802.11T	Это не стандарт, а рекомендация относительно проведения тестов и измерений
IEEE 802.11u	Описывает взаимодействие с не-802 сетями (например, с сотовыми сетями)
IEEE 802.11v	Описывает управление беспроводными сетями
IEEE 802.11x	Не используется, зарезервирован
IEEE 802.11w	Описывает защищенные управляющие кадры (Protected Management Frames)

Характеристики стандартов 802.11a/b/g/n приведены в табл. 6.2.

Таблица 6.2. *Характеристики стандартов Wi-Fi*

Стандарт	Частота, ГГц	Реальная скорость передачи, Мбит/с	Максимальная скорость передачи, Мбит/с	Радиус покрытия
802.11b	2,4	5	11	~30 м (внутри) ~100 м (снаружи)
802.11a	5	20	54	~35 м (внутри) ~110 м (снаружи)
802.11g	2,4	20	54	~35 м (внутри) ~110 м (снаружи)
802.11n	2,4	150	480	~70 м (внутри) ~160 м (снаружи)

Примечание

Радиус покрытия во многом зависит от точки доступа. По стандарту он составляет примерно 35 метров внутри помещения и около 100 метров снаружи. Но современные точки доступа позволяют охватывать значительно большую территорию. Например, точка доступа D-LINK DWL-2100AP обладает радиусом действия 100 и 400 метров (соответственно, внутри и снаружи). И это недорогая точка доступа — она относится к средней ценовой категории. А точка доступа ENCORE ENRXWI-SG обладает меньшим радиусом действия (но и стоит дешевле, чем точка доступа от D-Link): 30–50 метров внутри помещения и 50–200 метров снаружи. Однако даже эти значения превышают стандартные.

Множество стандартов удручает? Поэтому гении маркетинга решили назвать все стандарты семейства IEEE 802.11 одним красивым термином — Wi-Fi[1] (кратко

[1] От англ. *Wireless Fidelity* — беспроводная точность.

и созвучно с Hi-Fi). Кто эти гении? Специалисты группы WECA (Wireless Ethernet Compatibility Alliance, Альянс совместимости беспроводных Ethernet-сетей). WECA проводит тестирование и сертификацию оборудования различных производителей. Если на коробке с Wi-Fi-устройством вы увидели логотип Wi-Fi (рис. 6.1), значит, находящееся в коробке устройство совместимо с другими устройствами с таким же логотипом.

Рис. 6.1. Логотип Wi-Fi

6.5. Радиочастоты и каналы Wi-Fi

6.5.1. Стандарты 802.11b и 802.11g

Как уже было отмечено, сети 802.11b и 802.11g работают на частоте 2,4 ГГц, но это не значит, что рабочая частота именно 2,400 ГГц, — сеть может использовать диапазон частот 2,400–2,4835 ГГц. Именно в таком диапазоне частот работают Wi-Fi-сети в Европе и США. Исключение составляют Франция, Испания и Япония, где рабочий диапазон частот чуть другой:

- 2,445–2.475 ГГц — в Испании;
- 2,4465–2.4835 ГГц — во Франции;
- 2,471– 2.497 ГГц — в Японии.

Вообще, точные значения частот нам не интересны, поскольку все продаваемое в нашей стране оборудование сертифицировано и использует частоты 2,400–2,4835 ГГц.

Какую именно частоту из диапазона 2,400–2,4835 ГГц будет использовать ваша сеть? Рабочая частота сети определяется радиоканалом, на котором она работает (помните, говорили о каналах чуть ранее). В табл. 6.3 приведено распределение беспроводных каналов.

Таблица 6.3. *Распределение беспроводных каналов (для 802.11b и 802.11g)*

Канал	Частота, ГГц	Канал	Частота, ГГц
1	2,412	8	2,447
2	2,417	9	2,452
3	2,422	10	2,457
4	2,427	11	2,462
5	2,432	12	2.467
6	2,437	13	2,472
7	2,442	14	2,484

В Европе и США используется 13 каналов, в Японии — 14. Опять всю картину портит Франция — она использует всего 4 канала, но поскольку мы к французам не имеем никакого отношения, то и говорить больше о них не будем.

Итак, если ваша сеть работает на первом канале, то рабочей частотой будет 2,412 ГГц, если на третьем, то — 2,422 ГГц.

Помните, мы говорили о интерференции (наложении) радиосигналов двух рядом работающих сетей? По умолчанию беспроводные точки доступа настроены на работу на первом канале. Если рядом (на расстоянии, которое не превышает 35 метров) размещены две точки доступа, работающие на первом канале, то их радиосигналы будут накладываться. Чтобы две сети не мешали друг другу, одну из сетей нужно перенести на другой канал, например, на 5-й. Тогда одна сеть будет работать на частоте 2,412 ГГц, а другая — на 2,432 ГГц.

Дабы полностью исключить интерференцию, нужно, чтобы частоты сетей отличались на 25 МГц, или на 5 каналов. Оптимальная раскладка для трех рядом расположенных сетей: это каналы 1, 6 и 11. Если надо обеспечить совместную работу четырех рядом расположенных сетей, тогда можно использовать каналы 1, 5, 9, 13. В этом случае "расстояние" между ними будет равно четырем каналам — все сети будут ощущать небольшое вмешательство, но не критичное, и все они смогут работать почти с максимальной производительностью. Если вы хотите полностью исключить интерференцию сигналов четырех сетей, нужно также понизить мощность передатчика каждой точки доступа. Правда, в этом случае охватываемая территория может стать меньше (например, вместо 35 метров покрытия будет охвачено всего 30). Но в большинстве случаев и это не критично. Ведь средняя площадь квартиры или небольшого офиса — примерно 60–65 квадратных метров. Грубо говоря, нам нужно охватить "коробочку" размером 8×8 м. Даже если вы понизите мощность передатчика так, что он будет охватывать радиус в 20 метров, этого будет достаточно, чтобы охватить все помещение.

6.5.2. Стандарт 802.11a

Стандарт 802.11a использует диапазон частот 5,00–5,34 ГГц. Каналы для этого стандарта "шириной" в 20 МГц (а не в 25, как в случае с 802.11g). Распределение каналов для этого стандарта приведено в табл. 6.4.

Таблица 6.4. *Распределение беспроводных каналов для 802.11a*

Канал	Частота, ГГц	Канал	Частота, ГГц
34	5,17	46	5,23
36	5,18	48	5,24
38	5,19	52	5,26
40	5,20	56	5,28
42	5,21	60	5,30
44	5,22	64	5,32

В Европе используются каналы 36, 40, 44 и 48. Остальные каналы задействованы в неевропейских странах — например, каналы 34, 38, 42 и 46 заняты под Японию. В США используются "европейские" каналы плюс каналы 52, 56, 60 и 64.

6.6. Режимы работы сети

Обычно беспроводная сеть является *централизованной*, или, как ее еще называют, *инфраструктурной* (рис. 6.2). Центральным устройством выступает точка доступа. На рис. 6.2 показано, что ноутбуки для подключения к беспроводной сети используют встроенные адаптеры, а стационарные компьютеры — внешние. Чуть позже вы узнаете, почему для стационарных компьютеров лучше подходят внешние адаптеры.

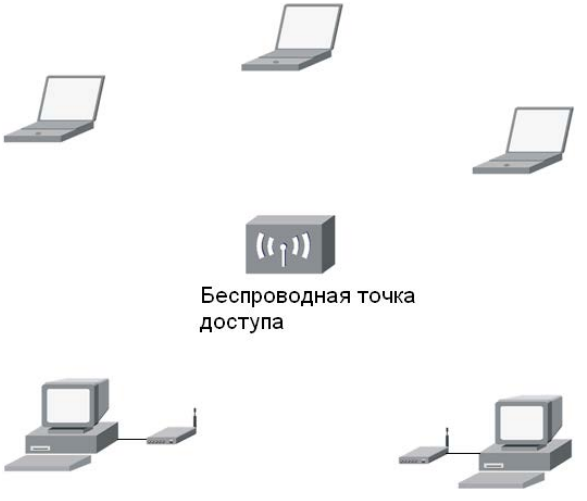


Рис. 6.2. Централизованная сеть

Но при желании можно организовать одноранговую беспроводную сеть без центрального устройства. Такие сети называются *ad hoc-сетями*. Сетевые адаптеры компьютеров переводятся в режим ad hoc и обмениваются данными напрямую, без участия точки доступа. Такой режим полезен, когда есть два компьютера (например, два ноутбука), но рядом нет никакой точки доступа, а передать данные нужно. В такой сети максимальная скорость передачи данных составляет всего 1 Мбит/с, но для обмена данными между всего двумя узлами этого вполне достаточно.

Напоследок рассмотрим несколько терминов, которые вы можете встретить при чтении документации по беспроводным сетям (например, при чтении руководства по точке доступа):

- BSS (Basic Service Set) — обычная беспроводная сеть с одной точкой доступа;
- ESS (Extended Service Set) — беспроводная сеть с двумя или больше точками доступа;
- IBSS (Independent Basic Service Set) — одноранговая беспроводная сеть без точки доступа.

В следующей главе будет больше практики — мы поговорим о выборе оборудования для нашей беспроводной сети.

6.7. Основные сетевые устройства беспроводной сети

Для построения беспроводной сети нам понадобится одна *точка доступа* (в англ. терминологии wireless access point) и несколько *беспроводных сетевых адаптеров* (в англ. терм. wireless adapter) — по количеству *стационарных* компьютеров. Напомню, что точка доступа выполняет роль центрального устройства сети. Попросту говоря, это тот же коммутатор, но для беспроводной сети. Такое сравнение сугубо ассоциативное, просто чтобы у вас сформировалось представление о функциях точки доступа.

Если вы планируете построить довольно большую беспроводную сеть, то вам понадобится несколько точек доступа. Просчитать зону покрытия относительно просто: в помещении радиус действия точки доступа составляет примерно 35 метров, снаружи — примерно 100 метров. Обычно радиосигналы точки доступа распространяются с одинаковой мощностью по всем направлениям, поскольку точки доступа по умолчанию оснащаются всенаправленными антеннами. Если нужно обеспечить покрытие сети только в одном направлении, то понадобится направленная антенна. В этом случае вы можете охватить примерно 70 метров внутри помещения и около 200 метров (иногда даже больше) — снаружи.

В помещениях, в плане близких к квадрату, лучше всего использовать всенаправленную антенну (рис. 6.3). Угол распространения радиосигнала направленной антенны составляет 45 градусов (рис. 6.4), это тоже нужно учитывать при построении сети. Хотя бывают антенны с другим углом апертуры (распространения радиосигнала), но об этом мы поговорим в следующей главе, когда будем планировать свою сеть.

Забегая вперед (вообще-то об этом нужно говорить при планировании сети, но ведь оборудование выбирается сейчас!), скажу, что бывают не очень приятные ситуации, связанные с расположением стационарных компьютеров. Такая ситуация изображена на рис. 6.5. Вы установили точку доступа, которая охватывает практически всю необходимую территорию, но вне зоны покрытия остался один стационарный компьютер. С такими компьютерами всегда сложнее — ведь ноутбук можно легко перенести на другое место, а вот проделать то же самое со стационарным компьютером не всегда просто.

Что делать? Покупать вторую точку доступа? Но это нерационально — ведь компьютер всего один. В этом случае поможет беспроводной сетевой адаптер с возможностью подключения внешней антенны. Такой сетевой адаптер подключается к тому самому неохваченному компьютеру, а к нему, в свою очередь, подключается направленная антенна, которая и направляется в сторону точки доступа. Все — проблема решена.

Что же касается беспроводных сетевых адаптеров, то вам нужно купить их столько, сколько у вас стационарных компьютеров. Как правило, все современные ноутбуки оснащены беспроводными адаптерами 802.11g, поэтому покупать беспроводные адаптеры для ноутбуков необходимости нет.

Итак, теперь мы знаем, что понадобится для нашей сети: пока одна точка доступа и несколько беспроводных сетевых адаптеров. Как правило, беспроводными сетевыми адаптерами оснащаются все современные ноутбуки и нетбуки, поэтому

выбирать адаптер вам не придется (что же касается стационарных компьютеров, то они, как правило, подключаются к сети по витой паре). А вот о выборе точки доступа нужно поговорить.

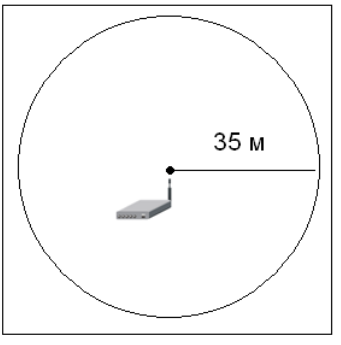


Рис. 6.3. Зона покрытия при использовании всенаправленной антенны

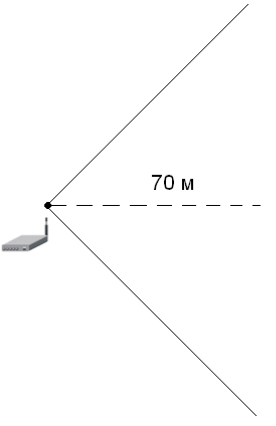


Рис. 6.4. Зона покрытия сети при использовании направленной антенны

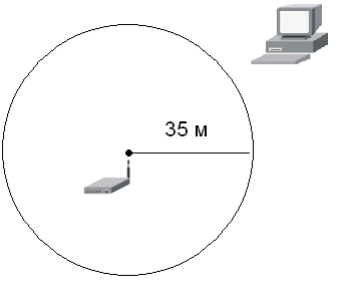


Рис. 6.5. Один компьютер остался вне зоны сети

6.8. Выбор точки доступа

При выборе точки доступа нужно учитывать следующие факторы:

- ❒ поддерживаемые точкой доступа стандарты;
- ❒ область применения точки доступа (внутреннее или наружное);
- ❒ радиус покрытия;
- ❒ тип антенны, наличие разъема для подключения внешней антенны;
- ❒ алгоритм шифрования;
- ❒ дополнительные функции.

Типичная точка доступа изображена на рис. 6.6.

Рис. 6.6. Точка доступа

6.8.1. Поддерживаемые точкой доступа стандарты

Современные точки доступа поддерживают стандарты 802.11b и 802.11g. Иногда встречаются комбинированные точки доступа, поддерживающие стандарты 802.11a и 802.11g. Но такие точки доступа — редкость, и их стоимость существенно превышает стоимость обычных точек доступа стандарта 802.11g. Необходимость в таких точках доступа может быть только, если вы настраиваете публичную сеть и хотите, чтобы к ней могли подключиться клиенты с адаптерами всех типов.

6.8.2. Область применения и радиус действия точки доступа

Для многих точек доступа указывается область внутреннего и наружного радиуса действия. Однако это не означает, что данная точка доступа может эксплуатироваться за пределами помещения. Значение наружного радиуса действия можно использовать только в ознакомительных целях для оценки мощности передатчика точки доступа. Если в характеристиках точки доступа, например, указано: "Радиус покрытия: внутри помещения: 30–50 м, на открытом пространстве: 50–200 м", то это означает, что в обычном помещении радиус действия составит от 30 до 50 метров в зависимости от материала стен и наличия в окружающей среде радиопомех. Но гарантировано, что вы получите радиус в 30 метров. А вот если в этом помещении убрать все стены и исключить радиопомехи, то точка доступа сможет охватить радиус в 200 метров.

Точки доступа для наружного размещения (в наименовании таких точек доступа часто указывается "outdoor") стоят, как правило, в несколько раз дороже, чем обычные. Это связано с особым корпусом точки доступа, который защищает ее от воздействия окружающей среды: дождя, мороза, влажности. "Комнатная" точка доступа может работать при температуре от 0 до 55 градусов. Наружная точка доступа (рис. 6.7) может работать и при отрицательных температурах. Обычная точка доступа, скорее всего, не выживет даже после дождя, а наружную от погодных условий защищает корпус.

Рис. 6.7. Наружная точка доступа

Зачем нужны наружные точки доступа? В Америке они обычно используются для построения так называемых *кампусных сетей* — сетей, которые развернуты на территории университетского городка (кампуса), причем такие сети обеспечивают доступ как внутри помещения, так и снаружи. Студентам некоторых отечественных вузов приходится только мечтать об Интернете в общежитиях, не говоря уже об Интернете за его пределами...

Другое дело, если у вас частный дом, и вы хотите, чтобы Интернет был и во дворе. Довольно удобно летом сидеть в беседке и работать в Интернете. А почему бы и нет? Сказано — сделано. Но в этом случае можно немного сэкономить. Разместите обычную ("внутреннюю") точку доступа ближе к наружной стене дома.

Если использовать некоторые модели с повышенным радиусом действия, то вы получите охват даже внутри помещения порядка 100 метров. А этого вполне хватит, чтобы охватить и дом, и двор.

6.8.3. Антенна точки доступа

Практически у всех точек доступа антенны внешние, и в некоторых случаях "на борту" точки доступа имеются даже две антенны (такие точки доступа обеспечивают наилучший прием, поэтому лучше отдать предпочтение подобным моделям). Вы можете изменять угол наклона каждой антенны для обеспечения наилучшего покрытия.

Забегая вперед, подскажу: если вы планируете разместить точку доступа на столе, то антенны нужно направить вверх, а если вам хочется разместить точку доступа под потолком, тогда антенны нужно направить вниз. Так будет обеспечен наилучший прием сигналов.

Желательно также, чтобы у точки доступа была возможность подключения дополнительной антенны. Мало ли когда это может понадобиться.

6.8.4. Алгоритм шифрования

Существуют два алгоритма шифрования данных, передаваемых по беспроводной сети: WEP (Wired Equivalent Privacy) и WPA (Wi-Fi Protected Access). Алгоритм WEP по своей надежности напоминает швейцарский сыр (или решето). В Интернете имеется масса программ для взлома WEP-защиты, так что взломать сеть, использующую WEP, может даже школьник. Ради справедливости нужно отметить, что и WPA не панацея, но этот метод шифрования намного надежнее, чем WEP, поэтому следует покупать точку доступа, которая поддерживает WPA.

Впрочем, WPA поддерживают практически все современные точки доступа, поэтому сейчас можно было бы и не упоминать об алгоритмах шифрования. Но я физически не могу ознакомиться со всеми моделями всех производителей. Может, где-то есть ультрапростые и ультрадешевые точки доступа, которые не поддерживают WPA, а вы присмотрели именно такое устройство из-за его дешевизны.

6.8.5. Дополнительные функции

Перед покупкой точки доступа очень важно ознакомиться с ее дополнительными возможностями. Как минимум, у каждой точки доступа должен быть порт для подключения к коммутатору проводной Ethernet-сети. Схема сети в этом случае будет такой, как показано на рис. 6.8.

Некоторые точки доступа сочетают в себе функции коммутатора. Правда, портов совсем немного (обычно 4 или 8), но для небольших домашних и офисных сетей — это решение. Тогда схема сети будет значительно проще (рис. 6.9).

Но обратите внимание — у нас маршрутизатор все еще выступает в виде отдельного устройства. Это может быть аппаратное устройство, к которому подключается DSL-модем, или же компьютер, к которому мы подключили DSL-модем и установили специальное программное обеспечение, выполняющее функции шлюза. Но можно сделать нашу сеть еще проще. Имеются точки доступа с функциями

и маршрутизатора, и брандмауэра. В этом случае мы получаем одно единое устройство, которое будет полностью обслуживать нашу сеть (рис. 6.10).

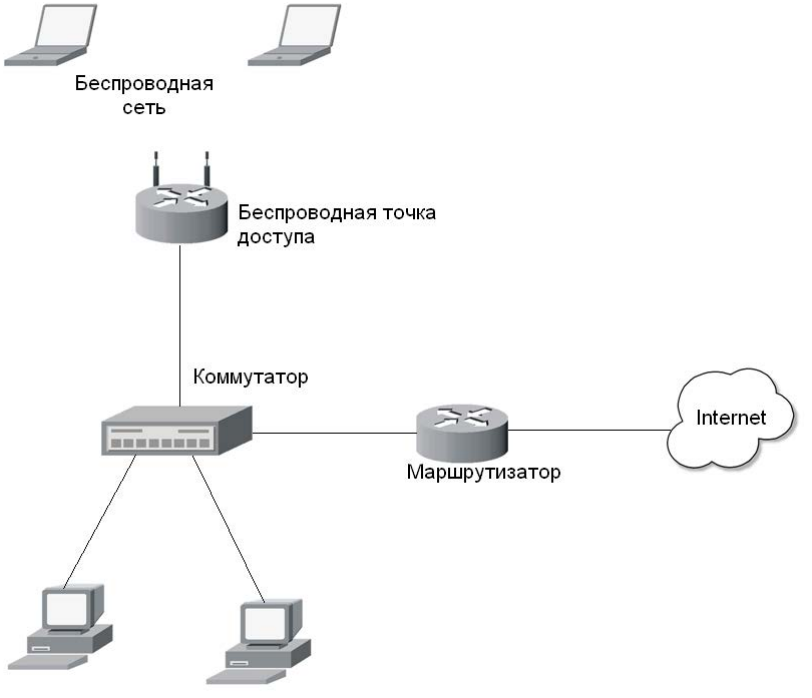


Рис. 6.8. Схема сети с обычной точкой доступа

Рис. 6.9. Небольшое упрощение схемы сети

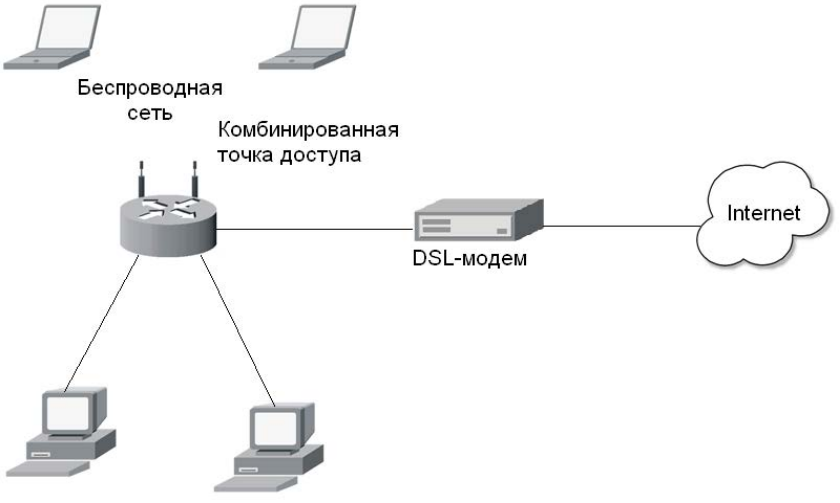


Рис. 6.10. Упрощение схемы сети с помощью комбинированной точки доступа

Теперь, когда вы знаете все нюансы выбора точки доступа, можете отправляться в магазин (конечно же, в интернет-магазин — там дешевле) за покупкой. Конкретные модели описывать я не стану, поскольку модельный ряд, как и цены, периодически обновляются.

6.9. Настройка беспроводной сети

6.9.1. Выбор расположения точки доступа

Настройка точки доступа — не очень сложный процесс. Обычно программа конфигурации позволяет установить основные параметры, что займет 5–10 минут, и после этого ваша сеть готова к работе. Но вот *как* она будет работать, напрямую зависит от размещения точки доступа. Если у вас небольшое, квадратное (или около того) в плане помещение, площадью примерно 60 кв. метров, то лучше всего разместить точку доступа в центре этого помещения. С учетом материала стен, наличия радиопомех и некоторых бытовых приборов (напр., микроволновок) можно рассчитывать на радиус действия точки доступа порядка 35 метров. Некоторые точки доступа в зависимости от мощности передатчика и типа антенны могут охватывать куда большие расстояния: от 45 до 100 метров. При наружном размещении точки можно рассчитывать на минимальный радиус действия 60 метров, максимальный — 300 метров (табл. 6.5).

Таблица 6.5. *Радиус действия точки доступа с всенаправленной антенной*

Использование точки доступа	Минимальный радиус, м	Максимальный радиус, м
Внутри помещения	35	100
За пределами помещения	60	300

При планировании размещения точки доступа я бы рекомендовал ориентироваться на минимальный радиус действия, поскольку максимальный зависит от слишком многих факторов, включая модель точки доступа, наличие помех, материал стен и даже время года! Так, если вы разворачиваете беспроводную сеть во дворе (то есть за пределами помещения), то зимой радиус действия будет больше, чем летом. Дополнительные помехи создает листва — зимой ее просто нет. Поэтому при построении наружных сетей старайтесь установить внешние антенны выше крон деревьев. Во всяком случае, всегда нужно помнить об эффекте листвы при планировании наружной сети.

Если вам требуется охватить большое помещение, протяженностью, скажем, в 70–80 метров, лучше не рассчитывать, что одна точка доступа обеспечит всю зону покрытия, — в этом случае следует разместить две точки доступа. Но разместить их нужно правильно. На рис. 6.11 приведены варианты не совсем правильного расположения точек доступа в прямоугольном помещении. На рис. 6.11, *а* они расположены слишком близко к граничным стенам помещения, поэтому охватывают соседнюю территорию, а в центре помещения образуется огромная "мертвая" зона, где нет беспроводной связи.

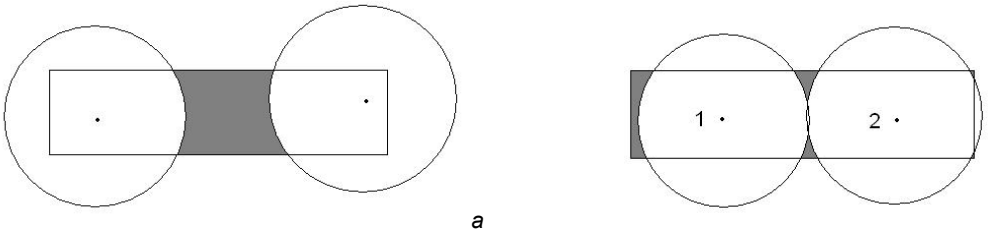


Рис. 6.11. Неправильное расположение точек доступа: *а* — "мертвая" зона в центре помещения; *б* — в центре сигнал слабый, слева — "мертвая" зона

На рис. 6.11, *б* точки доступа расположены намного лучше, но тоже не идеально. Здесь они смещены ближе к центру помещения. Поскольку помещение узкое, мощность передатчиков немного снижена — чтобы зона покрытия чуть меньше (по сравнению с расположением точек доступа на рис. 6.11, *а*) выходила за пределы помещения. Слева наблюдается небольшая "мертвая" зона, а в центре помещения — слабый сигнал. Чтобы избавиться от этих недостатков, нужно точку доступа 1 сдвинуть на пару метров влево. Точку доступа 2 тоже нужно сдвинуть немного влево, но при этом не перестараться, иначе справа появится такая же "мертвая" зона, как была до этого слева. Если важно покрытие именно в центре помещения, то обе точки доступа следует сдвинуть ближе к центру помещения. Понятно, что реальная конфигурация помещения может быть намного сложнее, и вполне вероятно, что вам понадобится даже три точки доступа, чтобы полностью покрыть всю территорию.

Не нужно забывать и о другом факторе — возможности одной точки доступа не безграничны. Предположим, что у нас есть идеально квадратное помещение размером 30×30 метров. Проблем с покрытием не будет — охватить такое помещение с легкостью сможет одна точка доступа. Но одна точка доступа сможет обслужить

примерно 20–30 клиентов. После этого ее производительность начнет снижаться. Другими словами, пользователи будут жаловаться, что сеть медленно работает. Поэтому на загруженных участках, где предполагается одновременная работа более 20 клиентов, лучше установить несколько точек доступа из расчета 20–30 клиентов на одну точку доступа.

Примечание

При выборе места расположения точки доступа не нужно забывать и об электрических розетках — ведь точки доступа надо подключить к сети напряжения. Если нет желания тянуть огромную "переноску" или делать еще одну розетку, можно выбрать точку доступа, которая получает питание по Ethernet-кабелю (технология PoE — Power over Ethernet).

6.9.2. Физическая установка точки доступа

Вообще, при физической установке точки доступа нужно руководствоваться инструкцией, к ней прилагаемой (ведь все точки доступа — разные). Но иногда такие инструкции оставляют желать лучшего (или же попросту инструкция на русском языке отсутствует).

Основные действия по установке точки доступа следующие:

1. Вскройте упаковку точки доступа и, если нужно, соберите ее — например, подключите антенны, смонтируйте подставку для точки доступа, если она есть в комплекте.
2. Установите точку доступа в заранее выбранное место. Вы уже, надеюсь, определились с местом, где она должна быть установлена?
3. Антенны установите под углом 90 градусов к корпусу. Если точка доступа расположена на столе или на полке, тогда антенны (или антенну, если она одна) нужно направить вверх. Если же точка доступа расположена под потолком, то антенны следует направить вниз.
4. Подключите кабель питания. Если ваша точка доступа поддерживает PoE (электропитание по кабелю локальной сети), тогда подключите к ней Ethernet-кабель.
5. Подключите Ethernet-кабель к ближайшему коммутатору, маршрутизатору или к другой точке доступа — все зависит от схемы вашей сети.
6. Включите точку доступа.

Вот некоторые дополнительные правила правильного размещения точки доступа:

- размещайте ее на ровной горизонтальной поверхности. Если вы крепите точку доступа к стене, то нужно смонтировать ее горизонтально, а не вертикально;
- не загораживайте вентиляционные отверстия точки доступа во избежание ее перегрева;
- желательно подключать точку доступа через стабилизатор напряжения или ИБП (источник бесперебойного питания) для уменьшения риска ее повреждения при скачках напряжения и разрядах молнии.

Внимание!

После физической установки точки доступа можно приступить к ее настройке. Однако тут есть один нюанс. Предположим, вы используете точку доступа, поддерживающую технологию PoE, и там, где вы собираетесь ее устанавливать, нет свободной электроро-

зетки. Другими словами, чтобы точка доступа смогла получить питание, вы сразу подключаете ее к Ethernet-кабелю, ведущему в вашу сеть. Не нужно этого делать! Ведь запустится DHCP-сервер точки доступа, что может привести к возникновению коллизий с основным DHCP-сервером сети. Сначала нужно запитать точку доступа от обычной розетки, подключиться к ней по Wi-Fi с любого ноутбука, настроить ее, а только потом уже подключать точку доступа к существующей сети.

6.9.3. Практическая настройка беспроводной сети

Точка доступа D-Link DSL-2640U

Рассмотрим построение реальной беспроводной сети на базе точки доступа D-Link DSL-2640U. Почему я остановил свой выбор на D-Link, не спрашивайте — просто примите как есть. Да и не худшая это модель. И, вообще, нельзя сказать, что D-Link — это плохо, а ZyXEL — хорошо, или наоборот. Нужно говорить о конкретных моделях. Среди моделей ZyXEL иногда встречаются не совсем удачные, а некоторые D-Link по своим функциям не уступают устройствам от ZyXEL, но при этом стоят ощутимо дешевле.

Полный комплект DSL-2640U изображен на рис. 6.12. Что сразу бросается в глаза — это четыре порта для организации локальной сети (то есть точка доступа поддерживает функции коммутатора), телефонный разъем RJ-11 (точка доступа выполняет функции DSL-модема). В комплекте также был DSL-сплиттер, кабель Ethernet (RJ-45) и два телефонных кабеля RJ-11 — мелочь, а приятно. Другими словами, все, что нужно для установки этой точки доступа, поставляется в комплекте с ней, и больше ничего не нужно покупать.

Рис. 6.12. Точка доступа DSL-2640U:
а — вид спереди; *б* — вид сзади; *в* — комплект поставки

Внимание!

На задней панели этой точки доступа (впрочем, как и других устройств от D-Link), имеется загадочный порт Console и кнопка Reset. Порт Console предназначен только для сервисного персонала D-Link, а вот с кнопкой Reset нужно быть предельно осторожным! Это не перезагрузка точки доступа, как можно подумать сразу (сначала я так и подумал!), — это кнопка сброса всех настроек. То есть после нажатия этой кнопки все параметры будут сброшены к заводским значениям. Для настроенной точки доступа нажатие этой кнопки означает, что придется ее настраивать сначала.

Как выяснилось из руководства, точка доступа, помимо своих собственных функций, выполняет также функции ADSL-модема, коммутатора, маршрутизатора и брандмауэра. Брандмауэр тоже непростой — кроме фильтрации пакетов (это стандартная функция брандмауэра), он умеет выполнять фильтрацию по MAC-адресу и содержит средства защиты от DoS-атак (атак на отказ).

Предварительная настройка

Итак, начнем настройку нашей сети. Первым делом установите точку доступа в предназначенное ей место. Затем подключите ADSL-сплиттер (он входит в комплект поставки) к телефонной линии. Подключите к ADSL-сплиттеру телефон и точку доступа с помощью обычного телефонного кабеля (тоже входит в комплект поставки).

Затем подключите к точке доступа все стационарные компьютеры с помощью обычного Ethernet-кабеля. Поскольку эта модель поддерживает только Fast Ethernet, можно обойтись витой парой 5-й категории (не 5E и не 6-й). О том, как самостоятельно обжать витую пару, рассказано в *главе 5*.

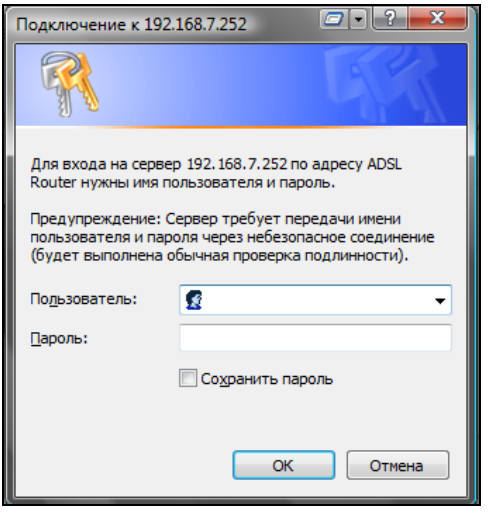


Рис. 6.13. Окно входа в программу управления точкой доступа

Практически все готово — можно приступать к настройке точки доступа. Первый доступ к ней лучше производить со стационарного компьютера или ноутбука, подключенного к точке доступа с помощью Ethernet-кабеля (он входит в состав поставки). При этом компьютер должен быть настроен на автоматическое получение IP-адреса (по умолчанию оно так и есть).

Откройте браузер и подключитесь к узлу 192.168.1.1 (по умолчанию у точки доступа именно такой IP-адрес), набрав в адресной строке: `http://192.168.1.1`. Вы увидите окно, подобное изображенному на рис. 6.13. Имя пользователя по умолчанию: `admin`, пароль: `admin` — введите их и нажмите кнопку **OK**.

Примечание

На рис. 6.13 вы видите другой IP-адрес — это потому, что я подключался к реальной и уже настроенной сети и не хотелось для создания иллюстрации изменять IP-адрес точки доступа.

Прежде всего зайдите на вкладку **Home** (рис. 6.14) и на левой панели окна нажмите кнопку **Wizard**. Убедитесь, что параметр **DSL Auto-connect** включен, и нажмите кнопку **Next**.

Примечание

Мастер настройки точки доступа позволяет за несколько минут настроить точку доступа, в чем мы сейчас и убедимся. Но все иллюстрации мастера приводить я не стану. Пользователям D-Link они мало помогут — ведь на прилагаемом к ней компакт-диске имеется руководство на русском языке, а пользователям других точек доступа иллюстрации программы настройки D-Link будут вообще бесполезны. Однако не нужно думать, что этот материал помещен в книгу зря. Этапы настройки и название конфигурационных параметров сходны для точек доступа всех производителей. Зная примерное название опции, вы сможете настроить точку доступа другого производителя "по образу и подобию". Настраивая D-Link, я буду комментировать процесс настройки и, кроме всего прочего, описывать опции, которым уделено мало внимания в руководстве пользователя.

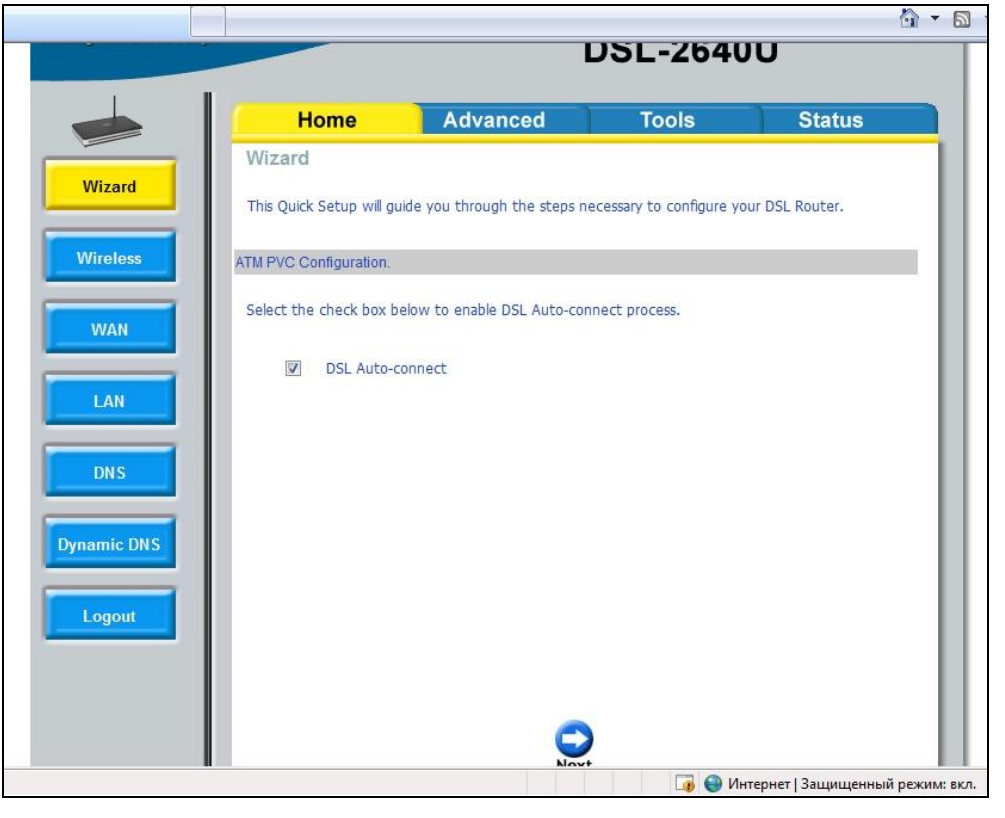


Рис. 6.14. Программа настройки точки доступа

Первым делом мастер попросит выбрать тип соединения (**Connection type**). По умолчанию выбран тип **PPPoA** (PPP over ATM), но отечественные провайдеры чаще используют другой тип соединения — **PPPoE** (PPP over Ethernet), его и нужно выбрать. Впрочем, не помешает заглянуть в договор со своим провайдером и уточнить тип соединения.

Следующий шаг — это ввод имени пользователя (**PPP username**) и его пароля (**PPP Password**). Для экономии трафика (если у вас не безлимитное соединение или соединение с почасовой оплатой) можно выбрать опцию **Dial on demand**. В этом случае соединение с Интернетом будет установлено, только если один из клиентов сети обратится к Интернету. В противном случае соединение будет установлено при включении точки доступа.

Опция **Keep Alive** разрешает поддерживать соединение с провайдером, даже если оно долгое время не использовалось. Если опции **Dial on demand** и **Keep Alive** включены, точка доступа поведет себя следующим образом: соединение с провайдером будет установлено не при запуске точки доступа, а при первом обращении к Интернету одного из клиентов. После этого соединение не будет разорвано, пока не выключат точку доступа или пока не произойдет разрыв соединения со стороны провайдера. Такая схема очень удобна для домашней сети, поскольку дома Интернет нужен не всегда. Если у вас не безлимитка, а тариф с почасовой оплатой, то при запуске точки доступа лучше сразу не устанавливать соединение. Например, вы просто хотите скачать файл с другого компьютера локальной сети. Включаете точку доступа, она "поднимает" интернет-соединение, а вам потом придется платить за фактически не использованное время доступа к Интернету. Так что пусть точка доступа соединяется с провайдером только тогда, когда вам, действительно, понадобится Интернет. Если вы уж совсем экономны, то можно выключить параметр **Keep Alive** — при длительном простое точка доступа отключится от Интернета.

Опция **Use static IP Address** позволяет установить статический IP-адрес для вашего интернет-соединения. Обычно в этом нет необходимости, поскольку IP-адрес динамически назначается DHCP-сервером провайдера. С другой стороны, если провайдер назначает IP-адреса статически (эта услуга провайдера, как правило, платная), такая возможность иногда будет довольно полезной.

Иногда нужно указать шлюз по умолчанию (этот факт можно уточнить в службе поддержки провайдера). Для этого включите параметры **Use the following default gateway** и **Use IP Address**. Опция **Use IP Address** позволяет задать IP-адрес шлюза.

На следующем шаге можно (и нужно!!!) включить NAT и брандмауэр. Включите параметры **Enable NAT** и **Enable Firewall** и нажмите **Next**.

Далее установите следующие параметры:

- **IP Address** — IP-адрес точки доступа (по умолчанию 192.168.1.1), в принципе можно не изменять, если у вас нет существующей Ethernet-сети, к которой вы подключаете точку доступа. Если такая сеть есть, надо ввести свободный IP-адрес, принадлежащий этой сети. В любом случае из соображений безопасности (об этом мы поговорим в *главе 17*) рекомендуется изменить IP-адрес точки доступа на какое-либо значение, отличное от установленного по умолчанию. IP-адрес точки доступа не должен принадлежать диапазону динамических IP-адресов, которые будут "раздаваться" DHCP-сервером (см. далее);

- **Subnet Mask** — маску подсети (по умолчанию 255.255.255.0) при настройке новой сети можно не изменять. Если вы подключаете точку доступа к существующей сети, нужно указать ее маску;
- **Disable DHCP** — позволяет отключить DHCP-сервер. Этот параметр нужно установить, если вы планируете использование отдельного DHCP-сервера (вы должны знать, как настроить такой сервер!);
- **Enable DHCP** — включить DHCP-сервер. Здесь же можно установить его параметры:
  - **Start IP Address** — начальный IP-адрес диапазона IP-адресов, из которого DHCP-сервер будет назначать адреса (по умолчанию 192.168.1.2). Если вы настраиваете новую сеть, это значение можно не изменять. А при подключении точки доступа к существующей сети вы должны знать, какой диапазон IP-адресов у вас занят, а какой — свободен;
  - **End IP Address** — конечный IP-адрес диапазона IP-адресов;
  - **Leased Time** — время аренды IP-адреса в часах (значение по умолчанию — 24 часа). Через 24 часа DHCP-сервер назначит всем клиентам, работающим 24 часа подряд, новые IP-адреса.

Совет

Как уже здесь несколько раз отмечалось, если вы настраиваете новую сеть, то можно оставить все параметры по умолчанию. Однако из соображений безопасности ряд параметров рекомендуется изменить — например, назначить точке доступа другой IP-адрес, скажем, 192.168.1.99. А диапазон IP-адресов DHCP-сервера установить так: 192.168.1.1 — 192.168.1.98. Вам мало 98 адресов? Не забывайте, что эта точка доступа может обслужить одновременно всего 30 беспроводных клиентов и 4 проводных. Как мне кажется, 98 адресов должно хватить с головой.

Следующий шаг в настройке точки доступа — это установка параметров беспроводной сети:

- параметр **Enable Wireless** позволяет включить функции беспроводной точки доступа. В принципе, вы можете использовать точку доступа в качестве маршрутизатора для проводных клиентов и отказаться от использования беспроводного доступа к сети, если он вам в данное время не нужен. Для этого нужно выключить параметр **Enable Wireless**. Но поскольку мы сейчас настраиваем именно беспроводную сеть, то отключение этого параметра выглядело бы странным;
- параметр **SSID** задает имя (идентификатор) беспроводной сети. Измените SSID, установленный по умолчанию! SSID не должен содержать вашего адреса, номера вашего офиса или квартиры, названия вашей компании. SSID A75SN привлечет внимание злоумышленника меньше, чем VaBankNetwork.

Вот и все! Теперь программа выведет установленные вами параметры. Если они верны, нажмите кнопку **Save/Reboot** для перезагрузки точки доступа с заданными параметрами.

Конечно, это только базовая настройка. Если вы планируете развернуть серьезную беспроводную сеть, настоятельно рекомендую прочитать мою книгу "Беспроводная сеть дома и в офисе"[2].

[2] **http://bhv.ru/books/book.php?id=185666**

Проблемы интерференции

Если вы — единственный владелец беспроводной сети в радиусе 50–70 метров и рядом нет устройств, работающих на частоте 2,4 ГГц (например, беспроводных телефонов, радиоуправляемых игрушек), то можно сказать с уверенностью, что проблемы интерференции вам не страшны. Но, как показывает практика, сейчас в каждом многоэтажном доме есть несколько беспроводных сетей. Следовательно, проблемы интерференции — это более чем реально. *Интерференция* (наложение) радиосигналов приводит к снижению производительности сети. Если наложение сигналов очень сильное, ваша беспроводная сеть вообще не будет работать — радиосигналы, отправленные беспроводными адаптерами, не дойдут до получателя.

Вокруг вас достаточно много источников интерференции. Это и микроволновка, и некоторые беспроводные телефоны, и радиоуправляемые игрушки. Но все эти источники — не проблема. Их легко обнаружить, следовательно, легко устранить проблему, просто выключив устройство-источник интерференции.

Просмотрите список доступных сетей. В нем может оказаться сеть вашего соседа. Посмотрите в свойствах сети номер канала, на котором она работает. Чтобы максимально исключить интерференцию, вы должны перевести свою сеть на канал, номер которого отличается от номера другой сети на 5 единиц. Например, если сеть соседа работает на канале 11, то вам нужно перевести свою сеть на канал 6.

Бывает иная ситуация. Сосед свою сеть скрыл, но она-то работает, и наложение сигналов все равно происходит. В этом случае попробуйте просто установить другой номер канала. Экспериментируйте. По умолчанию многие точки доступа используют или номер 1, или номер 11. Попробуйте установить номер канала 5 или 6. Можно, конечно, обойти ближайших соседей и узнать, используют ли они беспроводную сеть и на каком канале она работает.

Изменить номер канала можно в настройках точки доступа. В случае с D-Link перейдите на вкладку **Advanced** и нажмите кнопку **Wireless**. Номер канала точки доступа задается параметром **Channel**, а параметр **Preamble type** позволяет изменить тип преамбулы: **long** (длинная) и **short** (короткая). Напомню, длинная преамбула более универсальна (ее будут поддерживать как современные, так и "древние" клиенты), но короткая позволяет повысить производительность сети.

Снизить интерференцию можно также путем перемещения вашей точки доступа. Иногда стоит перенести точку доступа на метр в сторону, и результат будет довольно ощутим. Также помогает использование направленных антенн — они позволяют существенно сократить область пересечения сигналов двух сетей.

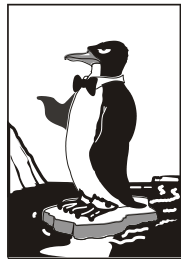
6.9.4. Настройка соединения Wi-Fi в Linux

Эта глава получилась и так достаточно большой, поэтому настройку соединения Wi-Fi в Linux мы рассмотрим в отдельной главе, 13-й. Думаю, так будет правильнее, тем более, что со следующей главы мы как раз и начинаем изучать работу в Linux.

ЧАСТЬ II

Знакомство с Linux

Во второй части книги мы рассмотрим установку Linux, работу в командной строке, файловую систему, а также управление пользователями и группами.



Глава 7

Особенности установки Linux

Установка Linux совсем не похожа на установку привычной многим операционной системы Windows. В этой главе мы поговорим об особенностях установки Linux, которые вы просто обязаны знать до ее начала. Зная эти особенности, установить Linux сможет даже школьник — ведь вся установка проходит в графическом режиме, да еще и на русском языке, что существенно облегчает весь процесс. Забегая вперед (об этом мы еще поговорим), хочу сразу предупредить, что Windows на компьютер нужно устанавливать до Linux, потому что загрузчик Linux без проблем загружает все имеющиеся версии Windows, а вот заставить загрузчик Windows загружать Linux довольно сложно. Поэтому, дабы не усложнять себе жизнь, сначала установите все нужные версии Windows, а затем устанавливайте все необходимые дистрибутивы Linux.

7.1. Системные требования

Современные дистрибутивы Linux не очень требовательны к системным ресурсам, хотя некоторые из них требуют для запуска в графическом режиме программы установки 256 Мбайт оперативной памяти (а некоторые даже больше — см. примечание далее), что, на мой взгляд, уже слишком! Так что, если у вас оперативки меньше (например, вы хотите создать шлюз из пылившегося в углу старенького компьютера), установка будет происходить в текстовом режиме.

> ***Проблема с Fedora 13***
>
> Fedora 13 вообще меня неприятно удивила. Попытался запустить ее установку на компьютере с 256 Мбайт ОЗУ, но инсталлятор запустился только в текстовом режиме. Ради интереса я попытался запустить установку в виртуальной машине с 384 Мбайт ОЗУ (иногда встречаются компьютеры, где установлено два модуля памяти: 256 Мбайт + 128 Мбайт, но найти такой мне было сложно, поэтому пришлось тестировать в VMware). И что вы думаете? Инсталлятор по-прежнему запустился в текстовом режиме. А "графика" пошла, когда было установлено 512 честных мегабайтов. Fedora 12 я на старые компьютеры поставить не пытался, вполне возможно, что такая же "особенность" есть и у двенадцатой версии дистрибутива.
>
> К чести других дистрибутивов нужно отметить, что последние версии, в частности Ubuntu 10.04 и openSUSE 11.3, запускаются в графическом режиме на компьютере с 256 Мбайт ОЗУ.

В части дискового пространства ориентируйтесь минимум на 4−5 Гбайт (это с небольшим запасом — ведь еще нужно оставить место для своих данных), что вполне приемлемо по нынешним меркам, учитывая, что после установки вы получаете

не "голую" систему, а уже практически готовую к работе — с офисными пакетами и программами мультимедиа. Если вы настраиваете сервер, то все офисные и мультимедиапрограммы, понятно, можно не устанавливать. Тогда для самой системы понадобится максимум 2 Гбайт (с графическим интерфейсом и необходимыми пакетами, содержащими программы-серверы), но не нужно забывать, что само слово "сервер" подразумевает достаточное количество дискового пространства. Получается, что потребуется 2 Гбайт — для самой системы и еще сколько-то гигабайт для данных, которые будет обрабатывать сервер.

7.2. Параметры ядра

Прежде чем поговорить о параметрах ядра, нужно вкратце рассмотреть процесс загрузки компьютера. После включения питания запускается процедура самотестирования POST (Power On Self Test), проверяющая основные компоненты системы: видеокарту, оперативную память, жесткие диски и т. д. После POST начинается загрузка операционной системы. Компьютер "ищет" на жестком диске (и других носителях) программу-загрузчик операционной системы. Если такая программа найдена, то ей передается управление, если же такая программа не найдена ни на одном из носителей, то выдается сообщение с просьбой вставить загрузочный диск.

В настоящее время популярны два загрузчика Linux: LILO и GRUB. GRUB — более современный загрузчик и используется по умолчанию в большинстве дистрибутивов. То есть после установки Linux начальным загрузчиком будет именно GRUB, если вы самостоятельно не выберете какой-либо другой. Некоторые дистрибутивы имеют собственные загрузчики — например, ASPLinux использует загрузчик ASPLoader.

Относительно недавно появилась новая версия GRUB — GRUB-PC, или GRUB-2. Особенность этой версии — возможность загружать Linux с раздела файловой системы ext4 и другой, более гибкий, файл конфигурации. Новая версия GRUB также будет рассмотрена далее в этой книге.

Задача загрузчика — предоставить пользователю возможность выбрать нужную операционную систему (ведь кроме Linux может быть установлена и другая операционная система) и передать ей управление. В случае с Linux загрузчик загружает ядро операционной системы и передает управление ему. Все последующие действия по загрузке системы выполняет ядро Linux (монтирует корневую файловую систему, запускает программу инициализации).

Ядро Linux — это святая святых операционной системы Linux. Ядро управляет всем: файловой системой, процессами, распределением памяти, устройствами и т. д. Если программе нужно выполнить какую-либо операцию, она обращается к ядру Linux. Например, если программа хочет прочитать данные из файла, то она сначала открывает файл, используя системный вызов `open()`, а затем читает данные из файла с помощью системного вызова `read()`. Для закрытия файла используется системный вызов `close()`. Конечно, на практике все выглядит сложнее, поскольку Linux — многопользовательская и многозадачная система. Это значит, что с системой могут работать одновременно несколько пользователей, и каждый из пользователей может запустить несколько процессов. Ясно, что программе нужно учиты-

вать "поправку на совместный доступ", то есть во время работы с файлом одного из пользователей программа должна установить блокировку доступа к этому файлу других пользователей. Впрочем, в такие нюансы мы сейчас вникать не будем.

Итак, ядро — это программа, пусть и самая главная программа в Linux. Как и любой другой программе ядру Linux можно передать параметры, влияющие на его работу. Передать параметры ядру Linux можно с помощью любого загрузчика Linux. При установке Linux, особенно если операционная система отказывается устанавливаться с параметрами по умолчанию, полезно передать ядру особые параметры. Например, на некоторых ноутбуках для установки Linux требуется передать ядру параметры `noauto` и `nopcmcia`. Первый параметр запрещает автоматическое определение устройств, а второй — проверку PCMCIA-карт.

Кроме параметров ядра, при установке можно передать параметры программе установки — например, параметр `vga` при установке Linux Mandriva означает, что программа установки должна работать при разрешении 640×480, что позволяет запустить установку на самых "древних" компьютерах или когда видеокарта не полностью совместима с Linux (такое редко, но бывает). Передать параметры программе установки можно так же, как и параметры ядру.

Для редактирования параметров ядра, например, в Fedora 13, нужно выбрать необходимый вариант установки (обычно выбирается первый, предлагающий установить или обновить существующую систему) и нажать клавишу <Tab> (рис. 7.1). После этого можно отредактировать параметры ядра (рис. 7.2).

В Ubuntu 10 для редактирования параметров ядра нужно выбрать необходимый вариант установки и нажать клавишу <F6> (рис. 7.3).

В Mandriva 2010 для ввода параметров ядра нужно нажать <F6>, а потом ввести параметры ядра в поле **Опция ядра** (**Boot options**) (рис. 7.4).

Рис. 7.1. Начальное меню при установке Fedora 13

Рис. 7.2. Редактирование параметров ядра в Fedora 13

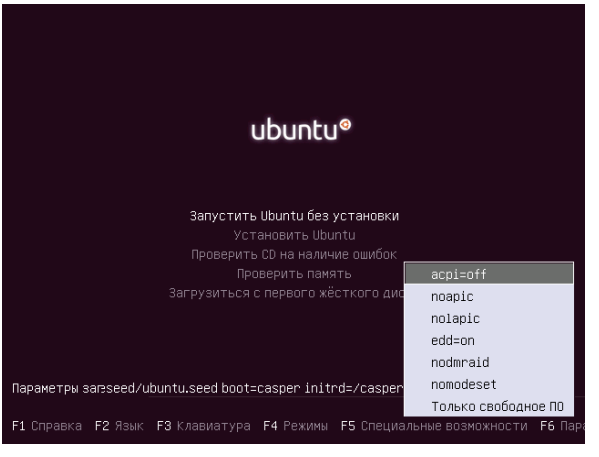


Рис. 7.3. Начальное меню при установке Ubuntu 10

Рис. 7.4. Начальное меню при установке Mandriva 2010.1 Spring

Примечание

Обратите внимание на рис. 7.3 и 7.4: меню загрузчика GRUB русифицированное. Однако сразу после загрузки с DVD меню будет на английском языке. Для выбора языка нужно нажать клавишу <F2> и выбрать русский язык из списка. Такая возможность есть не у всех дистрибутивов. Например, в Fedora выбор языка возможен только после запуска программы установки.

Более подробно о параметрах ядра вы сможете прочитать в *приложении 1*. Некоторые полезные параметры программы установки Fedora представлены в табл. 7.1.

Таблица 7.1. *Дополнительные параметры программы установки Fedora*

Параметр	Описание
`linux noprobe`	Запретить исследование "железа" вашего компьютера. Очень полезно, например, на ноутбуках, когда не хочет правильно определяться та или иная PCMCIA-карта
`linux mediachek`	Проверка носителя перед установкой. Бессмысленный параметр — ведь при установке программа все равно спросит вас, хотите ли вы проверить носитель
`linux rescue`	Запуск режима восстановления Linux
`linux askmethod`	Позволяет выбрать другой метод установки, например, установку по сети
`memtest86`	Запускает программу memtest86 — если у вас есть подозрение на неисправность оперативной памяти, что проявляется в непредсказуемых зависаниях и перезагрузках компьютера. Программа протестирует оперативную память и сообщит о возможных ошибках
`linux resolution=XxY`	Устанавливает разрешение программы установки, например, `linux resolution=1024x768`

В идеальных условиях ни один из этих параметров вводить не нужно — все и так пройдет успешно.

7.3. Проверка носителей

Некоторые дистрибутивы, в частности, Fedora (и прочие, основанные на этом дистрибутиве), предлагают выполнить проверку установочного DVD перед установкой (рис. 7.5).

Если поверхность DVD вызывает у вас сомнения, можно его проверить — зачем тратить время на установку, если на 99-м проценте программа установки сообщит вам, что ей не удается прочитать какой-то очень важный пакет, и система не может быть установлена? Если же DVD новый (только что купленный), можно отказаться от проверки носителя — вы сэкономите немного времени.

Рис. 7.5. Проверка носителя

7.4. Изменение таблицы разделов

Система Linux не может быть установлена в Windows-разделы (FAT32, NTFS). Для ее установки нужно создать Linux-разделы (файловая система ext3 или ext4). Понятно, что для этого на жестком диске должно иметься неразмеченное пространство. Если его нет, то придется или удалить один из Windows-разделов и на его месте создать Linux-раздел, или же уменьшить размер одного из Windows-разделов и создать разделы Linux на освободившемся месте.

Понятно, что удалять разделы не хочется — можно потерять данные. Поэтому обычно уменьшают размер Windows-раздела. Перед началом установки убедитесь, что в каком-либо разделе у вас есть 4–5 Гбайт свободного пространства.

> ***Совет***
>
> Если вы устанавливаете старый дистрибутив Linux, в котором все еще используется загрузчик LILO, то основной раздел Linux должен находиться ближе к началу диска. Дело в том, что Linux может загружаться с разделов, которые начинаются до 1024-го цилиндра (первый блок раздела должен находиться до 1024-го цилиндра). Это не проблема самой операционной системы, а требование данного загрузчика Linux. В некоторых случаях эту проблему удается обойти, а в некоторых — нет. Лучше лишний раз не тратить время зря и создать Linux-раздел так, чтобы он начинался как можно ближе к "началу" диска. После установки Linux сможет использовать (читать и записывать данные) любые разделы вне зависимости от начального номера цилиндра раздела.

Перед установкой Linux следует произвести дефрагментацию того Windows-раздела, который вы собрались уменьшать, чтобы упростить задачу программе установки по переносу ваших файлов.

В любом дистрибутиве программа установки системы Linux умеет автоматически разбивать жесткий диск — она сама создаст Linux-разделы без вашего участия. Например, в Fedora 13 вам доступны следующие варианты разметки диска (рис. 7.6):

- **Все пространство** — будет задействован весь жесткий диск. Используйте этот вариант, если устанавливаете Linux на новый компьютер;

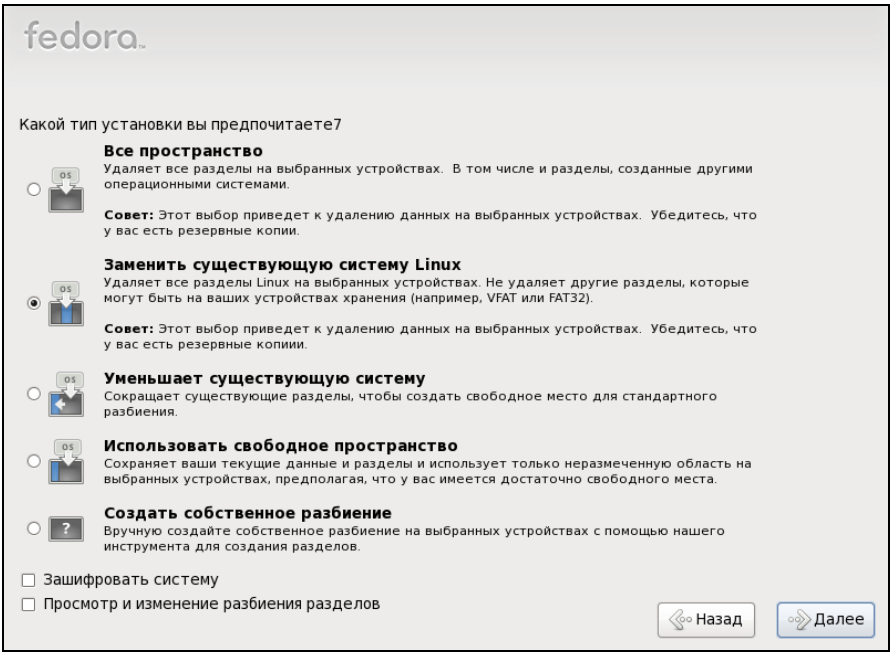


Рис. 7.6. Выбор типа разметки диска в Fedora 13

- **Заменить существующую систему Linux** — если на компьютере уже была установлена операционная система Linux, то выбор этого варианта уничтожит эту систему, а на ее место будет установлена Fedora 13;
- **Уменьшает существующую систему** — существующая система будет сжата и в освободившееся после сжатия пространство начнет устанавливаться Fedora. По своим последствиям этот вариант непредсказуем. На своей системе я его не проверял и вам не советую. Если все-таки спортивный интерес победит здравый смысл, сделайте резервную копию всех важных данных перед выбором этого варианта;
- **Использовать свободное пространство** — инсталлятор начнет устанавливать Linux на свободное (неразмеченное) пространство. Этот вариант я протестировал и обнаружил, что он работает некорректно. Система почему-то пытается использовать неразмеченное пространство, зарезервированное мной для первичного раздела (туда я планировал установить FreeBSD), при этом она совсем не хочет видеть свободное дисковое пространство в расширенном разделе;
- **Создать собственное разбиение** — этот вариант подходит для пользователей, которые понимают, что делают, и которым не все равно, что случится с их данными после установки Linux.

Mandriva предлагает следующие варианты:

- использовать для установки весь жесткий диск — если еще ни одна система не установлена;
- удалить Windows и установить Mandriva взамен;
- вручную разметить разделы, если какая-либо система уже установлена.

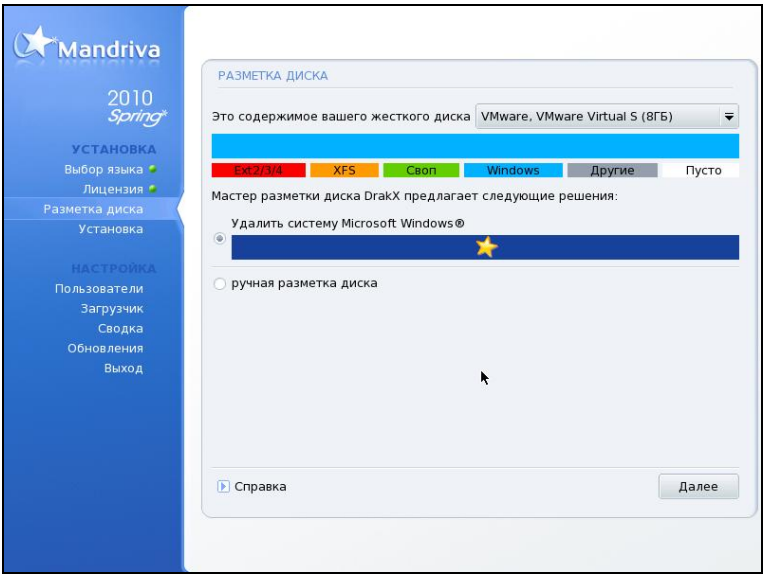


Рис. 7.7. Выбор типа разметки диска в Mandriva 2010.1 Spring

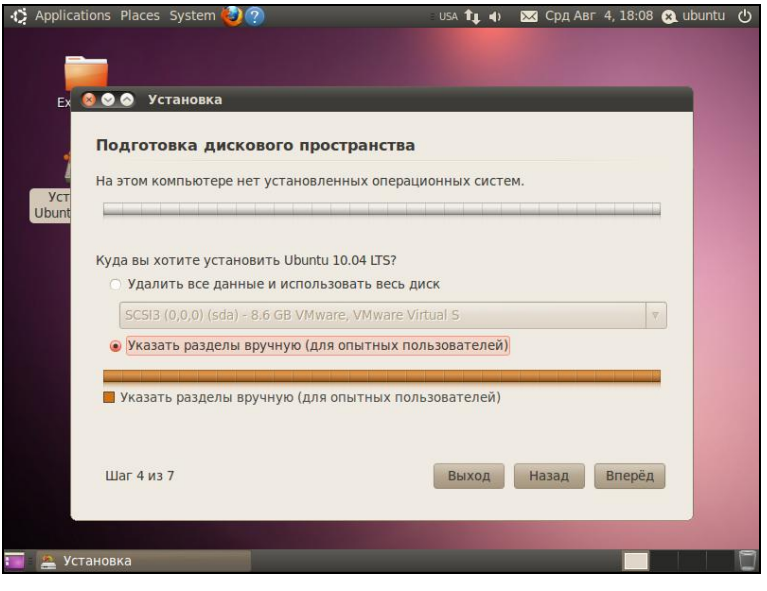


Рис. 7.8. Выбор типа разметки диска в Ubuntu 10

Поэтому если вы хотите сохранить свою Windows, то придется выбрать ручную разметку и собственноручно настроить разделы (рис. 7.7 и 7.8).

Примечание

Лично я люблю контролировать процесс разметки (не без преувеличения скажу, что это один из самых важных процессов), поэтому всегда выбираю ручную разметку.

Итак, если вы выбрали ручную разметку, вам следует изменить размер одного из существующих Windows-разделов и создать два Linux-раздела. Первый — корневой, его точка монтирования обозначается слешем — /. Второй — раздел подкачки (тип swap).

Размер раздела подкачки зависит от объема оперативной памяти:

- если ваш компьютер имеет менее 256 Мбайт оперативной памяти, то можно установить 512 или больше Мбайт для раздела подкачки;
- если объем ОЗУ 256−1024 Мбайт, можно установить размер раздела подкачки в пределах 256−512 Мбайт;
- если ОЗУ 1 Гбайт или более, можно вообще отказаться от раздела подкачки или установить чисто символический размер — 256 Мбайт. Даже если вам и не хватит виртуальной памяти (оперативной + подкачка), вы всегда сможете создать файл подкачки.

Примечание

В Linux можно создать специальный файл подкачки, который тоже будет использоваться в процессе свопинга. Как правило, файл подкачки создается, если размера раздела подкачки оказалось недостаточно, а заново переразбивать жесткий диск (с целью увеличения размера раздела подкачки) не хочется. О создании файла подкачки мы поговорим в *главе 9*.

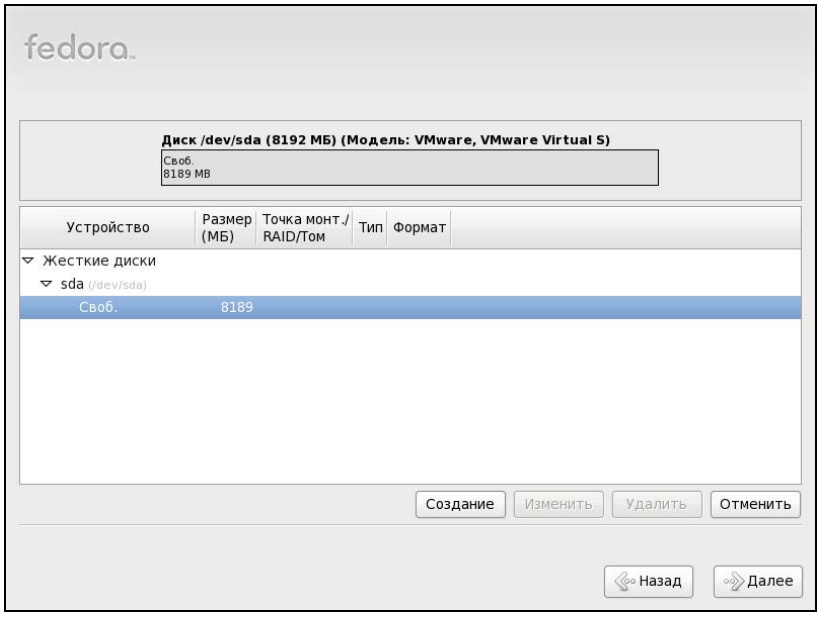


Рис. 7.9. Программа разметки диска в Fedora 13

Рассмотрим для примера работу программы разметки Fedora 13 (рис. 7.9). Выделите свободное пространство и нажмите кнопку **Создание**. Затем надо выбрать тип раздела (стандартный, программный RAID или том LVM), установить параметры

раздела и в открывшемся окне указать новый размер Windows-раздела. Для изменения параметров уже существующих разделов служит кнопка **Изменить**. Как получить свободное пространство, если жесткий диск уже размечен? Правильно, можно изменить его размер.

Примечание

В Mandriva кнопка **Изменить** называется **Изменить размер**.

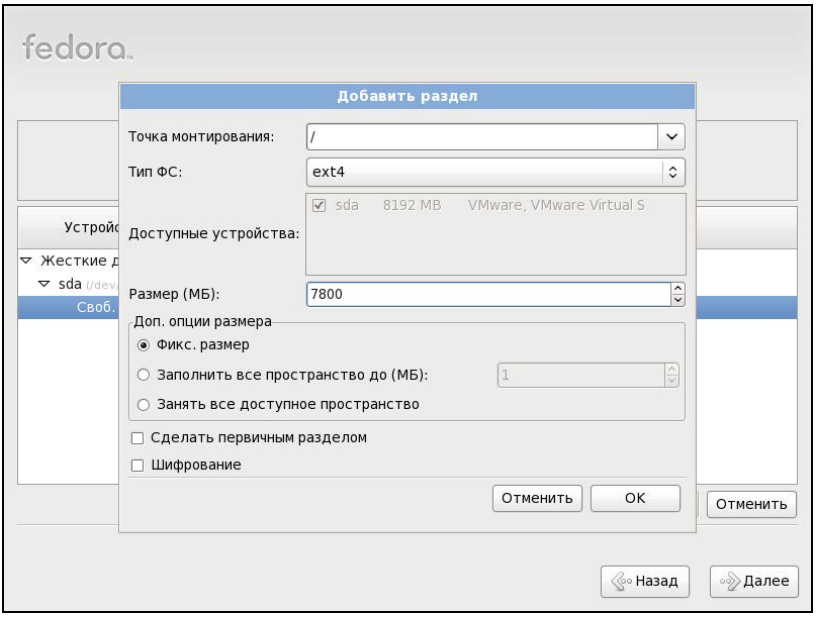


Рис. 7.10. Создание раздела в Fedora 13

После освобождения места вы увидите, что на диаграмме диска появилось свободное (неразмеченное) пространство. Нужно его выделить, нажать кнопку **Создание**, в открывшемся окне (рис. 7.10) определить параметры создаваемого раздела (файловая система в зависимости от дистрибутива выбирается ext3 или ext4) и нажать кнопку **OK** — этим вы создадите Linux-раздел.

Аналогично осуществляется разметка диска и в программе diskdrake из состава дистрибутива Mandriva 2010.1 Spring (рис. 7.11). При этом, хоть инсталлятор Mandriva и не предлагает автоматического решения вопроса размещения Windows и Linux на одном жестком диске, нужно отметить, что сама программа работы с разделами (diskdrake) работает лучше и корректнее, нежели другие программы, в том числе и GParted (GNOME Partition Editor), которая так популярна в Debian и Ubuntu. Поэтому, если вам нужна надежная программа разметки диска, можете смело использовать DVD с Mandriva (в режиме ручной разметки) вместо платных программ вроде Partition Magic. Во всяком случае diskdrake наиболее корректно изменяет размер раздела, что немаловажно.

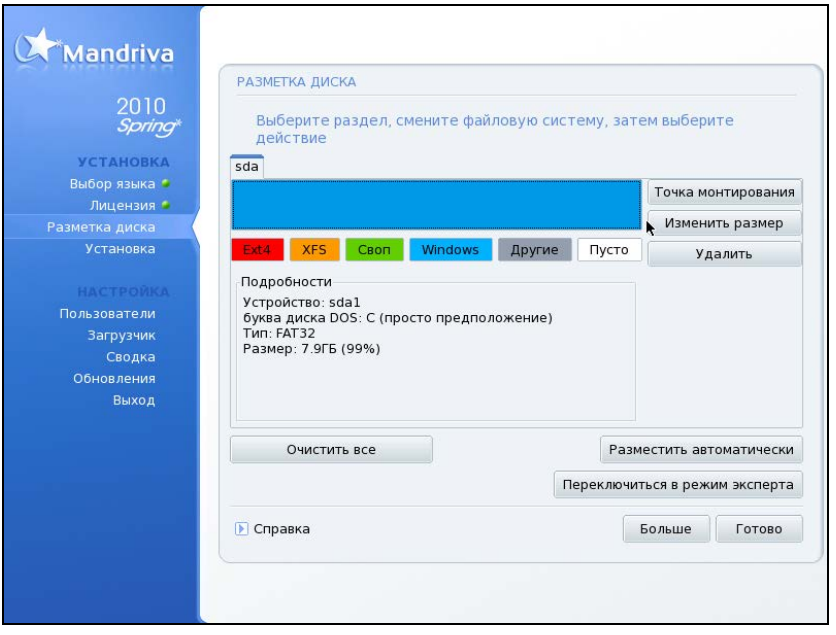


Рис. 7.11. diskdrake — программа разметки диска в Mandriva 2010

Примечание

Практически все современные дистрибутивы поддерживают шифрование файловой системы. При создании раздела вы можете предусмотреть такое шифрование — например, включить параметр **Зашифровать** или **Шифрование** (см. рис. 7.10) — в зависимости от дистрибутива он называется по-разному. Но нужно ли вам это? Если вы агент 007 — бесспорно, это очень полезная опция. А вот во всех остальных ситуациях в случае сбоя системы при попытке восстановления данных опция шифрования создаст только дополнительные проблемы.

7.5. Выбор групп пакетов

Некоторые дистрибутивы, например, Mandriva и Fedora, разрешают пользователю самому выбрать, какие группы пакетов нужно устанавливать, а какие — нет. Другие — например, Ubuntu и его клоны, не имеют такой возможности.

Если в вашем дистрибутиве можно выбирать пакеты самому, главное, о чем нужно заботиться, — это дисковое пространство. У меня как-то раз произошла анекдотическая ситуация: программа установки установила почти все пакеты, а потом лишь сообщила, что не хватает места на диске, и предложила… перезагрузку.

Серьезная недоработка программы установки Fedora (я номер версии даже не указываю — видимо, это "фирменная особенность" дистрибутива) — она не сообщает полный объем выбранных пакетов (рис. 7.12). В других дистрибутивах (openSUSE, Mandriva) с этим проще — вы знаете, сколько доступно места на диске, и видите, какой объем пакетов выбран (рис. 7.13).

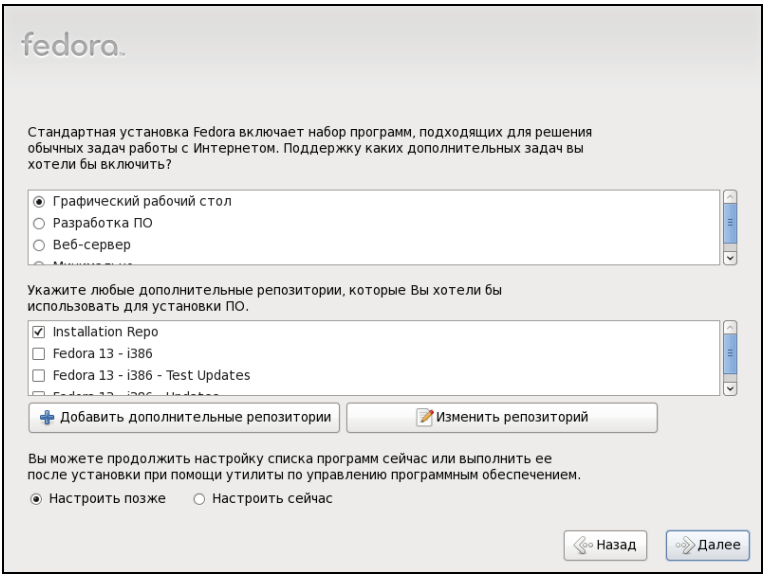


Рис. 7.12. Выбор групп пакетов в Fedora 13:
сколько места на диске займет система, неизвестно

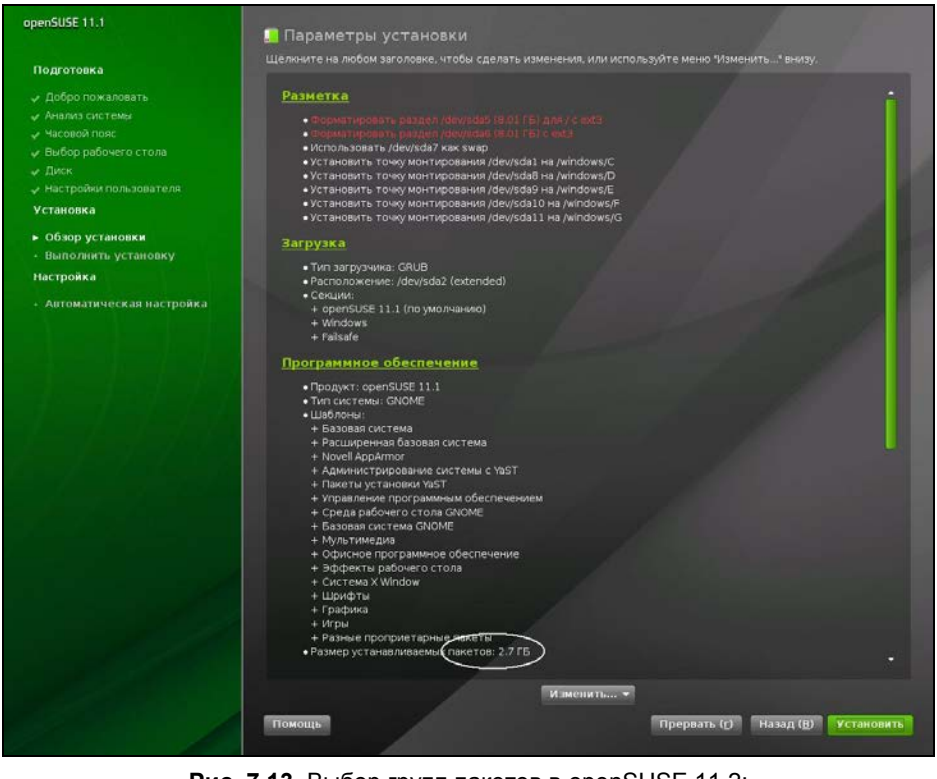


Рис. 7.13. Выбор групп пакетов в openSUSE 11.2:
система займет 2,7 Гбайт дискового пространства

7.6. Выбор графической среды

В Windows мы привыкли к тому, что у нас один-единственный графический интерфейс. Мы можем менять графическую тему, изменять настройки отдельных графических элементов, но чтобы мы ни делали (установку программ вроде Talisman не учитываем — это от лукавого!), графический интерфейс пользователя останется тот же.

В Linux все немного иначе. Имеется графическая подсистема — сервер X (X.Org), который предоставляет фундамент для построения графического интерфейса. А вот построением самого интерфейса пользователя занимаются графические среды, то есть графическая среда и определяет, как будет выглядеть интерфейс пользователя. При установке Linux можно выбрать предпочтительную графическую среду, например: GNOME или KDE — допустимо установить и обе, если имеется достаточно дискового пространства.

Первой графической средой для Linux, способной тягаться по комфорту использования с графическим интерфейсом Windows, стала KDE (1996 год). В 1999 году появилась графическая среда GNOME. С тех пор они конкурируют между собой.

Назначая графическую среду, вы выбираете не только, как станет выглядеть интерфейс пользователя, — вы определяете набор программ, с которыми будете работать. Дело в том, что среда KDE использует библиотеку Qt, а в основе GNOME лежит библиотека GTK. Следовательно, если вы выбрали KDE, то будут установлены программы, которые основаны на этой библиотеке. Если же вы выберете GNOME, то будут установлены приложения, основанные на GTK. Простейший пример: в качестве файлового менеджера при выборе KDE будет установлена программа Dolphin, а если выбрать GNOME, то — Nautilus.

Какую графическую среду предпочесть? Раньше я советовал выбирать KDE, потому что эта графическая среда была лучше русифицирована и более удобна в использовании для бывших Windows-пользователей. Сейчас у GNOME нет никаких проблем с русским языком, и в то же время GNOME так же удобна, как и KDE. Во всяком случае в последнее время я использую GNOME.

Текущая версия KDE — четвертая, KDE3 уже окончательно удалена из состава некоторых дистрибутивов как устаревшая. Впрочем, некоторые дистрибутивы все еще позволяют выбрать версию KDE. Текущая версия GNOME — 2.29, но уже не за горами третья версия — GNOME3.

7.7. Установка пароля root

Пользователь root — это главный пользователь в системе (как Администратор в Windows). Постарайтесь не забыть его пароль (рис. 7.14)!

В некоторых дистрибутивах окно для ввода пароля root совмещено с окном добавления пользователя (например, в Mandriva), некоторые дистрибутивы выводят отдельное окно для задания пароля root (Fedora), а openSUSE предлагает создать обычного пользователя, и при этом его пароль предлагается использовать в качестве пароля root (рис. 7.15). Это довольно удобно, но, с точки зрения безопасности, лучше, чтобы пароль root не совпадал с пользовательским паролем.

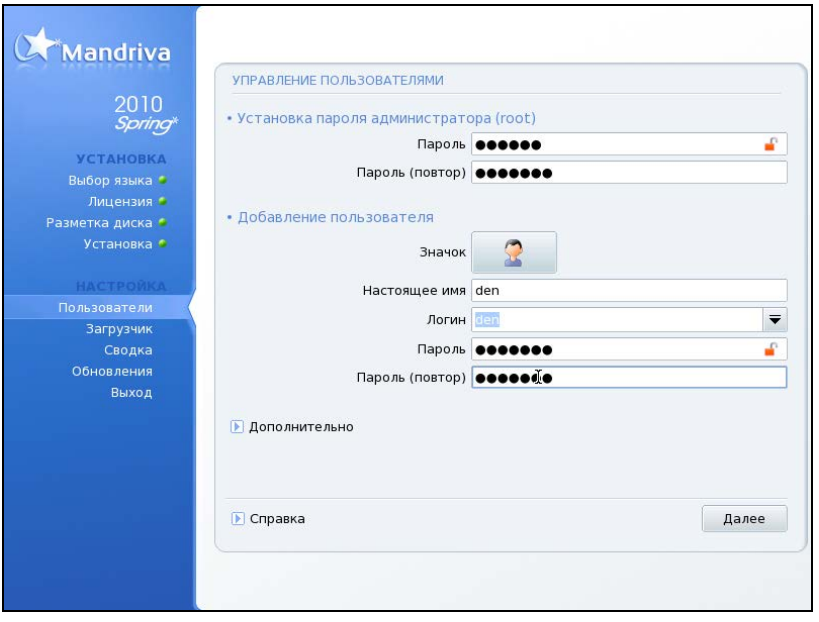


Рис. 7.14. Установка пароля root (Mandriva 2010)

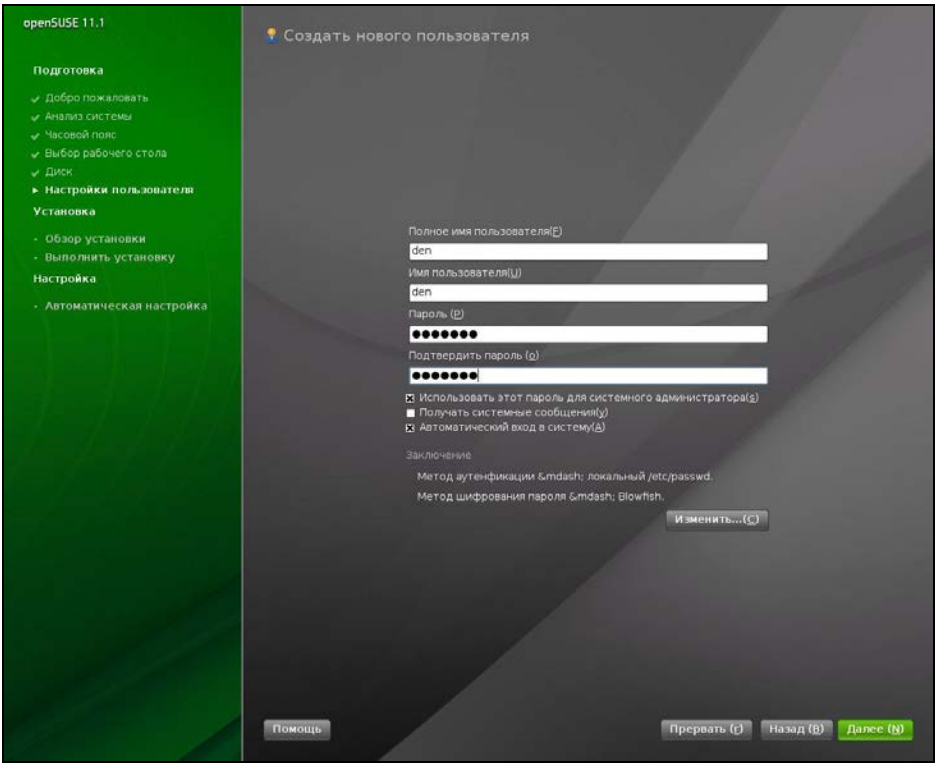


Рис. 7.15. Создание пользователя в openSUSE (при установке системы)

Примечание

При установке Fedora 13 сразу же активируется выбранная в начале установки раскладка клавиатуры (в наших краях — русская), поэтому при вводе имени и пароля пользователя может возникнуть проблема — переключиться на английскую раскладку не получится, потому что ее вообще не существует. Так что либо следует изначально выбрать английскую раскладку, либо вводить пароль цифрами.

7.8. Создание учетных записей пользователей

При установке системы вам нужно создать хотя бы одну пользовательскую учетную запись — ее вы будете использовать для входа в систему. Многие современные дистрибутивы запрещают вход в систему от имени root, поэтому вы будете использовать именно созданную при установке учетную запись пользователя.

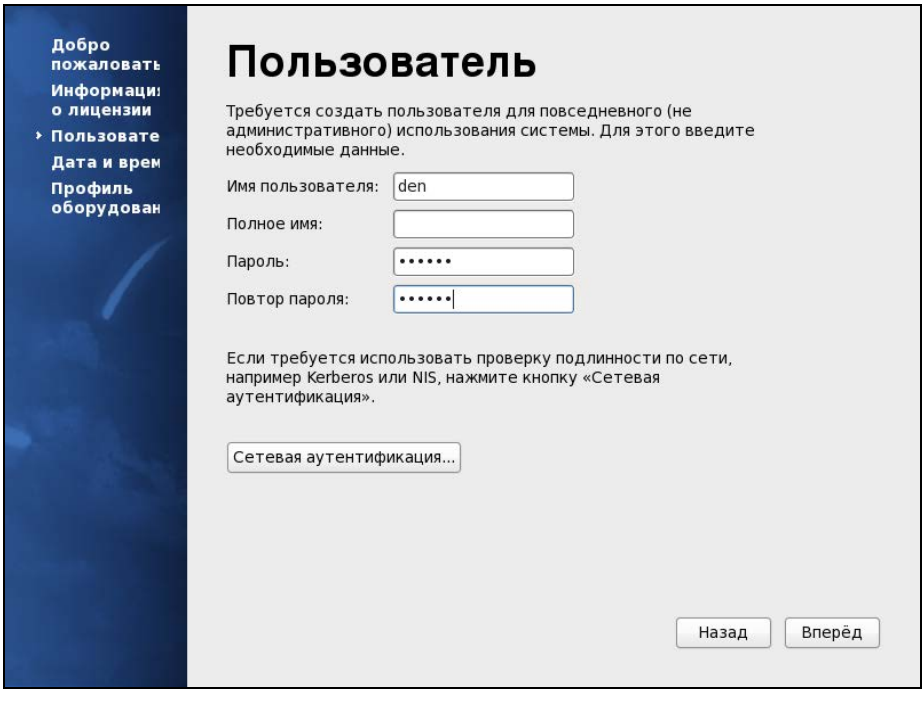


Рис. 7.16. Создание пользователя в Fedora (при первом запуске системы)

Кстати, openSUSE, Mandriva и Ubuntu предлагают создать учетную запись во время установки ОС (см. рис. 7.15), а Fedora — при первом запуске (рис. 7.16).

Примечание

На страничке **http://www.dkws.org.ua/index.php?page=show&file=video-lessons/index** вы найдете несколько полезных видеоуроков, в том числе и урок по установке Fedora 13.

7.9. Установка Linux по сети

7.9.1. Немного о загрузке и установке по сети

Большинство современных компьютеров умеют загружаться по сети. BIOS находит загрузочный PXE-сервер (Preboot Execution Environment) и загружает с него операционную систему. В этом случае компьютеру для загрузки операционной системы не нужен ни жесткий диск, ни любой другой носитель информации. Обычно такая схема используется на "тонких клиентах" — компьютерах, не имеющих жесткого диска (с целью удешевления), загрузка операционной системы на которых производится с центрального компьютера сети.

В этом разделе мы рассмотрим настройку и использование PXE-сервера, предназначенного для загрузки программы установки Linux. Установка по сети может понадобиться в двух случаях:

- **при установке Linux на нетбук (мини-ноутбук), не оснащенный приводом DVD** — покупать USB-привод DVD только для установки Linux не очень хочется, правда? Или старенький ноутбук оснащен только CD-приводом, тогда как большинство современных дистрибутивов распространяются на DVD;
- **при установке Linux на целый парк компьютеров** — тут все просто: компьютеров много, а диск всего один. Установка по сети позволит значительно сэкономить время. В среднем установка Linux (без настройки) занимает полчаса. 10 компьютеров — это уже более 5 часов работы. А вот при наличии загрузочного сервера, на настройку которого у вас уйдет минут 20, эти 10 компьютеров будут готовы к работе всего за 1 час.

Как видите, PXE-сервер — довольно полезная в хозяйстве вещь. В этой книге, правда, мы не будем рассматривать создание полноценного PXE-сервера.

7.9.2. Подготовка загрузочного сервера

Настройку загрузочного сервера рассмотрим на примере дистрибутива Ubuntu. Поскольку установка по сети — довольно специфическая операция, и она нужна далеко не всем пользователям, то не вижу особой необходимости рассматривать установку PXE-сервера в разных дистрибутивах — в другом дистрибутиве можно все сделать "по образу и подобию".

Установка DHCP-сервера

Первым делом нужно установить DHCP-сервер. В Ubuntu это делается командой:

```
$ sudo apt-get install dhcp-server
```

Затем откройте файл /etc/dhcp3/dhcpd/dhcpd.conf и добавьте в него следующие строки:

```
host pxeinstall {
  hardware ethernet xx:xx:xx:xx:xx:xx:xx;
  filename "pxelinux.0";
}
```

Об инструкции hardware следует сказать особо. По большому счету — она не нужна. Но если вы запускаете DHCP-сервер в реальной сети, где уже наверняка есть другой DHCP-сервер, а вам надо установить Linux всего на один компьютер, тогда замените символы x в инструкции hardware MAC-адресом сетевого адаптера, установленного на компьютере, на который нужно поставить Linux.

Если же вы настраиваете всю сеть компьютеров или же полноценный PXE-сервер, тогда можно инструкцию `hardware` удалить — чтобы ваш сервер могли использовать все компьютеры сети.

С другой стороны, указать MAC-адреса потенциальных клиентов — это хорошая идея с точки зрения безопасности. Но если вы разворачиваете свой PXE-сервер только для установки операционной системы, нет никакой необходимости тратить время на вычисления всех MAC-адресов в сети. Это когда надо настроить полноценный PXE-сервер, может, и следует указать адреса "тонких клиентов", чтобы никто другой не смог использовать ваш сервер для загрузки. Тут уже решать вам...

Сохраните файл конфигурации DHCP-сервера и перезапустите сервер:

```
$ sudo /etc/init.d/dhcpd restart
```

Настройка TFTP-сервера

Следующий шаг — настройка TFTP-сервера (Trivial File Transfer Protocol), на котором будет размещен образ операционной системы. В нашем случае — это установочный образ Ubuntu.

Установить TFTP-сервер можно командой:

```
$ sudo apt-get install tftpd-hpa
```

После установки сервера отредактируйте ваш файл /etc/inetd.conf. Убедитесь, что в нем есть следующая строка (и что она раскомментирована):

```
tftp   dgram   udp     wait    root  /usr/sbin/in.tftpd /usr/sbin/in.tftpd
-s /var/lib/tftpboot
```

Поскольку TFTP-сервер работает не автономно, а через inted, то для запуска TFTP-сервера нужно перезапустить inetd:

```
$ sudo /etc/init.d/inetd restart
```

Загрузка установочного образа

Теперь надо загрузить специальный установочный образ, рассчитанный на установку по сети. Подключитесь к Интернету и введите следующие команды:

```
$ mkdir net-install

$ sudo lftp -c "open
http://archive.ubuntu.com/ubuntu/dists/dapper/main/installer-
i386/current/images/; mirror net-install/"
```

Первая команда создаст каталог net-install, а вторая — загрузит в нее установочный образ Ubuntu.

Почти все готово. В каталог net-install будут загружены файлы, необходимые для установки Linux по сети. Но давайте вспомним наш файл /etc/inetd.conf (см. ранее). Конфигурация TFTP предполагает, что все файлы, доступные по протоколу

TFTP, должны быть расположены в каталоге /var/lib/tftpboot. Поэтому следует туда скопировать файлы из каталога net-install:

```
$ sudo cp -a net-install/* /var/lib/tftpboot
$ sudo cd /var/lib/tftpboot
$ sudo tar zxf netboot.tar.gz
```

Вот и все. Ваш PXE-сервер готов к работе.

7.9.3. Настройка клиента

Настраивать клиент, то есть компьютер, на который вы будете устанавливать Linux, очень просто. Достаточно зайти в BIOS и установить загрузку по сети. Но не все компьютеры умеют загружаться по сети.

Что делать, если у вас старый компьютер, который не умеет загружаться по сети? Можно попытаться перепрошить BIOS — новая версия наверняка будет поддерживать загрузку по сети. Если перепрошивать BIOS не хочется или вы не можете найти подходящую версию BIOS именно для вашего компьютера, тогда придется изготовить специальную загрузочную дискету. После этого нужно будет загрузиться с этой дискеты, а загрузчик уже сам найдет PXE-сервер и запустит процесс установки. Создать загрузочную дискету можно с помощью команды `mknbi`, страница руководства по этой команде находится тут: **http://manpages.ubuntu.com/manpages/intrepid/man1/mknbi.html**.

> ***Совет***
>
> Понимаю, что на большинстве современных ноутбуков уже нет дисководов для дискет. Поэтому взамен загрузочной дискеты лучше всего изготовить загрузочную флешку, для создания которой используется программа **Система | Администрирование | Startup disk creator**. С другой стороны, все современные машины поддерживают загрузку по сети, так что вам не стоит особо беспокоиться по этому поводу.

7.10. Проблемы при установке

7.10.1. Проблема с APIC

APIC (Advanced Programmable Interrupt Controller) — улучшенный программируемый контроллер прерываний. Поскольку контроллер прерываний "улучшенный", то проблем быть с ним не должно, но на практике это далеко не так. Одним словом, проблемы с APIC в Linux возникают довольно часто. При загрузке система может зависнуть. Вы можете увидеть сообщение о проблеме с APIC, а можете и не увидеть его. Если сообщение есть, то оно будет выглядеть примерно так:

kernel panic — not syncing: IO-APIC + timer doesn't work! Boot with apic=debug and send areport. Then try booting with the 'noapic' option

Решить проблему помогает параметр ядра `noapic`, позволяющий SMP-ядру не использовать расширенные возможности контроллера прерываний в многопроцессорных машинах. Обратите внимание — ядро само подсказало, чего ему не хватает!

Подробно о передаче параметров ядра мы поговорим в *приложении 1*. А пока, находясь в меню загрузчика GRUB, нажмите клавишу <e> (или <F5> в случае с openSUSE) для редактирования параметров ядра. Просто добавьте в список параметров команду `noapic` — проблема должна исчезнуть. Если данный параметр вам помог, нужно добавить его в файл /boot/grub/menu.lst или отключить APIC в BIOS.

7.10.2. Ошибка: *kernel panic:VFS: Unable to mount root fs*

Появление такого сообщения означает, что ядро не может подмонтировать корневую файловую систему. Понятно, что дальнейшее продолжение работы невозможно. Наиболее вероятная причина — повреждение установочного диска. Если с поверхностью диска все в порядке (она не поцарапана, отсутствуют следы грязи и/или жира), тогда причина в ошибке при записи DVD. Выход один — раздобыть другой DVD и загрузиться с него.

7.10.3. Проблемы с некоторыми LCD-мониторами

Если ваш LCD-монитор подключен к DVI-разъему видеокарты и с ним возникают проблемы (не поддерживается максимальное разрешение, низкое качество изображения, самопроизвольное выключение питания монитора), попробуйте передать ядру параметр `nofb`. Если это поможет решить проблему, "пропишите" данный параметр в конфигурационном файле загрузчика (об этом мы также поговорим далее).

Что делать, если параметр `nofb` не помог? Просто подключите монитор к аналоговому разъему видеокарты — все должно заработать нормально.

7.10.4. Сообщение *Probing EDD* и зависание системы

Некоторые дистрибутивы при загрузке могут вывести сообщение **Probing EDD**, и на этом загрузка остановится. Я столкнулся с этой проблемой при установке openSUSE 11.0 на ноутбук Toshiba. Но, судя по письмам пользователей, такая проблема проявляется и в Fedora 10 и в Mandriva 2009 при использовании определенных жестких дисков.

Если вы увидели это сообщение и система зависла, передайте ядру параметр `edd=off`.

7.10.5. Список известных проблем в Mandriva Linux 2009

Вы можете ознакомиться со списком известных проблем, обнаруженных в Mandriva 2009: возможно, ваша проблема уже решена:

http://wiki.mandriva.com/ru/Список_известных_проблем_в_Mandriva_Linux_2009

Поскольку этот список постоянно обновляется, не вижу смысла приводить его в книге — вы всегда сможете прочитать его обновленную версию в Интернете, да и скоро появится аналогичный список проблем для Mandriva 2010.

7.10.6. Не переключается раскладка в Fedora 13

После входа в свежеустановленную систему вам наверняка захочется запустить терминал и ввести пару команд. Но активна только русская раскладка (или та, которую вы выбрали при установке). Какие бы комбинации клавиш вы ни нажимали (<Ctrl>+<Shift>, <Alt>+<Shift>, <Shift>+<Shift> и др.) — ничего не помогает. Оказывается, в пользовательский профиль устанавливается лишь выбранная раскладка. Чтобы вводить латиницу, следует добавить соответствующую раскладку. Выполните команду меню **Система | Параметры | Клавиатура** и на вкладке **Раскладки** добавьте раскладку **Соединенные штаты/США**.

7.11. Вход в систему и завершение работы

По умолчанию в современных дистрибутивах при входе в систему запускается графический менеджер регистрации (рис. 7.17).

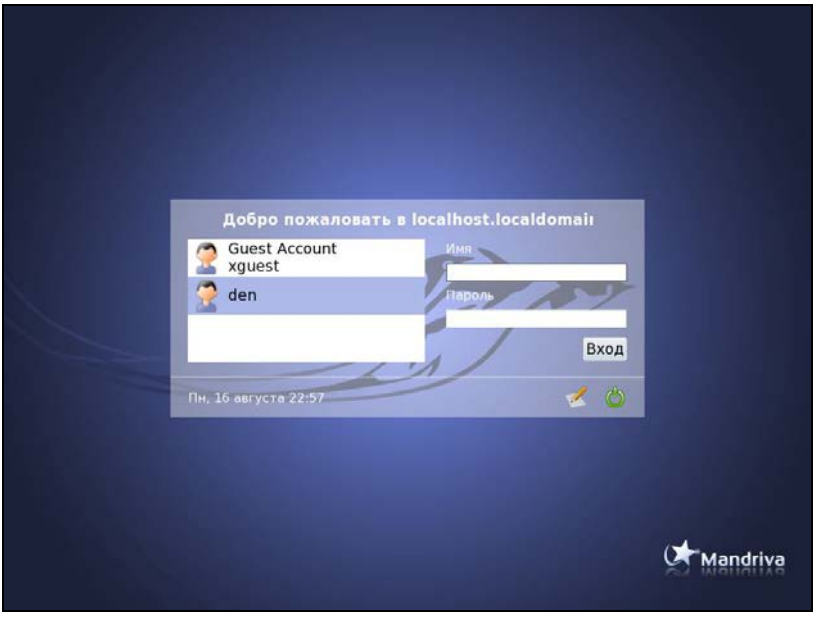


Рис. 7.17. Графический вход в систему (Mandriva 2010)

Примечание

Однако из всех правил могут быть исключения. Пример тому дистрибутив Slackware — в нем сначала нужно выполнить вход в консоли *(см. главу 8)*, а потом для запуска графического интерфейса ввести команду `startx`.

Для входа в систему следует указать имя пользователя и пароль. После этого загрузится KDE или GNOME (в зависимости от того, какая графическая среда уста-

новлена в вашем дистрибутиве по умолчанию). Для выбора другой графической среды надо нажать кнопку **Тип сеанса**, как показано на рис. 7.18.

Примечание

В Fedora и некоторых других дистрибутивах эта кнопка называется **Сеанс**, а в ряде дистрибутивов она может быть представлена графической пиктограммой.

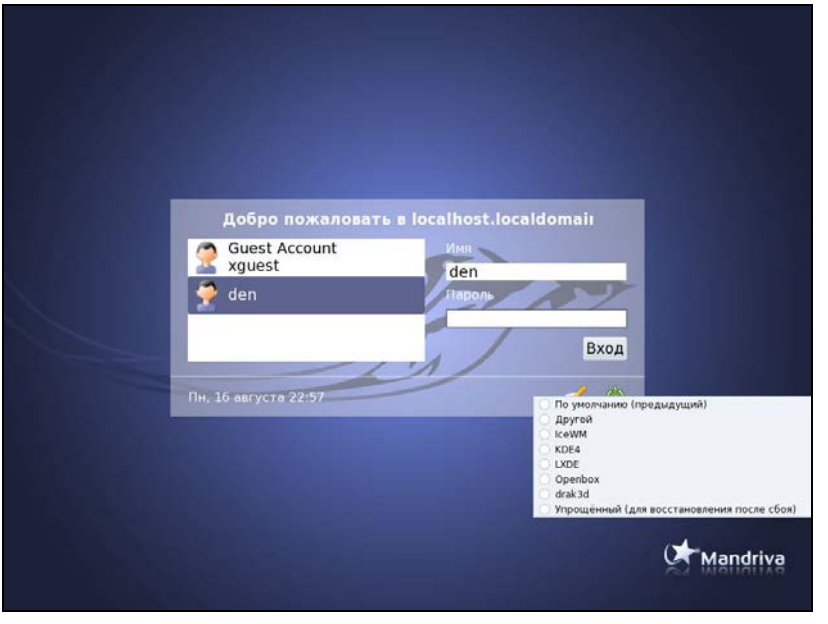
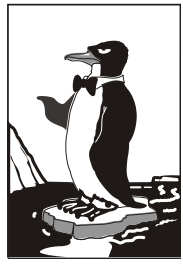


Рис. 7.18. Выбор типа сеанса (Mandriva 2010)



Глава 8

Командная строка Linux

8.1. Консоль

Настоящий линуксоид должен уметь работать в консоли. Ведь когда система Linux только появилась, существовала только консоль, о графическом интерфейсе не было и речи. Знаете, почему UNIX и Linux отталкивали обычных пользователей? Потому что не было хорошего графического интерфейса. Раньше в Linux работали одни профессионалы. Сейчас все изменилось — в Linux очень удобный графический интерфейс, который с удовольствием используют и профессионалы (дождались наконец-то!), забывая о командной строке. Многие дистрибутивы вообще ориентированы на работу в графическом режиме, а в официальных руководствах, которые можно найти в Интернете, о консоли вообще не упоминается. А ведь она есть! В этой главе мы поговорим о том, как правильно работать в консоли. Совсем необязательно работать полностью в текстовом режиме, вы можете использовать материал данной главы для эффективной работы с *терминалом* — эмулятором консоли.

Обычные пользователи в консоль ни ногой — даже принципиально, мол, зачем возвращаться в DOS? Под "DOS" имеется в виду командная строка Linux. Да, ее вид не очень дружелюбен, но это только так кажется на первый взгляд. Стоит вам поработать в консоли, и вы поймете все ее преимущества. Начнем с того, что командная строка Linux намного удобнее командной строки DOS (об этом мы еще поговорим). В консоли можно выполнять те же операции, что и в графическом режиме, причем намного быстрее. Хотите бороздить просторы Интернета? Пожалуйста, но без картинок. Не так красиво, но зато сэкономите трафик. А на обмен электронными сообщениями это никак не повлияет. В консоли можно работать и с документами, правда, о графике придется забыть. Консоль позволяет эффективно использовать ресурсы старых компьютеров — в графическом режиме на стареньком "Пентиуме" не поработаешь, зато в текстовом режиме его можно быстро превратить в очень полезный для всей сети компьютер — в шлюз, через который его более мощные собратья будут получать доступ к Интернету.

8.2. Переход в консоль и обратно

Итак, после входа в систему *(см. главу 7)* вы, как правило, находитесь в графическом режиме. Для того, чтобы перейти из графического режима в консоль (рис. 8.1), нажмите клавиатурную комбинацию <Ctrl>+<Alt>+<F*n*>, где *n* — номер консоли

(от 1 до 6). То есть, чтобы перейти на первую консоль, нужно нажать <Ctrl>+<Alt>+<F1>, на вторую — <Ctrl>+<Alt>+<F2> и т. д. Обратите внимание, что так можно перейти в консоль только из графического режима. Если вы уже находитесь в консоли, то для переключения между консолями служат комбинации клавиш <Alt>+<F1> ... <Alt>+<F6>, а также <Alt>+<F7> — для перехода в графический режим. Для лучшего запоминания эти комбинации клавиш приведены в табл. 8.1.

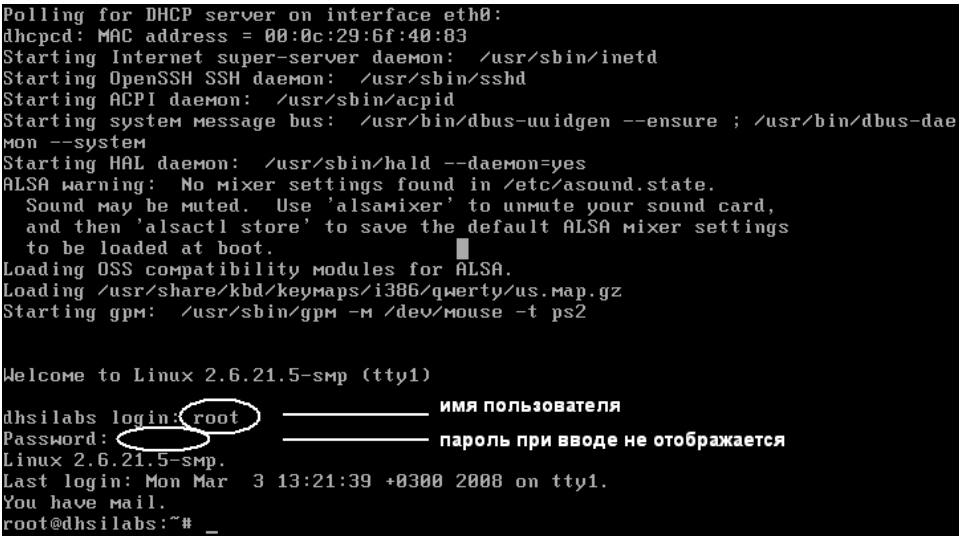


Рис. 8.1. Регистрация в консоли (Slackware)

Таблица 8.1. *Клавиши переключения между консолями и графическим режимом*

Комбинация клавиш	Предназначение
<Ctrl>+<Alt>+<F*n*> (*n* от 1 до 6)	Переключение из графического режима в консоль с номером *n*
<Alt>+<F*n*> (*n* от 1 до 6)	Переключение между консолями
<Alt>+<F7>	Переключение из консоли в графический режим

8.3. Выход из консоли и завершение работы (команды *poweroff*, *halt*, *reboot*, *shutdown*)

Для выхода из консоли (чтобы ею никто не воспользовался во время вашего отсутствия) предусмотрена команда `logout`, она же команда `exit`.

Для перезагрузки компьютера существует команда `reboot`. Кроме нее вы можете использовать еще две команды: `halt` и `poweroff`:

- команда `halt` завершает работу системы, но не выключает питание. Вы увидите сообщение **System is halted**, свидетельствующее о возможности выключения

питания. Эта команда предназначена для старых компьютеров, не поддерживающих расширенное управление питанием;

- команда `poweroff` завершает работу системы и выключает ее питание.

Самая "продвинутая" команда — `shutdown` — позволяет завершить работу и перезагрузить систему в назначенное время. Предположим, что вы хотите уйти пораньше, но компьютер нужно выключить ровно в 19.30 (вдруг некоторые пользователи задержались на работе, а вы выключите сервер, — некрасиво получится). Вот тут-то вам и поможет команда `shutdown`:

```
# shutdown -h 19:30 [сообщение]
```

> ***Примечание***
>
> Здесь и далее решетка (`#`) означает, что команда должна быть выполнена от имени пользователя root. Если перед командой ничего не указано или же указан символ доллара (`$`), команду можно выполнить от имени обычного пользователя.

Сообщение `[сообщение]` можно и не указывать, все равно Windows-пользователи его не увидят.

Если нужно завершить работу системы прямо сейчас, вместо времени укажите `now`:

```
# shutdown -h now
```

Для перезагрузки системы есть опция `-r`:

```
# shutdown -r now
```

8.4. Как работать в консоли

Работа в консоли заключается во вводе нужной команды. Вы вводите команду (например, создания каталога, просмотра файла, вызова редактора и т. д.) и нажимаете клавишу <Enter>. Команда содержит как минимум имя запускаемой программы. Кроме имени программы команда может содержать параметры, которые будут переданы программе, а также символы перенаправления ввода/вывода (об этом чуть позже). Естественно, вам нужно знать имя программы, а также параметры, которые нужно ей передать.

Если вы помните название программы, а назначение параметров забыли, вспомнить поможет команда `man`. Man (от англ. *manual*) — это справочная система Linux. В ней есть информация о каждой программе, которая установлена в вашей системе. Откуда система знает все обо всех программах? Все очень просто. Разработчики программ под Linux договорились, что вместе с программой будет поставляться специальный man-файл — файл справочной системы. Понятно, если разработчик недобросовестный, он может и не создать файл справочной системы, но это происходит очень редко. Чтобы получить справку по какой-нибудь программе, нужно ввести команду:

```
man имя_программы
```

Вы никак не можете запомнить, как пишется та или иная команда? Если вы помните хотя бы, на какую букву она начинается, то воспользуйтесь функцией автодополнения командной строки — введите первые буквы команды и нажмите кла-

вишу <Tab>. При первом нажатии система попытается дополнить команду, если это возможно. Иногда дополнить команду невозможно. Например, вы ввели букву `a` и нажали клавишу <Tab>. Ясное дело, в системе есть несколько команд, которые начинаются на букву "a". Тогда система не дополнит командную строку. Если вы хотите просмотреть все команды на букву "a", тогда нажмите еще раз клавишу <Tab>.

> ***Примечание***
>
> Описанная здесь функция автодополнения работает в командной оболочке bash (которая используется по умолчанию). В следующей главе будут рассмотрены особенности и других оболочек.

Вам лень писать (даже с автодополнением) длинные команды? Тогда можно создать псевдонимы команд. Для этого в файл .bash_profile добавьте строки вида:

```
alias псевдоним='команда'
```

Например:

```
alias cfg-net='system-config-network'
```

Для того чтобы изменения вступили в силу, выйдите из консоли (команда `logout`) и заново зарегистрируйтесь.

Пожалуй, для полноценной работы с консолью вам нужно знать еще одну команду — `clear`. Данная команда очищает консоль (терминал). Очень полезная команда, особенно когда вы хотите все начать с "чистого листа".

8.5. Графические терминалы

Понимаю, что большинство дистрибутивов оснащены графическим интерфейсом, который к тому же запускается по умолчанию. Поэтому большинство пользователей не будут жертвовать удобным и привычным интерфейсом ради консоли.

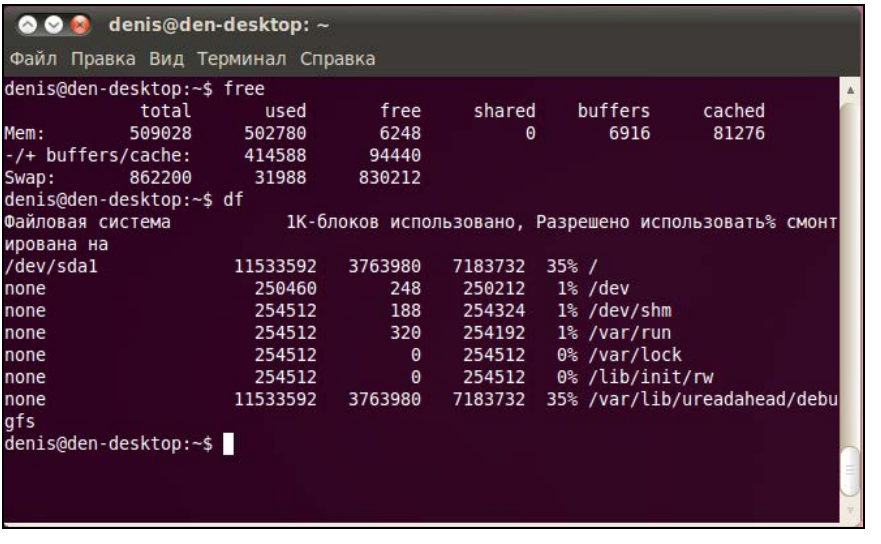


Рис. 8.2. Терминал

Вместо того чтобы переключиться в консоль, можно использовать терминалы — эмуляторы консоли. Терминал — это графическая программа (рис. 8.2), в окне которой вы можете вводить команды и видеть результат их выполнения. Запустить терминал можно через меню GNOME/KDE (**Система | Стандартные | Терминал** или **Система | Системные | Терминал** — в зависимости от дистрибутива).

8.6. Перенаправление ввода/вывода

С помощью перенаправления ввода/вывода мы можем перенаправить вывод одной программы в файл или на стандартный ввод другой программы. Например, у вас не получается настроить сеть, и вы хотите перенаправить вывод команды `ifconfig` в файл, а затем разместить этот файл на форуме, где вам помогут разобраться с этой проблемой. А можно перенаправить список всех процессов (командой `ps -ax`) команде `grep`, которая найдет в списке интересующий вас процесс.

Рассмотрим следующую команду:

```
echo "some text" > file.txt
```

Символ `>` означает, что вывод команды, находящейся слева от этого символа, будет записан в файл, находящийся справа от символа, при этом файл будет перезаписан.

Чуть ранее мы говорили о перенаправлении вывода программы `ifconfig` в файл. Команда будет выглядеть так:

```
ifconfig > ifconfig.txt
```

Если вместо `>` указано `>>`, то исходный файл не будет перезаписан, а вывод команды добавится в конец файла:

```
echo "some text" > file.txt
echo "more text" >> file.txt
cat file.txt
some text
more text
```

Кроме символов `>` и `>>` для перенаправления ввода/вывода часто употребляется вертикальная черта `|`. Предположим, что мы хотим вывести содержимое файла big_text:

```
cat big_text
```

Но в файле big_text много строк, поэтому мы ничего не успеем прочитать. Следовательно, целесообразно отправить вывод команды `cat` какой-то программе, которая будет выводить файл постранично, например,

```
cat big_text | more
```

Конечно, этот пример не очень убедительный, потому что для постраничного вывода гораздо удобнее команда `less`:

```
less big_text
```

Вот еще один интересный пример. Допустим, мы хотим удалить файл file.txt без запроса — для этого можно указать команду:

```
echo y | rm file.txt
```

Команда `rm` запросит подтверждение удаления (нужно нажать клавишу <Y>), но за нас это сделает команда `echo`.

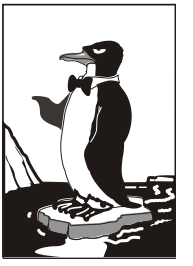
И еще один пример. Пусть имеется большой файл, и нам нужно найти в нем все строки, содержащие подстроку 555-555. Чтобы не делать это вручную, можно воспользоваться командой:

```
cat file.txt | grep "555-555"
```

8.7. Команды Linux

В Linux очень много команд. Интересующимся рекомендую свою книгу "Руководство по командам и shell-программированию в Linux"[1]. Краткое руководство по полезным командам вы найдете в *приложении 3*.

[1] Колисниченко Д. Н. Руководство по командам и shell-программированию в Linux. — СПб.: БХВ-Петербург, 2011, ISBN 978-5-9775-0619-9.



Глава 9

Файловая система

9.1. Файловые системы, поддерживаемые Linux

Linux поддерживает много различных файловых систем. Начинающий пользователь просто теряется, когда видит такое многообразие выбора, — ведь в качестве корневой файловой системы доступны: ext2, ext3, ext4, XFS, ReiserFS, JFS.

Ранее "родной" файловой системой Linux была ext2 (файловая система ext использовалась разве что в самых первых версиях Linux), затем ей на смену пришла *журналируемая* версия файловой системы — ext3. Сегодня все современные дистрибутивы по умолчанию используют следующее поколение файловой системы — ext4. В этой главе помимо всего прочего вы найдете сравнение последних версий файловых систем Linux: ext3 и ext4.

> ***Примечание***
>
> Linux до сих пор поддерживает файловую систему ext, но она считается устаревшей, и рекомендуется воздержаться от ее использования.

Итак, в качестве корневой файловой системы и файловой системы других Linux-разделов могут быть использованы файловые системы ext3, ext4, XFS, ReiserFS, JFS. Все перечисленные файловые системы (кроме ext2) ведут журналы своей работы, что позволяет восстановить данные в случае сбоя. Осуществляется это следующим образом — перед тем как выполнить операцию, *журналируемая* файловая система записывает эту операцию в журнал, а после выполнения операции удаляет запись из журнала. Представим, что после занесения операции в журнал произошел сбой (например, выключили электропитание). Позже, когда сбой будет устранен, файловая система по журналу выполнит все действия, которые в него занесены. Конечно, и это не всегда позволяет уберечься от последствий сбоя — стопроцентной гарантии никто не дает, но все же такая схема работы лучше, чем вообще ничего.

Файловые системы ext2 и ext3 совместимы. По сути, ext3 — та же ext2, только с журналом. Раздел ext3 могут читать программы (например, Total Commander и Ext2Fsd в Windows), рассчитанные на ext2. В свою очередь, ext4 — это усовершенствованная версия ext3.

> ***Примечание***
>
> Особенности новой файловой системы ext4 рассмотрены в *разд. 9.5*.

В современных дистрибутивах по умолчанию уже используется файловая система ext4 (хотя в некоторых дистрибутивах все еще может применяться ext3). При необходимости можно выбрать другие файловые системы. Далее мы рассмотрим их особенности, чтобы понять, нужно ли их использовать или же остановить свой выбор на стандартной ext3/ext4.

- *Файловая система XFS* была разработана компанией Silicon Graphics в 2001 году. Основная особенность данной системы — высокая производительность (до 7 Гбайт/с). XFS может работать с блоками размером от 512 байтов до 64 Кбайт. Ясно, что если у вас много маленьких файлов, то в целях экономии места можно установить самый маленький размер блока. А если вы работаете с файлами большого размера (например, мультимедиа), то нужно выбрать самый большой размер блока — так файловая система обеспечит максимальную производительность (конечно, если "железо" позволяет). Учитывая высокую производительность этой файловой системы, ее нет смысла устанавливать на домашнем компьютере, поскольку все ее преимущества будут сведены на нет. А вот если вы будете работать с файлами очень большого размера, XFS проявит себя с лучшей стороны.
- *Файловая система ReiserFS* считается самой экономной, поскольку позволяет хранить несколько файлов в одном блоке (другие файловые системы могут хранить в одном блоке только один файл или одну его часть). Например, если размер блока равен 4 Кбайт, а файл занимает всего 512 байт (а таких файлов очень много в разных каталогах), то 3,5 Кбайт этого блока просто не будут использоваться. А вот ReiserFS позволяет задействовать буквально каждый байт жесткого диска!

  Но у этой файловой системы есть два больших недостатка: она неустойчива к сбоям, и ее производительность сильно снижается при фрагментации. Поэтому, если вы выбираете данную файловую систему, покупайте UPS (источник бесперебойного питания) и почаще дефрагментируйте жесткий диск.
- *Файловая система JFS* (разработка IBM) сначала появилась в операционной системе AIX, а потом была модифицирована под Linux. Основные достоинства этой файловой системы — надежность и высокая производительность (выше, чем у XFS). Но у нее маленький размер блока (от 512 байтов до 4 Кбайт). Следовательно, она хороша на сервере баз данных, но не при работе с данными мультимедиа, поскольку блок в 4 Кбайт для работы, например, с видео в реальном времени будет маловат.

9.1.1. Выбор файловой системы

Учитывая производительность рассматриваемых файловых систем, можно дать следующие рекомендации:

- для рабочей станции и сервера общего назначения оптимальными файловыми системами являются ext3/ext4 или ReiserFS (в крайнем случае);
- на сервере баз данных можно использовать JFS — в этом случае (особенно, если база данных огромная) будет наблюдаться определенный прирост производительности;
- файловая система XFS — это удел станции мультимедиа, на обычной рабочей станции или обычном сервере ее использовать не следует.

Но производительность — это не единственный критерий выбора файловой системы, особенно для сервера. Да, производительность учитывать нужно, но кроме того нельзя пренебрегать и следующими факторами:

- **надежностью** — все-таки мы выбираем файловую систему для сервера, а не для домашнего компьютера;
- **наличием программ для восстановления файловой системы в случае сбоя** — сбой может произойти даже в случае использования самой надежной файловой системы, поэтому наличие программного комплекса для восстановления файловой системы не будет лишним;
- **максимальным размером файла** — сервер обрабатывает огромные объемы информации, поэтому данный критерий для нас также важен.

Файловые системы ext3/ext4, ReiserFS и XFS одинаково надежны, а вот надежность JFS иногда оставляет желать лучшего. Учитывая это, а также то, что программы для восстановления файловой системы имеются только для ext*, на сервере лучше использовать все-таки ext3/ext4.

Если вы уже интересовались характеристиками файловых систем, то могли в некоторых источниках встретить неправильную информацию о максимальном размере файла для файловой системы ext3. Так, иногда сообщается, что максимальный размер файла для ext3 равен 2 Гбайт, что делает ее непригодной для использования на сервере. Это не так. Раньше, во времена ext2 и ядер 2.2 и 2.4, действительно, существовало такое ограничение, но для ext2. Файловая система ext3 поддерживает файлы размером до 1 Тбайт, а максимальный размер тома (раздела) равен 4 Тбайт, что вполне достаточно даже для сервера. Если же вам нужна поддержка больших объемов данных, то рекомендую обратить внимание на другие файловые системы, например на ReiserFS (максимальный размер файла — 16 Тбайт) или на XFS/JFS (размер файла вообще исчисляется в *петабайтах*).

9.1.2. Linux и файловые системы Windows

Linux почти безо всяких ограничений поддерживает файловые системы FAT12 (DOS), FAT16 (или просто FAT, как в Windows 95) и FAT32 (Windows 98 и все последующие версии). Вы можете из Linux читать в файловых системах Windows файлы и каталоги, изменять, создавать новые файлы и каталоги, удалять их — в общем все, что можно делать в файловой системе непосредственно в Windows.

Однако файловые системы Windows не поддерживают установку прав доступа, поэтому можно даже не пытаться установить в Linux права доступа к файлу, который находится на Windows-разделе, — у вас ничего не получится.

О файловой системе NTFS — отдельный разговор. По умолчанию (без перекомпиляции ядра) Linux умеет только читать данные, расположенные в NTFS-разделе. Однако даже после перекомпиляции ядра останется ряд ограничений на запись в NTFS-раздел: например, вы не можете создавать новые файлы, а можете только редактировать уже имеющиеся. Кстати, поддержка NTFS современным ядром до сих пор экспериментальна, то есть в один не совсем прекрасный момент при попытке записи вы можете потерять данные в вашем NTFS-разделе.

Я вас напугал? Существуют решения (мы их рассмотрим в этой книге), позволяющие снять бóльшую часть ограничений на запись в NTFS-разделы. Конечно, все эти решения не идеальные: что-то работает, но ужасно медленно, что-то снимает далеко не все ограничения на запись, но, тем не менее, все же есть возможность записывать данные в NTFS-раздел без потери данных.

9.1.3. Сменные носители

Linux превосходно работает со сменными CD/DVD- и USB-дисками, в большинстве случаев даже выполняется автоматическое монтирование и размонтирование сменных носителей (хотя эта функция доступна не во всех дистрибутивах). С другой стороны, автоматическое монтирование сменных носителей на сервере — это от лукавого, на домашнем компьютере — да, но не на сервере. О монтировании, в том числе автоматическом, мы поговорим чуть позже в этой главе.

9.2. Особенности файловой системы Linux

9.2.1. Имена файлов

По сравнению с Windows в Linux несколько другие правила построения имен файлов, вам придется с этим смириться. Начнем с того, что в Linux нет такого понятия, как расширение имени файла. В Windows, например, для файла Document1.doc именем файла является фрагмент Document1, а doc — это расширение. В Linux же Document1.doc — это имя файла, и никакого расширения нет.

Максимальная длина имени файла — 254 символа. Имя может содержать любые символы (в том числе и кириллицу), кроме / \ ? < > * " |. Но кириллицу в именах файлов я бы не рекомендовал вообще. Впрочем, если вы уверены, что не будете эти файлы передавать Windows-пользователям (на дискете, по электронной почте), — используйте на здоровье. А при обмене файлами по электронной почте имя файла лучше писать латиницей — кодировка-то у всех разная, и вместо русскоязычного имени получатель увидит абракадабру.

Также вам придется привыкнуть к тому, что система Linux чувствительна к регистру в имени файла: FILE.txt и FiLe.Txt — это два разных файла.

Разделение элементов пути осуществляется символом / (прямой слэш), а не \ (обратный слэш), как в Windows.

9.2.2. Файлы и устройства

Пользователи Windows привыкли к тому, что файл — это именованная область данных на диске. Отчасти так оно и есть. Отчасти — потому, что приведенное определение файла было верно для DOS (Disk Operating System) и Windows.

В Linux же понятие файла значительно шире. Сейчас Windows-пользователи будут очень удивлены: в Linux есть файлы устройств, позволяющие обращаться с устройством, как с обычным файлом. Файлы устройств находятся в каталоге /dev (от *devices*). Да, через файл устройства мы можем обратиться к устройству! Если вы работали в DOS, то, наверное, помните, что что-то подобное было и там — сущест-

вовали зарезервированные имена файлов: PRN (принтер), CON (клавиатура при вводе, дисплей при выводе), LPT*n* (параллельный порт, *n* — номер порта), COM*n* (последовательный порт).

Примечание

Кому-то может показаться, что разработчики Linux "украли" идею специальных файлов у Microsoft — ведь Linux появилась в начале 90-х, а DOS — в начале 80-х годов прошлого века. На самом деле это не так. Наоборот, Microsoft позаимствовала идею файлов устройств из операционной системы UNIX, которая была создана еще до появления DOS. Однако сейчас не время говорить об истории развития операционных систем, поэтому лучше вернемся к файлам устройств.

Вот самые распространенные примеры файлов устройств:

- /dev/sd*x* — файл жесткого диска или USB-накопителя;
- /dev/sd*xN* — файл устройства раздела на жестком диске, *N* — это номер раздела;
- /dev/mouse — файл устройства мыши;
- /dev/modem — файл устройства модема (на самом деле является ссылкой на файл устройства ttyS*n*);
- /dev/ttyS*n* — файл последовательного порта, *n* — номер порта (ttyS0 соответствует COM1, ttyS1 — COM2 и т. д.).

В свою очередь, файлы устройств бывают двух типов: блочные и символьные. Обмен информации с блочными устройствами, например с жестким диском, осуществляется блоками информации, а с символьными — отдельными символами. Пример символьного устройства — последовательный порт.

9.2.3. Корневая файловая система и монтирование

Наверняка на вашем компьютере установлена система Windows. Выполните команду **Пуск | Компьютер** (рис. 9.1).

Скорее всего, вы увидите пиктограмму гибкого диска (имя устройства — A:), пиктограммы разделов жесткого диска (пусть будет три раздела — C:, D: и E:), пиктограмму привода CD/DVD (F:).

Пояснение

На рис. 9.1 диск A: отсутствует — мой ноутбук не оснащен дисководом для гибких дисков.

Таким способом, с помощью буквенных обозначений A:, C:, D: и т. д. в Windows обозначаются корневые каталоги разделов жесткого диска и сменных носителей.

В Linux существует понятие *корневой файловой системы*. Допустим, вы установили Linux в раздел с именем /dev/sda3. В этом разделе и будет развернута корневая файловая система вашей Linux-системы. Корневой каталог обозначается прямым слэшем (/), то есть для перехода в корневой каталог в терминале (или консоли) нужно ввести команду `cd /`.

Понятно, что на вашем жестком диске есть еще разделы. Чтобы получить доступ к этим разделам, вам нужно *подмонтировать* их к корневой файловой системе. После монтирования вы можете обратиться к содержимому разделов через точку

монтирования — назначенный вами при монтировании специальный каталог, например /mnt/cdrom. Монтированию файловых систем посвящен *разд. 9.3*, поэтому сейчас не будем говорить об этом процессе подробно.

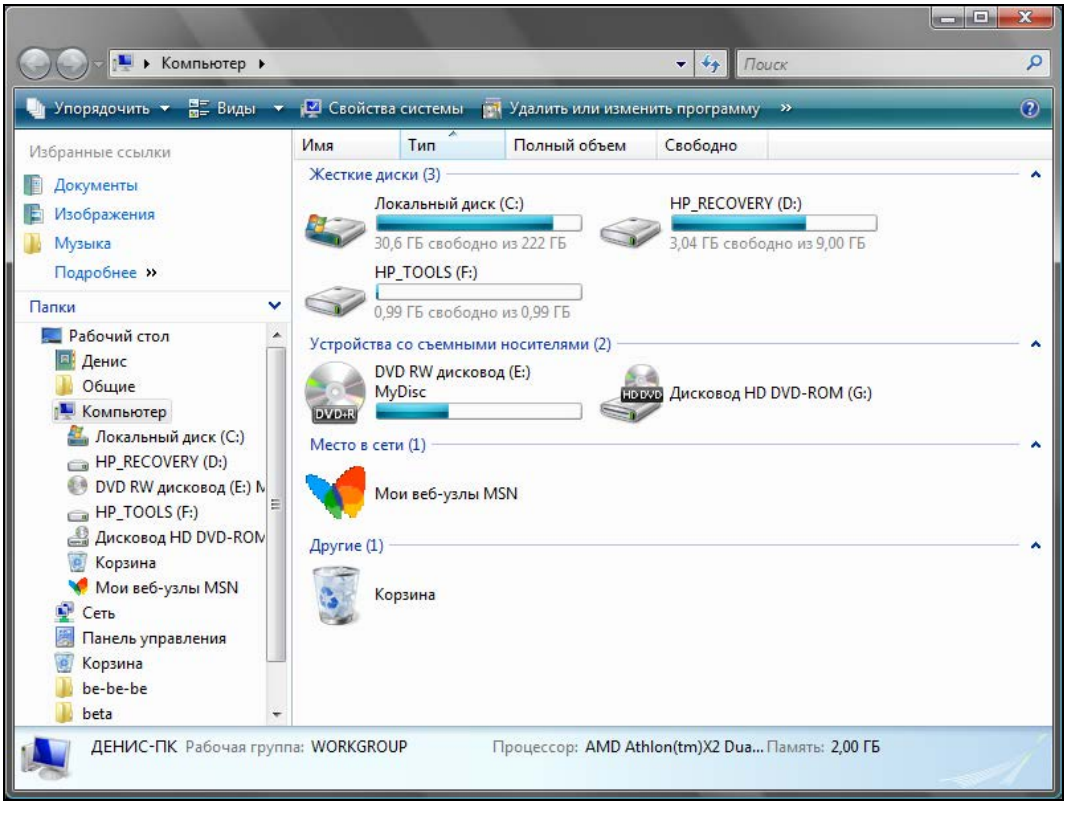


Рис. 9.1. Окно **Компьютер**

9.2.4. Стандартные каталоги Linux

Файловая система любого дистрибутива Linux содержит следующие каталоги:

- / — корневой каталог;
- /bin — содержит стандартные программы Linux (`cat`, `cp`, `ls`, `login` и т. д.);
- /boot — каталог загрузчика, содержит образы ядра и Initrd (RAM-диска инициализации), может содержать конфигурационные и вспомогательные файлы загрузчика;
- /dev — содержит файлы устройств;
- /etc — содержит конфигурационные файлы системы;
- /home — содержит домашние каталоги пользователей;
- /lib — библиотеки и модули;

- /lost+found — восстановленные после некорректного размонтирования файловой системы файлы и каталоги;
- /media — в современных дистрибутивах содержит точки монтирования сменных носителей (CD-, DVD-, USB-накопителей);
- /misc — может содержать все, что угодно, равно как и каталог /opt;
- /mnt — обычно содержит точки монтирования;
- /proc — каталог псевдофайловой системы procfs, предоставляющей информацию о процессах;
- /root — каталог суперпользователя root;
- /sbin — каталог системных утилит, выполнять которые имеет право пользователь root;
- /tmp — каталог для временных файлов;
- /usr — содержит пользовательские программы, документацию, исходные коды программ и ядра;
- /var — постоянно изменяющиеся данные системы, например очереди системы печати, почтовые ящики, протоколы, замки и т. д.

В зависимости от дистрибутива, в корневом каталоге могут находиться дополнительные каталоги (либо, наоборот, отсутствовать некоторые каталоги, приведенные в списке). Например, в Debian и Ubuntu обязательно присутствует каталог /opt (ранее он служил для установки альтернативного программного обеспечения, сейчас его можно использовать для чего угодно), кроме того в Ubuntu имеется каталог /cdrom (который по непонятным мне причинам не задействован).

Внимание!

Описание команд Linux для работы с файлами и каталогами приведено в *приложении 3 (разд. П3.2.1 и П3.2.2)*.

9.2.5. Ссылки: жесткие и символические

В Linux допускается, чтобы один и тот же файл существовал в системе под разными именами. Для этого используются ссылки. Ссылки бывают двух типов: жесткие и символические. Жесткие ссылки жестко привязываются к файлу — вы не можете удалить файл, пока на него указывает хотя бы одна жесткая ссылка. А вот если на файл указывают символические ссылки, его удалению ничто не помешает.

Жесткие ссылки не могут указывать на файл, который находится за пределами файловой системы. Предположим, у вас два Linux-раздела: один корневой, а второй используется для домашних файлов пользователей и монтируется к каталогу /home корневой файловой системы. Так вот, вы не можете создать в корневой файловой системе ссылку, которая ссылается на файл в файловой системе, подмонтированной к каталогу /home. Это очень важная особенность жестких ссылок. Если нужно создать ссылку на файл, который находится за пределами файловой системы, следует использовать символические ссылки.

Внимание!

Описание команды `ln` для работы со ссылками приведено в *приложении 3 (разд. П3.2.3)*.

9.2.6. Задание прав доступа к файлам и каталогам

Для каждого каталога и файла вы можете задать права доступа. Точнее, права доступа автоматически задаются при создании каталога/файла, а вам при необходимости можно их изменить. Какая может быть необходимость? Например, вам нужно, чтобы к вашему файлу-отчету смогли получить доступ пользователи — члены вашей группы. Или вы создали обычный текстовый файл, содержащий инструкции командного интерпретатора. Чтобы этот файл стал сценарием, вам нужно установить право на выполнение для этого файла.

Существуют три права доступа: чтение (`r`), запись (`w`), выполнение (`x`). Для каталога право на выполнение означает право на просмотр содержимого каталога.

Вы можете установить разные права доступа для владельца (то есть для себя), для группы владельца (то есть для всех пользователей, входящих в одну с владельцем группу) и для прочих пользователей. Пользователь root может получить доступ к любому файлу или каталогу вне зависимости от прав, которые вы установили.

Чтобы просмотреть текущие права доступа, введите команду:

```
ls -l <имя файла/каталога>
```

Например,

```
ls -l video.txt
```

В ответ программа выведет следующую строку:

```
-r--r----- 1 den group 300 Apr 11 11:11 video.txt
```

В этой строке фрагмент `-r--r-----` описывает права доступа:

- первый символ — это признак каталога. Сейчас перед нами файл. Если бы перед нами был каталог, то первый символ был бы символом `d` (от *directory*);
- последующие три символа (`r--`) определяют *права доступа владельца файла или каталога*. Первый символ — это чтение, второй — запись, третий — выполнение. Как можно видеть, владельцу разрешено только чтение этого файла, запись и выполнение запрещены, поскольку в правах доступа режимы `w` и `x` не определены;
- следующие три символа (`r--`) задают *права доступа для членов группы владельца*. Права такие же, как и у владельца: можно читать файл, но нельзя изменять или запускать;
- последние три символа (`---`) задают *права доступа для прочих пользователей*. Прочие пользователи не имеют права ни читать, ни изменять, ни выполнять файл. При попытке получить доступ к файлу они увидят сообщение **Access denied**.

Примечание

После прав доступа команда `ls` выводит имя владельца файла, имя группы владельца, размер файла, дату и время создания, а также имя файла.

Внимание!

Описание команды `chmod` для задания прав доступа к файлам и каталогам приведено в *приложении 3 (разд. П3.2.4)*.

9.2.7. Специальные права доступа (SUID и SGID)

Мы рассмотрели обычные права доступа к файлам, но в Linux есть еще так называемые *специальные права доступа*: SUID (Set User ID root) и SGID (Set Group ID root).

Данные права доступа позволяют обычным пользователям запускать программы, требующие для своего запуска привилегий пользователя root. Например, демон pppd требует привилегий root, но чтобы каждый раз при установке PPP-соединения (модемное, ADSL-соединение) не входить в систему под именем root, достаточно установить специальные права доступа для демона pppd. Делается это так:

```
chmod u+s /usr/sbin/pppd
```

Однако не нужно увлекаться такими решениями, поскольку каждая программа, для которой установлен бит SUID, является потенциальной "дырой" в безопасности вашей системы. Для выполнения программ, требующих прав root, намного рациональнее использовать программы `sudo` и `su` (описание которых можно получить по командам `man sudo` и `man su`).

Внимание!

В *приложении 3* приведены описания команды `chown`, позволяющей сменить владельца файла *(разд. П3.2.5)*, и команды `chattr`, позволяющей изменить атрибуты файла или запретить любое изменение, переименование и удаление файла *(разд. П3.2.6)*.

Совет

В *разд. П3.2.7–П3.2.13 приложения 3* приведены описания ряда весьма полезных "особых" команд — рекомендую обязательно с ними познакомиться.

9.3. Монтирование файловых систем

9.3.1. Команды *mount* и *umount*

Чтобы работать с какой-либо файловой системой, необходимо *примонтировать* ее к корневой файловой системе. Например, вставив в дисковод дискету, нужно подмонтировать файловую систему дискеты к корневой файловой системе — только так мы сможем получить доступ к файлам и каталогам, которые на этой дискете записаны. Аналогичная ситуация с жесткими, оптическими дисками и другими носителями данных.

Если вы хотите заменить сменный носитель данных (дискету, компакт-диск), вам нужно сначала размонтировать файловую систему, затем извлечь носитель данных, установить новый и заново смонтировать файловую систему. В случае с дискетой о размонтировании должны помнить вы сами, поскольку при размонтировании выполняется синхронизация буферов ввода/вывода и файловой системы, то есть данные физически записываются на диск, если это еще не было сделано. А компакт-диск система не разрешит вам извлечь, если он не размонтирован. В свою очередь, размонтировать файловую систему можно, только когда ни один процесс ее не использует.

При завершении работы системы (перезагрузке, выключении компьютера) размонтирование всех файловых систем выполняется автоматически.

Команда монтирования (ее нужно выполнять с привилегиями root) выглядит так:

```
# mount [опции] <устройство> <точка монтирования>
```

Точка монтирования — это каталог, через который будет осуществляться доступ к монтируемой файловой системе. Например, если вы подмонтировали компакт-диск к каталогу /mnt/cdrom, то получить доступ к файлам и каталогам, записанным на компакт-диске, можно будет через точку монтирования (именно этот каталог /mnt/cdrom). Точкой монтирования может быть любой каталог корневой файловой системы, хоть /aaa-111. Главное, чтобы этот каталог существовал на момент монтирования файловой системы.

В некоторых современных дистрибутивах запрещен вход в систему под именем суперпользователя — root. Поэтому для выполнения команд с привилегиями root вам нужно использовать команду `sudo`. Например, чтобы выполнить команду монтирования привода компакт-диска, вам нужно ввести команду:

```
sudo mount /dev/hdc /mnt/cdrom
```

Перед выполнением команды `mount` команда `sudo` попросит вас ввести пароль root. Если введенный пароль правильный, то будет выполнена команда `mount`.

Для размонтирования файловой системы используется команда `umount`:

```
# umount <устройство или точка монтирования>
```

9.3.2. Файлы устройств и монтирование

В этой главе мы уже говорили о файлах устройств. Здесь мы вернемся к ним снова, но в контексте монтирования файловой системы.

Как уже было отмечено, для Linux нет разницы между устройством и файлом. Все устройства системы представлены в корневой файловой системе как обычные файлы. Например, /dev/fd0 — это ваш дисковод для гибких дисков, /dev/sda (ранее — /dev/hda) — жесткий диск. Файлы устройств хранятся в каталоге /dev.

Примечание

В современных дистрибутивах имена вида /dev/hd*x* уже не используются (см. далее).

Жесткие диски

С жесткими дисками сложнее всего, поскольку одно и то же устройство может в разных версиях одного и того же дистрибутива называться по-разному. Например, мой IDE-диск, подключенный как первичный мастер, в Fedora 5 все еще назывался /dev/hda, а, начиная с Fedora 8, он стал называться /dev/sda. Дело в том, что ранее накопители, подключающиеся к интерфейсу IDE (PATA), назывались /dev/hd*x*, а SCSI/SATA-накопители — /dev/sd*x* (где в обоих случаях *x* — буква устройства). После перехода к менеджеру устройств udev[1] и принятия глобального уникального

[1] udev — это менеджер устройств, используемый в ядрах Linux версии 2.6. Пришел на смену более громоздкой псевдофайловой системе devfs. Управляет всеми манипуляциями с файлами из каталога /dev.

идентификатора устройств (UUID) все дисковые устройства, вне зависимости от интерфейса подключения (PATA, SATA, SCSI), стали называться /dev/sd*x*, где *x* — буква устройства. И поскольку все современные дистрибутивы поддерживают udev и UUID, не удивляйтесь, если вдруг ваш старенький IDE-винчестер в новом дистрибутиве будет назван /dev/sda.

С одной стороны, это вносит некоторую путаницу *(см. разд. 9.3.5)*. С другой — все современные компьютеры оснащены именно SATA-дисками (так как PATA-диски уже устарели, а SCSI — дорогие), так что многие пользователи просто ничего не заметят.

Рассмотрим ситуацию с жесткими дисками чуть подробнее. Пусть у нас есть устройство /dev/sda. На жестком диске, понятное дело, может быть несколько разделов. В нашем случае на диске имеются три раздела (логических диска), которые в Windows называются C:, D: и E:. Диск C: обычно является загрузочным (активным), поэтому этот раздел будет записан в самом начале диска. Нумерация разделов жесткого диска в Linux начинается с 1, и в большинстве случаев диску C: будет соответствовать имя /dev/sda1 — первый раздел на первом жестком диске.

Резонно предположить, что двум оставшимся разделам (D: и E:) будут присвоены имена /dev/sda2 и /dev/sda3. Это может быть и так, и не так. Как известно, на жестком диске могут существовать или четыре первичных раздела, или три первичных и один расширенный. В расширенном разделе могут разместиться до 11 логических дисков (разделов). Таким образом, раздел может быть первичным (primary partition), расширенным (extended partition) или логическим (logical partition).

Для возможных четырех первичных разделов диска в Linux зарезервированы номера 1, 2, 3, 4. Если разделы D: и E: нашего диска первичные, то да — им будут присвоены имена /dev/sda2 и /dev/sda3. Но в большинстве случаев эти разделы являются логическими и содержатся в расширенном разделе. Логические разделы именуются, начиная с 5, а это означает, что если разделы D: и E: — логические, им будут присвоены имена /dev/sda5 и /dev/sda6 соответственно.

> ***Примечание***
>
> В Windows расширенному разделу не присваивается буква, потому что этот раздел не содержит данных пользователя, а только информацию о логических разделах.

Узнать номер раздела очень просто — достаточно запустить утилиту `fdisk`, работающую с таблицей разделов диска. Чтобы узнать номера разделов первого жесткого диска (/dev/sda), введите команду:

```
# /sbin/fdisk /dev/sda
```

Вы увидите приглашение `fdisk`. В ответ на приглашение введите `p` и нажмите клавишу <Enter>. Откроется таблица разделов (рис. 9.2). Для выхода из программы введите `q` и нажмите клавишу <Enter>.

На рис. 9.2 изображена таблица разделов моего первого жесткого диска. Первый раздел (это мой диск C:, где установлена система Windows) — первичный. Сразу после него расположен расширенный раздел (его номер — 2). Следующий за ним — логический раздел (номер 5). Разделы с номерами 3 и 4 пропущены, потому что их нет на моем жестком диске. Это те самые первичные разделы, которые я не создал — они мне не нужны.

```
1) программами, запускаемым при загрузке (напр., старые версии LILO)
2) загрузкой и программами разметки из других ОС
   (напр., DOS FDISK, OS/2 FDISK)

Команда (m для справки): p

Диск /dev/sda: 160.0 ГБ, 160041885696 байт
255 heads, 63 sectors/track, 19457 cylinders
Units = цилиндры of 16065 * 512 = 8225280 bytes
Disk identifier: 0xe905e905

Устр-во Загр     Начало       Конец       Блоки   Id  Система
/dev/sda1   *           1         543     4361616    b  W95 FAT32
/dev/sda2             544       19457   151926705    f  W95 расшир. (LBA)
/dev/sda5             544        1021     3839503+  83  Linux
/dev/sda6            1022        1759     5927953+  83  Linux
/dev/sda7            1760        1825      530113+  82  Linux своп / Solaris
/dev/sda8            1826        5963    33238453+   b  W95 FAT32
/dev/sda9            5964       10101    33238453+   b  W95 FAT32
/dev/sda10          10102       14268    33471396    b  W95 FAT32
/dev/sda11          14269       16949    21535101    b  W95 FAT32
/dev/sda12          16950       19457    20145478+   b  W95 FAT32

Команда (m для справки):
```

Рис. 9.2. Таблица разделов жесткого диска

Приводы оптических дисков

Любое из устройств sd*x* может быть приводом для чтения CD или DVD. Если система видит, что устройство является приводом CD-ROM, то автоматически создается ссылка /dev/cdrom. А если ваш привод умеет также читать и DVD, то в каталоге /dev появится еще одна ссылка — /dev/dvd. Например, мой DVD-RW подключен как первичный подчиненный (/dev/sdb), и в каталоге /dev ему соответствуют три файла: /dev/sdb, /dev/cdrom, /dev/dvd. Обратиться к устройству можно, используя любой из этих файлов.

Для монтирования привода для чтения оптических дисков нужно ввести одну из трех команд:

```
# mount /dev/sdb /mnt/cdrom
# mount /dev/cdrom /mnt/cdrom
# mount /dev/dvd /mnt/cdrom
```

После этого обратиться к файлам, записанным на диске, можно будет через каталог /mnt/cdrom. Напомню, что каталог /mnt/cdrom должен существовать.

Дискеты и USB-накопители

Аналогичная ситуация и с дискетами. В системе может быть установлено два дисковода для дискет: первый (/dev/fd0) и второй (/dev/fd1).

Для их монтирования можно использовать команды:

```
# mount /dev/fd0 /mnt/floppy
# mount /dev/fd1 /mnt/floppy
```

В Windows-терминологии устройство /dev/fd0 — это диск A:, а устройство /dev/fd1 — диск B:.

USB-накопители определяются как жесткие диски. Если в вашей системе есть один жесткий диск, то его имя будет /dev/sda. Когда вы подключите флешку или

другой USB-накопитель (мобильный жесткий диск, кардридер и т. п.), то в системе появится новое устройство — /dev/sdb. Осталось только подмонтировать его. Далее мы поговорим о монтировании USB-накопителей более подробно.

Об автоматическом монтировании

Нужно отметить, что CD/DVD-приводы, а также USB-накопители монтируются автоматически (к каталогу /media/<ID накопителя>), поэтому все, что здесь сказано о монтировании таких носителей, — исключительно для общего развития.

9.3.3. Опции монтирования файловых систем

Теперь, когда мы знаем номер раздела, можно подмонтировать его файловую систему. Делается это так:

```
# mount <раздел> <точка монтирования>
```

Например:

```
# mount /dev/sda5 /mnt/win_d
```

У команды `mount` довольно много опций, но на практике наиболее часто используются только некоторые из них: `-t`, `-r`, `-w`, `-a`.

- Параметр `-t` позволяет задать тип файловой системы. Обычно программа сама определяет файловую систему, но иногда это у нее не получается. Тогда мы должны ей помочь. Формат использования этого параметра следующий:

```
# mount -t <файловая система> <устройство> <точка монтирования>
```

  Например:

```
# mount -t iso9660 /dev/hdc /mnt/cdrom
```

  Вот опции для указания наиболее популярных монтируемых файловых систем:

  - `ext2`, `ext3` или `ext4` — файловая система Linux;
  - `iso9660` — указывается при монтировании CD-ROM;
  - `vfat` — FAT, FAT32 (поддерживается Windows 9*x*, ME, XP);
  - `ntfs` — NT File System (поддерживается Windows NT, XP), будет использована стандартная поддержка NTFS, при которой NTFS-раздел доступен только для чтения;
  - `ntfs-3g` — будет использован модуль ntfs-3g, входящий в большинство современных дистрибутивов. Данный модуль позволяет производить запись информации на NTFS-разделы.

Примечание

Если в вашем дистрибутиве нет модуля ntfs-3g, то есть при попытке указания данной файловой системы вы увидели сообщение об ошибке, тогда вы можете скачать его с сайта **www.ntfs-3g.org**. На данном сайте доступны как исходные коды, так и уже откомпилированные для разных дистрибутивов пакеты.

- Параметр `-r` монтирует указанную файловую систему в режиме "только чтение".
- Параметр `-w` монтирует файловую систему в режиме "чтение/запись". Данный параметр используется по умолчанию для файловых систем, поддерживающих

запись (например, NTFS по умолчанию запись не поддерживает, как и файловые системы CD/DVD-дисков).

- Параметр `-a` используется для монтирования всех файловых систем, указанных в файле /etc/fstab (кроме тех, для которых указано `noauto` — такие файловые системы нужно монтировать вручную). При загрузке системы вызывается команда `mount` с параметром `-a`.

Если вы не можете смонтировать NTFS-раздел с помощью опции `ntfs-3g`, то, вероятнее всего, он был неправильно размонтирован (например, работа Windows не была завершена корректно). В этом случае для монтирования раздела нужно использовать опцию `-o force`, например:

```
sudo mount -t ntfs-3g /dev/sdb1 /media/usb -o force
```

9.3.4. Монтирование разделов при загрузке

Если вы не хотите при каждой загрузке монтировать постоянные файловые системы (например, ваши Windows-разделы), то их нужно прописать в файле /etc/fstab. Обратите внимание: в этом файле не нужно прописывать файловые системы сменных носителей (дисковода, CD/DVD-привода, Flash-диска). Следует отметить, что программы установки некоторых дистрибутивов, например Mandriva, читают таблицу разделов и автоматически заполняют файл /etc/fstab. В результате все ваши Windows-разделы доступны сразу после установки системы. К сожалению, не все дистрибутивы могут похвастаться такой интеллектуальностью, поэтому вам нужно знать формат файла fstab:

```
устройство точка_монтирования тип_ФС опции флаг_РК флаг_проверки
```

Здесь: `тип_ФС` — это тип файловой системы, а `флаг_РК` — флаг резервного копирования. Если он установлен (`1`), то программа `dump` заархивирует данную файловую систему при создании резервной копии. Если не установлен (`0`), то резервная копия этой файловой системы создаваться не будет. `Флаг проверки` устанавливает, будет ли данная файловая система проверяться на наличие ошибок программой `fsck`. Проверка производится в двух случаях:

- если файловая система размонтирована некорректно;
- если достигнуто максимальное число операций монтирования для этой файловой системы.

Поле опций содержит важные параметры файловой системы. Некоторые из них представлены в табл. 9.1.

Таблица 9.1. *Опции монтирования файловой системы в файле /etc/fstab*

Опция	Описание
`auto`	Файловая система должна монтироваться автоматически при загрузке. Опция используется по умолчанию, поэтому ее указывать не обязательно
`noauto`	Файловая система не монтируется при загрузке системы (при выполнении команды `mount -a`), но ее можно смонтировать вручную с помощью все той же команды `mount`

Таблица 9.1 *(окончание)*

Опция	Описание
`defaults`	Используется стандартный набор опций, установленных по умолчанию
`exec`	Разрешает запуск выполняемых файлов для данной файловой системы. Эта опция используется по умолчанию
`noexec`	Запрещает запуск выполняемых файлов для данной файловой системы
`ro`	Монтирование в режиме "только чтение"
`rw`	Монтирование в режиме "чтение/запись". Используется по умолчанию для файловых систем, поддерживающих запись
`user`	Данную файловую систему разрешается монтировать/размонтировать обычному пользователю (не root)
`nouser`	Файловую систему может монтировать только пользователь root. Используется по умолчанию
`umask`	Определяет маску прав доступа при создании файлов. Для файловых систем не Linux'а маску нужно установить так: `umask=0`
`utf8`	Применяется только на дистрибутивах, которые используют кодировку UTF-8 в качестве кодировки локали. В старых дистрибутивах (где используется KOI8-R) для корректного отображения русских имен файлов на Windows-разделах нужно задать параметры `iocharset=koi8-u`, `codepage=866`

Примечание

Редактировать файл /etc/fstab, как и любой другой файл из каталога /etc, можно в любом текстовом редакторе (например, gedit, kate), но перед этим нужно получить права root (командой `su` или `sudo`).

Рассмотрим небольшой пример:

```
/dev/sdc /mnt/cdrom auto umask=0,user,noauto,ro,exec 0 0
/dev/sda1 /mnt/win_c vfat umask=0,utf8 0 0
```

Первая строка — это строка монтирования файловой системы компакт-диска, а вторая — строка монтирования диска C:.

- Начнем с первой строки. `/dev/hdc` — это имя устройства CD-ROM. Точка монтирования — `/mnt/cdrom`. Понятно, что этот каталог должен существовать. Обратите внимание: в качестве файловой системы не указывается жестко iso9660, поскольку компакт-диск может быть записан в другой файловой системе, поэтому в качестве типа файловой системы задано `auto`, то есть автоматическое определение. Теперь идет довольно длинный набор опций. Ясно, что `umask` установлен в ноль, поскольку файловая система компакт-диска не поддерживает права доступа Linux. Параметр `user` говорит о том, что данную файловую систему можно монтировать обычному пользователю. Параметр `noauto` запрещает автоматическое монтирование этой файловой системы, что правильно — ведь на момент монтирования в приводе может и не быть компакт-диска. Опция `ro` разрешает монтирование в режиме "только чтение", а `exec` разрешает запускать

исполнимые файлы. Понятно, что компакт-диск не нуждается ни в проверке, ни в создании резервной копии, поэтому два последних флага равны нулю.

- Вторая строка проще. Первые два поля — это устройство и точка монтирования. Третье — тип файловой системы. Файловая система постоянна, поэтому можно явно указать тип файловой системы (`vfat`), а не `auto`. Опция `umask`, как и в предыдущем случае, равна нулю. Указание опции `utf8` позволяет корректно отображать русскоязычные имена файлов и каталогов.

9.3.5. Подробно о UUID и файле /etc/fstab

Пока вы еще не успели забыть формат файла /etc/fstab, нужно поговорить о UUID (Universally Unique Identifier), или о *длинных именах* дисков. В некоторых дистрибутивах, например в Ubuntu, вместо имени носителя (первое поле файла fstab) указывается его ID, поэтому fstab выглядит устрашающе, например вот так:

```
# /dev/sda6
UUID=1f049af9-2bdd-43bf-a16c-ff5859a4116a / ext3 defaults 0  1
# /dev/sda1
UUID=45AE-84D9  /media/hda1  vfat  defaults,utf8,umask=007 0 0
```

В SUSE идентификаторы устройств указываются немного иначе:

```
/dev/disk/by-id/scsi-SATA_WDC_WD1600JB-00_WD-WCANM7959048-part5 / ext3
acl,user_xattr 1 1
/dev/disk/by-id/scsi-SATA_WDC_WD1600JB-00_WD-WCANM7959048-part7 swap swap
defaults 0 0
```

Понятно, что использовать короткие имена вроде /dev/sda1 намного проще, чем идентификаторы в стиле `1f049af9-2bdd-43bf-a16c-ff5859a4116a`. Использование имен дисков еще никто не отменял, поэтому вместо идентификатора носителя можете смело указывать его файл устройства — так вам будет значительно проще!

Но все же вам нужно знать соответствие длинных имен коротким именам устройств. Ведь система использует именно эти имена, а в файле /etc/fstab не всегда указывается, какой идентификатор принадлежит какому короткому имени устройства (или указывается, но не для всех разделов).

Узнать "длинные имена" устройства можно с помощью простой команды:

```
ls -l /dev/disk/by-uuid/
```

Результат выполнения этой команды приведен на рис. 9.3.

```
[den@localhost ~]$ ls -l /dev/disk/by-uuid/
итого 0
lrwxrwxrwx 1 root root 10 Фев 12 15:25 1c3b8bd3-c26f-449e-9eba-8f254fefe814 -> .
./../sda1
lrwxrwxrwx 1 root root 10 Фев 12 15:25 3106fa17-65ef-42e9-a1d4-2313daee96b5 -> .
./../sda2
[den@localhost ~]$ ls -l /dev/disk/by-label
итого 0
lrwxrwxrwx 1 root root 10 Фев 12 15:25 SWAP-sda2 -> ../../sda2
lrwxrwxrwx 1 root root 10 Фев 12 15:25 \x2f -> ../../sda1
[den@localhost ~]$
```

Рис. 9.3. Соответствие длинных имен дисков коротким

Спрашивается, зачем введены длинные имена, если короткие имена были удобнее, во всяком случае для пользователей? Оказывается, разработчики Linux в первую очередь и заботились как раз о пользователях. Возьмем обычный IDE-диск. Как известно, его можно подключить либо к первичному (primary), либо к вторичному (secondary), если он есть, контроллеру. В зависимости от положения перемычки выбора режима винчестер может быть либо главным устройством (master), либо подчиненным (slave). Таким образом, в зависимости от контроллера, к которому подключается диск, изменяется его короткое имя — sda (primary master), sdb (primary slave), sdc (secondary master), sdd (secondary slave). То же самое происходит с SATA/SCSI-винчестерами — при изменении параметров подключения изменяется и короткое имя устройства.

При использовании же длинных имен идентификатор дискового устройства остается постоянным вне зависимости от типа подключения устройства к контроллеру. Именно поэтому длинные имена дисков часто также называются *постоянными* именами (persistent name). Вы, например, могли ошибочно подключить жесткий диск немного иначе, и разделы, которые назывались, скажем, /dev/sda*N*, стали называться /dev/sdb*N*. Понятно, что загрузить Linux с такого диска не получится, поскольку везде будут указаны другие имена устройств. Если же используются длинные имена дисков, система загрузится в любом случае, как бы вы ни подключили жесткий диск. Удобно? Конечно.

Но это еще не все. Постоянные имена — это только первая причина. Вторая причина заключается в обновлении библиотеки libata. В новой версии libata все PATA-устройства именуются не как hd*x*, а как sd*x*, что (как отмечалось в этой главе ранее) вносит некую путаницу. Длинные имена дисков от такой замены не изменяются, поэтому они избавляют пользователя от беспокойства по поводу того, что его старый IDE-диск вдруг превратился в диск SATA/SCSI.

При использовании UUID однозначно идентифицировать раздел диска можно несколькими способами:

- `UUID=45AE-84D9 /media/sda1 vfat defaults,utf8,umask=007,gid=46 0 0` — здесь с помощью параметра UUID указывается идентификатор диска;
- `/dev/disk/by-id/scsi-SATA_WDC_WD1600JB-00_WD-WCANM7959048-part7 swap swap defaults 0 0` — здесь указывается длинное имя устройства диска;
- `LABEL=/ / ext3 defaults 1 1` — самый компактный третий способ, позволяющий идентифицировать устройства по их метке.

> ***Примечание***
>
> Первый способ получения длинного имени в англоязычной литературе называется by-uuid, то есть длинное имя составляется по UUID, второй способ называется by-id, то есть по аппаратному идентификатору устройства. Третий способ называется by-label — по метке. Просмотреть соответствие длинных имен коротким можно с помощью команд:
>
> ```
> ls -l /dev/disk/by-uuid
> ls -l /dev/disk/by-id
> ls -l /dev/disk/by-label
> ```
>
> Но есть еще и четвертый способ, который называется by-path. В этом случае имя генерируется по sysfs. Данный способ является наименее используемым, поэтому вы редко столкнетесь с ним.

Узнать метки разделов можно с помощью команды:

```
ls -lF /dev/disk/by-label
```

Установить метку можно с помощью команд, указанных в табл. 9.2.

Таблица 9.2. *Команды для установки меток разделов*

Файловая система	Команда
ext2/ext3/ext4	`# e2label /dev/XXX <метка>`
ReiserFS	`# reiserfstune -l <метка> /dev/XXX`
JFS	`# jfs_tune -L <метка> /dev/XXX`
XFS	`# xfs_admin -L <label> /dev/XXX`
FAT/FAT32	Только средствами Windows
NTFS	`# ntfslabel /dev/XXX <метка>`

В файле /etc/fstab вы можете использовать длинные имена в любом формате. Можно указывать имена устройств в виде: `/dev/disk/by-uuid/*`, `/dev/disk/by-id/*` или `/dev/disk/by-label/*`, можно использовать параметры `UUID=идентификатор` или `LABEL=метка`. Используйте тот способ, который вам больше нравится.

9.3.6. Монтирование Flash-дисков

В последнее время стала очень популярна Flash-память. Уже сегодня Flash-память, точнее Flash-диски (они же USB-диски), построенные с использованием Flash-памяти, практически вытеснили обычные дискеты — они очень компактны и позволяют хранить довольно большие объемы информации. Сегодня никого не удивишь небольшим брелоком, вмещающем до 8 Гбайт информации.

Принцип использования Flash-диска очень прост — достаточно подключить его к шине USB, и через несколько секунд система определит диск. После этого с ним можно будет работать как с обычным диском. Да, Flash-диски не очень шустры, но молниеносной реакции от них никто и не ожидает — во всяком случае на фоне обычных дискет они выглядят настоящими спринтерами.

Технология Flash-памяти нашла свое применения в различных портативных устройствах — от мобильных телефонов до цифровых фотоаппаратов. Вы можете подключить мобильник к компьютеру и работать с ним как с обычным диском — записывать на него мелодии и картинки. Аналогичная ситуация и с цифровым фотоаппаратом — когда вы фотографируете, то фотографии и видеоролики записываются на его Flash-память. Потом вам нужно подключить его к компьютеру и просто скопировать фотографии. Вы также можете записать фотографии (или другие файлы — не имеет значения) на фотоаппарат, используя его встроенную Flash-память как большую дискету — для переноса своих файлов.

Все современные дистрибутивы умеют автоматически монтировать Flash-диски. После монтирования открывается окно с предложением просмотреть содержимое диска или же импортировать фотографии (в зависимости от типа подключенного устройства: обычный USB-диск или фотоаппарат).

Понятно, что нам, как настоящим линуксоидам, интересно, как самостоятельно смонтировать Flash-диск. Оказывается, тут все просто. USB-диск — это обычный накопитель, и его можно увидеть в каталоге /dev/disk/by-id. Напомню, что способ by-id подразумевает получение длинного имени по аппаратному идентификатору устройства, а поэтому с помощью каталога /dev/disk/by-id проще всего найти длинное имя USB-диска среди имен других накопителей — в его начале будет префикс `usb`. Введите команду:

```
ls –l /dev/disk/by-id | grep usb
```

Результат выполнения этой команды представлен на рис. 9.4.

Исходя из рис. 9.4, для монтирования Flash-диска нужно выполнить команду:

```
# mount /dev/sdb1 /mnt/flash
```

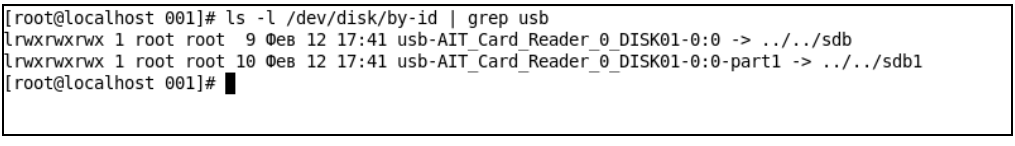


Рис. 9.4. USB-диск найден

9.4. Настройка журнала файловой системы ext3

Журналируемая файловая система имеет три режима работы: `journal`, `ordered` и `writeback`. Первый режим является самым медленным, но он позволяет минимизировать потери ваших данных в случае сбоя системы (или отключения питания). В этом режиме в системный журнал записывается все, что только можно, — это позволяет максимально восстановить файловую систему в случае сбоя.

В последовательном режиме (`ordered`) в журнал заносится информация только об изменении метаданных (служебных данных файловой системы). Данный режим используется по умолчанию и является компромиссным вариантом между производительностью и отказоустойчивостью.

Самым быстрым является режим обратной записи (`writeback`). Но использовать его я вам не рекомендую, поскольку особого толку от него не будет. Проще тогда уже при установке Linux выбрать файловую систему ext2 вместо ext3/ext4.

Если отказоустойчивость для вас на первом месте — выбирайте режим `journal`, во всех остальных случаях лучше выбрать `ordered`. Выбор режима осуществляется редактированием файла /etc/fstab. Например:

```
# Режим ordered используется по умолчанию,
# поэтому ничего указывать не нужно
/dev/sda1 / ext3 defaults 1 0
# На этом разделе важные данные — используем режим journal
/dev/sda2 /var ext3 data=journal 1 0
# Здесь ничего важного нет — режим writeback
/dev/sda3 /opt ext3 data=writeback 0 0
```

После изменения этого файла выполните команду:

```
# mount -a
```

Данная команда заново смонтирует все файловые системы, чтобы изменения вступили в силу.

9.5. Файловая система ext4

Файловая система ext4 заслуживает отдельного разговора. Все, что было сказано о файловых системах ранее, справедливо и для ext4, но у новой файловой системы есть ряд особенностей, о которых мы сейчас и поговорим.

Поддержка ext4, как стабильной файловой системы, появилась в ядре Linux версии 2.6.28. Если сравнивать эту файловую систему с ext3, то производительность и надежность новой файловой системы существенно увеличена, а максимальный размер раздела доведен до 1024 петабайт (1 эксбибайт). Максимальный размер файла — более 2 Тбайт. Ресурс Phoronix (**www.phoronix.com**) произвел тестирование новой файловой системы на SSD-накопителе (такие накопители устанавливаются на современные нетбуки) — результат, как говорится, налицо — ext4 почти в два раза превзошла файловые системы ext3, XFS, JFS и ReiserFS.

Впрочем, когда я установил Fedora 11 на рабочую станцию, прироста производительности при работе с файлами мне почувствовать не удалось. Однако производительность — это не конек ext4. Но обо всем по порядку.

9.5.1. Сравнение ext3 и ext4

Описание особенностей файловой системы ext4 и ее преимуществ по сравнению с ext3 сведены в табл. 9.3.

Таблица 9.3. *Особенности ext4*

Особенность	Комментарий
Увеличенный размер файла и файловой системы	Для ext3 максимальный размер файловой системы составляет 32 Тбайт, а файла — 2 Тбайт, но на практике ограничения были более жесткими. Так, в зависимости от архитектуры, максимальный размер тома составлял до 2 Тбайт, а максимальный размер файла — до 16 Гбайт. В случае с ext4 максимальный размер тома составляет 1 эксбибайт (EiB) — это 2^{60} байт. Максимальный размер файла составляет 16 Тбайт. Такие объемы информации пока не нужны обычным пользователям, однако весьма пригодятся на серверах, работающих с большими дисковыми массивами
Экстенты	Основной недостаток ext3 — ее метод выделения места на диске. Дисковые ресурсы выделялись с помощью битовых карт свободного места, а такой способ не отличается ни скоростью, ни масштабируемостью. Получилось, что ext3 более эффективна для небольших файлов, но совсем не подходит для хранения больших файлов

Таблица 9.3 *(окончание)*

Особенность	Комментарий
Экстенты	Для улучшения выделения ресурсов и более эффективной организации данных в ext4 были введены *экстенты*. Экстент — это способ представления непрерывной последовательности блоков памяти. Благодаря использованию экстентов сокращается количество метаданных (служебных данных файловой системы), поскольку вместо информации о том, где находится каждый блок памяти, экстент содержит информацию о том, где находится большой список непрерывных блоков памяти. Для эффективного представления маленьких файлов в экстентах применяется уровневый подход, а для больших файлов используются деревья экстентов. Например, один индексный дескриптор может ссылаться на четыре экстента, каждый из которых может ссылаться на другие индексные дескрипторы и т. д. Такая структура является мощным механизмом представления больших файлов, а также более защищена и устойчива к сбоям
Отложенное выделение пространства	Файловая система ext4 может отложить выделение дискового пространства до последнего момента, что увеличивает производительность системы
Контрольные суммы журналов	Контрольные суммы журналов повышают надежность файловой системы
Большее количество каталогов	В ext3 могло быть максимум 32 000 каталогов, в ext4 количество каталогов не ограничивается
Дефрагментация "на лету"	Файловая система ext3 не особо склонна к фрагментации, но все же такое неприятное явление имеется. В ext4 производится дефрагментация "на лету", что позволяет повысить производительность системы в целом
Наносекундные временные метки	В большинстве файловых систем временные метки (timestamp) устанавливаются с точностью до секунды, в ext4 точность повышена до наносекунды. Кроме того, ext4 поддерживает временные метки до 25 апреля 2514 года, в отличие от ext3 (18 января 2038 г.)

9.5.2. Совместимость с ext3

Файловая система ext4 является прямо и обратно совместимой с ext3, однако все же существуют некоторые ограничения. Предположим, что у нас на диске имеется файловая система ext4. Ее можно смонтировать и как ext3, и как ext4 (это и есть прямая совместимость) — и тут ограничений никаких нет. А вот с обратной совместимостью не все так безоблачно — если файловую систему ext4 смонтировать как ext3, то она будет работать без экстентов, что снизит ее производительность.

9.5.3. Переход на ext4

Если вы при установке системы выбрали файловую систему ext3, то перейти на ext4 можно без потери данных и в любой удобный для вас момент. Откройте терминал и введите команду:

```
sudo tune2fs -O extents,uninit_bg,dir_index /dev/имя_устройства
```

На момент ввода этой команды устройство должно быть размонтировано.

> ***Внимание!***
>
> Если нужно преобразовать в ext4 корневую файловую систему, то данную команду нужно вводить с LiveCD, поддерживающего ext4.

После этого проверим файловую систему:

```
sudo fsck -pf /dev/имя_устройства
```

Затем смонтируем файловую систему так:

```
mount -t ext4 /dev/имя_устройства /точка_монтирования
mount -t ext4 /dev/disk/by-uuid/UUID-устройства /точка_монтирования
```

Если раздел автоматически монтируется через /etc/fstab, не забудьте исправить файловую систему на ext4:

```
UUID=UUID-раздела  /точка ext4    defaults,errors=remount-ro,relatime
     0  1
```

Если вы изменили тип файловой системы корневого раздела, то необходимо отредактировать файл /boot/grub/menu.lst и добавить опцию `rootfstype=ext4` в список параметров ядра, например:

```
title              Linux
root        (hd0,1)
kernel      /boot/vmlinuz-2.6.28.1 root=UUID=879f797c-944d-4c28-a720-
249730705714 ro quiet splash rootfstype=ext4
initrd             /boot/initrd.img-2.6.28.1
quiet
```

> ***Совет***
>
> Рекомендую прочитать статью Тима Джонса "Анатомия ext4": **http://www.ibm.com/developerworks/ru/library/l-anatomy-ext4/index.html**.

9.6. Псевдофайловые системы

В Linux довольно популярны *псевдофайловые* системы. Слово "псевдо", как мы знаем, означает "почти", то есть псевдофайловая система — не совсем файловая система в прямом смысле этого слова. Псевдофайловые системы также называются *виртуальными* файловыми системами, поскольку работают на уровне виртуальной файловой системы (Virtual File System layer). Для большинства пользователей виртуальная файловая система выглядит как обычная файловая система — можно открыть тот или иной файл и посмотреть, что в нем записано, можно записать информацию в файл. Ради интереса зайдите в каталог /proc (это каталог псевдофайловой системы proc) и посмотрите на размер любого файла — например, на размер файла /proc/filesystems. Его размер будет равен 0, как и остальных файлов этой файловой системы, но если открыть сам файл, то вы увидите, что информация в нем есть. Это объясняется тем, что содержимое файла формируется при обращении к нему, то есть "на лету". Другими словами, виртуальная файловая система находится в оперативной памяти, а не на жестком диске. Информация попадает в файл на основании сведений, полученных от ядра.

В большинстве современных дистрибутивов используются виртуальные файловые системы sysfs и proc. Откройте файл /etc/fstab и вы увидите строки монтирования этих файловых систем:

```
sysfs     /sys       sysfs   defaults        0 0
proc      /proc      proc    defaults        0 0
```

9.6.1. Виртуальная файловая система sysfs

Виртуальная (псевдофайловая) система sysfs экспортирует в пространство пользователя информацию о ядре Linux, об имеющихся в системе устройствах и их драйверах. Впервые sysfs появилась в ядре версии 2.6. Зайдите в каталог /sys. Названия подкаталогов говорят сами за себя:

- block — содержит каталоги всех блочных устройств, имеющихся в системе в данное время (под устройством подразумевается совокупность физического устройства и его драйвера). Когда вы подключаете Flash-диск, то в любом случае в каталоге /sys/devices/ появляется новое устройство, но в каталоге /sys/block это устройство появится только при наличии соответствующих драйверов (в данном случае usb-storage);
- bus — перечень шин, поддерживаемых ядром (точнее, зарегистрированных в ядре). В каждом каталоге шины есть подкаталоги devices и drivers. В каталоге devices находятся ссылки на каталоги всех устройств, которые описаны в системе (то есть находящихся в каталоге /sys/devices);
- class — по этому каталогу можно понять, как устройства формируются в классы. Для каждого устройства в каталоге class есть свой отдельный каталог (под устройством, как и в случае с каталогом block, подразумевается совокупность устройства и его драйвера);
- devices — содержит файлы и каталоги, которые полностью соответствуют внутреннему дереву устройств ядра;
- drivers — каталоги драйверов для загруженных устройств. Подкаталог drivers каталога шины содержит драйверы устройств, работающих на данной шине.

9.6.2. Виртуальная файловая система proc

Виртуальная (псевдофайловая) система /proc — это специальный механизм, позволяющий посылать информацию ядру, модулям и процессам (кстати, потому данная файловая система так и называется: proc — это сокращение от process). Также, используя /proc, вы можете получать информацию о процессах и изменять параметры ядра и его модулей "на лету". Для этого в /proc есть файлы, позволяющие получать информацию о системе, ядре или процессе, и есть файлы, с помощью которых можно изменять некоторые параметры системы. Первые файлы мы можем только просмотреть, а вторые — просмотреть и, если нужно, изменить.

Просмотреть информационный файл можно командой `cat`:

```
cat /proc/путь/<название_файла>
```

Записать значение в один из файлов proc можно так:

```
echo "данные" > /proc/путь/название_файла
```

Информационные файлы

В табл. 9.4 представлены некоторые (самые полезные) информационные proc-файлы: с их помощью вы можете получить информацию о системе.

Таблица 9.4. *Информационные proc-файлы*

Файл	Описание
/proc/version	Содержит версию ядра
/proc/cmdline	Список параметров, переданных ядру при загрузке
/proc/cpuinfo	Информация о процессоре
/proc/meminfo	Информация об использовании оперативной памяти (почти то же, что и команда `free`)
/proc/devices	Список устройств
/proc/filesystems	Файловые системы, которые поддерживаются вашей системой
/proc/mounts	Список подмонтированных файловых систем
/proc/modules	Список загруженных модулей
/proc/swaps	Список разделов и файлов подкачки, которые активны в данный момент

Файлы, позволяющие изменять параметры ядра

Каталог /proc/sys/kernel содержит файлы, с помощью которых вы можете изменять важные параметры ядра. Конечно, все файлы мы рассматривать не будем, а рассмотрим лишь те, которые используются на практике (табл. 9.5).

Таблица 9.5. *Файлы каталога /proc/sys/kernel*

Файл	Каталог
/proc/sys/kernel/ctrl-alt-del	Если данный файл содержит значение 0, то при нажатии клавиатурной комбинации <Ctrl>+<Alt>+<Del> будет выполнена так называемая "мягкая перезагрузка", когда управление передается программе init и последняя "разгружает" систему, как при вводе команды `reboot`. Если этот файл содержит значение 1, то нажатие <Ctrl>+<Alt>+<Del> равносильно нажатию кнопки Reset. Сами понимаете, значение 1 устанавливать не рекомендуется
/proc/sys/kernel/domainname	Здесь находится имя домена, например, dkws.org.ua
/proc/sys/kernel/hostname	Содержит имя компьютера, например, den
/proc/sys/kernel/panic	При критической ошибке ядро "впадает в панику" — работа системы останавливается, а на экране красуется надпись **kernel panic** и выводится текст ошибки. Данный файл содержит значение в секундах, которое система будет ждать, пока пользователь прочитает это сообщение, после чего компьютер будет перезагружен. Значение 0 (по умолчанию) означает, что перезагружать компьютер вообще не нужно

Таблица 9.5 *(окончание)*

Файл	Каталог
/proc/sys/kernel/printk	Данный файл позволяет определить важность сообщения об ошибках. По умолчанию файл содержит значения 6 4 1 7. Это означает, что сообщения с уровнем приоритета 6 и ниже (чем ниже уровень, тем выше важность сообщения) будут выводиться на консоль. Для некоторых сообщений об ошибках уровень приоритета не задается. Тогда нужно установить уровень по умолчанию. Это как раз и есть второе значение — 9. Третье значение — это номер самого максимального приоритета, а последнее значение задает значение по умолчанию для первого значения. Обычно изменяют только первое значение, дабы определить, какие значения должны быть выведены на консоль, а какие — попасть в журнал демона syslog

Файлы, изменяющие параметры сети

В каталоге /proc/sys/net вы найдете файлы, изменяющие параметры сети (табл. 9.6).

Таблица 9.6. *Файлы каталога /proc/sys/net*

Файл	Описание
/proc/sys/net/core/message_burst	Опытные системные администраторы используют этот файл для защиты от атак на отказ (DoS). Один из примеров DoS-атаки — когда система заваливается сообщениями атакующего, а полезные сообщения системой игнорируются, потому что она не успевает реагировать на сообщения злоумышленника. В данном файле содержится значение времени (в десятых долях секунды), необходимое для принятия следующего сообщения. Значение по умолчанию — 50 (5 секунд). Сообщение, попавшее в "перерыв" (в эти 5 секунд), будет проигнорировано
/proc/sys/net/core/message_cost	Чем выше значение в этом файле, тем больше сообщений будет проигнорировано в перерыв, заданный файлом message_burst
/proc/sys/net/core/netdev_max_backlog	Задает максимальное число пакетов в очереди. По умолчанию 300. Используется, если сетевой интерфейс передает пакеты быстрее, чем система может их обработать
/proc/sys/net/core/optmem_max	Задает максимальный размер буфера для одного сокета

Файлы, изменяющие параметры виртуальной памяти

В каталоге /proc/sys/vm вы найдете файлы, с помощью которых можно изменить параметры виртуальной памяти:

- в файле buffermem находятся три значения (разделяются пробелами): минимальный, средний и максимальный объем памяти, которую система может использовать для буфера. Значения по умолчанию: 2 10 60;

- в файле kswapd тоже есть три значения, которые можно использовать для управления подкачкой:
  - первое значение задает максимальное количество страниц, которые ядро будет пытаться переместить на жесткий диск за один раз;
  - второе значение — минимальное количество попыток освобождения той или иной страницы памяти;
  - третье значение задает количество страниц, которые можно записать за один раз. Значения по умолчанию 512 32 8.

Файлы, позволяющие изменить параметры файловых систем

Каталог /proc/sys/fs содержит файлы, изменяющие параметры файловых систем. В частности:

- файл file-max задает максимальное количество одновременно открытых файлов (по умолчанию 4096);
- в файле inode-max содержится максимальное количество одновременно открытых индексных дескрипторов — *инодов* (максимальное значение также равно 4096);
- в файле super-max находится максимальное количество используемых суперблоков;

> ***Пояснение***
>
> Поскольку каждая файловая система имеет свой суперблок, легко догадаться, что количество подмонтируемых файловых систем не может превысить значение из файла super-max, которое по умолчанию равно 256, чего в большинстве случаев вполне достаточно. Наоборот, можно уменьшить это значение, дабы никто не мог подмонтировать больше файловых систем, чем нужно (если монтирование файловых систем разрешено обычным пользователям).

- в файле super-nr находится количество открытых суперблоков в текущий момент. Данный файл нельзя записывать, его можно только читать.

Как сохранить изменения

Итак, вы изменили некоторые параметры системы с помощью /proc, и теперь вам нужно их сохранить. Чтобы сохранить измененные параметры, их нужно прописать в файле /etc/sysctl.conf. Вот только формат этого файла следующий: надо отбросить /proc/sys/ в начале имени файла, а все, что останется, записать через точку, а затем через знак равенства указать значение параметра. Например, для изменения параметра /proc/sys/vm/buffermem нужно в файле etc/sysctl.conf прописать строку:

```
vm.buffermem = 2 11 60
```

Если в вашем дистрибутиве нет файла /etc/sysctl.conf, тогда пропишите команды вида `echo "значение" > файл` в сценарий инициализации системы.

9.7. Программы для разметки диска

9.7.1. Программа fdisk

В этом разделе будут рассмотрены две программы для разметки диска: классическая программа fdisk и более продвинутая parted. Реально для разметки диска (если вам придется это делать) вы будете использовать программу parted, поскольку она умеет изменять размеры разделов, что пригодится при переразметке диска. А вот fdisk можно использовать разве что для разметки новых жестких дисков. Изменить размер раздела без потери данных fdisk не может — вам придется удалить один из разделов, а вместо него создать несколько разделов меньшего размера. Только так, и не иначе. Зато fdisk установлена по умолчанию во всех дистрибутивах, и ее не нужно доустанавливать самостоятельно.

Введите команду (можно использовать короткие имена):

```
# fdisk <имя_устройства>
```

Например, если вы подключили винчестер как вторичный мастер, то команда будет следующей:

```
# fdisk /dev/sda
```

Чтобы убедиться, что диск не размечен, введите команду `p`. Программа выведет пустую таблицу разделов (рис. 9.5).

Самое время создать раздел. Для этого используется команда `n` (рис. 9.6). Кстати, для справки можете ввести команду `m`, которая выведет список доступных команд программы fdisk (рис. 9.7).

```
Command (m for help): p

Disk /dev/sda: 1825 MB, 1825360896 bytes
64 heads, 63 sectors/track, 884 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes

   Device Boot      Start         End      Blocks   Id  System

Command (m for help): _
```

Рис. 9.5. Таблица разделов пуста

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-884, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-884, default 884): +700M
```

Рис. 9.6. Создание нового раздела

```
64 heads, 63 sectors/track, 884 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes

   Device Boot      Start         End      Blocks   Id  System

Command (m for help): m
'Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)

Command (m for help): _
```

Рис. 9.7. Список команд программы fdisk

```
Command (m for help): p

Disk /dev/sda: 1825 MB, 1825360896 bytes
64 heads, 63 sectors/track, 884 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1               1         340      685408+  83  Linux
/dev/sda2             341         884     1096704   83  Linux

Command (m for help): _
```

Рис. 9.8. Создание второго раздела, вывод таблицы разделов

```
 0  Empty           1e  Hidden W95 FAT1 80  Old Minix       be  Solaris boot
 1  FAT12           24  NEC DOS         81  Minix / old Lin bf  Solaris
 2  XENIX root      39  Plan 9          82  Linux swap / So c1  DRDOS/sec (FAT-
 3  XENIX usr       3c  PartitionMagic  83  Linux           c4  DRDOS/sec (FAT-
 4  FAT16 <32M      40  Venix 80286     84  OS/2 hidden C:  c6  DRDOS/sec (FAT-
 5  Extended        41  PPC PReP Boot   85  Linux extended  c7  Syrinx
 6  FAT16           42  SFS             86  NTFS volume set da  Non-FS data
 7  HPFS/NTFS       4d  QNX4.x          87  NTFS volume set db  CP/M / CTOS / .
 8  AIX             4e  QNX4.x 2nd part 88  Linux plaintext de  Dell Utility
 9  AIX bootable    4f  QNX4.x 3rd part 8e  Linux LVM       df  BootIt
 a  OS/2 Boot Manag 50  OnTrack DM      93  Amoeba          e1  DOS access
 b  W95 FAT32       51  OnTrack DM6 Aux 94  Amoeba BBT      e3  DOS R/O
 c  W95 FAT32 (LBA) 52  CP/M            9f  BSD/OS          e4  SpeedStor
 e  W95 FAT16 (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi eb  BeOS fs
 f  W95 Ext'd (LBA) 54  OnTrackDM6      a5  FreeBSD         ee  EFI GPT
10  OPUS            55  EZ-Drive        a6  OpenBSD         ef  EFI (FAT-12/16/
11  Hidden FAT12    56  Golden Bow      a7  NeXTSTEP        f0  Linux/PA-RISC b
12  Compaq diagnost 5c  Priam Edisk     a8  Darwin UFS      f1  SpeedStor
14  Hidden FAT16 <3 61  SpeedStor       a9  NetBSD          f4  SpeedStor
16  Hidden FAT16    63  GNU HURD or Sys ab  Darwin boot     f2  DOS secondary
17  Hidden HPFS/NTF 64  Novell Netware  b7  BSDI fs         fd  Linux raid auto
18  AST SmartSleep  65  Novell Netware  b8  BSDI swap       fe  LANstep
1b  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid ff  BBT
1c  Hidden W95 FAT3 75  PC/IX
Hex code (type L to list codes): _
```

Рис. 9.9. Коды файловых систем

После ввода команды n программа попросит вас уточнить, какого типа должен быть раздел. Можно выбрать первичный или расширенный раздел. В нашем случае больше подойдет первичный, поэтому вводим букву p. Затем нужно ввести номер раздела. Поскольку это первый раздел, то вводим 1. Потом fdisk попросит ввести номер первого цилиндра. Это первый раздел, поэтому вводим номер 1. После ввода первого цилиндра нужно ввести номер последнего цилиндра. Чтобы не высчитывать на калькуляторе номер цилиндра, намного проще ввести размер раздела. Делается это так: +*<размер>*M. После числа должна идти именно буква M, иначе размер будет воспринят в байтах, а этого нам не нужно. Например, если вы хотите создать раздел размером 10 Гбайт, то введите +10240M.

Для создания второго раздела опять введите команду n. Программа вновь попросит тип раздела, номер первого цилиндра (это будет номер последнего цилиндра первого раздела плюс 1) и размер раздела. Если вы хотите создать раздел до "конца" диска, то просто введите номер последнего цилиндра.

Теперь посмотрим на таблицу разделов. Для этого опять введите команду p (рис. 9.8).

По умолчанию программа fdisk создает Linux-разделы. Если вы собираетесь работать только в Linux, можно оставить и так, но ведь не у всех есть Linux. Если вы снимете этот винчестер, чтобы, например, переписать у товарища большие файлы, то вряд ли сможете комфортно с ним работать. Прочитать данные (например, с помощью Total Commander) вам удастся, а что-либо записать — уже нет. Поэтому давайте изменим тип разделов. Для этого используется команда t. Введите эту команду. Программа запросит у вас номер раздела и тип файловой системы. С номером раздела все ясно, а вот с кодом файловой системы сложнее. Введите L, чтобы просмотреть доступные файловые системы (рис. 9.9).

Код FAT32 — b. Введите его, и вы увидите сообщение программы, что тип файловой системы изменен (рис. 9.10).

```
Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): b
Changed system type of partition 2 to b (W95 FAT32)

Command (m for help): _
```

Рис. 9.10. Тип файловой системы изменен

Еще раз введите команду p, чтобы убедиться, что все нормально. Для сохранения таблицы разделов введите w, а для выхода без сохранения изменений — q.

9.7.2. Программа parted

Утилита parted (название представляет собой сокращение от PARTition EDitor) является консольной программой, которая используется для создания, удаления, копирования, изменения размера и размещения разделов диска.

Программа поддерживает следующие таблицы разделов: BSD, MAC, MSDOS, PC98, SUN, GPT. Кроме того, программа поддерживает прямой доступ (raw

access) к диску, что полезно при работе с логическими томами (LVM) и RAID-массивами.

Программа parted поддерживает множество файловых систем, но не для всех файловых систем доступны все выполняемые программой действия. В табл. 9.7 представлена информация о действиях, которые можно выполнить над той или иной файловой системой.

Таблица 9.7. *Действия, поддерживаемые программой parted*

Файловая система	Обнаружение	Создание	Изменение размера	Копирование	Проверка
ext3	+		+	+	+
ext2	+	+	+	+	+
fat32	+	+	+	+	+
fat16	+	+	+	+	+
ntfs	+	+	+	+	+
linux-swap	+	+	+	+	+
ReiserFS	+	+	+	+	+
JFS	+				
XFS	+				
UFS	+				

Программа не умеет создавать разделы ext3 и ext4, но может создать раздел ext2, который можно без особых проблем преобразовать в ext3 или ext4. Разделы типов JFS, XFS и UFS только обнаруживаются программой, но она не может выполнять над ними никаких действий.

Примечание

Для работы с NTFS-разделами обязательна установка пакета linux-ntfs.

Запустим parted (рис. 9.11):

```
# parted <имя устройства>
```

Например:

```
# parted /dev/sda
```

```
denix@denis-desktop:~$ sudo parted /dev/sda
GNU Parted 1.8.8.1.159-1e0e
Использование /dev/sda
Добро пожаловать в GNU Parted! Наберите 'help' для получения списка команд.
(parted)
```

Рис. 9.11. Программа parted

Примечание

Не забывайте, что программа должна быть выполнена от имени root! Получить права root можно с помощью команды `sudo`, например: `sudo parted /dev/sda`.

Введите команду `print` для просмотра имеющихся разделов:

```
(parted) print
Disk geometry for /dev/sda: 0.000-9990.109 megabytes
Disk label type: msdos
Minor    Start       End          Type              Filesystem        Flags
1        0.031     512.000     primary          linux-swap
2      512.000    9990.109     primary          ext2                   boot
```

Первая колонка — это номер раздела, вторая и третья — смещение (в мегабайтах) от "начала" диска. Следующая колонка — тип раздела, затем — тип файловой системы. Последняя колонка — флаги, например: `boot` — загрузочный раздел.

Введите команду `help`, чтоб увидеть список команд parted (рис. 9.12). В табл. 9.8 приведено их описание.

Таблица 9.8. *Основные команды parted*

Команда	Описание
`check n`	Проверить раздел с номером `n`
`cp [устройство] n m`	Копировать файловую систему из раздела `n` в раздел `m`. `устройство` — номер устройства, где находится раздел `n`. Если устройство не задано, то считается, что используется текущее устройство
`mklabel тип`	Создает новую метку диска
`mktable тип`	Создает новую таблицу разделов
`mkfs n тип_фс`	Создает файловую систему заданного типа на разделе n
`mkpart тип [фс] нач кон`	Создать раздел указанного типа. `[фс]` — необязательный параметр, задающий тип файловой системы. Параметры `нач` и `кон` задают начало и конец раздела
`move n нач кон`	Переместить раздел с номером `n`. `нач` и `кон` — конечные "координаты" раздела, его начало и конец, заданные как смещение от "начала" диска в мегабайтах
`print [devices|free|all|n]`	Отображает таблицу разделов (если параметры не заданы), список устройств (`devices`), свободное место (`free`), все найденные разделы (`all`) или информацию о разделе с номером `n`
`quit`	Выход из программы
`rescue нач кон`	Восстанавливает потерянный раздел в промежутке, заданном параметрами `нач` и `кон`
`resize n нач кон`	Изменяет размер раздела `n`
`rm n`	Удаляет раздел с номером `n`

***Таблица 9.8** (окончание)*

Команда	Описание
`select устройство`	Выбирает устройство для редактирования, вам нет необходимости выходить из программы для изменения устройства
`set n флаг состояние`	Изменяет состояние флага для раздела `n`. Доступные флаги описаны в табл. 9.9
`unit устройство`	Устанавливает устройство по умолчанию
`version`	Выводит версию parted

```
GNU Parted 1.8.8.1.159-1e0e
Использование /dev/sda
Добро пожаловать в GNU Parted! Наберите 'help' для получения списка команд.
(parted) help
  check НОМЕР                      производит простую проверку файловой системы
  cp [ИЗ_УСТРОЙСТВА] ИЗ_НОМ В_НОМ скопировать файловую систему на другой раздел
  help [КОМАНДА]                   распечатать общую справку или справку по
        КОМАНДЕ
  mklabel,mktable ТИП_МЕТКИ         создать новую метку диска (таблицу раздела)
  mkfs НОМЕР ТИП_ФС                создать файловую систему ТИП_ФС на разделе
        НОМЕР
  mkpart ТИП_РАЗД [ТИП_ФС] НАЧ КОН создать раздел
  mkpartfs ТИП_РАЗД ТИП_ФС НАЧ КОН создать раздел с файловой системой
  move НОМЕР НАЧ КОН               переместить файловую систему НОМЕР
  name НОМЕР ИМЯ                   назначает имя разделу НОМЕР на ИМЯ
  print [devices|free|list,all|НОМЕР]     отображает таблицу разделов, доступные
        устройства, свободное место, все найденные разделы или определённый
        раздел
  quit                             выйти из программы
  rescue НАЧАЛО КОНЕЦ              восстановить потерянный раздел в промежутке
        от НАЧАЛА до КОНЦА
  resize НОМЕР НАЧ КОН             изменить размер файловой системы на разделе
        НОМЕР
  rm НОМЕР                         удалить раздел НОМЕР
  select УСТРОЙСТВО                выбор устройства для редактирования
  set НОМЕР ФЛАГ СОСТОЯНИЯ         изменить ФЛАГ на разделе НОМЕР
  toggle [НОМЕР [ФЛАГ]]            переключает состояния ФЛАГА на разделе НОМЕР
  unit УСТРОЙСТВО                  установить устройство по умолчанию на
        УСТРОЙСТВО
  version                                  отображает текущую версию GNU Parted
        и информацию о лицензии
(parted)
```

Рис. 9.12. Команда help

Примечание

Помнить формат каждой команды необязательно. Если вы введете команду, но не укажете ее параметры, то они будут запрошены.

***Таблица 9.9.** Флаги разделов*

Флаг	Тип раздела	Описание
`boot`	Mac, msdos, pc98	Флаг загрузочного раздела. Нужен для некоторых операционных систем
`lba`	msdos	Нужен для MS-DOS, MS Windows 9*x* и MS Windows ME, чтобы эти ОС использовали для раздела режим линейной адресации (LBA)

Таблица 9.9 *(окончание)*

Флаг	Тип раздела	Описание
`swap`	Mac	Устанавливается, если раздел является разделом подкачки Linux
`root`	Mac	Устанавливается, если раздел является корневым разделом Linux
`raid`	msdos	Раздел используется в RAID-массиве
`lvm`	msdos	Раздел используется как физический том в LVM
`hidden`	msdos, pc98	Скрытый раздел, устанавливается, если нужно скрыть от операционных систем семейства Microsoft

Как видите, программа parted намного более эффективна, чем fdisk. Любителям графического интерфейса можно порекомендовать графическую версию этой программы — gparted (рис. 9.13).

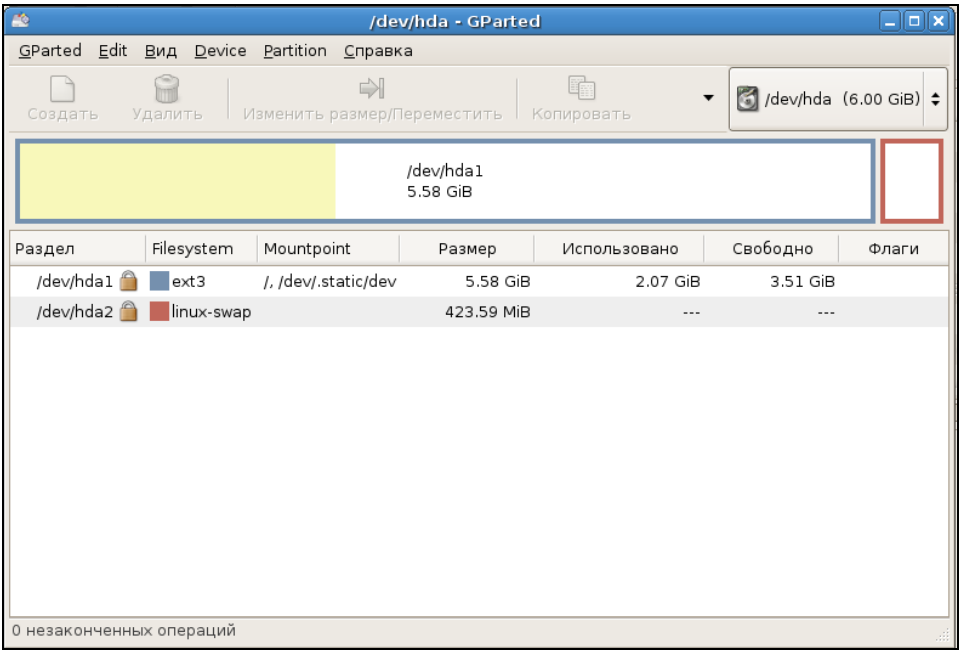
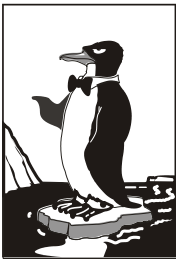


Рис. 9.13. Программа gparted



Глава 10

Команды управления пользователями

10.1. Многопользовательская система

Linux, как и UNIX, является многозадачной многопользовательской операционной системой. Это означает, что в один момент с системой могут работать несколько пользователей, и каждый пользователь может запустить несколько приложений. При этом вы можете зайти в систему локально, а кто-то — удаленно, используя один из протоколов удаленного доступа (telnet, ssh) или по FTP. Согласитесь, очень удобно. Предположим, что вы забыли распечатать очень важный документ, а возвращаться домой уже нет времени. Если ваш компьютер должным образом настроен и подключен к Интернету, вы можете получить к нему доступ (даже если компьютер выключен, достаточно позвонить домой и попросить кого-то включить его, а к Интернету компьютер подключится автоматически). После чего зайдите в систему по ssh (или подключитесь к графическому интерфейсу, если вы предпочитаете работать в графическом режиме) и скопируйте нужный вам файл. Даже если кто-то в момент вашего подключения уже работает с системой, вы не будете мешать друг другу.

Вы можете обвинить меня в рекламе Linux: мол, эта возможность была и в Windows 98, если установить соответствующее программное обеспечение вроде Remote Administrator. Должен отметить, что в Windows все иначе. Да, Remote Administrator предоставляет удаленный доступ к рабочему столу, но если за компьютером уже работает пользователь, то вы вместе работать не сможете — вы будете мешать ему, а он вам. Ведь все, что будете делать вы, будет видеть он, а все, что будет делать он, вы увидите у себя на экране, то есть рабочий стол получится как бы общий. Если вы предварительно не предупредите пользователя о своем удаленном входе, он даже может подумать, что с системой что-то не то. Помню, со мной так и было — пользователь, работавший за компьютером, закрывал окна, которые я открывал, работая в удаленном режиме. Пришлось мне самому пойти к компьютеру того пользователя и попросить его не мешать.

В Linux же все так, как и должно быть. Несколько пользователей могут работать с системой и даже не подозревать о существовании друг друга, пока не введут соответствующую команду (`who`).

10.2. Пользователь root

10.2.1. Максимальные полномочия

Пользователь root обладает максимальными полномочиями в системе. Система полностью подвластна этому пользователю. Любая команда будет безоговорочно выполнена системой. Поэтому работать под именем пользователя root нужно с осторожностью. Всегда думайте над тем, что собираетесь сделать. Если вы дадите команду на удаление корневой файловой системы, система ее выполнит. Если же вы попытаетесь выполнить определенную команду, зарегистрировавшись под именем обычного пользователя, система сообщит вам, что у вас нет полномочий.

Представим, что кто-то решил пошутить и выложил в Интернете (записал на диск или прислал по электронной почте — не важно) вредоносную программу. Если вы ее запустите от имени пользователя root, система может быть уничтожена. Запуск этой же программы от имени обычного пользователя ничего страшного не произведет — система просто откажется ее выполнять.

Или же все может быть намного проще — вы ошибочно введете команду, которая разрушит вашу систему. Или просто отойдете ненадолго от своего компьютера, а тут сразу же появится "доброжелатель", — имея полномочия пользователя root, уничтожить систему можно одной командой. Именно поэтому практически во всех современных дистрибутивах вход под именем root затруднен. В одних дистрибутивах вы не можете войти как root в графическом режиме (но можете войти в консоли, переключившись на первую консоль с помощью комбинации клавиш <Ctrl>+<Alt>+<F1>), а в других вообще не можете войти в систему как root: ни в графическом режиме, ни в консоли (пример такого дистрибутива — Ubuntu).

Отсюда можно сделать следующие выводы:

- старайтесь реже работать пользователем root;
- всегда думайте, какие программы вы запускаете под именем root;
- если программа, полученная из постороннего источника, требует root-полномочий, это должно насторожить;
- создайте обычного пользователя (даже если вы сами являетесь единственным пользователем компьютера) и рутинные операции (с документами, использование Интернета и т. д.) производите от имени этого пользователя;
- если полномочия root все же нужны, совсем необязательно заходить в систему под этим пользователем, достаточно запустить терминал и выполнить команду `sudo` (см. далее). После этого в терминале можно выполнять команды с правами root. Если вы закроете терминал, то больше не сможете работать с правами root. Очень удобно — ведь обычно права root нужны для одной-двух операций (например, выполнить команду установки программы или создать/удалить пользователя).

10.2.2. Как работать без root

Некоторые операции, например, установка программного обеспечения, изменение конфигурационных файлов, требуют полномочий root. Чтобы их временно

получить, нужно использовать команды `sudo` или `su` (эти команды, скорее всего, вы будете запускать в терминале).

Команда *sudo*

Команда `sudo` позволяет запустить любую команду с привилегиями root. Использовать ее нужно так:

```
sudo <команда_которую_нужно_выполнить_с_правами_root>
```

Например, вам необходимо изменить файл /etc/apt/sources.list. Для этого используется команда:

```
sudo nano /etc/apt/sources.list
```

> ***Пояснение***
>
> Программа `nano` — это текстовый редактор, мы ему передаем один параметр — имя файла, который нужно открыть.

Если ввести эту же команду, но без `sudo` (просто `nano /etc/apt/sources.list`), текстовый редактор тоже запустится и откроет файл, но сохранить изменения вы не сможете, поскольку у вас не хватит полномочий.

Программа `sudo` перед выполнением указанной вами команды запросит у вас пароль:

```
sudo nano /etc/apt/sources.list
```

```
Password:
```

Вы должны ввести свой *пользовательский пароль* — тот, который применяете для входа в систему, но не пароль пользователя root (кстати, мы его и не знаем).

> ***Примечание***
>
> Использовать команду `sudo` имеют право не все пользователи, а только те, которые внесены в файл /etc/sudoers. Администратор системы (пользователь root) может редактировать этот файл с помощью команды `visudo`. Если у вас дистрибутив, который запрещает вход под учетной записью root (следовательно, у вас нет возможности отредактировать файл sudoers), то в файл sudoers вносятся пользователи, которых вы добавили при установке системы.

Проблемы с *sudo* в Ubuntu и Kubuntu

Если вы в терминале хотите запустить графическую программу с правами root (например, `gedit`), желательно использовать не программу `sudo`, а программу `gksudo` или `gksu` (для Ubuntu) или `kdesu` (для Kubuntu). Программа `sudo` не всегда корректно работает с графическими приложениями, поэтому рано или поздно вы можете получить сообщение **Unable to read ICE authority file**, и после этого вообще станет невозможным запуск графических программ с правами root. Если это все же произошло, поправить ситуацию можно, удалив файл .{ICE,X}authority из вашего домашнего каталога:

```
rm ~/.{ICE,X}authority
```

Напомню, что тильда здесь означает домашний каталог текущего пользователя.

Графические приложения с правами root проще запускать, используя главное меню. Но не все приложения есть в главном меню, или не все приложения вызы-

ваются с правами root — например, в главном меню есть команда вызова текстового редактора, но нет команды для вызова текстового редактора с правами root. Поэтому намного проще нажать клавиатурную комбинацию <Alt>+<F2> и в открывшемся диалоговом окне **Выполнить программу** ввести команду в соответствующее поле (рис. 10.1).

```
gksu <команда>
```

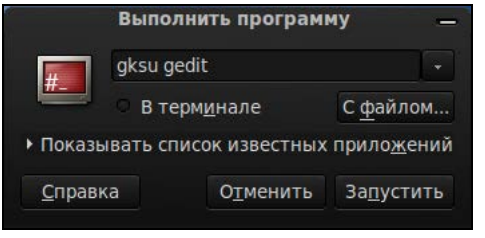


Рис. 10.1. Быстрое выполнение программы

Ввод серии команд *sudo*

Вам надоело каждый раз вводить `sudo` в начале команд? Тогда выполните команду:

```
sudo -i
```

Данная команда запустит оболочку root, то есть вы сможете вводить любые команды, и они будут выполнены с правами root. Обратите внимание, что изменится приглашение командной строки (рис. 10.2). До этого приглашение имело вид `$`, что означало, что вы работаете от имени обычного пользователя, а после выполнения программы приглашение изменилось на # — это верный признак того, что каждая введенная команда будет выполнена с правами root.

```
denix@denix-desktop:~$ sudo -i
[sudo] password for denix:
root@denix-desktop:~#
```

Рис. 10.2. Оболочка root

Опция `-i` позволяет так же удобно вводить команды, как если бы вы использовали команду `sudo`.

Преимущества и недостатки *sudo*

Рассмотрим теперь преимущества и недостатки использования команды `sudo`.

К преимуществам `sudo` можно отнести следующие соображения:

- вам не нужно помнить несколько паролей (то есть ваш пароль и пароль пользователя root) — вы помните только свой пароль и вводите его, когда нужно;

- с помощью `sudo` вы можете выполнять практически те же действия, что и под именем root, но перед каждым действием у вас будет запрошен пароль, что позволит еще раз подумать о правильности своих действий;
- каждая команда, введенная с помощью `sudo`, записывается в журнал /var/log/auth.log, поэтому в случае чего вы хотя бы будете знать, что случилось, прочитав этот журнал. У вас также будет храниться история введенных команд с полномочиями root, в то время как при работе под именем root никакой журнал не ведется;
- предположим, некто захотел взломать вашу систему. Этот некто не знает, какие учетные записи есть в вашем компьютере, зато уверен, что учетная запись root есть всегда. Знает он также, что, завладев паролем к этой учетной записи, можно получить неограниченный доступ к системе. Но не к вашей системе — у вас учетная запись root отключена!
- вы можете разрешать и запрещать другим пользователям использовать полномочия root (позже мы разберемся, как это сделать), не предоставляя пароль root, то есть практически нет риска скомпрометировать учетную запись root (впрочем, риск есть всегда, ведь при неправильно настроенной системе с помощью команды `sudo` можно легко изменить пароль root).

Но у `sudo` есть и недостатки:

- неудобно использовать перенаправление ввода/вывода, например, команда:

```
sudo ls /etc > /root/somefile
```

работать не будет, вместо нее нужно использовать команду:

```
sudo bash -c "ls /etc > /root/somefile"
```

Длинновато, правда?

- имеются и неудобства, связанные с технологией NSS (Name Service Switch) LDAP (Lightweight Directory Access Protocol). К счастью, она используется не очень часто, поэтому основной недостаток `sudo` будет связан только с перенаправлением ввода/вывода.

Команда *su*

Команда `su` позволяет получить доступ к консоли root любому пользователю (даже если пользователь не внесен в файл /etc/sudoers) при условии, что он знает пароль root. Понятно, что в большинстве случаев этим пользователем будет сам пользователь root — не будете же вы всем пользователям доверять свой пароль? Поэтому команда `su` предназначена, в первую очередь, для администратора системы, а `sudo` — для остальных пользователей, которым иногда нужны права root (чтобы они меньше отвлекали администратора от своей работы).

Использовать команду `su` просто:

```
su
```

После этого нужно ввести пароль пользователя root, и вы сможете работать в консоли как обычно. Использовать `su` удобнее, чем `sudo`, потому что вам не нужно вводить `su` перед каждой командой, которая должна быть выполнена с правами root.

Чтобы закрыть сессию `su`, нужно или ввести команду `exit`, или просто закрыть окно терминала.

В случае, если вы запускаете какую-нибудь графическую программу, требующую привилегий root, тогда вы увидите окно с требованием ввести свой пароль, подобное изображенному на рис. 10.3.

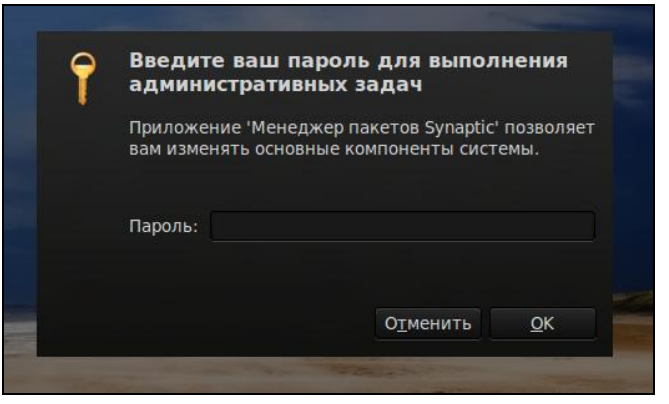


Рис. 10.3. Требование ввести пароль

10.2.3. Переход к традиционной учетной записи root

Как уже было отмечено, во многих дистрибутивах использование учетной записи root ограничено. В одних дистрибутивах она отключена, и для получения необходимых полномочий нужно использовать команду `sudo`, в других невозможно войти в графическом режиме как root.

Тем не менее, возможность перейти к традиционной учетной записи root, то есть заходить в систему под именем root, как вы заходите под именем обычного пользователя, имеется всегда.

Традиционная учетная запись root в Ubuntu

Вы все-таки хотите использовать обычную учетную запись root? Для этого достаточно задать пароль для пользователя root. Делается это командой:

```
sudo passwd root
```

Сначала программа запросит ваш пользовательский пароль, затем новый пароль root и его подтверждение:

Enter your existing password:
Enter password for root:
Confirm password for root:

После этого вы сможете входить в систему под учетной записью root.

Для отключения учетной записи root используется команда:

```
sudo passwd -l root
```

Внимание!

Помните, что после закрытия учетной записи root у вас могут быть проблемы с входом в систему в режиме восстановления, поскольку пароль root уже установлен (то есть он

не пустой, как по умолчанию), но в то же время учетная запись закрыта. Поэтому если вы уже включили учетную запись root, то будьте внимательны и осторожны. А вообще лучше ее не включать, а пользоваться командой `sudo -i`.

Традиционная учетная запись root в Mandriva

В Ubuntu учетная запись root отключена честно. В Linux Mandriva 2010 отключена лишь возможность графического входа в систему под именем root. Другими словами, вы можете переключиться в консоль, нажав <Ctrl>+<Alt>+<F1>, и войти в систему под именем root.

Сейчас мы разберемся, как же войти под именем root в графическом режиме. За регистрацию пользователей в системе в графическом режиме отвечает KDM (KDE Display Manager). Он-то и не пускает пользователя root в систему.

Для изменения поведения KDM нужно открыть его конфигурационный файл. Это нужно сделать с привилегиями root:

```
su
kwrite /etc/kde/kdm/kdmrc (для Mandriva 2008)
kwrite /etc/alternatives/kdm4-config (для Mandriva 2009/2010)
```

В этом файле найдите строку:

```
AllowRootLogin=false
```

Значение директивы `AllowRootLogin` измените на `true`:

```
AllowRootLogin=true
```

После этого можно будет войти в систему под именем root. Кстати, при входе в систему вы получите предупреждение, а фон графического стола станет красным, извещая вас об опасности такого решения.

Вход в качестве root в Fedora

Как и в Mandriva, в Fedora 9 и 10 вход пользователя root ограничен менеджером рабочего стола. Введите команду:

```
su -c 'gedit /etc/pam.d/gdm'
```

Так вы запустите с правами root текстовый редактор gedit для редактирования файла /etc/pam.d/gdm. Найдите в этом файле следующую строку

```
auth required pam_succeed_if.so user != root quiet
```

Закомментируйте ее (поставьте знак `#` перед ней) или вообще удалите эту строку.

В Fedora 11 и 12 дополнительно нужно открыть файл /etc/pam.d/gdm-password и найти следующую строку:

```
pam_succeed_if.so user != root quiet
```

Эту строку тоже нужно или закомментировать или удалить.

Если вы используете вход в систему по отпечатку пальца, тогда откройте файл gdm-fingerprint и закомментируйте в нем следующую строку:

```
pam_succeed_if.so user != root quiet
```

После этого сохраните файлы и завершите сеанс пользователя. После перезагрузки GDM вы сможете войти в систему как root[1].

[1] Просмотреть видеоурок, демонстрирующий разрешение входа root в графическом режиме, можно по адресу: **http://dkws.org.ua/video-lessons/login_as_root_in_fedora_10.avi**.

10.3. Создание, удаление и модификация пользователей стандартными средствами

10.3.1. Команды *adduser* и *passwd*

Для добавления нового пользователя выполните следующие команды (от имени root):

```
# adduser <имя пользователя>
# passwd <имя пользователя>
```

Первая команда (`adduser`) добавляет пользователя, а вторая (`passwd`) изменяет его пароль. Ясно, что и в первом, и во втором случае вы должны указать одно и то же имя пользователя.

В некоторых дистрибутивах, например, в Ubuntu и Debian, сценарий `adduser` не только добавляет пользователя, но позволяет указать дополнительную информацию о пользователе и сразу же задать пароль пользователя (рис. 10.4).

```
root@denis-desktop:/home/denix# adduser denis
Добавляется пользователь `denis' ...
Добавляется новая группа `denis' (1001) ...
Добавляется новый пользователь `denis' (1001) в группу `denis' ...
Создаётся домашний каталог `/home/denis' ...
Копирование файлов из `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for denis
Enter the new value, or press ENTER for the default
        Full Name []: Denis
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Данная информация корректна? [Д/н] y
root@denis-desktop:/home/denix#
```

Рис. 10.4. Добавление нового пользователя в Ubuntu

Примечание

В некоторых дистрибутивах (например, в openSUSE) вместо команды `adduser` используется команда `useradd`. Программы `adduser` и `useradd` обычно находятся в каталоге /usr/sbin.

Обратите внимание — если пароль слишком прост для подбора, программа `passwd` выдаст соответствующее предупреждение: **BAD PASSWORD** и сообщит, чем же наш пароль плохой (в нашем случае в основе пароля лежит словарное слово, что делает пароль легким для подбора).

Примечание

Команду `passwd` может использовать не только администратор, но и сам пользователь для изменения собственного пароля.

10.3.2. Команда *usermod*

Для модифицирования учетной записи пользователя можно использовать команду `usermod`. Формат вызова команды следующий:

```
usermod [параметры] LOGIN
```

Параметры команды `usermod` приведены в табл. 10.1.

Таблица 10.1. *Параметры команды* `usermod`

Параметр	Описание
`-a, --append`	Добавляет пользователя в дополнительную группу. Используется только с параметром `-G`
`-c, --commentКОММЕНТАРИЙ`	Изменяет комментарий пользователя. Комментарий удобнее изменять с помощью команды `chfn`
`-d, --homeКАТАЛОГ`	Изменяет домашний каталог пользователя
`-e, --expiredateДАТА`	Устанавливает дату отключения учетной записи. Дата устанавливается в формате ГГГГ-ММ-ДД
`-f, --inactiveДНИ`	Устанавливает число дней после даты отключения учетной записи, на протяжении которых можно изменить пароль. Если число дней равно 0, то учетная запись блокируется сразу после устаревания пароля
`-g, --gidГРУППА`	Задает идентификатор (число, GID) начальной группы пользователя
`-G, --groupsГ1[,Г2,...[,ГN]]]`	Задает список дополнительных групп, в которые будет входить пользователь. Список групп задается через запятую без дополнительных пробелов. Если пользователь уже является членом группы, которой нет в списке, он будет удален из этой группы. Поведение программы можно изменить, добавив параметр `-a`, в этом случае пользователь просто будет добавлен в дополнительные группы, а если он уже являлся членом других групп, которых нет в списке, то он не будет удален из них
`-l, --loginНОВОЕ_ИМЯ`	Изменяет имя пользователя
`-L, --lock`	Блокирует пароль пользователя. Не используйте этот параметр вместе с параметрами `-p` или `-U`. Параметр `-L` блокирует только вход по паролю, но есть и другие способы аутентификации, поэтому если вам нужно полностью заблокировать учетную запись, используйте параметр `-e`
`-o, --non-unique`	При использовании с параметром `-u` позволяет задавать неуникальный идентификатор пользователя
`-p, --passwordПАРОЛЬ`	Задает новый пароль пользователя. Нужно указать не сам пароль, а его хэш. Поэтому для смены пароля проще использовать команду `passwd`
`-s, --shellОБОЛОЧКА`	Задает новую оболочку пользователя

Таблица 10.1 *(окончание)*

Параметр	Описание
`-u, --uidUID`	Задает идентификатор пользователя. Идентификатор должен быть уникальным, если не задан параметр `-o`. Идентификаторы 0 — 999 зарезервированы для системных учетных записей
`-U, --unlock`	Разблокирует пароль пользователя

10.3.3. Команда *userdel*

Для удаления пользователя используется команда `userdel`:

```
# userdel <имя пользователя>
```

10.3.4. Подробно о создании пользователей

Давайте разберемся, что же происходит при создании новой учетной записи пользователя.

Во-первых, создается запись в файле /etc/passwd. Формат записи следующий:

```
имя_пользователя:пароль:UID:GID:полное_имя:домашний_каталог:оболочка
```

Рассмотрим фрагмент этого файла (две строки):

```
root:x:0:0:root:/root:/bin/bash
den:x:500:500:Denis:/home/den:/bin/bash
```

- первое поле — это логин пользователя, который он вводит для регистрации в системе. Пароль в современных системах в этом файле не указывается, а второе поле осталось просто для совместимости со старыми системами. Пароли хранятся в файле /etc/shadow, о котором мы поговорим чуть позже;
- третье и четвертое поле — это UID (User ID) и GID (Group ID) — идентификаторы пользователя и группы соответственно. Идентификатор пользователя root всегда равен 0, как и идентификатор группы root. Список групп вы найдете в файле /etc/groups;
- пятое поле — это настоящее имя пользователя. Может быть не заполнено, а может содержать фамилию, имя и отчество пользователя — все зависит от педантичности администратора системы, то есть от вас. Если вы работаете за компьютером в гордом одиночестве, то, думаю, свою фамилию вы не забудете. А вот если ваш компьютер — сервер сети, тогда просто необходимо указать Ф. И. О. каждого пользователя, а то, когда придет время обратиться к пользователю по имени, вы его знать не будете (попробуйте запомнить 500 фамилий и имен!);
- шестое поле содержит имя домашнего каталога. Обычно это каталог /home/<имя_пользователя>;
- последнее поле — это имя командного интерпретатора, который будет обрабатывать введенные вами команды, когда вы зарегистрируетесь в консоли.

Пояснение

В целях безопасности пароли были перенесены в файл /etc/shadow (доступен для чтения/записи только пользователю root), где они и хранятся в закодированном виде (используется алгоритм MD5 или Blowfish в некоторых системах). Узнать, с помощью какого алгоритма зашифрован пароль, очень просто: посмотрите на шифр — если он достаточно короткий и не начинается с символа $, то применен алгоритм DES (самый слабый и ненадежный — как правило, используется в старых дистрибутивах). Если же шифр начинается с символов 1, то это MD5, а если в начале шифра имеются символы $2a$, то это Blowfish.

Во-вторых, при создании пользователя создается каталог /home/<имя пользователя>, в который копируется содержимое каталога /etc/skel. Каталог /etc/skel содержит "джентльменский набор" — файлы конфигурации по умолчанию, которые должны быть в любом пользовательском каталоге. Название каталога skel (от skeleton) полностью оправдывает себя — он действительно содержит "скелет" домашнего каталога пользователя.

Примечание

Файл /etc/passwd можно редактировать с помощью обычного текстового редактора. То есть вы можете очень легко, не прибегая к помощи ни графического конфигуратора, ни команды `usermod`, изменить параметры учетной записи любого пользователя, например, задать для него другую оболочку или прописать его настоящую фамилию. Однако нужно быть осторожным при изменении домашнего каталога пользователя! Если вы это сделали, то, чтобы у пользователя не возникло проблем с правами доступа для нового каталога, нужно выполнить команду:

```
chown –R <пользователь> <каталог>
```

10.4. Группы пользователей

Иногда пользователей объединяют в *группы*. Группы позволяют более эффективно управлять правами пользователей. Например, у нас есть три пользователя: igor, pavel, alex, которые должны совместно работать над проектом. Их достаточно объединить в одну группу — тогда пользователи будут иметь доступ к домашним каталогам друг друга (по умолчанию один пользователь не имеет доступ к домашнему каталогу другого пользователя, поскольку пользователи находятся в разных группах).

Создать группу, а также поместить пользователя в группу, позволяют графические конфигураторы. Вы можете использовать их — они очень удобные, но если вы хотите стать настоящим линуксоидом, то должны знать, что доступные в системе группы указываются в файле /etc/group. Добавить новую группу в систему можно с помощью команды `groupadd`, но, как правило, проще добавить в текстовом редакторе еще одну запись в файл /etc/group, а изменить группу пользователя еще проще — для этого достаточно отредактировать файл /etc/passwd.

10.5. Команды квотирования

Квотирование — это механизм ограничения дискового пространства пользователей. Linux — это многопользовательская система, поэтому без ограничения дискового пространства вам не обойтись. Когда используешь компьютер только сам,

то все дисковое пространство доступно вам и только вам. А вот когда пользователей несколько, нужно ограничить доступное пространство, чтобы один из пользователей не "узурпировал" все место на диске. Как именно вы будете ограничивать дисковое пространство, решаете вы — можно поделить дисковое пространство поровну между пользователями, можно одним пользователям отдать больше места, а другим — меньше.

На домашнем компьютере квотирование вряд ли понадобится, а на сервере, как правило, для каталога /home отводится отдельный раздел жесткого диска. Поэтому будем считать, что у нас есть отдельный раздел, который монтируется к каталогу /home.

Перед настройкой квот нужно установить пакет quota. Больше ничего устанавливать не нужно.

Чтобы пользователи не потеряли свои данные, перезагрузитесь в однопользовательский режим (параметр ядра `single`). Вот теперь можно приступать к редактированию квот. Первым делом разрешим устанавливать квоты на разделе, который содержит файлы пользователей. Откройте /etc/fstab:

```
# nano /etc/fstab
```

Далее добавьте параметр `usrquota` к списку параметров раздела:

```
/dev/sda5       /home           ext4    defaults,usrquota       0   2
```

Параметр `usrquota` включает поддержку квот для отдельных пользователей, если вам нужна поддержка квот групп пользователей, тогда добавьте параметр `grpquota`.

Далее перемонтируем /home, так как мы только что изменили его параметры:

```
# mount -o remount /home
```

Механизм квотирования требует создания файлов aquota.user и aquota.group, но поскольку мы не будем устанавливать квоты для групп, а только для пользователей, то создадим только файл aquota.user:

```
# touch /home/aquota.user
# chmod 600 /home/aquota.user
```

Теперь введите команду:

```
# quotacheck -vagum
```

Поскольку мы создали файл aquota.user вручную, то вы увидите сообщение об ошибке, но это только в первый раз — далее все будет нормально:

```
quotacheck: WARNING - Quotafile /home/aquota.user was probably truncated.
Can't save quota settings...
        quotacheck: Scanning /dev/sda5 [/home] quotacheck: Old group file
not found. Usage will not be substracted.
        done
        quotacheck: Checked 3275 directories and 54301 files
```

Теперь отредактируем квоты для пользователя user:

```
# edquota -u user
```

Будет запущен текстовый редактор по умолчанию, и вы увидите следующий текст:

```
Disk quotas for user user (uid 1001):
        Filesystem   blocks      soft       hard     inodes     soft   hard
        /dev/sda5      16          0          0          5        0      0
```

Примечание

По умолчанию используется редактор vi, который, мягко говоря, не очень удобный. Для изменения редактора по умолчанию установите переменную окружения `EDITOR`. Например, `EDITOR=nano`.

Разберемся, что здесь есть что:

- blocks — место в блоках, используемое пользователем (1 блок = 1 Кбайт);
- soft — максимальное дисковое пространство (в блоках по 1 Кбайт), которое может занимать пользователь. Если вы включите период отсрочки (grace period), то пользователь получит только лишь сообщение о превышении квоты;
- hard — жесткое ограничение, эту квоту пользователь превысить не может, даже если включен период отсрочки. Предположим, что вы хотите "отдать" пользователю 500 Мбайт. В качестве жесткой квоты можно установить значение 500 Мбайт (или 500000 блоков), а в качестве "мягкой" квоты установить значение 495 Мбайт (495000 блоков). Когда пользователь превысит 495 Мбйт, он получит сообщение о превышении квоты, а вот когда будет превышена жесткая квота, то пользователь больше не сможет сохранять файлы в своем домашнем каталоге;
- inodes — число используемых пользователем файлов.

Отредактируйте квоты так:

```
Disk quotas for user user (uid 1001):
        Filesystem   blocks     soft       hard     inodes     soft   hard
        /dev/sda5      16       495000     500000     5          0      0
```

Теперь сохраните файл, выйдите из редактора и введите команду:

```
# edquota -t
```

Сейчас мы установим период отсрочки:

```
Grace period before enforcing soft limits for users:
        Time units may be: days, hours, minutes, or seconds
        Filesystem             Block grace period     Inode grace period
        /dev/sda8                     7days                  7days
```

Вы должны вместо `7days` вписать свой период отсрочки, при этом используйте названия единиц изменения время на английском:

- `seconds` — секунды;
- `minutes` — минуты;
- `hours` — часы;
- `days` — дни;
- `weeks` — недели;
- `months` — месяцы.

Например:

- `24hours` — 24 часа;
- `2days` — 2 дня;
- `1weeks` — 1 неделя.

После этого включим квотирование для наших файловых систем:

```
# quotaon файловая_система
```

Например,

```
# quotaon /
```

После этого перезагружаем систему:

```
# reboot
```

При загрузке вы увидите сообщение: **Turning on user and group quotas for local filesystems** (Включаем квоты пользователей и групп для локальных файловых систем) — это означает, что механизм квотирования правильно работает.

Для просмотра квот используется команда `repquota`, например:

```
# repquota /home
```

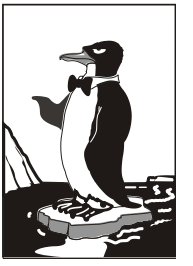
Наверняка использовать редактор vi вам не очень нравится. Значительно упрощают задание квот так называемые *прототипы*. Например, вы задали ограничение для пользователя den. Но у вас есть еще несколько пользователей, для которых нужно задать такие же ограничения. Вы можете использовать квоту пользователя den в качестве прототипа:

```
# edquota -p den user1
# edquota -p den user2
...
```

ЧАСТЬ III

Настройка сети в Linux

После установки системы и создания учетных записей пользователей можно приступить к настройке сети в Linux. Настройка сети может быть произведена как с помощью графических конфигураторов, так и вручную — путем редактирования конфигурационных файлов. Какой вариант выбрать, зависит только от вашего решения и от используемого дистрибутива.



Глава 11

Настройка локальной сети

11.1. Несколько слов о монтаже сети

Существует много сетевых технологий, но в этой книге мы будем рассматривать настройку локальной сети, построенной на технологии Fast Ethernet. Зато мы рассмотрим ее полностью — от обжатия кабеля до конфигурирования сети в Linux. Монтаж Ethernet был подробно описан в *главе 5*, поэтому сейчас мы не будем рассматривать этот процесс подробно. Я лишь напомню основные характеристики стандарта Fast Ethernet:

- скорость передачи данных — 100 Мбит/с;
- метод доступа к среде передачи данных — CSMA/CD;
- среда передачи данных — витая пара UTP 3, 4 или 5-й категории (лучше 5-й), оптоволоконный кабель;
- максимальное количество компьютеров — 1024;
- максимальная длина сети — 200 м (272 м для оптоволокна).

Прежде всего, вам нужно убедиться, что компьютеры, предназначенные для соединения в сеть, оснащены сетевыми адаптерами, поддерживающими технологию Fast Ethernet. Как правило, сейчас сетевые адаптеры интегрированы в материнскую плату и устанавливать их отдельно не нужно. Но встречаются материнские платы и без интегрированных сетевых адаптеров. В этом случае вам нужно их купить. Стоят они очень дешево — от 150 рублей за штуку (рис. 11.1). А за 500–600 рублей можно купить сетевой адаптер, поддерживающий технологию Gigabit Ethernet — модификацию Fast Ethernet, позволяющую передавать данные со скоростью до 1000 Мбит/с. Правда, коммутаторы (см. далее) для Gigabit Ethernet стоят чуть дороже, чем для Fast Ethernet. Но если финансы позволяют, и вы собираетесь строить сеть с нуля, то лучше ориентироваться на Gigabit Ethernet как на более современную технологию. Тем более, что оборудование для Gigabit Ethernet стоит уже дешевле, чем, скажем, год или два назад.

Установка сетевого адаптера проблем не вызывает — просто вставьте ваш сетевой адаптер в свободный разъем шины PCI (все адаптеры Fast Ethernet выполнены как платы расширения именно для шины PCI). Существуют и USB-сетевые адаптеры, позволяющие подключиться к сети, не разбирая компьютер. Но такие адаптеры стоят очень дорого и встречаются пока редко. Тем более, точно не известно, как будет работать Linux с таким вот чудом научно-технического прогресса (рис. 11.2).

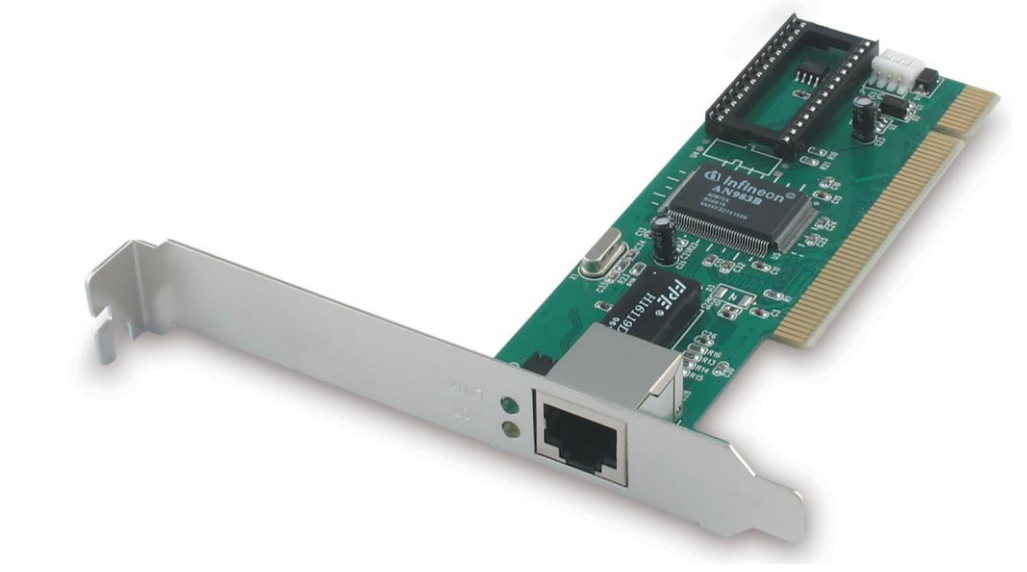


Рис. 11.1. Сетевой адаптер Fast Ethernet

Ясно, что устанавливать сетевой адаптер полагается при выключенном компьютере — шина PCI пока еще не поддерживает "горячей замены". После этого нужно подключить к сетевому адаптеру коннектор сетевого кабеля. Коннекторы крепятся на концах отрезков кабеля (кабель "обжимается"), как правило, администратором сети. Процесс обжима кабеля подробно описан в *главе 5*.

Рис. 11.2. USB-сетевой адаптер

Пояснение

Обжать кабель — значит, особым образом закрепить на его концах специальные наконечники-коннекторы.

Если вы пропустили *главу 3*, то давайте сначала вкратце рассмотрим оборудование, необходимое для создания Ethernet-сети, а затем сразу приступим к настройке сети в Linux. Но *главы 3*, *4* и *5* все же настоятельно рекомендую прочитать.

Для создания сети Fast Ethernet вам нужны следующие устройства:

- сетевые адаптеры — с ними мы уже разобрались;
- коммутатор (switch) — его можно купить в любом компьютерном магазине. Дизайном и количеством портов коммутаторы могут отличаться друг от друга. На рис. 11.3 изображен 24-портовый коммутатор, больше подходящий для корпоративной сети (и внешним видом и возможностью помещения в специальную стойку), нежели для дома. А для домашней сети можно найти более "симпатичное" устройство;

- сетевой кабель (витая пара 5-й категории) — приобретайте именно такой тип кабеля и такой длины, чтобы нормально хватило для соединения каждого компьютера сети с коммутатором;
- коннекторы RJ-45 — таких коннекторов вам понадобится в два раза больше, чем компьютеров, поскольку каждый отрезок кабеля нужно будет обжать с двух концов. Но я рекомендую купить еще несколько лишних штук — если вы будете обжимать кабель впервые, думаю, без брака не обойдется. Не пожалейте пару копеек, а то придется сбегать в магазин еще раз;
- инструмент (специальные обжимные щипцы) для обжимки витой пары — хороший инструмент стоит относительно дорого (примерно как коммутатор), а плохой лучше не покупать. Если не хотите выкладываться, возьмите у кого-нибудь на пару дней.

Рис. 11.3. Коммутатор (switch)

11.2. Файлы конфигурации сети в Linux

Прежде чем приступить к настройке сети, следует ознакомиться с файлами конфигурации сети, которые имеются в любом дистрибутиве Linux, вне зависимости от его версии (табл. 11.1).

Таблица 11.1. *Общие файлы конфигурации сети в Linux*

Файл	Описание
/etc/aliases	База данных почтовых псевдонимов. Формат этого файла очень прост: `псевдоним пользователь`
/etc/aliases.db	Системой на самом деле используется не файл /etc/aliases, а файл /etc/aliases.db, который создается программой newaliases по содержимому файла /etc/aliases. Поэтому после редактирования этого файла не забудьте выполнить от имени root команду `newaliases`
/etc/hosts.conf	Содержит параметры разрешения доменных имен. Например, директива `order hosts,bind` означает, что сначала поиск IP-адреса по доменному имени будет произведен в файле /etc/hosts, а затем лишь будет произведено обращение к DNS-серверу, заданному в файле /etc/resolv.conf Директива `multi on` означает, что одному доменному имени могут соответствовать несколько IP-адресов

Таблица 11.1 *(продолжение)*

Файл	Описание
/etc/hosts	В этом файле можно прописать IP-адреса и имена узлов локальной сети, но обычно здесь указывается только IP-адрес узла localhost (127.0.0.1), потому что сейчас даже в небольшой локальной сети устанавливается собственный DNS-сервер
/etc/hosts.allow	Содержит IP-адреса узлов, которым разрешен доступ к сервисам данного узла
/etc/hosts.deny	Содержит IP-адреса узлов, которым запрещен доступ к сервисам данного узла
/etc/hostname	В Debian/Ubuntu содержит имя узла
/etc/iftab	Содержит таблицу интерфейсов, то есть соответствие имен интерфейсов и их MAC-адресов
/etc/motd	Файл задает сообщение дня (Message of the day). Данный файл используется многими сетевыми сервисами (например, FTP- и SSH-серверами), которые при регистрации пользователя могут выводить сообщение из этого файла
/etc/network/interfaces	В Debian и Ubuntu используется для ручной настройки сетевых интерфейсов (то есть не с помощью NetworkManager). Вообще принято настраивать сетевые интерфейсы с помощью NetworkManager, но некоторые администраторы предпочитают отключать NetworkManager и настраивать сетевые интерфейсы вручную — по старинке
/etc/rc.config	В openSUSE содержит имя компьютера, IP-адрес интерфейса и другую сетевую информацию
/etc/resolv.conf	Задает IP-адреса серверов DNS. Формат файла прост: `nameserver IP-адрес` Всего можно указать четыре DNS-сервера. В Ubuntu этот файл автоматически перезаписывается при установке соединения с Интернетом — сюда записываются адреса DNS-серверов, полученных от провайдера, что не совсем хорошо, особенно, когда вы настроили собственный DNS-сервер и желаете его использовать[1]
/etc/route.conf	В старых версиях SUSE данный файл содержит описание статических маршрутов, в том числе и маршрут по умолчанию *(см. главу 14)*
/etc/services	База данных сервисов, задающая соответствие символьного имени сервиса (например, pop3) и номера порта (110/tcp, tcp — это наименование протокола)
/etc/sysconfig/network	Параметры сетевого интерфейса в Fedora, Red Hat и других дистрибутивах, основанных на Fedora/Red Hat, например, ASP Linux, Mandriva

[1] О моей борьбе с перезаписью этого файла можно прочитать статью по адресу: **http://www.dkws.org.ua/index.php?page=show&file=a/ubuntu/static-dns-ubuntu9**.

Таблица 11.1 *(окончание)*

Файл	Описание
/etc/sysconfig/static-routes	Статические маршруты в Fedora/CentOS/ASP Linux
/etc/sysconfig/network/routes	Статические маршруты в современных версиях openSUSE *(см. главу 14)*
/etc/sysconfig/network-scripts/ifcfg-имя	Параметры конкретного сетевого интерфейса, например, параметры интерфейса eth0 хранятся в файле /etc/sysconfig/network-scripts/ifcfg-eth0 (дистрибутив Fedora)
/etc/sysconfig/network/ifcfg-имя	Параметры конкретного сетевого интерфейса (имя — имя сетевого интерфейса). Дистрибутив openSUSE
/etc/xinetd.conf	Файл конфигурации суперсервера xinetd, предназначенного для запуска сетевых сервисов, которые не работают в автономном режиме

11.3. Настройка сети с помощью конфигуратора

Настроить сеть в Linux можно за несколько минут. Ведь в большинстве случаев ваш сетевой адаптер поддерживается ядром, поэтому для настройки сети достаточно лишь указанной здесь командой запустить соответствующий конфигуратор:

- `drakconnect` — в Linux Mandriva;
- `system-config-network` — в Fedora и ASP Linux;
- `network-admin` — в старых версиях Debian и Ubuntu;
- `netconfig` — Slackware.

> ***Примечание***
>
> В новых версиях Ubuntu, Denix и Fedora используется конфигуратор NetworkManager.

А если в вашей сети организован DHCP-сервер, то настраивать сеть в современных дистрибутивах вовсе не придется — Linux автоматически распознает ваш адаптер, активирует соответствующие модули ядра и установит сетевые параметры, полученные от DHCP-сервера. Настраивать сеть придется в двух случаях:

- если у вас небольшая сеть, использующая статические IP-адреса — ради всего 2–3 компьютеров вы не стали настраивать DHCP-сервер;
- если вы настраиваете сеть "с нуля", и компьютер, на который вы установили Linux, как раз и будет тем DHCP-сервером, который потом станет настраивать остальные узлы сети.

> ***Примечание***
>
> Даже если у вас небольшая домашняя сеть из 2–3 компьютеров, совсем не обязательно отсутствие DHCP-сервера. Часто DHCP-сервер "крутится" на точке доступа Wi-Fi или на ADSL-модеме, совмещающем также и функции коммутатора. Современные сетевые устройства позволяют существенно снизить стоимость монтажа сети, осо-

бенно сети домашней. Так, вы можете купить точку доступа с четырьмя Ethernet-портами (к которым могут подключаться стационарные компьютеры) и ADSL-модемом. По сути, это единственное устройство обеспечивает все необходимые функции: ноутбуки будут подключаться по Wi-Fi, стационарные компьютеры — к встроенным портам Ethernet, а само подключение к Интернету будет происходить через встроенный ADSL-модем.

Вот только на предприятии от подобных устройств толку мало, разве что в самых небольших офисах, поскольку количество Ethernet-портов редко превышает 4, чего явно недостаточно для предприятия. Поэтому понадобятся дополнительные устройства — как минимум еще один коммутатор для подключения остальных компьютеров.

11.3.1. Настройка сети в Linux Mandriva

Настройку сети мы будем производить на примере последней версии Mandriva — 2010.1 Spring. В принципе, конфигуратор настройки сети практически не изменился, поэтому все иллюстрации будут также соответствовать и предыдущей версии — 2010.0. Перед началом настройки убедитесь, что сетевой кабель подключен и что запущен сервис network, обеспечивающий поддержку сети.

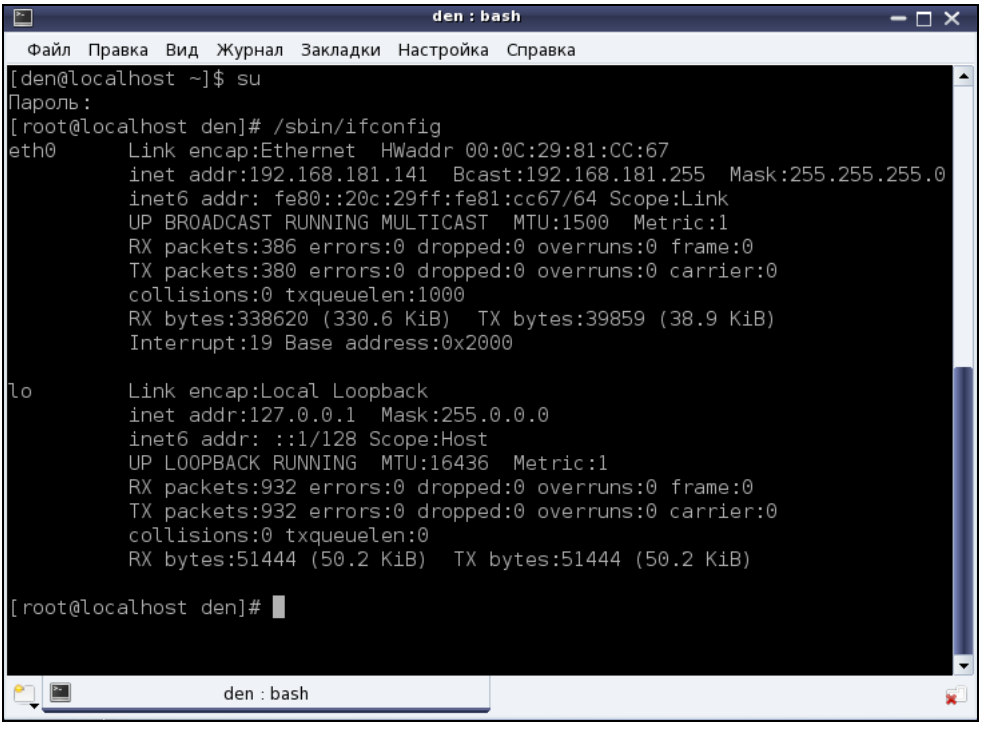


Рис. 11.4. Вывод команды `ifconfig`

Убедиться в этом очень просто — достаточно от имени пользователя root выполнить команду `/sbin/ifconfig`. Если в выводе команды вы увидите информацию об интерфейсе lo — все нормально (рис. 11.4).

Пояснение

Интерфейс lo — это интерфейс обратной петли, использующийся преимущественно для тестирования поддержки сети.

Можно также запустить от имени root конфигуратор drakxservices и убедиться, что сервис network запущен.

Если интерфейса lo нет в выводе программы, значит, вам нужно запустить сервис network:

```
# service network start
```

Для настройки локальной сети запустите конфигуратор drakconnect (рис. 11.5) и выберите тип соединения **Проводная связь (Ethernet)**[2].

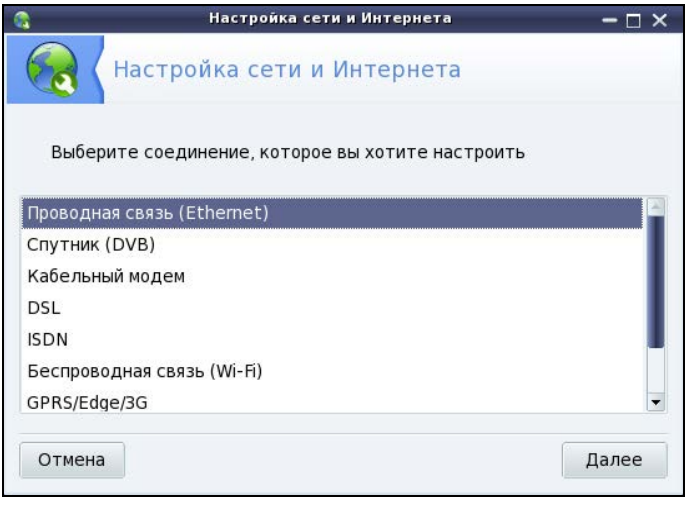


Рис. 11.5. Конфигуратор drakconnect — создание соединения по локальной сети

Конфигуратор предложит вам выбрать устройство, которое будет использоваться для этого соединения, попросту говоря — сетевую плату (рис. 11.6). Если в вашем компьютере несколько сетевых плат, нужно выбрать именно ту, к которой подсоединен сетевой кабель, ведущий к сети, подключение к которой вы хотите настроить.

Примечание

Если вы заметили, то до этого момента ничего не было сказано ни о моделях сетевых плат, ни о поддержке сетевых плат операционной системой. А дело в том, что Linux поддерживает **практически** все сетевые платы. Во всяком случае, неподдерживаемая сетевая плата мне еще не попадалась.

Впрочем, на ноутбуке Acer E525 сетевой адаптер в Linux не определяется, соответственно, к Интернету вы сразу подключиться не сможете. Чтобы исправить данную проблему, нужно установить дополнительный драйвер, скачать который можно по адресу: **http://partner.atheros.com/Drivers.aspx**.

[2] В предыдущей версии Mandriva — **Ethernet**, в еще более древней версии — **Соединение по локальной сети**.

Для установки драйвера введите в терминале следующие команды:

```
tar -xvzf AR81Family-linux-v1.0.0.10.tar.gz
cd src
make
sudo make install
sudo modprobe atl1e
```

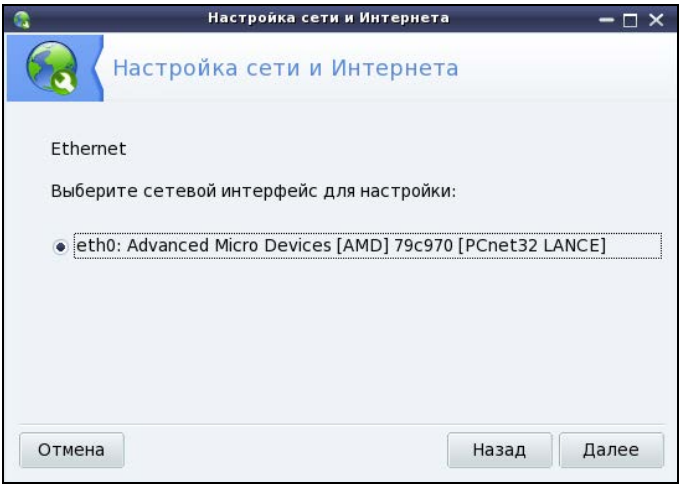


Рис. 11.6. Выбор сетевой платы

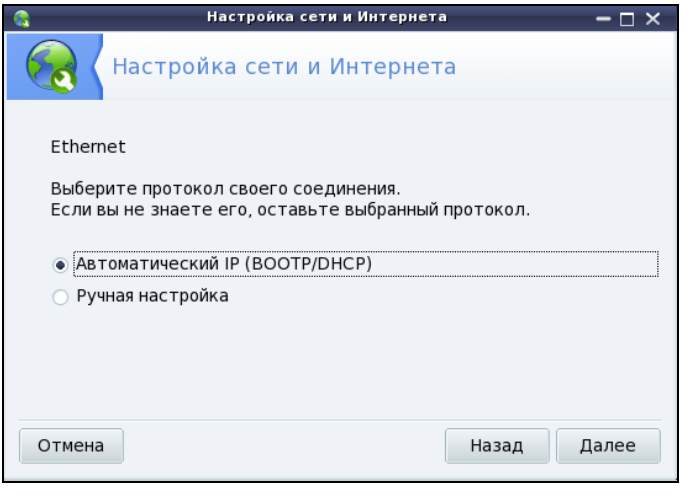


Рис. 11.7. Автоматическая или ручная настройка?

Следующий этап — выбор типа настройки (рис. 11.7): автоматический (с помощью DHCP) или ручной — в этом случае параметры TCP/IP вам нужно будет ввести вручную. Выбирать наугад не нужно — уточните тип настройки у администра-

тора. Если в вашей сети развернут DHCP-сервер, то никаких параметров сети вам вводить не понадобится — в общем, на этом настройка вашей сети и закончится, поэтому далее мы будем рассматривать именно ручное конфигурирование сети.

> ***Примечание***
>
> Если вы выберете автоматическую настройку, то конфигуратор предложит вам изменить только параметры DNS: имя компьютера и IP-адреса DNS-серверов. Эту информацию можно или ввести вручную или получить от DHCP (конфигуратор допускает выбор любого варианта — на ваше усмотрение).

Получите у системного администратора значения параметров сети (IP-адрес сетевого интерфейса, маску сети, IP-адрес шлюза и адреса DNS-серверов), введите IP-адрес сетевого интерфейса и проверьте предложенную конфигуратором маску сети (рис. 11.8). Нужно отметить, что конфигуратор сам пытается вычислить маску сети по введенному IP-адресу и в большинстве случаев у него это получается. В этом же окне можно ввести IP-адрес шлюза (если он есть в вашей сети), а также IP-адреса серверов DNS. В самом нижнем поле следует ввести имя узла (хоста).

Обратите внимание: что угодно вводить нельзя — имя узла должно быть зарегистрировано на DNS-сервере вашей сети.

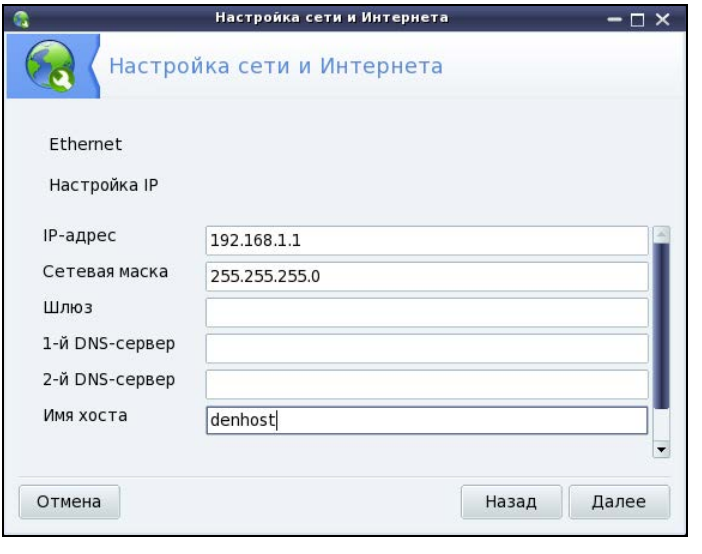


Рис. 11.8. Параметры TCP/IP

Если вы сам себе администратор, тогда для настройки локальной сети вы можете использовать следующие параметры:

- IP-адреса в диапазоне: 192.168.0.1–192.168.0.254;
- маска сети: 255.255.255.0 (сеть класса C);
- IP-адрес шлюза равен IP-адресу компьютера, подключенного к Интернету;
- если вы настраиваете шлюз, то есть компьютер, который будет предоставлять доступ к Интернету другим компьютерам сети, то в его настройках IP-адрес

шлюза указывать не нужно, а в качестве DNS-серверов можно указать IP-адрес этого компьютера (если вы планируете настройку собственного DNS-сервера) или IP-адреса DNS-серверов провайдера;

- имена узлов можно установить любые — главное, чтобы эти имена были уникальными (как и IP-адреса). Далее можно или настроить сервер DNS или, если сеть небольшая, прописать соответствие IP-адресов именам компьютеров в файле /etc/hosts. После редактирования этого файла (а редактировать его можно как в любом текстовом редакторе, так и с помощью конфигуратора сети) его нужно скопировать на все компьютеры сети.

Если вы хотите, чтобы соединение устанавливалось при загрузке системы (в большинстве случаев желательно, чтобы это было так), установите соответствующий флажок (рис. 11.9). Разрешать управлять соединением другим пользователям не стоит — ведь это соединение по локальной сети. Другое дело — модемное или DSL-соединение, которое нужно включать и останавливать по нескольку раз в день. Можно также включить подсчет трафика, а просмотреть информацию о трафике можно будет через средство мониторинга сети. Кстати, конфигуратор позволяет включить подсчет трафика и для автоматически настраиваемого интерфейса.

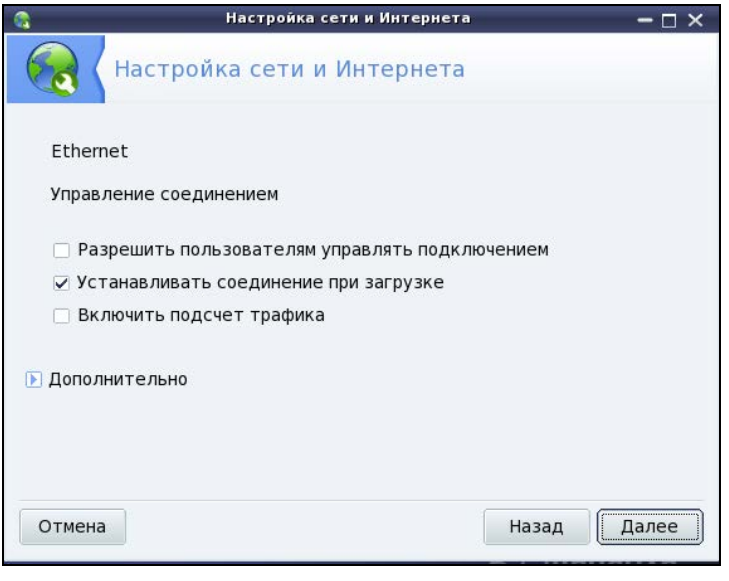


Рис. 11.9. Дополнительные параметры соединения

Далее конфигуратор предложит вам запустить созданное соединение — соглашайтесь. Все! При успешном "поднятии" сети (или автоматически или вручную), вы увидите сообщение, подобное изображенному на рис. 11.10.

Сеть настроена, можно приступить к тестированию ее работы, то есть проверить правильность настроек.

Прежде всего убедимся, что интерфейс eth0 (это ваша первая сетевая плата) поднят (то есть включен и работает нормально). Введите команду:

```
ifconfig
```

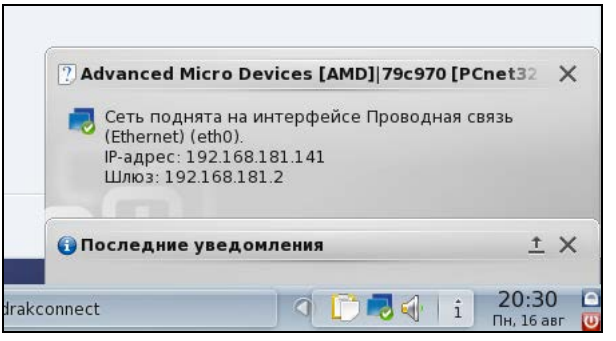


Рис. 11.10. Сеть успешно поднята по DHCP: компьютеру присвоен IP-адрес 192.168.181.141

В ее выводе (рис. 11.11) вы увидите информацию об интерфейсе eth0 (а также о других активных интерфейсах). Здесь же вы можете узнать IP-адрес интерфейса, маску сети, аппаратный MAC-адрес сетевой платы (HWaddr), количество принятых и переданных байтов (RX и TX соответственно).

> ***Внимание!***
>
> Если вы изменили имя узла, то нужно перезагрузить компьютер (командой `reboot`) или, хотя бы, X.Org во избежание проблемы с графической подсистемой X.Org, которая не сможет нормально работать после изменения имени компьютера. Для перезапуска X.Org нужно завершить сеанс пользователя и снова войти в систему. Можно использовать также комбинацию клавиш <Ctrl>+<Alt>+<Backspace>, но это решение грубое и больше подходит для аварийного завершения X.Org в случае его зависания.

Итак, мы убедились, что интерфейс eth0 поднят, теперь пропингуем[3] свой узел по IP-адресу (рис. 11.12):

```
# ping 192.168.1.1
```

Для завершения работы программы ping нажмите комбинацию клавиш <Ctrl>+<C>.

Если ошибок не случилось, можно пропинговать удаленный узел, например, ваш шлюз. Если произойдет ошибка при попытке пропинговать удаленный узел, это еще не означает, что ваш компьютер сконфигурирован неверно — вполне может быть, что удаленный компьютер просто выключен.

Напоследок пропингуйте узел, находящийся за пределами вашей сети:

```
# ping www.mail.ru
```

Этим вы убьете сразу двух зайцев. Во-первых, убедитесь, что работает служба DNS — ведь перед тем, как пинговать, системе нужно получить IP-адрес удаленного узла. Во-вторых, увидите, что маршрутизация нормально работает, и у вас есть доступ к Интернету. Если же пропинговать удаленный узел не удалось, вот наиболее вероятные причины сбоя:

- ❒ вы ошиблись при указании сетевых параметров — проверьте их;
- ❒ вы указали неправильный IP-адрес или имя компьютера — проверьте его;

[3] Пропинговать — послать на проверяемый адрес специальный тестовый сигнал (ping).

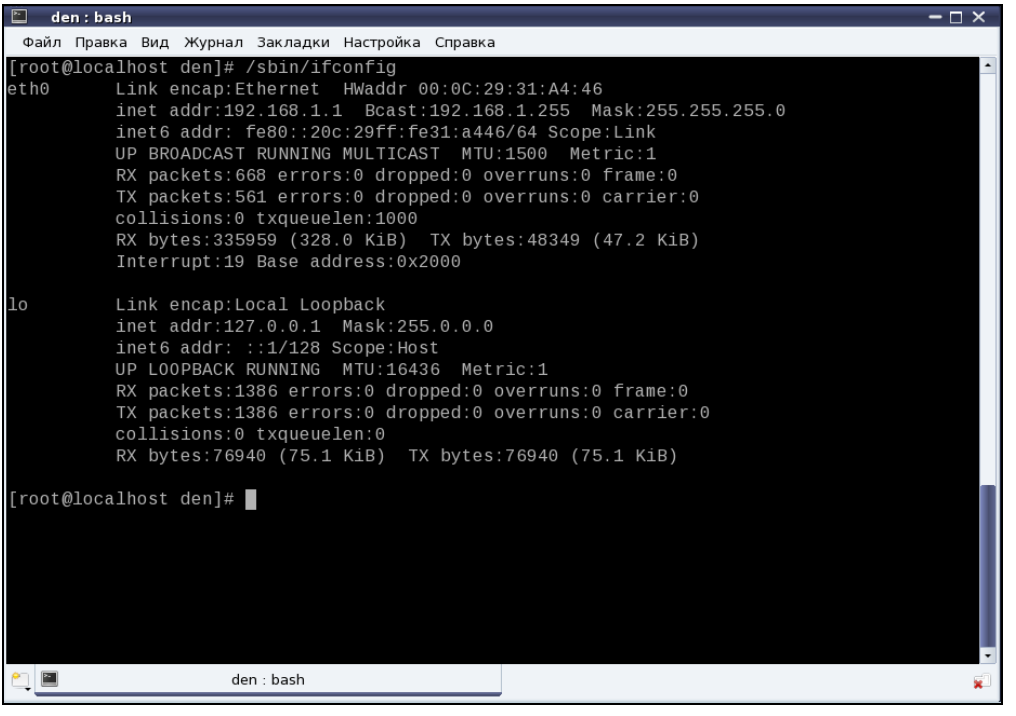


Рис. 11.11. Информация об интерфейсе eth0

```
[root@localhost den]# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.66 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.171 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.115 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2652ms
rtt min/avg/max/mdev = 0.115/0.651/1.669/0.720 ms
[root@localhost den]#
```

Рис. 11.12. Пингуем адрес 192.168.1.1: сразу после настройки сети

- удаленный компьютер просто выключен или временно недоступен, например, из-за сбоя интернет-канала, по которому удаленный компьютер подключается к Всемирной сети (такое бывает чаще, чем можно предположить);
- в вашей сети не настроен или не работает шлюз.

Пояснение

Дело в том, что когда пакет адресуется компьютеру, находящемуся за пределами локальной сети, он посылается на шлюз, а уже потом шлюз передает его удаленному компьютеру. Если сеть настраивали не вы, вполне вероятно, что шлюз уже настроен администратором сети, и вы сразу получите доступ к Интернету — вам нужно лишь правильно указать параметры сети. А вот если вы сам себе администратор, то вам нужно настроить брандмауэр на шлюзе (компьютере, подключенном к Интернету) так, чтобы он предоставлял другим компьютерам локальной сети доступ к Интернету.

Других причин недоступности удаленного компьютера не должно быть, если исключить неисправность сетевого оборудования.

Изменить параметры сетевого интерфейса можно с помощью конфигуратора drakconf, для запуска которого нажмите комбинацию клавиш <Alt>+<F2> и введите команду:

```
drakconf
```

После этого перейдите в раздел **Сеть и Интернет** и выберите конфигуратор для изменения параметров сетевого интерфейса (**Настройка сетевого интерфейса**). Там же вы найдете и конфигуратор для удаления сетевых интерфейсов. Как видите, все просто.

11.3.2. Настройка сети в Fedora

Последовательность действий по настройке сети в Fedora такая же, как и в Linux Mandriva, только используются другие конфигураторы. Первым делом командой `/sbin/ifconfig` убедитесь, что подключен сетевой кабель и активен интерфейс lo. С другой стороны, не припомню, чтобы на работающей Linux-машине интерфейс lo не был бы активен.

> ***Примечание***
>
> Честно говоря, не помню, в какой версии Fedora появился диспетчер сети NetworkManager (кажется, в 9-й). Поначалу эта программа "глючила" так, что многие администраторы попросту отказывались от нее и настраивали сеть вручную с помощью конфигуратора system-config-network. Позже конфигуратор NetworkManager (впрочем, это не обычный конфигуратор в прямом смысле слова — это системная служба в сочетании с графическим интерфейсом настройки сети) появился в других дистрибутивах, в частности, в Ubuntu.
>
> Сейчас вроде бы NetworkManager работает вполне достойно, но ради экономии количества страниц в книге (значит, и ваших денег!) мы рассмотрим конфигуратор system-config-network, которым также можно настроить сеть в Fedora (и не только в последних версиях, но и в самых ранних). Мы также узнаем, как отключить NetworkManager в Fedora, если у вас с ним возникнут проблемы. А вот с самим NetworkManager мы познакомимся на примере дистрибутива Ubuntu/Denix *(см. разд. 11.3.3)* — он там такой же, как и в Fedora.
>
> Для предельной точности отмечу, что в книге рассматриваются последние версии дистрибутивов Fedora 13 и Ubuntu 10.04. Я специально указываю номера версий, чтобы не получать от читателей письма примерно такого содержания: "А почему в таком-то окне такая-то кнопка называется так, а на рисунке — иначе?" А все потому, что по непонятным мне причинам разработчики программ для Linux частенько любят менять названия всевозможных кнопок, хотя внешний вид окон и действия кнопок остаются теми же.
>
> И еще одно: конфигуратор system-config-network используется только для задания статического IP-адреса. Впрочем, Fedora, как и любой другой дистрибутив, отлично дружит с DHCP, поэтому в случае его наличия вообще не придется запускать какой-либо конфигуратор для настройки сети.

Итак, приступим к рассмотрению system-config-network. Введите команду:

```
# system-config-network
```

Откроется окно конфигуратора сети (рис. 11.13). Если соединение по локальной сети уже у вас создано (что происходит при загрузке), выделите его и нажмите кнопку **Изменить**. После чего установите параметры сети.

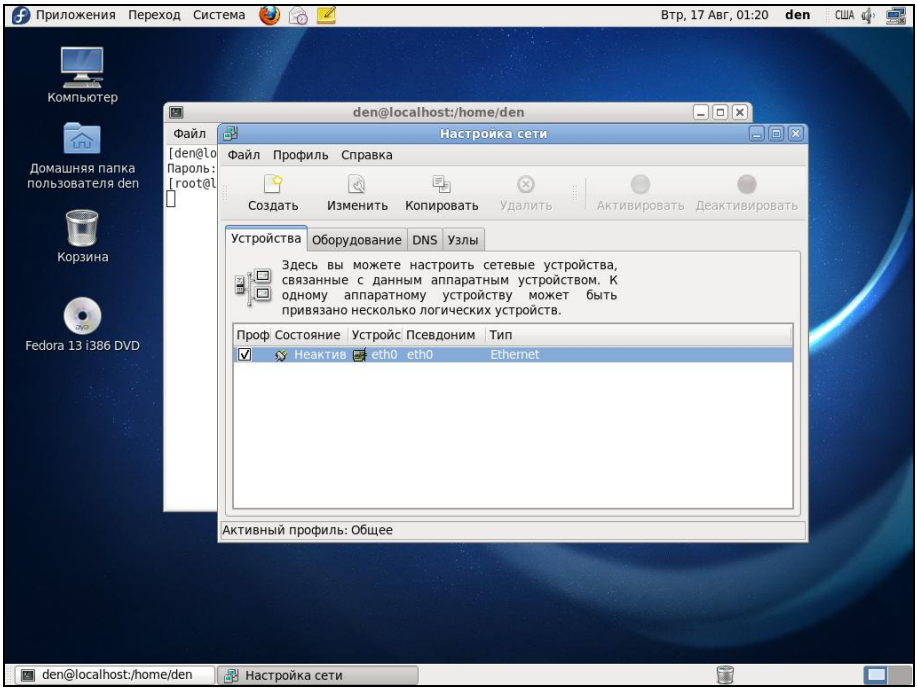


Рис. 11.13. Окно **Настройка сети**

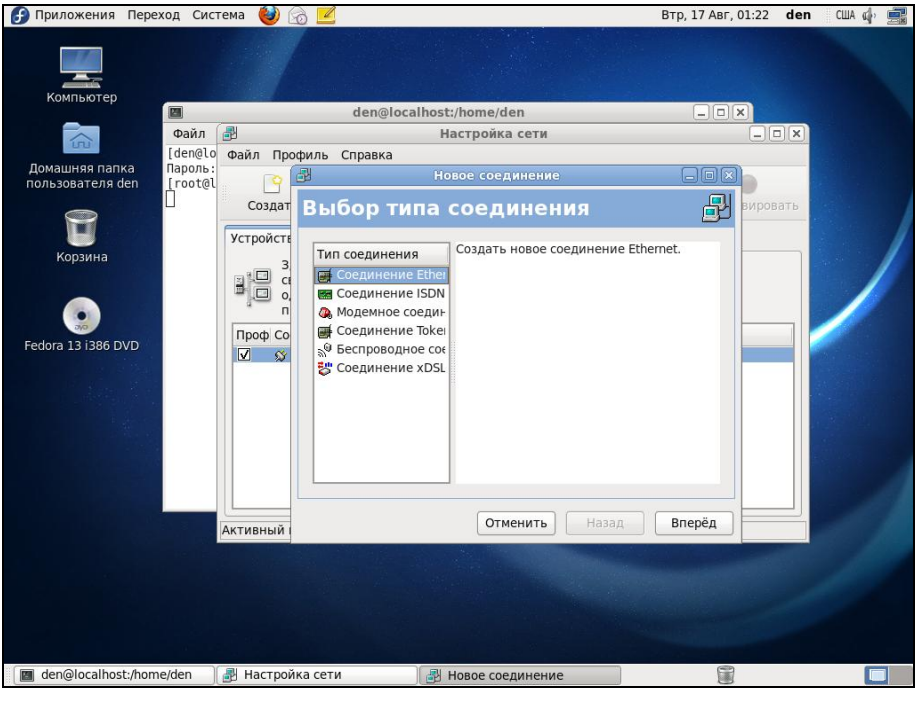


Рис. 11.14. Создание Ethernet-соединения

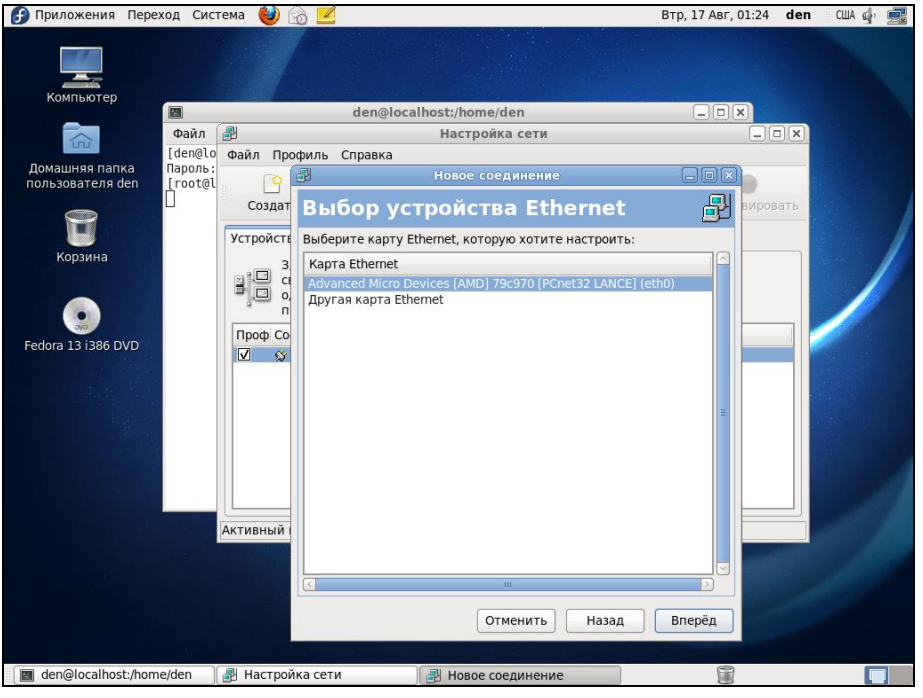


Рис. 11.15. Выбор сетевой платы

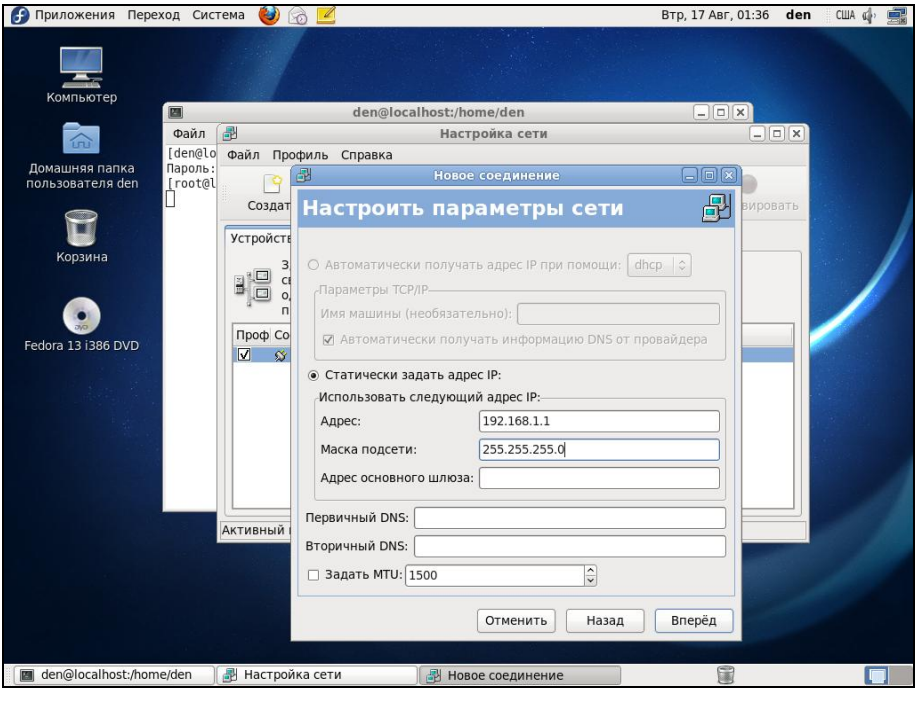


Рис. 11.16. Ввод параметров сети

Если же соединений в окне конфигуратора сети нет, нажмите кнопку **Создать**, а затем выберите **Соединение Ethernet** и нажмите кнопку **Вперед** (рис. 11.14).

Следующий шаг — это выбор сетевой платы (рис. 11.15). Выделите сетевую плату, через которую осуществляется настраиваемое соединение с сетью. Если у вас всего одна сетевая плата, просто подтвердите выбор.

Теперь введите параметры сети: IP-адрес, маску сети и IP-адрес шлюза по умолчанию (рис. 11.16).

На этом настройка сетевого интерфейса завершена. Проверьте введенные вами данные и, если все правильно, нажмите кнопку **Вперед** — откроется основное окно **Настройка сети** конфигуратора сети system-config-network, в котором будет отображен только что созданный вами интерфейс.

Сразу после настройки сетевой интерфейс неактивен. Нажмите кнопку **Активировать** для его активации.

> ***Примечание***
>
> Кнопки **Активировать** и **Деактивировать** станут активными только, если сервис NetworkManager отключен! Ведь именно он по умолчанию управляет сетевыми настройками. А после его отключения "бразды правления" передаются конфигуратору system-config-network. Спрашивается, а почему я тогда вообще рассматриваю system-config-network, если он устарел? Тому есть три причины. Первая — если вам нужно задать статический адрес. Вторая — если у вас старый дистрибутив Fedora. Третья — если у вас в сети есть DHCP-сервер, и вы хотите использовать NetworkManager, то вам вообще ничего не нужно делать. Только подключите сетевой кабель — и все.

Изменить параметры интерфейса можно, нажав кнопку **Изменить**. В открывшемся окне вы сможете переназначить различные параметры сети, в том числе выбрать использование протокола DHCP для автоматического конфигурирования интерфейса.

Конфигуратор drakconf позволяет установить параметры DNS сразу при конфигурировании каждого сетевого интерфейса. С одной стороны — это удобно. С другой — несколько неправильно, потому что установки DNS общие для всех интерфейсов. Если вы зададите одни параметры DNS при настройке одного интерфейса и совершенно другие параметры DNS при настройке другого интерфейса, последние указанные параметры перезапишут параметры, заданные ранее. Разработчики Fedora поступили правильно — они вынесли параметры DNS на отдельную страничку конфигуратора (рис. 11.17). Теперь ясно, что параметры одни для всех, а не разные для каждого интерфейса, как можно было подумать в Linux Mandriva.

На вкладке **DNS** (см. рис. 11.17) вы можете установить имя локального узла, IP-адреса трех серверов DNS (при непосредственной правке файла /etc/resolv.conf можно записать четыре директивы `nameserver`), а также указать путь поиска домена (это директива `search`).

Вкладка **Узлы** (рис. 11.18) предоставляет вам возможность редактирования файла /etc/hosts, в котором хранятся соответствия IP-адресов доменным именам. В данный файл для ускорения процесса разрешения доменного имени (хотя правильнее для этого использовать кэширующий DNS-сервер) можно внести IP-адреса, к которым вы обращаетесь чаще всего, например, **www.mail.ru**, **www.google.com** и т. д. Только не забывайте со временем обновлять эту информацию, поскольку IP-адреса могут периодически меняться.

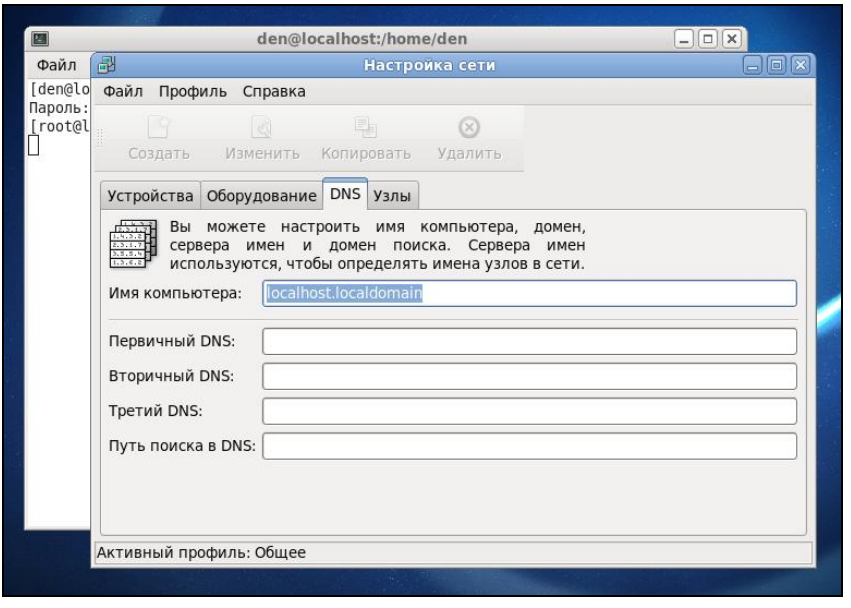


Рис. 11.17. Редактирование параметров DNS

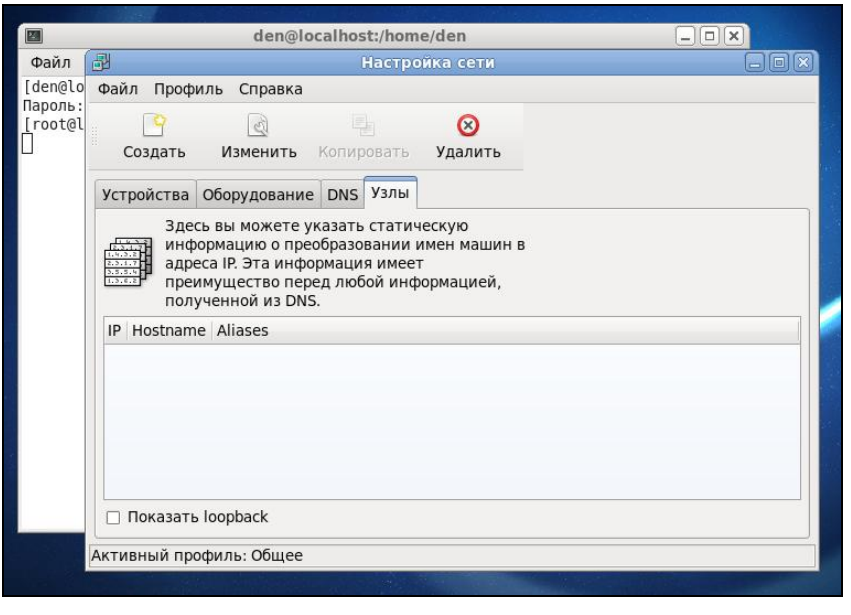


Рис. 11.18. Редактирование файла /etc/hosts

Для добавления записи в файл /etc/hosts нажмите в окне кнопку **Создать**. Откроется небольшое окошко, в котором нужно будет ввести IP-адрес узла, его доменное имя и псевдоним (обычно — сокращенное имя). Например, если имя узла **den.mycompany.com.ru**, то сокращенное имя можно установить типа `den`.

Настало время проверить работу сетевого интерфейса. Для этого сначала введем команду `ifconfig`, чтобы убедиться, что сетевой интерфейс активен, а затем пропингуем сетевой интерфейс по его адресу, который вы узнаете из вывода `ifconfig` — хотя и так должны его помнить, ведь вы только что настраивали сеть!

Как уже отмечалось, в Fedora 12 и 13 сервис NetworkManager работает без особых нареканий. А вот в Fedora 9 и 10 мой сетевой интерфейс отказывался подниматься до тех пор, пока я не отключил NetworkManager и не вернулся к старому доброму сервису network (кстати, в Mandriva 2010.1 сервис NetworkManager отсутствует, а до сих пор используется по умолчанию сервис network — наверное, не зря). Отключить NetworkManager и включить сервис network можно следующими командами (возможно, они вам пригодятся):

```
# /etc/init.d/NetworkManager stop
# /sbin/chkconfig --level 235 NetworkManager off
# /etc/init.d/network start
Bringing up loopback interface:                    [  OK  ]
Bringing up interface eth0:                        [  OK  ]
Bringing up interface isp:                         [  OK  ]
# /sbin/chkconfig --level 235 network on
```

Приведенные команды:

- останавливают сервис NetworkManager;
- отключают NetworkManager на уровнях запуска 2, 3 и 5;
- запускают сервис network;
- включают сервис network на уровнях запуска 2, 3 и 5.

11.3.3. Настройка сети в Debian, Ubuntu и Denix. Конфигураторы nm-connection-editor (NetworkManager) и network-admin

В старых версиях Ubuntu (кажется, до версии 8.10) и Debian для настройки используется конфигуратор network-admin, запустить который можно так:

```
sudo network-admin
```

Хотя в этом случае я предпочитаю редактировать файл /etc/network/interfaces вручную (чуть позже я приведу ссылку на свою статью с подробным описанием этого файла).

В новых версиях Ubuntu и Denix (это мой собственный дистрибутив на базе Ubuntu; а книга — лучшее средство "пропиарить" свое детище) используется конфигуратор nm-connection-editor (NetworkManager Connection Editor, редактор соединений NetworkManager), запустить который можно или командой меню **Система | Параметры | Сетевые соединения** или командой:

```
sudo nm-connection-edtior
```

Да, вы правильно догадались — данный конфигуратор является графическим интерфейсом для сервиса NetworkManager. Точно такой же конфигуратор используется в Fedora.

Конфигуратор nm-connection-editor позволяет настроить Ethernet-соединения, беспроводные соединения (Wi-Fi), мобильные широкополосные соединения (GPRS/EDGE/3G), VPN (виртуальную частную сеть) и DSL-соединения. Нужно отметить, что этот конфигуратор намного лучше старого network-admin.

> ***Внимание!***
>
> Если у вас используется DHCP-сервер, то вообще ничего не нужно настраивать — все будет настроено автоматически (в том числе и для Wi-Fi-соединения). В крайнем случае для Wi-Fi придется ввести пароль доступа, если, конечно, система корректно распознала ваш Wi-Fi-адаптер.

После запуска конфигуратора (рис. 11.19) вы увидите список созданных сетевых интерфейсов. Если нужно установить какие-то определенные параметры интерфейса, выделите интерфейс и нажмите кнопку **Изменить**.

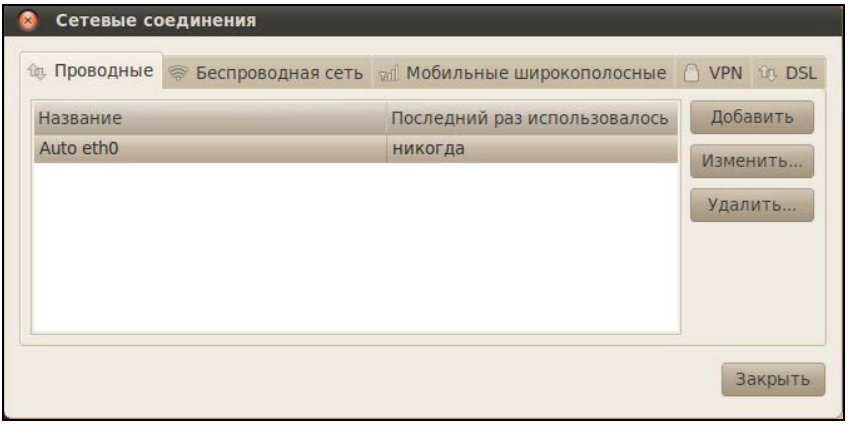


Рис. 11.19. Конфигуратор nm-connection-editor

Рассмотрим окно изменения параметров сетевого интерфейса (рис. 11.20):

- на вкладке **Проводные** можно просмотреть (и даже изменить — в случае необходимости) MAC-адрес сетевого интерфейса и изменить MTU (Maximum Transfer Unit);
- вкладка **Защита 802.1x** используется для задания специальных опций защиты интерфейса (используется редко);
- на вкладке **Параметры IPv4** можно изменить сетевые параметры, относящиеся к протоколу IPv4. Чтобы задать статический IP-адрес, выберите метод **Вручную**, затем нажмите кнопку **Добавить** и добавьте IP-адрес;
- Ubuntu поддерживает концепцию VLAN, позволяющую одному сетевому интерфейсу присвоить несколько IP-адресов. Если вы хотите использовать DHCP, но хотите указать свои DNS-серверы, то выберите метод **Автоматически (DHCP, только адрес)**;
- на вкладке **Параметры IPv6** можно указать параметры, относящиеся к протоколу IPv6, если вы таковой используете.

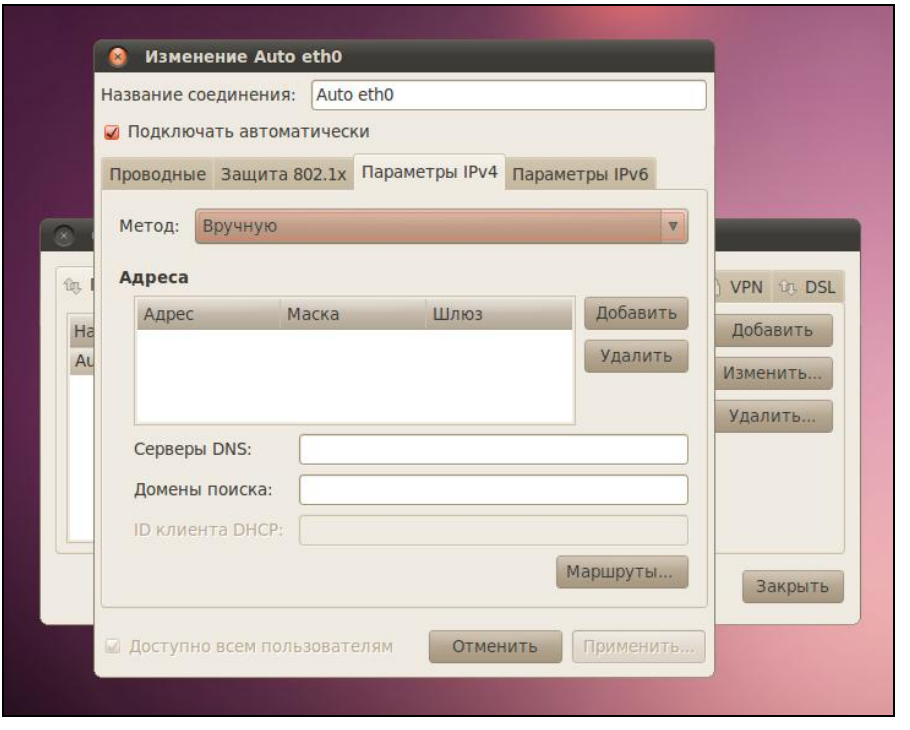


Рис. 11.20. Изменение параметров сетевого интерфейса

Если в вашей сети нет DHCP-сервера, выполняющего автоматическую настройку рабочих станций, тогда перейдите на вкладку **Параметры IPv4** и выберите конфигурацию **Вручную.** После этого введите свой IP-адрес, маску сети и IP-адрес шлюза (gateway). Всю эту информацию вы сможете узнать у администратора сети.

Как уже отмечалось, Ubuntu поддерживает технологию VLAN (Virtual LAN), что позволяет одному сетевому адаптеру назначить несколько IP-адресов. На практике данная возможность используется редко, но вы должны знать, что поддержка VLAN в Ubuntu есть. Дополнительную информацию о VLAN можно получить в следующих моих статьях:

- **http://www.xakep.ru/magazine/xa/121/122/1.asp**;
- **http://www.dkws.org.ua/index.php?page=show&file=a/ubuntu/network-interfaces**.

В заключение этого раздела приведу несколько полезных ссылок:

- **http://dkws.org.ua/index.php?page=show&file=a/ubuntu/static-ip-ubuntu9** — если у вас возникнут проблемы с установкой статического IP-адреса;
- **http://dkws.org.ua/index.php?page=show&file=a/ubuntu/static-dns-ubuntu9** — как установить вручную IP-адрес DNS-сервера в Ubuntu 9.04;
- **http://dkws.org.ua/index.php?page=show&file=a/ubuntu/network-interfaces** — если вы решите отказаться от NetworkManager и использовать старый сервис network, настоятельно рекомендую ознакомиться с форматом файла конфигура-

ции /etc/network/interfaces (кстати, этот файл конфигурации используется для задания сетевых параметров в Debian);

- **http://dkws.org.ua/index.php?page=show&file=a/ubuntu/vpn-ubuntu9** — как настроить VPN-соединение в Ubuntu (хотя это и не относится к настройке локальной сети, но, думаю, вам пригодится).

11.3.4. Конфигуратор netconfig в Slackware

Конфигуратор netconfig в Slackware можно запускать даже в консоли (рис. 11.21). Он поочередно задаст вам ряд вопросов: от имени компьютера до IP-адреса шлюза. По сути, его работа ничем не отличается от работы прочих рассмотренных здесь конфигураторов, просто у него несколько своеобразный интерфейс пользователя.

Рис. 11.21. Конфигуратор netconfig

11.4. Утилиты для диагностики соединения

Причины отказа сети могут быть физическими или программными. Физические связаны с неработающим сетевым оборудованием или повреждением среды передачи данных. Программные — с неправильной настройкой сетевого интерфейса. В большинстве случаев избавиться от программных проблем помогает конфигуратор сети — вы его еще раз запускаете и правильно настраиваете сетевые интерфейсы. Если сомневаетесь в своих действиях, обратитесь за помощью к более опытному коллеге.

Для диагностики работы сети мы будем использовать стандартные сетевые утилиты, которые входят в состав любого дистрибутива Linux. Предположим, что у нас не работает PPPoE/DSL-соединение. Проверить, "поднят" ли сетевой интерфейс, можно с помощью команды `ifconfig`. На рис. 11.22 показано, что сначала я предпринял попытку установить соединение (ввел команду `sudo pon dsl-provider`), а затем вызвал `ifconfig` для того, чтобы убедиться, установлено ли соединение. В случае, если соединение не было бы установлено, интерфейс ppp0 в списке бы отсутствовал.

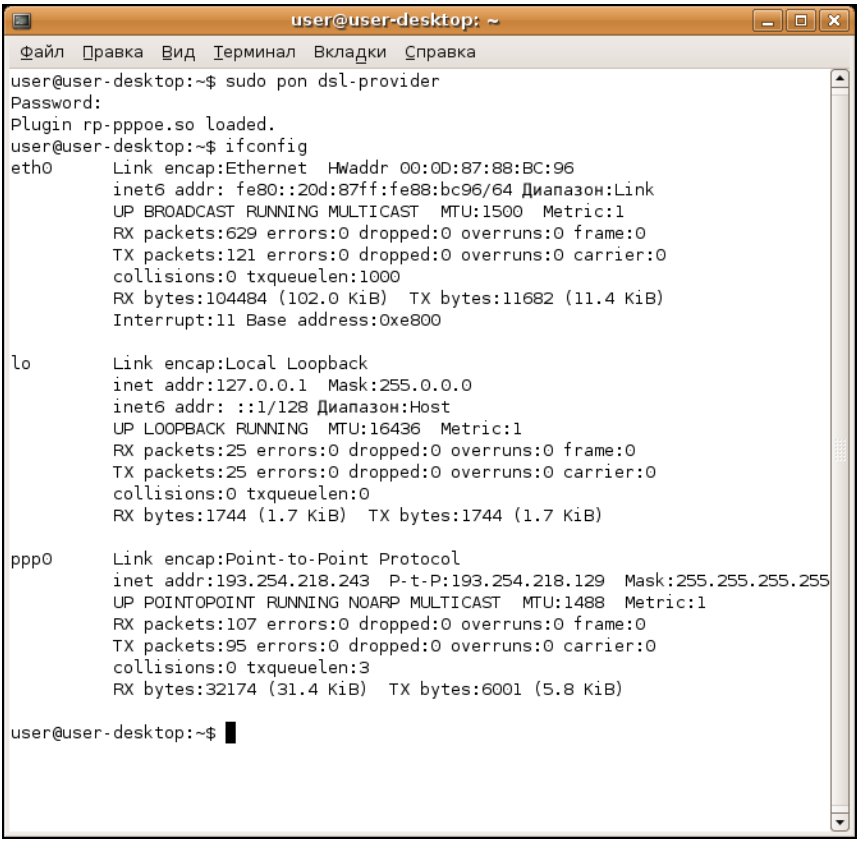


Рис. 11.22. Программа `ifconfig`

Пояснение

Интерфейс eth0 относится к первой сетевой плате (вторая называется eth1, третья — eth2 и т. д.), а интерфейс lo — это интерфейс обратной петли, который служит для тестирования программного обеспечения (у вас он всегда будет "поднят").

Если же интерфейс не поднят, нам нужно просмотреть файл /var/log/messages сразу после попытки установки сообщения:

```
tail -n 10 /var/log/messages
```

Данная команда просматривает "хвост" файла протокола (выводит последние 10 сообщений). В случае удачной установки соединения сообщения в файле протокола будут примерно следующими:

```
Feb  6 14:28:33 user-desktop pppd[5176]: Plugin rp-pppoe.so loaded.
Feb  6 14:28:33 user-desktop kernel: [17179852.932000] CSLIP: code copyright 198 9 Regents of the University of California
Feb  6 14:28:33 user-desktop kernel: [17179852.944000] PPP generic driver versio n 2.4.2
Feb  6 14:28:33 user-desktop pppd[5183]: pppd 2.4.4b1 started by root, uid 0
Feb  6 14:28:33 user-desktop pppd[5183]: PPP session is 2838
```

```
Feb  6 14:28:33 user-desktop kernel: [17179852.984000] NET: Registered protocol family 24
Feb  6 14:28:33 user-desktop pppd[5183]: Using interface ppp0
Feb  6 14:28:33 user-desktop pppd[5183]: Connect: ppp0 <--> eth0
Feb  6 14:28:33 user-desktop pppd[5183]: Remote message: Login ok
Feb  6 14:28:33 user-desktop pppd[5183]: PAP authentication succeeded
Feb  6 14:28:33 user-desktop pppd[5183]: peer from calling number 00:15:F2:60:28
:97 authorized
Feb  6 14:28:33 user-desktop pppd[5183]: local  IP address 193.254.218.243
Feb  6 14:28:33 user-desktop pppd[5183]: remote IP address 193.254.218.129
Feb  6 14:28:33 user-desktop pppd[5183]: primary   DNS address 193.254.218.1
Feb  6 14:28:33 user-desktop pppd[5183]: secondary DNS address 193.254.218.27
```

Первая строчка — сообщение о том, что загружен модуль поддержки PPPoE. Следующие два сообщения информируют нас о поддержке нашим компьютером протоколов CSLIP и PPP. Затем сообщается, что демон pppd запущен, от чьего имени он запущен (root) и приводится версия самого pppd. Далее сообщается имя используемого интерфейса (ppp0) и имя вспомогательного интерфейса (помните, что протокол PPPoE подразумевает передачу кадров PPP по Ethernet) — eth0. Следующие два сообщения свидетельствуют об удачной регистрации:

```
Feb  6 14:28:33 user-desktop pppd[5183]: Remote message: Login ok
Feb  6 14:28:33 user-desktop pppd[5183]: PAP authentication succeeded
```

Затем система сообщает нам наш IP-адрес, адрес удаленного компьютера, который произвел аутентификацию, а также IP-адреса серверов DNS.

А вот пример неудачной попытки соединения:

```
Feb  6 09:23:48 user-desktop pppd[6667]: PPP session is 2336
Feb  6 09:23:48 user-desktop pppd[6667]: Using interface ppp1
Feb  6 09:23:48 user-desktop pppd[6667]: Connect: ppp1 <--> eth0
Feb  6 09:23:48 user-desktop pppd[6667]: Remote message: Login incorrect
Feb  6 09:23:48 user-desktop pppd[6667]: Connection terminated.
```

Причина неудачи понятна — имя пользователя или пароль неправильные, о чем красноречиво свидетельствует сообщение **Login incorrect**. Для того чтобы изменить имя пользователя или пароль, запустите конфигуратор pppoeconf. Но не спешите это делать — если в предыдущий раз соединение было установлено (а настройки соединения вы не изменяли), возможно, нужно обратиться к провайдеру — это явный признак неправильной работы оборудования на стороне провайдера.

Вот еще один пример, характерный для PPPoE:

```
Feb  6 09:23:48 user-desktop pppd[6667]: PPP session is 2336
Feb  6 09:23:48 user-desktop pppd[6667]: Using interface ppp1
Feb  6 09:23:48 user-desktop pppd[6667]: Connect: ppp1 <--> eth0
Feb  6 09:23:48 user-desktop pppd[6667]: Connection terminated.
```

Это явный пример неправильной работы оборудования провайдера. Возможно, нужно перезагрузить точку доступа (access point), т. е. просто выключите и включите ее. Если это не помогает, тогда обращайтесь к провайдеру.

Наиболее простая ситуация, когда сеть вообще не работает. В этом случае очень легко обнаружить причину неисправности. Если работает устройство, значит, повреждена среда передачи данных (сетевой кабель). В случае с модемной линией нужно проверить, нет ли ее обрыва. В случае с витой парой обрыв маловероятен (хотя возможен), поэтому нужно проверить, правильно ли обжат кабель (возможно, нужно обжать витую пару заново).

Намного сложнее ситуация, когда сеть то работает, то нет. Например, вы не можете получить доступ к какому-нибудь узлу, хотя пять минут назад все работало отлично. Если исключить неправильную работу удаленного узла, к которому вы подключаетесь, следует поискать решение в маршруте, по которому пакеты добираются от вашего компьютера до удаленного узла. Сначала пропингуем удаленный узел. Для этого используется команда `ping` (прервать выполнение команды `ping` можно с помощью нажатия комбинации клавиш <Ctrl>+<C>):

```
ping dkws.org.ua

PING dkws.org.ua (213.186.114.75) 56(84) bytes of data.
64 bytes from wdt.org.ru (213.186.114.75): icmp_seq=1 ttl=58 time=30.7 ms
64 bytes from wdt.org.ru (213.186.114.75): icmp_seq=2 ttl=58 time=24.8 ms
64 bytes from wdt.org.ru (213.186.114.75): icmp_seq=5 ttl=58 time=12.2 ms
64 bytes from wdt.org.ru (213.186.114.75): icmp_seq=6 ttl=58 time=159 ms
64 bytes from wdt.org.ru (213.186.114.75): icmp_seq=7 ttl=58 time=19.3 ms
64 bytes from wdt.org.ru (213.186.114.75): icmp_seq=9 ttl=58 time=29.0 ms
...
```

В этом случае все нормально. Но иногда ответы от удаленного сервера то приходят, то не приходят. Чтобы узнать, в чем причина (где именно теряются пакеты), нужно выполнить трассировку узла:

```
tracepath dkws.org.ua
```

В других дистрибутивах вместо команды `tracepath` используется команда `traceroute`, а в Windows — `tracert`. На рис. 11.23 изображено выполнение команды `tracepath`. Сразу видно, что есть определенные проблемы с прохождением пакетов до удаленного узла.

Понятно, что по пути пакеты теряются. Для того чтобы выяснить причину, вам нужно обратиться к администратору того маршрутизатора, который не пропускает дальше пакеты. Причина именно в нем. В данном случае, как видно из рисунка, пакеты доходят до маршрутизатора dc-m7i-1-ge.interfaces.dc.utel.ua, а после него движение пакетов прекращается.

Если соединение установлено (о чем свидетельствует наличие поднятого интерфейса в выводе `ifconfig`), а Web-страницы не открываются, попробуйте пропинговать любой удаленный узел по IP-адресу. Если не знаете, какой узел пинговать (т. е. не помните ни одного IP-адреса), пропингуйте узел 213.186.114.75. Если вы получите ответ, а странички по-прежнему не открываются, когда вы вводите символьное имя, значит, у вас проблемы с DNS — сервер провайдера почему-то не передал вашему компьютеру IP-адреса DNS-серверов. Позвоните провайдеру, выяс-

ните причину этого, а еще лучше уточните IP-адреса серверов DNS и укажите их в файле /etc/resolv.conf. Формат этого файла прост:

```
nameserver IP-адрес
```

Например:

```
nameserver 193.254.218.1
nameserver 193.254.218.27
```

Всего можно указать до четырех серверов DNS.

Если же не открывается какая-то конкретная страничка, а все остальные работают нормально, тогда понятно, что причина в самом удаленном сервере, а не в ваших настройках.

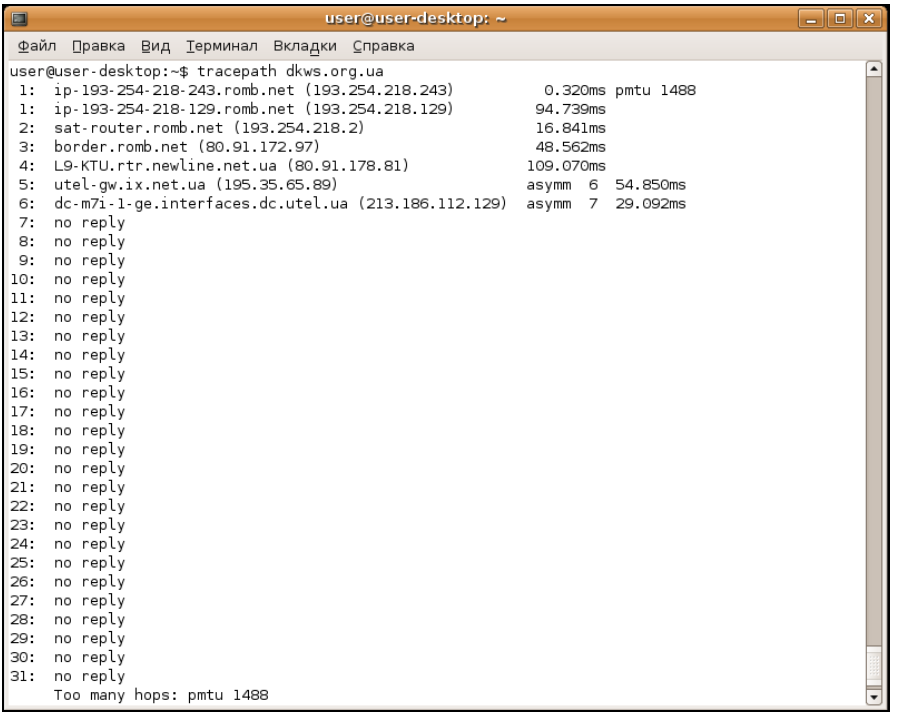


Рис. 11.23. Проблема с прохождением пакетов

11.5. Для фанатов, или как настроить сеть вручную

Иногда мои книги критикуют за то, что при настройке сети я использую только графические конфигураторы. С одной стороны, конфигураторы просты и удобны. Ведь в Windows вы пользуетесь панелью управления, а не редактором реестра, хотя можно изменять сетевые настройки и через regedit. С другой стороны, редактирование конфигурационных файлов позволяет глубже познать Linux. Если вам инте-

ресно, в какие файлы сохраняются сетевые настройки после нажатия кнопки **OK** в окне конфигуратора, тогда данный раздел — для вас. И самое время сейчас снова обратиться к данным табл. 11.1 — когда вы будете знать что и к чему, используя эти данные, вы быстро вспомните, какой конфигурационный файл нужно редактировать. Далее в этом разделе мы рассмотрим конфигурационные файлы конкретных дистрибутивов.

А если вы не считаете, что на это нужно тратить свое время (ведь за считанные секунды можно все настроить конфигуратором), можете смело приступать к чтению следующей главы. Хотя я вовсе не исключаю и такого развития ситуации — вы с интересом прочитаете этот раздел, но все-таки будете использовать конфигураторы, потому что это сильно упрощает процесс.

11.5.1. Конфигурационные файлы Fedora

Мне не нравится дистрибутив Fedora. Но не принимать его во внимание я не могу, поскольку — это классика дистрибутивостроения. Это все равно, что говорить об автомобилестроении и не упомянуть марку "Форд" — это тоже классика. Но в последнее время наблюдается не очень хорошая тенденция — все классические марки портятся. Раньше я с удовольствием работал в Red Hat и восхищался "Фордами". Но мне не нравятся ни современные "Форды", ни современная реализация Red Hat — Fedora.

После такой преамбулы приступим, все же, к рассмотрению конфигурационных файлов этого дистрибутива. Начнем с файла /etc/sysconfig/network. В нем можно задать имя машины, шлюз по умолчанию и включить IP-переадресацию *(см. главу 14)*. Пример этого файла приведен в листинге 11.1.

Листинг 11.1. Файл /etc/sysconfig/network

```
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=den.dkws.org.ua
# Дополнительно
DHCP_HOSTNAME=den.dkws.org.ua
GATEWAY=192.168.0.1
GATEWAYDEV=eth0
FORWARD_IPV4=no
```

В большинстве случаев хватает первых трех параметров:

- параметр `NETWORKING` определяет, будет ли включена поддержка сети. Обычно нужно включить такую поддержку сети (`yes`), поскольку даже функции печати в Linux требуют поддержки сети;
- параметр `NETWORKING_IPV6` включает поддержку IPv6. Поскольку этот протокол еще не используется, то следует задать значение `no`;
- параметр `HOSTNAME` задает имя узла.

В ряде ситуаций могут потребоваться и дополнительные параметры:

- параметр `DHCP_HOSTNAME` задает имя узла при использовании DHCP. Если вы не задали значение параметра `DHCP_HOSTNAME`, то DHCP-сервер может назначить узлу другое имя. Если же значение задано, то DHCP не будет изменять имя узла;
- параметр `GATEWAY` задает шлюз по умолчанию *(см. главу 14)*. В этом конфигурационном файле указывать шлюз по умолчанию не обязательно, поскольку его можно указать в файле /etc/sysconfig/network-scripts/ifcfg-eth0 — конфигурационном файле сетевого интерфейса eth0;
- параметр `GATEWAYDEV` указывает имя интерфейса для доступа к шлюзу. Часто этот параметр опускается;
- последний параметр, `FORWARD_IPV4`, позволяет превратить ваш компьютер в шлюз. Подробно об этом рассказано в *главе 14*.

После редактирования файла /etc/sysconfig/network нужно перейти в каталог /etc/sysconfig/network-scripts/, в котором содержатся конфигурационные файлы для каждого сетевого интерфейса. Например, конфигурация интерфейса eth0 содержится в файле /etc/sysconfig/network-scripts/ifcfg-eth0. Конфигурации интерфейсов могут различаться в зависимости от того, как настраивается интерфейс: автоматически по DHCP или же сетевая информация присваивается статически. Как правило, на рабочих станциях, сетевая информация присваивается автоматически — по DHCP. А вот на серверах (в том числе и на DHCP-сервере) сетевая информация указывается статически — вручную.

В листинге 11.2, *а* приведена конфигурация интерфейса, настраиваемого по DHCP.

Листинг 11.2, *а*. Конфигурация интерфейса, настраиваемого по DHCP

```
DEVICE=eth0
BOOTPROTO=dhcp
HWADDR=XX:XX:XX:XX:XX:XX
ONBOOT=yes
TYPE=Ethernet
IPV6INIT=no
```

Здесь параметр `DEVICE` задает имя устройства (`eth0`), параметр `BOOTPROTO` — тип конфигурации (по протоколу DHCP). Параметр `HWADDR` позволяет изменить аппаратный MAC-адрес сетевого адаптера. Как правило, этот параметр указывается только тогда, когда нужно изменить MAC-адрес. В обычных условиях он не нужен. Параметр `ONBOOT` определяет, будет ли "поднят" интерфейс при загрузке (`yes` — да, `no` — нет). Последние два параметра необязательны (`TYPE` — задает тип интерфейса, `IPV6INIT` — включает для данного интерфейса протокол IPv6).

Пример статической настройки интерфейса приведен в листинге 11.2, *б*.

Листинг 11.2, *б*. Статическая настройка интерфейса

```
DEVICE=eth0
BOOTPROTO=none
HWADDR=XX:XX:XX:XX:XX:XX
ONBOOT=yes
#
NETMASK=255.255.255.0
IPADDR=192.168.0.10
GATEWAY=192.168.0.1
#
NETWORK=192.168.0.0
BROADCAST=192.168.0.255
USERCTL=no
```

Первые четыре параметра нам знакомы. Разница лишь в том, что параметр `BOOTPROTO` содержит значение `none` вместо `dhcp`. Параметр `NETMASK` задает сетевую маску, параметр `IPADDR` — IP-адрес узла, `GATEWAY` — шлюз по умолчанию для данного сетевого интерфейса.

Также можно задать необязательные параметры `NETWORK` (адрес сети), `BROADCAST` (широковещательный IP-адрес) и `USERCTL`. Если последний параметр включен (`yes`), то интерфейсом могут управлять не-root пользователи. Обычно в этом нет необходимости, поэтому присваивается значение `no`.

С остальными файлами вы знакомы из табл. 11.1:

- /etc/resolv.conf — конфигурация DNS (здесь указываются DNS-серверы);
- /etc/hosts — статическая таблица поиска имен узлов, применяется, если ваша сеть не использует DNS;
- /etc/sysconfig/static-routes — данный файл по умолчанию отсутствует, он содержит список статических маршрутов и подробно описан в *главе 14*.

11.5.2. Конфигурационные файлы openSUSE

В openSUSE все конфигурационные файлы, относящиеся к настройкам сети, находятся в каталоге /etc/sysconfig/network:

- /etc/sysconfig/network/ifcfg-имя — содержит параметры сетевого интерфейса (здесь имя — это имя сетевого интерфейса);
- /etc/sysconfig/network/ifroute-имя — содержит маршруты для конкретного интерфейса *(см. главу 14)*;
- /etc/sysconfig/network/routes — список статических маршрутов *(см. главу 14)*;
- /etc/sysconfig/network/config — различные переменные.

Основные файлы — это файлы /etc/sysconfig/network/ifcfg-имя. Рассмотрим пример файла /etc/sysconfig/network/ifcfg-eth0, задающего параметры сетевого интерфейса eth0 (листинг 11.3).

Листинг 11.3. Файл /etc/sysconfig/network/ifcfg-eth0

```
BOOTPROTO='dhcp'
IPADDR=''
MTU=''
NAME='79c970 [PCnet32 LANCE]'
NETMASK=''
NETWORK=''
STARTMODE='auto'
USERCONTROL='no'
```

В файле конфигурации сетевого интерфейса может быть множество самых разных параметров. Все возможные параметры с пояснениями и допустимыми значениями описаны в файле ifcfg.template. Здесь мы рассмотрим только параметры, указанные в листинге 11.3.

Параметр `BOOTPROTO` задает протокол конфигурации интерфейса. Для автоматического назначения IP-адреса по DHCP служит значение `dhcp`. Если нужно назначить адрес вручную, то используется значение `static`. Есть еще два полезных значения:

- `autoip` — производится поиск свободного IP-адреса, найденный IP-адрес назначается статически;
- `dhcp+autoip` — основной способ — DHCP, но если DHCP-сервер отсутствует, то работает вариант autoip.

Назначение остальных параметров ясно — это IP-адрес, размер MTU (Maximum Transmission Unit, максимальный блок передачи), описание устройства (ни на что не влияет), сетевая маска, адрес сети.

Параметр `STARTMODE` задает режим запуска интерфейса:

- `auto` — автоматический запуск при загрузке системы;
- `manual` — интерфейс будет подниматься вручную;
- `off` — интерфейс не используется.

Есть и другие режимы запуска — о них вы прочитаете в файле ifcfg.template. Последний параметр — `USERCONTROL` — запрещает управление интерфейсом не-root пользователям.

Следует упомянуть и такую полезную опцию: `DHCLIENT_SET_HOSTNAME`. Она определяет, будет ли DHCP-клиент изменять имя узла, что полезно, если не нужно изменять имя узла каждый раз при получении нового IP-адреса (значение `no`).

Также можно установить значение no для опции `DHCLIENT_SET_HOSTNAME` в файле /etc/sysconfig/network/dhcp. Разница заключается в том, что в первом случае вы изменяете параметр `DHCLIENT_SET_HOSTNAME` локально — только для конкретного интерфейса, а во втором случае глобально — для всех интерфейсов.

А где же хранится имя узла? Привычного файла /etc/hostname я не нашел. Пришлось действовать старым проверенным способом: вызвать конфигуратор, установить имя узла, а потом смотреть, какой файл изменился. Меня ждал небольшой

сюрприз. Да, файла /etc/hostname нет, но зато есть файл /etc/HOSTNAME (все буквы прописные) — этот файл я просто не заметил. В нем и хранятся имя узла и имя домена.

11.5.3. Конфигурационные файлы Debian/Ubuntu

Основной конфигурационный файл Debian (и Ubuntu при выключенном NetworkManager) — /etc/network/interfaces. В нем можно изменить все — от IP-адреса интерфейса до параметров маршрутизации *(см. главу 14)*. Файл /etc/network/interfaces подробно описан в моей статье **http://dkws.org.ua/index.php?page=show&file=a/ubuntu/network-interfaces**, и нет смысла ее сюда переписывать.

Кроме файла /etc/network/interfaces вам еще пригодится файл /etc/hostname, содержащий имя узла.

Файл /etc/resolv.conf, как и в других дистрибутивах, содержит параметры DNS. Но этот файл перезаписывается системой при перезагрузке. Если у вас рабочая система, то такое поведение — оптимально. А вот на сервере хотелось бы больше контроля. О том, как побороть перезапись этого файла, рассказано в другой моей статье: **http://dkws.org.ua/index.php?page=show&file=a/ubuntu/static-dns-ubuntu9**.

11.6. Команда *mii-tool*

Современные сетевые адаптеры поддерживают несколько скоростей передачи данных: 10, 100 и 1000 Мбит/с, а также два режима передачи данных: полудуплексный и полнодуплексный.

Помню, настраивал PPPoE-соединение в Windows XP. Соединение отказывалось работать на штатной скорости адаптера 100 Мбит/с — происходили постоянные срывы связи через произвольные интервалы времени с момента установки соединения. Пришлось "зажать" сетевой адаптер до скорости 10 Мбит/с — после этого проблема исчезла. На скорости самого соединения это никак не отразилось, поскольку оно было ограничено провайдером — 5 Мбит/с.

До сих пор для меня загадка, почему все не работало по умолчанию. Возможно, дело в самом сетевом адаптере. А может, даже в коммутаторе. Ведь по умолчанию и сетевая плата, и порт коммутатора находятся в режиме автоматического согласования, когда оба устройства пытаются подобрать совместимые параметры. Как следствие — высокая потеря пакетов. Лучший способ — зафиксировать скорость и режим работы сетевого адаптера и порта коммутатора.

В Windows изменение скорости и режима работы сетевого адаптера осуществляется в окне изменения параметров сетевого адаптера. А в Linux нужно использовать команду `mii-tool`. Для изменения режима работы порта коммутатора используется Web-интерфейс коммутатора (как правило, дешевые коммутаторы не позволяют изменять свои параметры), и о том, как это сделать, вы сможете прочитать в документации по коммутатору.

Для просмотра параметров сетевого интерфейса служит команда:

```
# mii-tool -v eth0
```

Вывод будет примерно такой:

```
eth0: negotiated 100baseTx-FD flow-control, link ok
product info: vendor 88:58:43, model 0 rev 0
basic mode: autonegotiation enabled
basic status: autonegotiation complete, link ok
capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow control
link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow control
```

Сейчас сетевой адаптер работает в режиме автоматического согласования режима (`autonegotiation`), текущий статус — автосогласование завершено, связь установлена. Поле `capabilities` содержит список поддерживаемых режимов, а поле `link partne`r содержит список режимов, поддерживаемых коммутатором.

Для установки режима используется опция `-force`:

```
# mii-tool -force=режим интерфейс
```

Например:

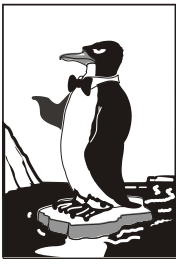
```
# mii-too -force=10baseT-FD eth0
```

11.7. Перед тем как перейти к следующей главе

Напоследок отмечу, что в большинстве случаев вообще сеть настраивать не придется — ведь DHCP-сервер сейчас не роскошь. Именно из-за этого конфигуратор сети openSUSE здесь не рассмотрен (хотя вы без проблем разберетесь с ним, запустив Центр управления YaST). Впрочем, с этим конфигуратором мы познакомимся в следующей главе, где будет рассматриваться настройка DSL-соединения, — там его описание более уместно, чем здесь.

Спрашивается, зачем была нужна эта глава, если все настраивается автоматически? Да, пользователю, может, и не обязательно все это знать, а вот администратор обязан разбираться в таких тонкостях. Пользователи сейчас немного расслабились, осознав, что Linux — это просто. А расслабляться вредно. И когда требуется присвоить статический IP-адрес (например, при настройке того же DHCP-сервера — он должен иметь статический адрес), они начинают "плавать".

Самое интересное, что это ненадуманная проблема. Именно поэтому на главную страницу своего сайта (**http://dkws.org.ua**) я вынес ссылки на статьи, где объясняется, как присвоить статический адрес — чтобы не плодились темы на форуме. По запросу `статический ip адрес в ubuntu` Google выдает более 7000 результатов. А все из-за незнания. Надеюсь, эта глава полностью заполнила пробел в ваших знаниях по настройке локальной сети в Linux.



Глава 12

Настройка ADSL-доступа к Интернету

DSL (Digital Subscriber Line) — цифровая абонентская линия, позволяющая производить двунаправленный обмен данными по телефонной линии. Существуют несколько вариантов DSL-линий: ADSL, VDSL, SDSL, RADSL. Наиболее распространены ADSL-линии. ADSL (Asymmetric DSL) — асимметрическая цифровая линия. Для передачи данных используется витая пара телефонной сети. Скорость передачи данных зависит от расстояния, например, 1,5 Мбит/с при расстоянии в 5–6 км. Но обычно скорость ограничивается провайдером и зависит от тарифного плана. Самый доступный тарифный план подразумевает скорость передачи данных 64 Кбит/с.

12.1. Причина популярности DSL-соединений

Почему ADSL-соединения стали такими популярными? Основная причина популярности — это скорость и дешевизна. Именно эти два фактора. Даже в самом "дешевом" варианте обеспечивается скорость передачи данных 64 Кбит/с. Это в два раза быстрее, чем модем (конечно, в идеальных условиях из модема можно "выжать" 56 Кбит/с, но на практике это получается далеко не всегда). И при этом никаких разрывов соединений!

Да, за подключение к провайдеру нужно заплатить определенную сумму (напомню, что модемное подключение бесплатно), но, поверьте, оно того стоит. Также понадобится специальный ADSL-модем, который стоит дороже обычного модема, но в большинстве случаев есть возможность взять модем в аренду у провайдера, а стоимость такой аренды просто смешна.

Дешево, быстро — это все просто замечательно. Но есть и еще одно преимущество — когда вы работаете в Интернете, ваш телефон не занят, в отличие от модемного соединения.

Однако и здесь не без неожиданностей — ADSL-соединение возможно не на каждой телефонной линии. Ваша телефонная линия должна быть цифровой, иначе ничего не получится.

12.2. Физическое подключение ADSL-модема

ADSL-модем подключается к телефонной линии через специальное устройство — ADSL-сплиттер, который обычно входит в комплект поставки модема. К ADSL-

сплиттеру также подключается и обычный параллельный телефон. В свою очередь компьютер соединяется с ADSL-модемом с помощью Ethernet-кабеля (витой пары), также входящей в комплект поставки. Схема подключения изображена на рис. 12.1.

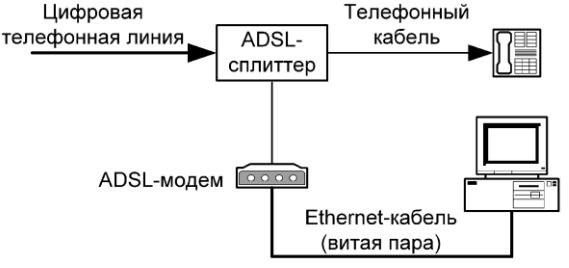


Рис. 12.1. Схема подключения ADSL-модема

Внимание!

Если у вас есть дополнительные параллельные телефоны, то подключать их к телефонной линии напрямую не допускается! Подключать параллельные телефоны можно только через ADSL-сплиттер.

12.3. Настройка DSL-соединения в Fedora

В Fedora соединение проще всего настроить конфигуратором `system-config-network`. Запустите его и нажмите кнопку **Создать**, затем выберите опцию **Соединение xDSL** (рис. 12.2).

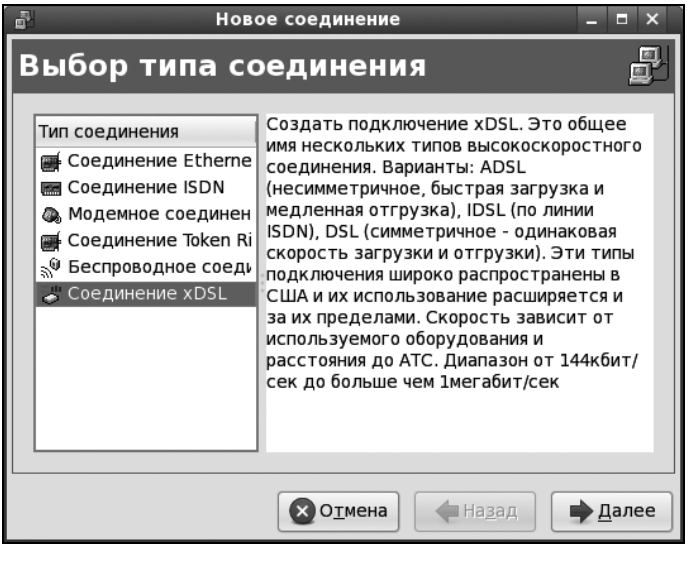


Рис. 12.2. Создание DSL-соединения в Fedora

Следующий шаг еще более важный — вам нужно выбрать устройство, которое соединено с точкой доступа, ввести имя провайдера, имя пользователя и пароль (рис. 12.3).

После этого нужно нажать кнопку **Далее**, а затем кнопку **Применить**. Установить соединение можно командой `system-config-network` — выбрать ваше соединение и нажать кнопку **Активировать** (рис. 12.4). Для разрыва соединения служит кнопка **Деактивировать**.

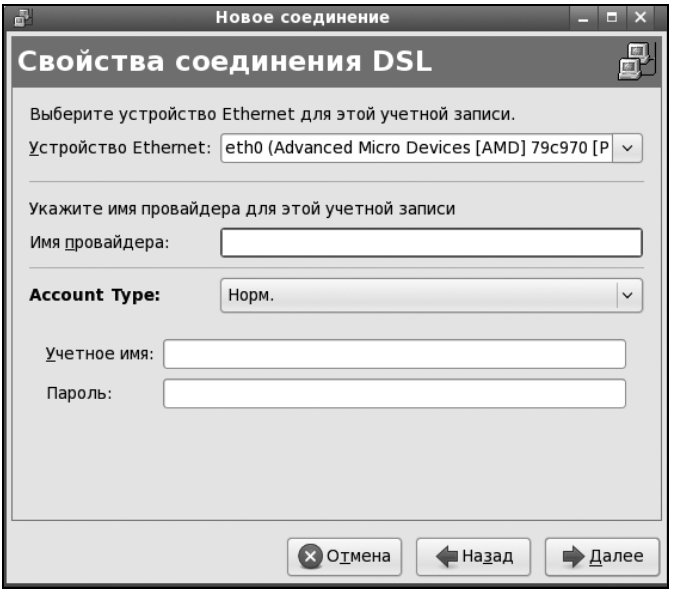


Рис. 12.3. Ввод параметров соединения

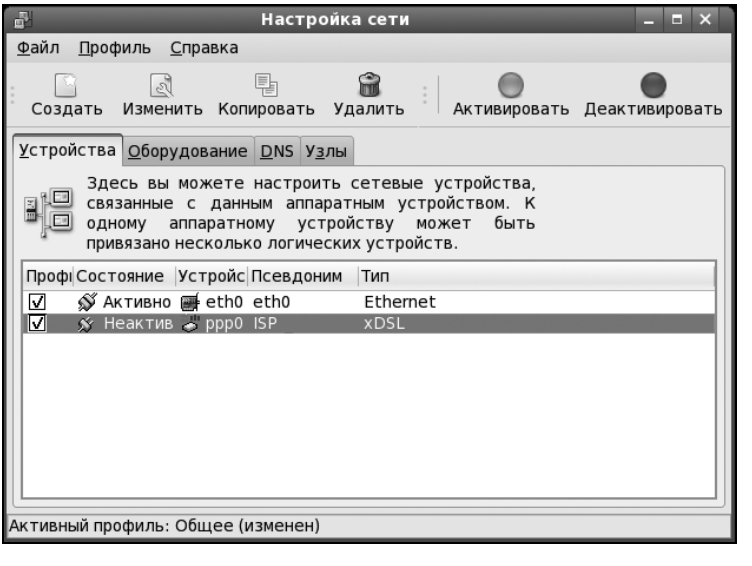


Рис. 12.4. Включение/выключение соединения

Для запуска конфигуратора system-config-network нужны полномочия root. Но обычные пользователи (у которых нет таких полномочий) тоже могут настроить соединение с Интернетом — с помощью программы NetworkManager (запускается из меню **Система | Параметры | Сетевые соединения**), которая установлена в последних версиях Fedora.

Некоторые администраторы предпочитают отключить NetworkManager — действительно, на сервере он не нужен (подробно об этом рассказано в *главе 11*). Для его отключения нужно ввести следующие команды:

```
# service NetworkManager stop
# chkconfig --level 2345 NetworkManager off
```

Первая команда выключает сервис NetworkManager (если он вообще запущен), а вторая — отключает запуск этого сервиса на уровнях запуска 2, 3, 4 и 5.

12.4. Настройка DSL-соединения в openSUSE

Запустите **Центр управления** и выберите **DSL**. Пользователям радиодоступа к Интернету (технология Radio Ethernet) тоже нужно использовать конфигуратор DSL — настройка Radio Eterhet осуществляется аналогично настройке DSL.

Примечание

Для непосредственного запуска (не через Центр управления) конфигуратора модема используется команда `/sbin/yast2 dsl`.

Конфигуратор попытается найти DSL-устройства. Это может занять некоторое время, так что придется немного подождать (рис. 12.5).

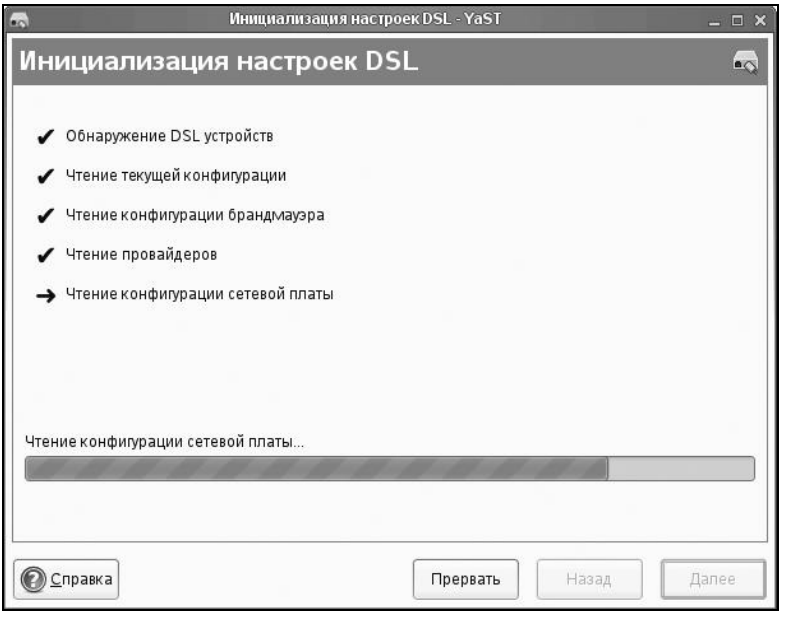


Рис. 12.5. Поиск DSL-устройств

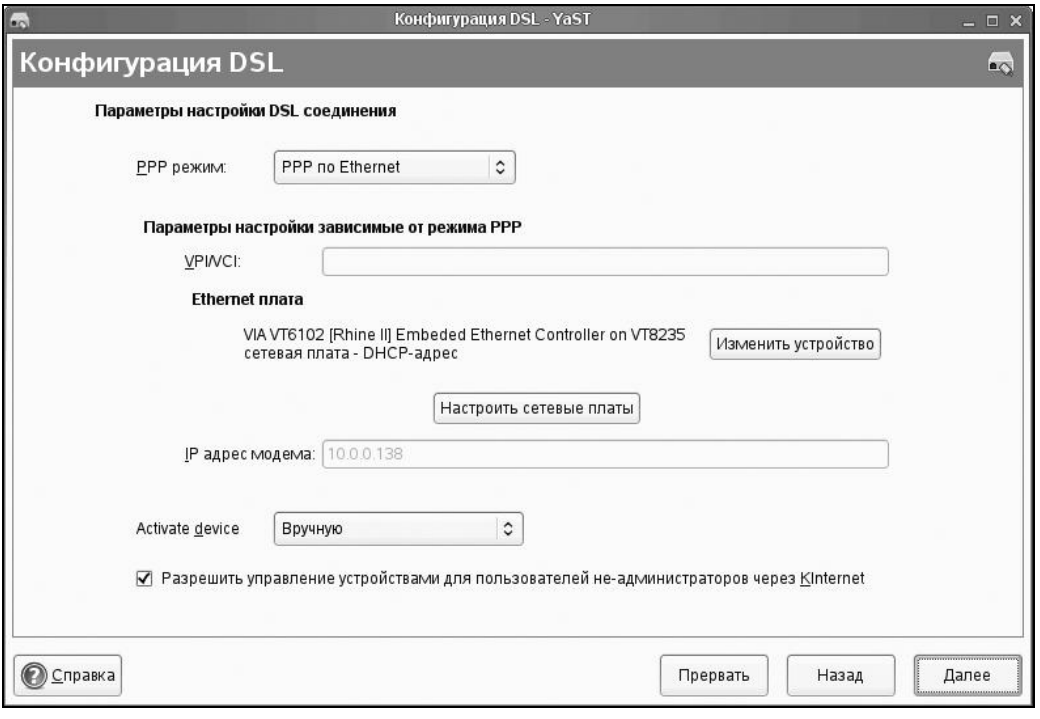


Рис. 12.6. Параметры DSL-соединения

Далее вы увидите пустое окно обзора настроек DSL (как будто не найдено ни одного DSL-устройства). Не пугайтесь — так и должно быть. Просто нажмите кнопку **Далее**. Вам нужно задать параметры DSL-соединения (рис. 12.6), а именно: выбрать режим PPP, сетевую плату, к которой подключен DSL-модем, режим активации устройства и обязательно разрешить управление соединением через KInternet (иначе вы просто не сможете использовать эту программу).

- Начнем с режима PPP — обычно используется режим **PPP по Ethernet**. Технология ADSL (как и другие технологии, например, Radio Ethernet), использует протокол PPPoE (Point to Point Protocol over Ethernet).

> ***Пояснение***
>
> Протокол PPP используется обычным модемным соединением, а протокол PPPoE обеспечивает передачу PPP-кадров по сетевой плате (Ethernet) — это и есть суть режима PPP по Ethernet.

- Сетевая плата обычно выбирается конфигуратором правильно, поэтому ее не нужно изменять, тем более, что в большинстве случаев найденная сетевая плата является единственным сетевым адаптером в системе.
- Режим активации устройства (**Activate device**) позволяет определить, как будет активироваться устройство — вручную или автоматически при запуске системы. Тут решать вам — можно запускать DSL-соединение и при запуске системы, но тогда отпадает необходимость в использовании KInternet.

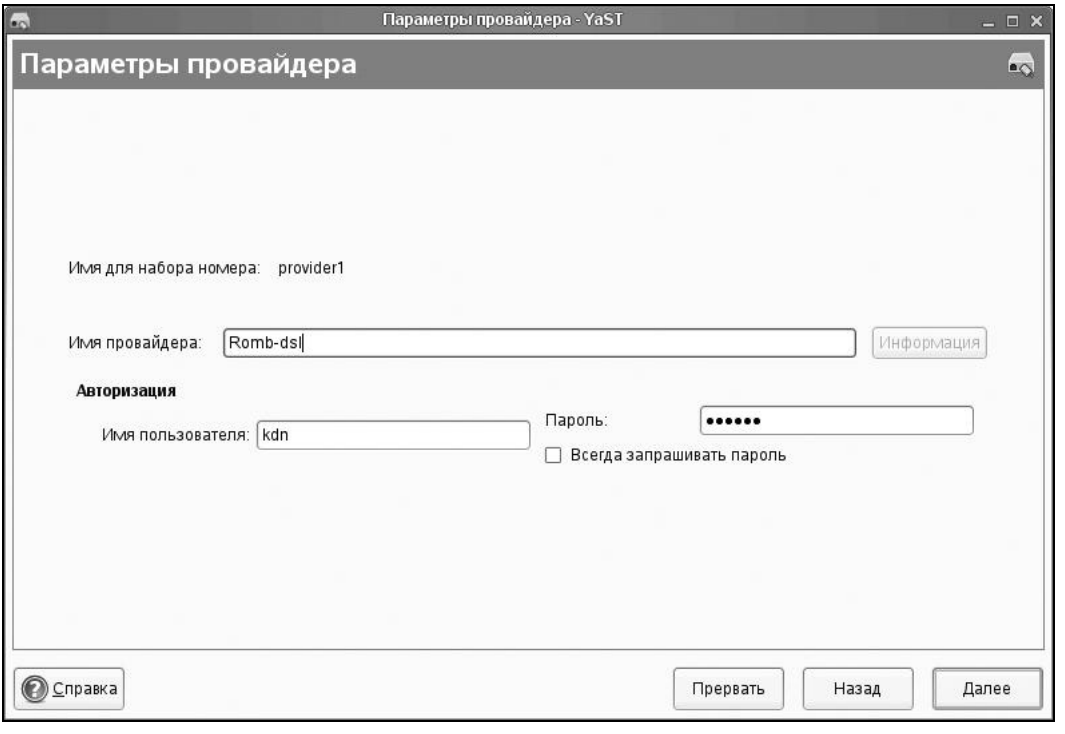


Рис. 12.7. Информация о провайдере

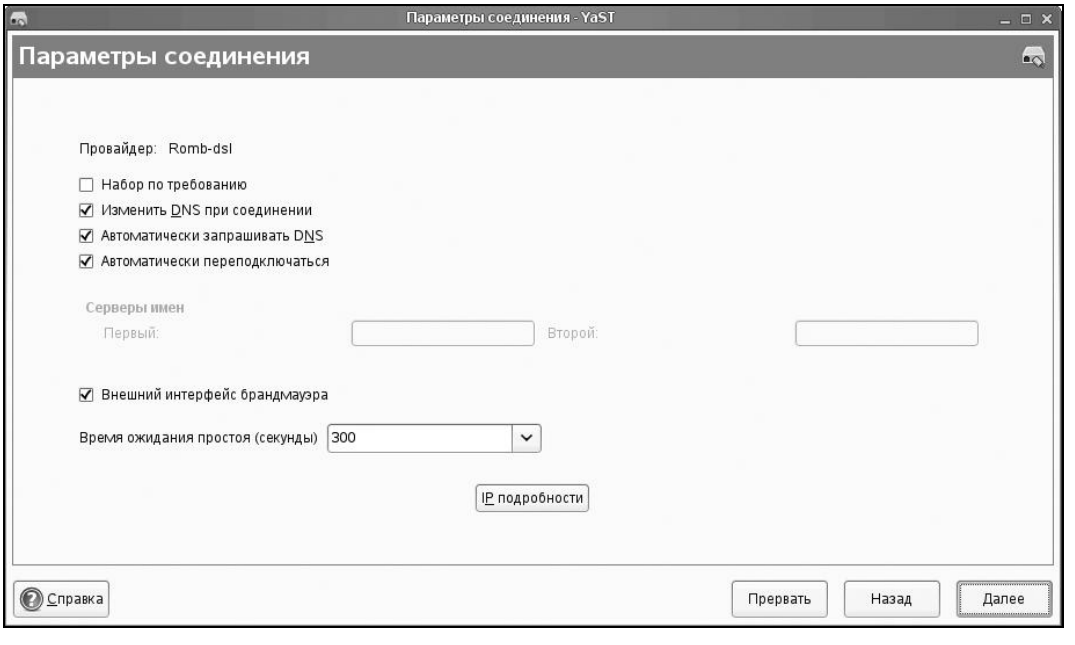


Рис. 12.8. Параметры соединения

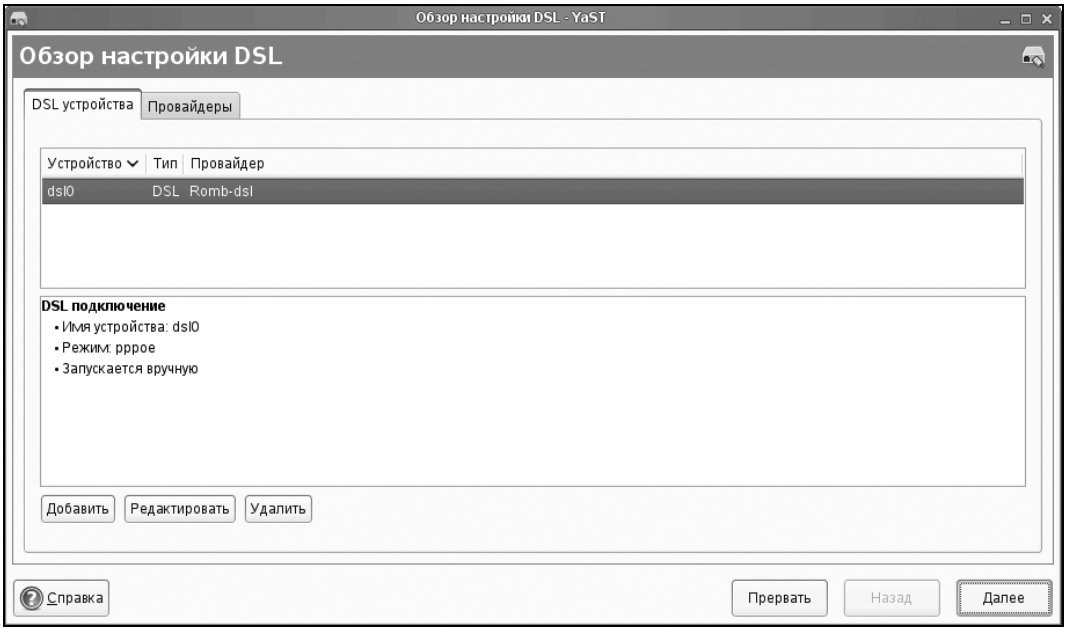


Рис. 12.9. Созданное соединение

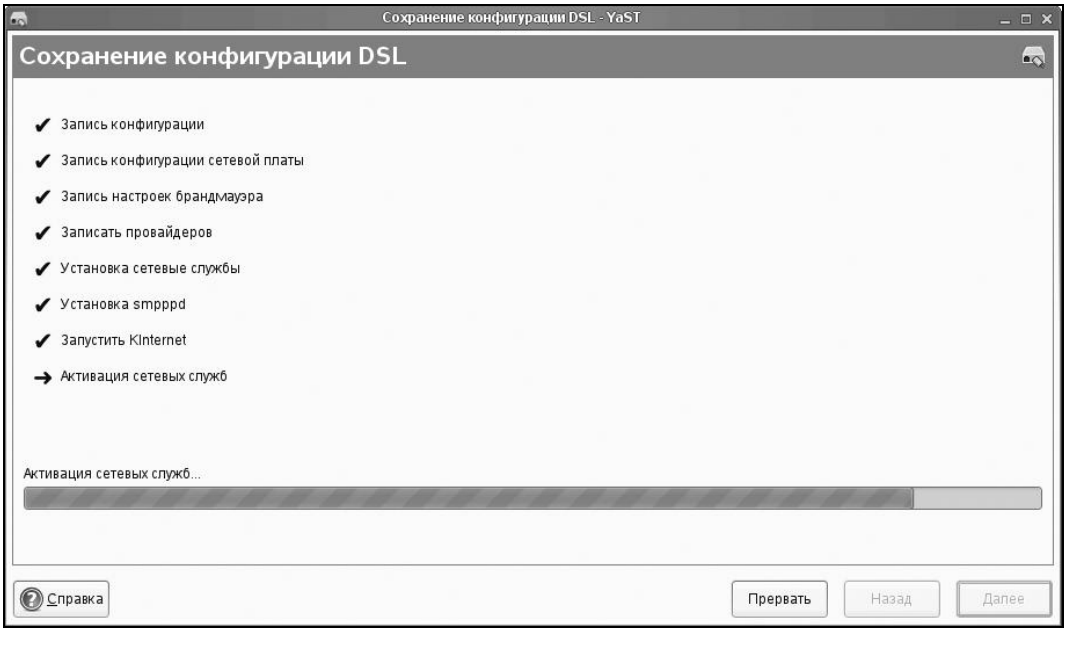


Рис. 12.10. Сохранение конфигурации

Следующий этап настройки DSL-соединения — это выбор провайдера. Вашего провайдера не будет в списке, поэтому сразу нажимайте **новый**, вводите имя провайдера, имя пользователя и пароль (рис. 12.7).

Теперь следует определить некоторые параметры соединения. Параметры, предложенные конфигуратором (рис. 12.8), вполне приемлемы и устроят большинство пользователей, поэтому просто просмотрите их и нажмите кнопку **Далее**.

Вы вернетесь в окно обзора DSL-соединений, которое теперь не будет пустым — в нем появится только что созданное соединение (рис. 12.9).

Все, что вам осталось — это нажать кнопку **Далее** и подождать, пока YaST сохранит конфигурацию системы (рис. 12.10).

Для подключения к Интернету нужно щелкнуть по значку KInternet (рис. 12.11, *а*). Но если вы до этого настраивали модемное соединение, вам теперь нужно выбрать DSL-подключение. Для этого щелкните правой кнопкой мыши по значку KInternet и выберите команду меню **Интерфейс | dsl0**. Вот теперь можно щелкнуть по значку левой кнопкой для установки соединения. Для отключения, как обычно, нужно снова щелкнуть по значку Kinternet (рис. 12.11, *б*).

Рис. 12.11. Программа Kinternet: *а* — соединения нет; *б* — соединение установлено

12.5. Настройка DSL-соединения в Ubuntu

В дистрибутивах Debian и Ubuntu для настройки DSL-соединений используется конфигуратор `pppoeconf`. Откройте терминал и введите команду:

```
sudo pppoeconf
```

Примечание

Начиная с Ubuntu 8.10, появился новый графический конфигуратор сети — NetworkManager. Для его запуска выполните команду **Система | Параметры | Сетевые соединения** и перейдите на вкладку **DSL**. Но вы, по желанию, можете использовать и программу pppoeconf — она также работает в новых версиях Ubuntu. Тут уж дело вкуса: некоторые пользователи предпочитают графические конфигураторы, а кто-то — работает в консоли. Лично я настраивал DSL-соединение в своем Ubuntu 10.04 с помощью pppoeconf.

Согласно спецификации PPPoE существуют две стадии: стадия поиска и стадия сессии. На первой стадии производится отправка специальных пакетов PADI (PPPoE Active Discovery Initiation), которые позволяют найти активные концентраторы доступа PPPoE (рис. 12.12). Стадия сессии — это само соединение и передача информации.

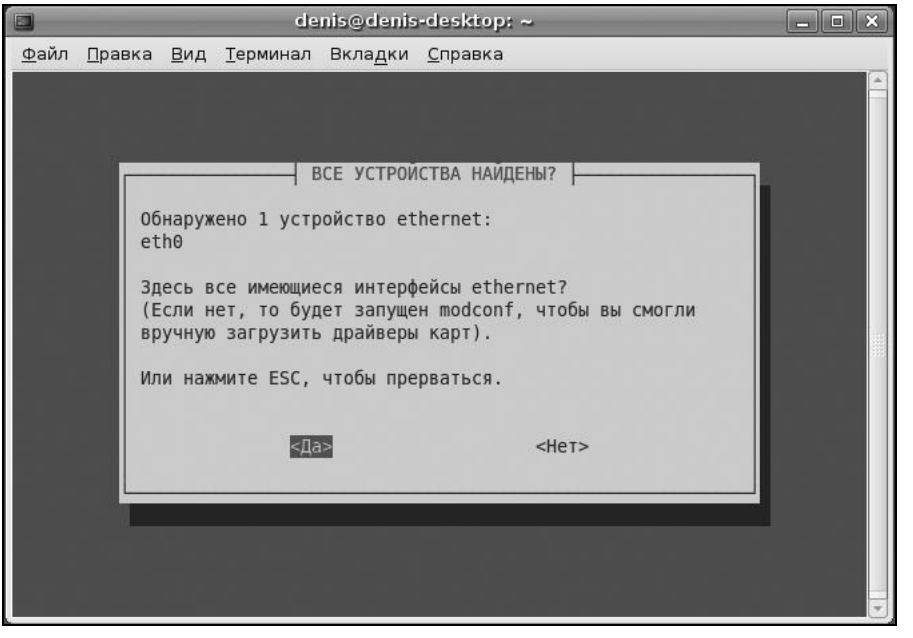


Рис. 12.12. Конфигуратор `pppoeconf` нашел Ethernet-устройство

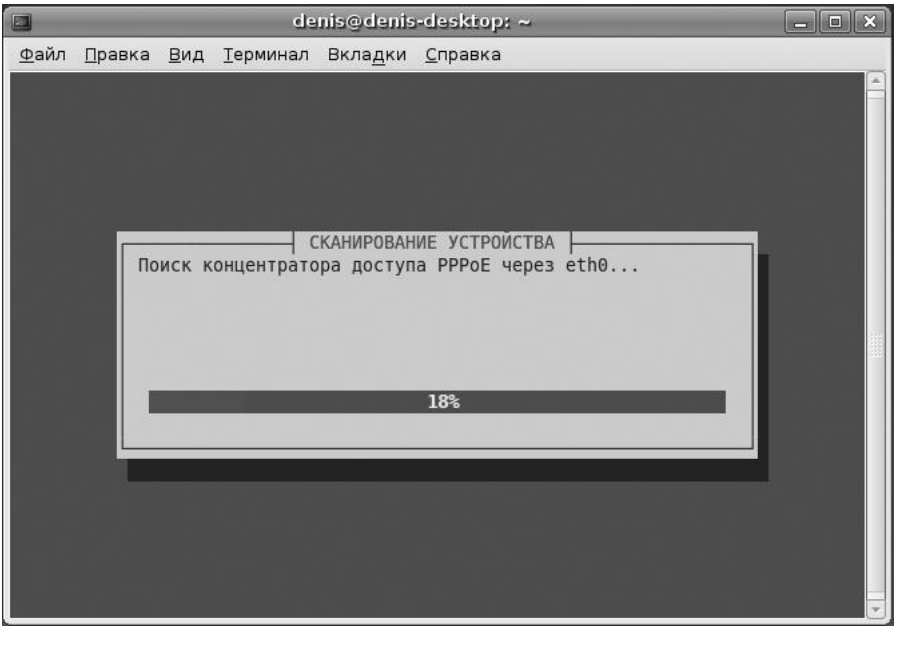


Рис. 12.13. Поиск активного концентратора доступа

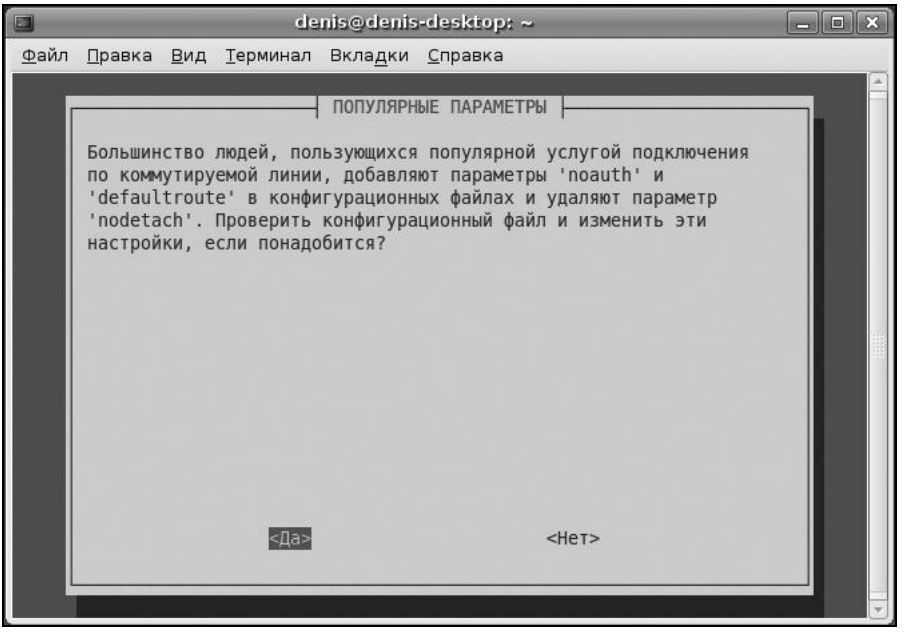


Рис. 12.14. Популярные опции соединения

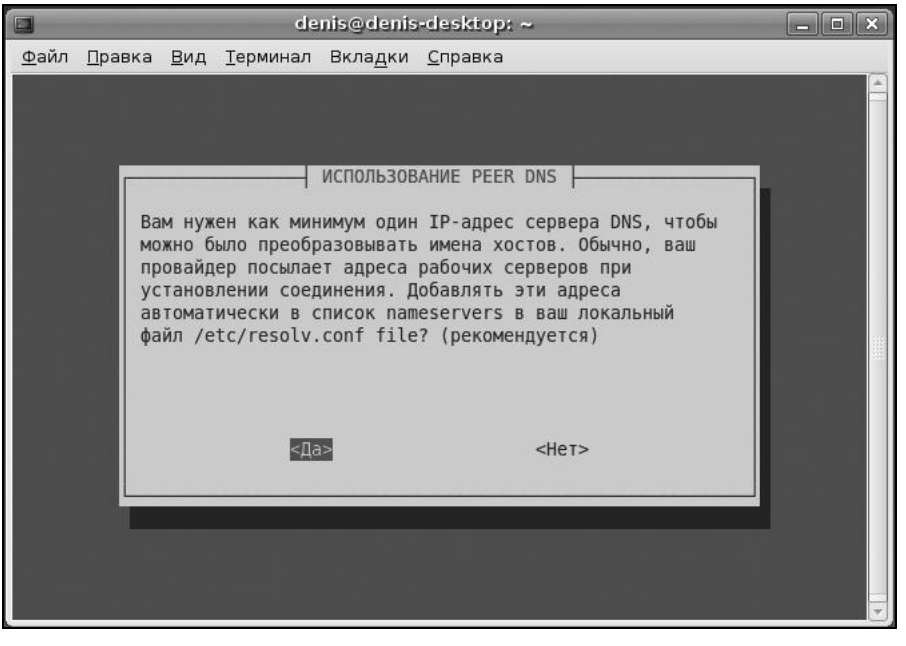


Рис. 12.15. Добавляем IP-адреса DNS-серверов в файл /etc/resolv.conf

Затем конфигуратор попытается найти активный концентратор доступа (рис. 12.13). После того как концентратор доступа будет найден, программа предложит вам установить популярные опции соединения (`noauth` и `defaultroute`): не стоит от них отказываться, поскольку их использует большинство провайдеров (рис. 12.14).

Следующие два шага — ввод имени пользователя и пароля, которые используются для аутентификации на сервере провайдера. После этого программа предложит вам добавить полученные от провайдера IP-адреса DNS-серверов в файл /etc/resolv.conf. Не стоит отказываться и от этого (рис. 12.15).

На следующий вопрос (рис. 12.16) можно просто ответить **Да**, не вникая в подробности. Если же вам интересно, прочитайте следующее примечание.

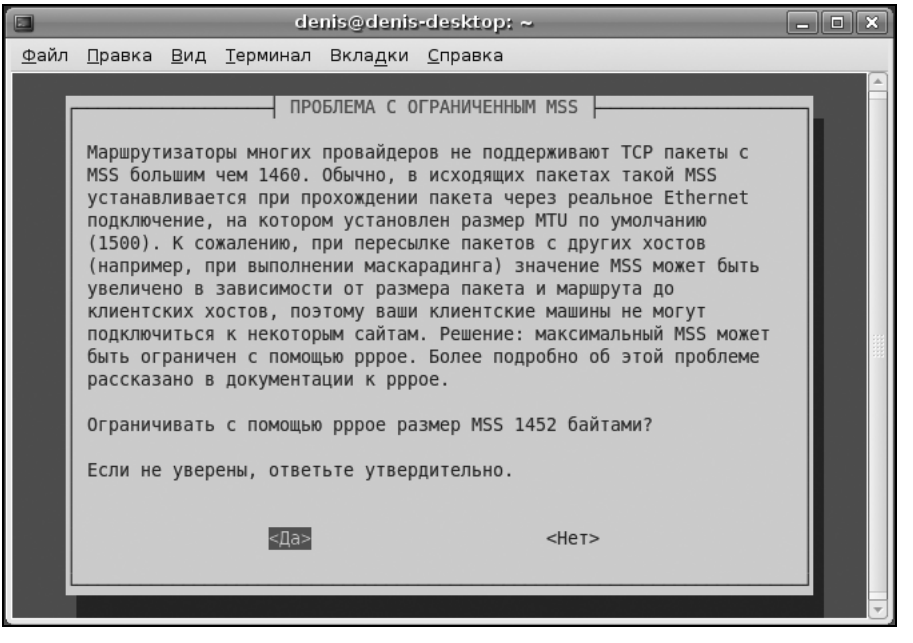


Рис. 12.16. Установка размера MSS

Примечание

Параметр MTU (Maximum Transmit Unit) задает максимальный размер пакета. По умолчанию данное значение может быть установлено автоматически, но не всегда оптимально. Если размер пакета окажется по размеру больше, чем позволяет машрутизатор провайдера, то пакет будет разделен на несколько пакетов, что, естественно, скажется на скорости и пропускной способности соединения. Если размер пакета будет меньше, чем положено, то это тоже не хорошо — канал будет использован нерационально, ведь будут проходить полупустые кадры. Поскольку мы работаем по протоколу PPPoE, то нужно учитывать несколько факторов. Максимальный размер кадра Ethernet составляет 1518 байтов, из которых 18 уходит на заголовок и контроль, поэтому для полезных данных остается 1500 байтов. Обычно данное значение и указывается для Ethernet. Но ведь по Ethernet мы собираемся передавать пакеты PPP, а PPPoE отбирает еще 6 байтов, PPP — 2 байта. Получается, что для PPPoE значение

> MTU должно быть равно 1492. При установке TCP-соединения каждая сторона устанавливает параметр MSS (Maximum Segment Size) — максимальный размер TCP-сегмента. По умолчанию его размер равен MTU минус размер заголовков TCP/IP, которые занимают еще 40 байтов. То есть размер MMS для PPPoE равен 1452 байта (для обычного Ethernet — 1460). Вот откуда взялось значение 1452.

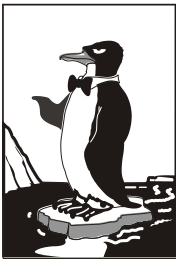
Следующий вопрос — хотите ли вы устанавливать соединение при загрузке системы. Тут уж решайте сами. А после этого программа спросит вас, хотите ли вы установить соединение немедленно. Конечно, да! Можно сразу запускать браузер и заходить на любимую страничку.

Для включения/отключения DSL-соединения используются следующие команды:

```
sudo pon dsl-provider
sudo poff dsl-provider
```

12.6. Настройка DSL-соединения в Mandriva

Для экономии места в книге (а значит, и для экономии ваших денег) подробно настройку DSL-соединения в Mandriva рассматривать мы не будем. Настройка производится с помощью конфигуратора drakconnect. Ответив на несложные вопросы конфигуратора, вы за считанные секунды настроите соединение с Интернетом.



Глава 13

Подключение к сети Wi-Fi

13.1. О настройке Wi-Fi в Linux

О подключении к сети Wi-Fi в Linux можно написать отдельную книжку. Сложность настройки Wi-Fi заключается в том, что последовательность действий при этом в различных дистрибутивах разная. В этой главе мы рассмотрим настройку Wi-Fi в дистрибутивах Ubuntu 8.10–10.*x*/Denix/Fedora 10–13 — то есть в тех, где имеется конфигуратор NetworkManager. Если на вашем компьютере установлен один из этих дистрибутивов, значит, в 99% случаев у вас не будет проблем с настройкой беспроводной сети. Если же у вас другой дистрибутив или старая версия одного из упомянутых (например, Ubuntu 8.04 или Fedora 8), тогда вам нужно или сменить/обновить дистрибутив, или же ознакомиться с материалом *разд. 13.3* — возможно, Wi-Fi и удастся настроить. Может, впрочем, получиться и так, что вы только провозитесь лишнее время, а потом все равно обновите версию дистрибутива.

> ***Пояснение***
>
> Denix — дистрибутив, разработанный автором этой книги. Дистрибутив основан на Ubuntu 10, поэтому полностью совместим с ним. Подробно об этом дистрибутиве можно узнать по адресу **http://denix.dkws.org.ua**.

13.2. Простая настройка (Ubuntu/Denix/Fedora)

Включите ваш беспроводной адаптер и щелкните на значке соединения на панели GNOME. Из рис. 13.1 видно, что обнаружена сеть D_DOT, но автоматического подключения к ней не произошло, поскольку сеть защищена паролем. Справа от названия сети выводится индикатор уровня сигнала. Чуть ниже вы видите команды:

- **Подключиться к скрытой беспроводной сети** — для подключения к скрытой сети нужно знать ее SSID и пароль. Как уже отмечалось ранее, иногда администраторы из соображений безопасности отключают широковещание SSID, и сеть становится скрытой. Подключиться к такой сети можно, только если вы знаете SSID и пароль сети;
- **Создать новую беспроводную сеть** — можно создать одноранговую ad hoc-сеть (без точки доступа).

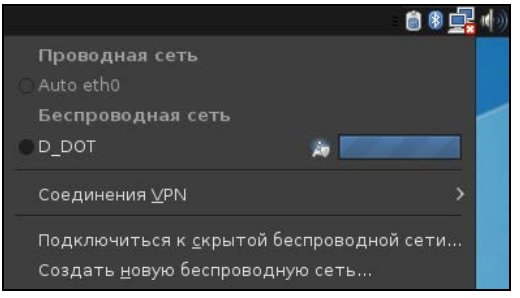


Рис. 13.1. Обнаружена защищенная паролем беспроводная сеть

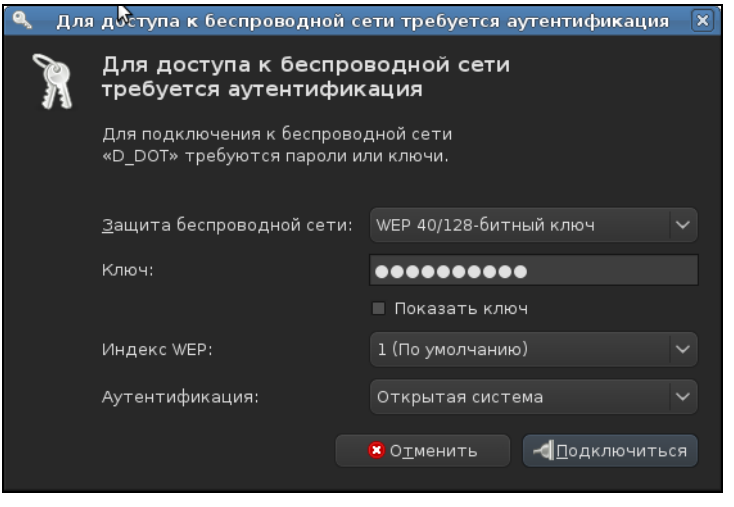


Рис. 13.2. Ввод пароля для подключения к сети

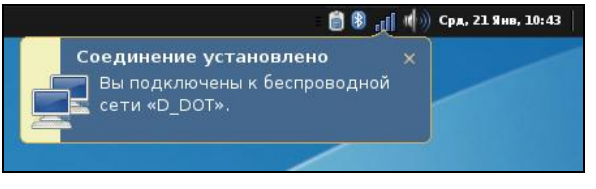


Рис. 13.3. Мы подключены к беспроводной сети

Выберите сеть из списка, изображенного на рис. 13.1. Вы увидите окно ввода пароля (ключа) для подключения к сети (рис. 13.2). Введите пароль (вы должны узнать его заранее), и если он верный, то последует уведомление о подключении к сети (рис. 13.3).

Список беспроводных сетей можно просмотреть, выполнив команду **Система | Параметры | Сетевые соединения** и перейдя на вкладку **Беспроводная сеть** (рис. 13.4).

Как видите, в современном дистрибутиве Linux беспроводная сеть настраивается так же легко, как и в Windows. В следующем пункте мы рассмотрим "тяжелый случай", когда нужно установить отсутствующий драйвер беспроводного адаптера.

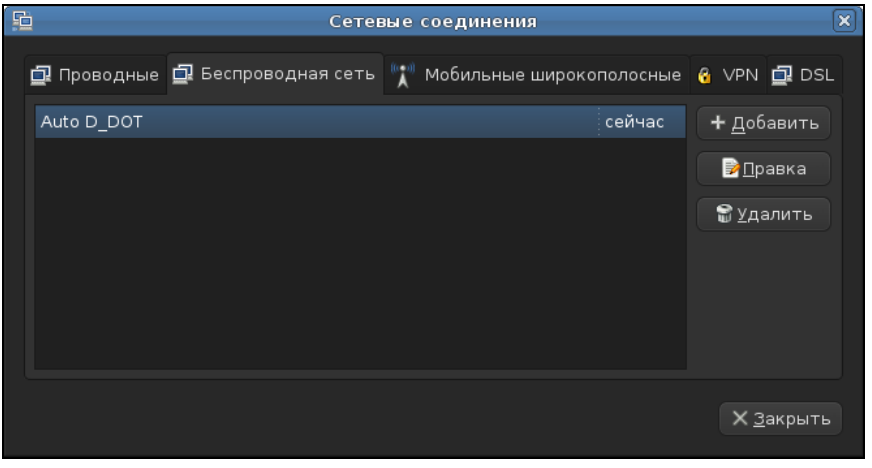


Рис. 13.4. Список беспроводных сетей

13.3. "Тяжелый случай"

В дистрибутиве Ubuntu 8.04 по умолчанию у меня определился только Bluetooth-адаптер, а вот чтобы заработал адаптер Wi-Fi, пришлось повозиться. Впрочем, может быть, вам повезет больше, чем мне, и ваш беспроводной адаптер будет определен по умолчанию или с наименьшими усилиями и затратами времени. На пошаговые инструкции здесь даже и не надейтесь. В этом разделе вы получите только минимально необходимый список средств, позволяющих настроить ваш беспроводной адаптер Wi-Fi.

Первым делом откройте терминал и введите команду `iwconfig`. Она покажет ваши беспроводные интерфейсы. Если беспроводной адаптер не будет обнаружен, вы получите вывод, изображенный на рис. 13.5.

Вся проблема заключается в отсутствии подходящих драйверов беспроводных адаптеров Wi-Fi для Linux. Основная ваша задача — сделать так, чтобы система увидела ваш беспроводной адаптер как сетевой интерфейс. Далее все настраивается довольно легко с помощью программы network-manager-gnome или другой утилиты, позволяющей задать параметры беспроводного доступа к сети (выбрать сеть, установить ее SSID, ввести пароль и т. д.).

Где взять драйверы? В некоторых случаях они уже будут в составе вашего дистрибутива. Все, что вам тогда нужно сделать, — это просто настроить беспроводную сеть. Иногда драйвер адаптера можно найти на сайте производителя, но в большинстве случаев там вы найдете драйверы только для Windows. Сожалею, но, скорее всего, драйверов для Linux вы там не найдете. Впрочем, зная производителя адаптера, всегда можно попытаться найти для него драйвер на сайте **http://linux-wless.passys.nl/**.

Если Linux-драйвера нет, придется воспользоваться Windows-драйвером. Тут следует иметь в виду, что если для того или иного устройства имеются Windows-драйверы, то их тоже можно использовать в Linux с помощью программы ndiswrapper. Можете ее считать "эмулятором драйверов для Linux".

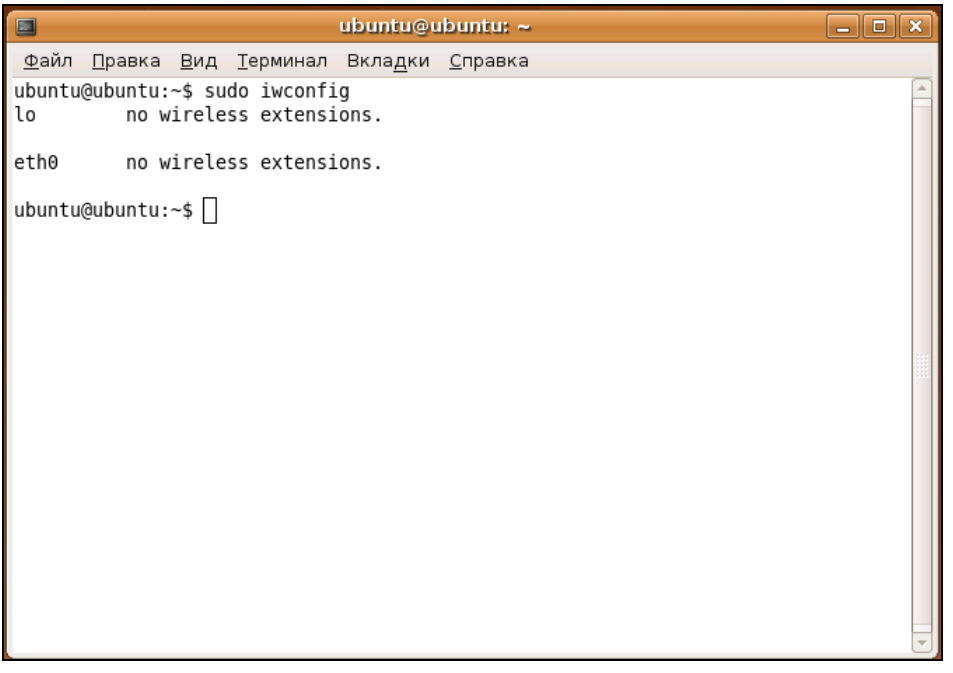


Рис. 13.5. Беспроводные адаптеры не обнаружены

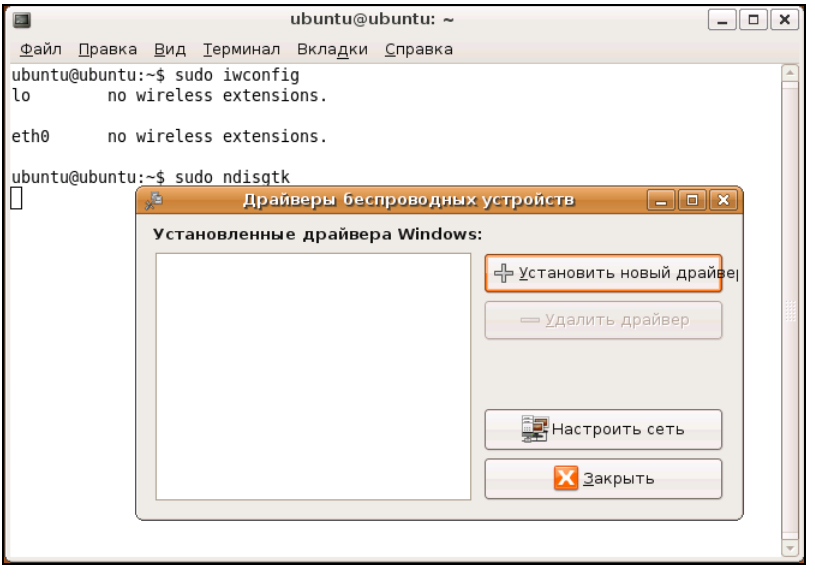


Рис. 13.6. Программа ndisgtk

Как уже отмечалось, драйверы для Windows можно найти или на компакт-диске, который поставляется вместе с адаптером или ноутбуком, или на сайте производителя. В моем случае мне пришлось скачать драйвер для Windows XP с сайта HP.

Итак, когда Windows-драйвер будет найден, нужно установить программы ndiswrapper и ndisgtk (это графическая оболочка для ndiswrapper). Сделать это можно с помощью вашего менеджера пакетов. В большинстве случаев программа ndiswrapper или уже установлена в системе, или находится на дистрибутивном DVD, поэтому, даже если у вас нет соединения с Интернетом, программа все равно будет установлена.

Запустите программу ndisgtk и нажмите кнопку **Установить новый драйвер** (рис. 13.6). Затем укажите путь к INF-файлу драйвера (рис. 13.7).

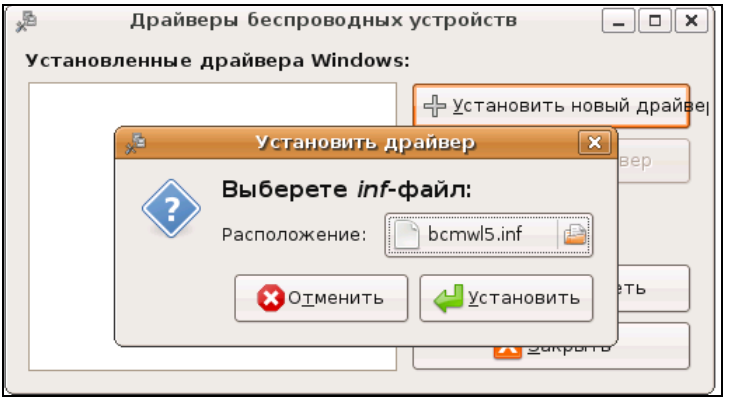


Рис. 13.7. Выберите INF-файл драйвера

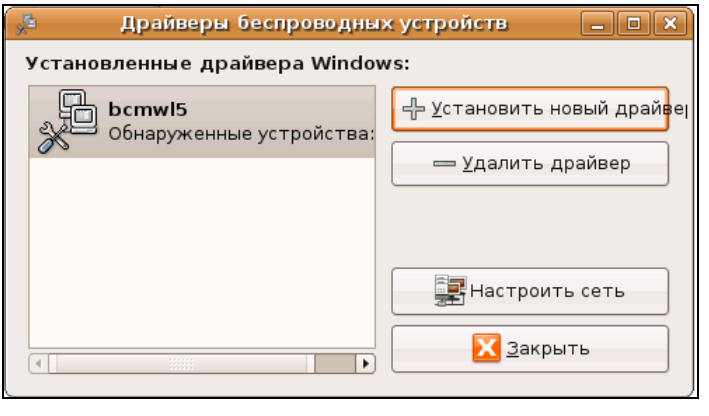


Рис. 13.8. Содержимое каталога, содержащего драйвер моего беспроводного адаптера

После этого драйвер появится в окне программы ndisgtk (рис. 13.8). Перезагрузите компьютер и введите команду `iwconfig wlan0`. Вы увидите примерно такой вывод:

```
iwconfig wlan0
wlan0     IEEE 802.11g  ESSID:"MyHome.Net"
          Mode:Managed  Frequency:2.462 GHz Access Point: xx:xx:xx:xx:xx:xx
```

Bit Rate:54 Mb/s Tx-Power:10 dBm Sensitivity=0/3
RTS thr:4096 B Fragment thr:4096 B
Power Management:off
Link Quality:100/100 Signal level:-42 dBm Noise level:-128 dBm
Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
Tx excessive retries:0 Invalid misc:0 Missed beacon:0

Теперь вам нужно щелкнуть правой кнопкой на значке сетевого соединения (он находится возле значка регулятора громкости и системной даты) и выбрать команду **Установить беспроводную сеть**. Используя открывшееся окно, вы без проблем подключитесь к сети.

13.4. Возможные осложнения

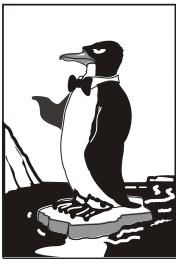
У вас могут возникнуть сложности, как при установке драйвера, так и при настройке самого беспроводного интерфейса. Сначала рассмотрим набор команд, которые могут помочь настроить драйвер:

- `uname -a` — получить версию ядра. При установке настоящих Linux-драйверов (а не Windows-драйверов через ndiswrapper) нужно, чтобы *модуль* (так в Linux называются драйверы) был откомпилирован для соответствующей версии ядра;
- `lspci`, `lsusb`, `lshw` — помогают идентифицировать ваше устройство и выводят, соответственно, список PCI-устройств, список USB-устройств и список оборудования компьютера;
- `lsmod` — выводит список загруженных модулей (драйверов устройств).

Если у вас не получается установить драйвер имеющегося у вас устройства, можно попытаться приобрести и установить другой USB-адаптер, но перед покупкой убедитесь, что сможете его настроить.

Для настройки сетевого интерфейса пригодятся следующие команды:

- `iwconfig` — просмотреть информацию обо всех беспроводных интерфейсах;
- `iwlist scan` — найти беспроводные сети;
- `sudo dhclient wlan0` — обновить IP-адрес и другие сетевые параметры интерфейса wlan0 (имя может быть другим), предварительно получив их от DHCP-сервера;
- `route` — просмотр таблицы маршрутизации;
- `sudo /etc/init.d/networking restart` — перезапуск сети;
- `dmesg | less` — просмотреть сообщения ядра;
- `sudo killall NetworkManager` — остановить NetworkManager;
- `iwevent` — просмотреть события беспроводной сети;
- `sudo /etc/init.d/dbus restart` — перезапустить все сетевые демоны.



Глава 14

Маршрутизация

14.1. Выбор маршрута, или краткое введение в маршрутизацию

Для начала уясним, что *маршрутизация* — это процесс перенаправления пакета по сетям, находящимся между отправителем и получателем. Представьте, что вам нужно поехать в гости к другу в город, где вы никогда не были. Понимаю, что на дворе XXI век, и GPS-навигатор больше не принадлежность Джеймса Бонда, но все же о навигаторах на минуту забудем. Итак, вам нужно выяснить, как проехать в город, где живет ваш друг. Если вы сами живете в относительно большом городе, то первым делом следует узнать, как выехать из своего города, — можно начать движение в произвольном направлении, но потом, чтобы вырулить на нужную дорогу, придется проехать лишний путь, чего бы не хотелось. Поэтому спрашиваем, например, у таксиста, в каком направлении ехать. После того как вы выбрались из своего города и знаете примерное направление, вы себе спокойно едете, пока не начнете сомневаться в правильности маршрута. Тогда вы остановитесь на придорожной АЗС или посту ДПС и узнаете, куда вам ехать дальше. Возможно, придется проехать еще через несколько городов и в каждом городе спросить, куда ехать. Можно и не спрашивать — если есть знаки. Одним словом, либо человек, либо дорожный знак укажут вам дорогу. Когда вы приедете в город друга, вам надо будет узнать, где находится улица, на которой он живет. А когда вы окажетесь на нужной вам улице, наверняка попросите прохожих подсказать, где находится дом с искомым номером.

Маршрутизация пакетов выполняется примерно так же. В приведенном примере "пакетом" были именно вы, а роль маршрутизаторов играли люди, которые подсказывали, куда вам ехать.

В TCP/IP-сетях информация о маршрутах имеет вид правил — например, чтобы добраться к сети А, нужно отправить пакеты через компьютер Д. Ничего удивительного и необычного — примерно так же выглядит и информация о маршрутах на дороге: чтобы доехать до города А нужно проехать через город Д. Кроме набора маршрутов существует также и стандартный маршрут — по нему отправляют пакеты, предназначенные для отправки в сеть, маршрут к которой явно не указан. Компьютер, на который отправляются эти пакеты, называется *шлюзом по умолчанию* (default gateway). Получив пакет, шлюз решает, что с ним сделать: или отправить

дальше, если ему известен маршрут в сеть получателя пакета, или же уничтожить пакет, как будто бы его никогда и не было. В общем, что сделать с пакетом — это личное дело шлюза по умолчанию, все зависит от его набора правил маршрутизации. Наше дело маленькое — отправить пакет на шлюз по умолчанию.

Данные о маршрутах хранятся в таблице маршрутизации ядра Linux. Каждая запись этой таблицы содержит несколько параметров: адрес сети назначения, сетевую маску и т. д. Если пакет не удалось отправить ни по одному маршруту (в том числе и по стандартному), отправителю пакета передается ICMP-сообщение "сеть недоступна" (network unreachable). Далее мы подробно рассмотрим работу с таблицей маршрутизации ядра.

14.2. Таблица маршрутизации ядра. Установка маршрута по умолчанию

Для просмотра таблицы маршрутизации используются команды `netstat -r` и `netstat -rn`. Можно также по старинке воспользоваться командой `route` без параметров. Разница между командами `netstat -r` и `netstat -rn` заключается в том, что параметр `-rn` запрещает поиск доменных имен в DNS, поэтому все адреса будут представлены в числовом виде (подобно команде `route` без параметров). А вот разница между выводом `netstat` и `route` заключается в представлении маршрута по умолчанию (`netstat` выводит адрес 0.0.0.0, а `route` — метку `default`) и в названии полей самой таблицы маршрутизации. На рис. 14.1 изображен вывод команд `netstat -rn` и `route`.

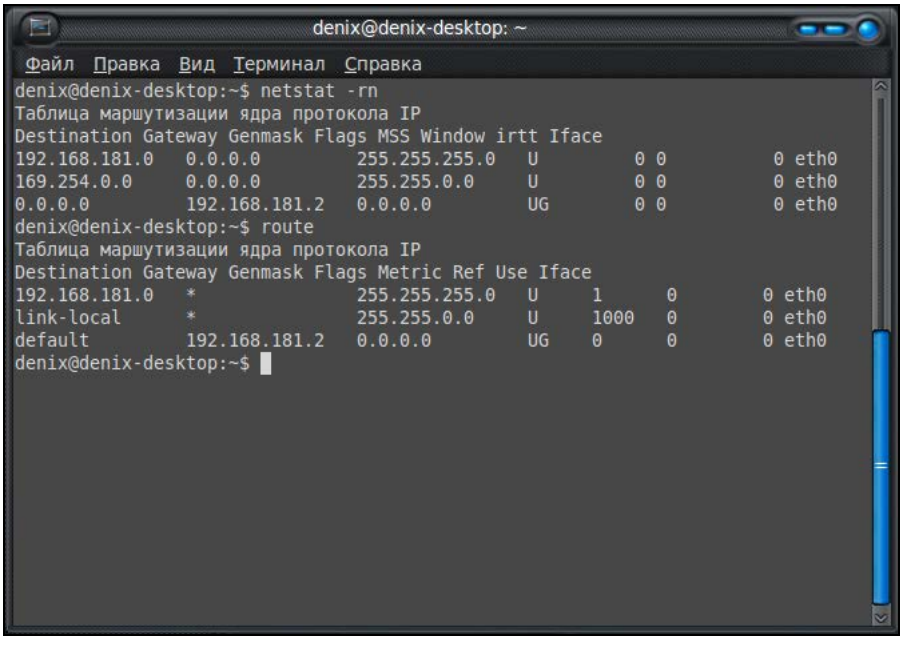


Рис. 14.1. Команды `netstat -rn` и `route`

Какую команду использовать — решать вам. Раньше я использовал `route` и для просмотра, и для редактирования таблицы маршрутизации. Теперь для просмотра таблицы я использую команду `netstat -rn`, а для ее изменения — команду `route`.

На рис. 14.1 представлены две сети 192.168.181.0 и 169.254.0.0 — обе на интерфейсе eth0. Такая ситуация сложилась из-за особенностей NAT/DHCP виртуальной машины VMware, в которой была запущена Linux. В реальных условиях обычно будет по одной подсети на одном интерфейсе. С другой стороны, рис. 14.1 демонстрирует поддержку VLAN, когда один интерфейс может использоваться двумя подсетями. Шлюз по умолчанию — компьютер с адресом 192.168.181.2, о чем свидетельствует таблица маршрутизации.

Поля таблицы маршрутизации представлены в табл. 14.1.

Таблица 14.1. *Поля таблицы маршрутизации*

Поле	Описание
`Destination`	Адрес сети назначения
`Gateway`	Шлюз по умолчанию
`Genmask`	Маска сети назначения
`Flags`	Поле `Flags` содержит флаги маршрута: • `U` — маршрут активен; • `H` — маршрут относится не к сети, а к хосту; • `G` — данная машина является шлюзом, поэтому при обращении к ней нужно заменить MAC-адрес машины получателя на MAC-адрес шлюза (если MAC-адрес получателя почему-то известен); • `D` — динамический маршрут, установлен демоном маршрутизации; • `M` — маршрут, модифицированный демоном маршрутизации; • `C` — запись кэширована; • `!` — запрещенный маршрут
`Metric`	Метрика маршрута, то есть расстояние к цели в хопах (переходах). Один хоп (переход) означает один маршрутизатор
`Ref`	Количество ссылок на маршрут. Не учитывается ядром Linux, но в других операционных системах, например, в FreeBSD, вы можете столкнуться с этим полем
`Use`	Содержит количество пакетов, прошедших по этому маршруту
`Iface`	Используемый интерфейс
`MSS`	Максимальный размер сегмента (Maximum Segment Size) для TCP-соединений по этому маршруту
`Window`	Размер окна по умолчанию для TCP-соединений по этому маршруту

Таблица 14.1 *(окончание)*

Поле	Описание
`irtt`	Протокол TCP гарантирует надежную доставку данных между компьютерами. Такая гарантия обеспечивается повторной отправкой пакетов, если они были потеряны. При этом ведется счетчик времени — сколько нужно ждать, пока пакет дойдет до назначения и придет подтверждение о получении пакета. Если время вышло, а подтверждение таки не было получено, то пакет отправляется еще раз. Это время и называется round-trip time (время "путешествия туда-обратно"). Параметр `irtt` — это начальное время rtt. В большинстве случаев подходит значение по умолчанию, но для некоторых медленных сетей, например, для сетей пакетного радио, значение по умолчанию слишком короткое, что вызывает ненужные повторы. Параметр `irtt` можно увеличить командой route. По умолчанию его значение — 0

Добавить маршрут в таблицу маршрутизации можно статически (с помощью команды `route`), динамически или комбинированно (например, статические маршруты добавляются при запуске системы, а динамические — по мере работы системы). Статические маршруты добавляются, как правило, командой `route`, запущенной из сценария инициализации системы. Например, следующая команда задает шлюз по умолчанию для интерфейса eth0:

```
# route add default gw 192.168.181.2 eth0
```

Но после перезагрузки системы добавленная нами запись исчезнет из таблицы маршрутизации. Можно добавить данную команду в сценарии инициализации системы, но это будет некорректно. Есть более корректный способ установки шлюза по умолчанию. В Fedora, Red Hat и других совместимых с ними дистрибутивах (CentOS, ASP Linux) нужно отредактировать файл /etc/sysconfig/network. Переменная `GATEWAY` содержит IP-адрес шлюза по умолчанию. Пример этого файла приведен в листинге 14.1.

Листинг 14.1. Файл /etc/sysconfig/network: основные сетевые параметры в Fedora

```
NETWORKING=yes
FORWARD_IPV4=yes
HOSTNAME=den.dkws.org.ua
GATEWAY=0.0.0.0
```

> ***Примечание***
>
> С некоторыми конфигурационными файлами, рассматриваемыми в этой главе, вы уже знакомы *(см., например, главу 11)*, но здесь они рассматриваются в разрезе маршрутизации.

Параметр `NETWORKING` определяет, будет ли включена поддержка сети (`yes` — поддержка сети включена, `no` — выключена). Параметр `FORWARD_IPV4` определяет, будет ли включено перенаправление пакетов. На компьютере, являющемся шлю-

зом, данный параметр должен быть включен (значение `yes`), на остальных компьютерах сети — выключен (значение `no`).

Параметр `HOSTNAME` задает имя узла, `GATEWAY` — шлюз по умолчанию. Если компьютер является шлюзом, то обычно для этого параметра устанавливается IP-адрес 0.0.0.0.

В SUSE для задания шлюза по умолчанию нужно отредактировать файл /etc/route.conf или /etc/sysconfig/network/routes (современные версии openSUSE). В него нужно добавить строку вида:

```
default       адрес              [маска]       [интерфейс]
```

Например:

```
default       192.168.181.2
```

Маску и интерфейс указывать необязательно. В этом же файле можно указать все остальные маршруты, то есть по сути, данный файл хранит таблицу маршрутизации. Маршрут по умолчанию, как правило, указывается последним. Пример файла конфигурации /etc/sysconfig/network/routes (/etc/route.conf) приведен в листинге 14.2.

Листинг 14.2. Файл /etc/route.conf

```
#
# /etc/sysconfig/network/routes (/etc/route.conf)
#
# Данный файл содержит описание статических маршрутов
#
# Назначение Шлюз            Маска           Устройство
#
192.168.0.0   0.0.0.0       255.255.255.128  eth0
default       192.168.0.1
```

Кроме файла route.conf в SUSE вы можете редактировать файл /etc/rc.config, содержащий всю информацию о сетевых интерфейсах. Здесь важно отметить, что речь идет о старых версиях SUSE. В *главе 11* мы рассматривали конфигурационные файлы современных версий openSUSE.

В Debian и Ubuntu вам нужно редактировать файл /etc/network/interfaces. Шлюз по умолчанию задается параметром `gateway`. В листинге 14.3 приведен пример файла /etc/network/interfaces.

Напомню, что подробно синтаксис этого файла описан в моей статье по адресу: **http://dkws.org.ua/index.php?page=show&file=a/ubuntu/network-interfaces**. Но позволю себе несколько комментариев. Как видно из листинга 14.3, производится конфигурация интерфейса eth0, IP-адрес задается статически (static), присваивается IP-адрес 192.168.1.11, маска 255.255.255.0. Шлюз по умолчанию — это компьютер с IP-адресом 192.168.1.1.

Листинг 14.3. Файл /etc/network/interfaces

```
iface eth0 inet static
address 192.168.1.11
netmask 255.255.255.0
gateway 192.168.1.1
```

14.3. Изменение таблицы маршрутизации. Команда *route*

Мы уже знакомы с командой `route`, но использовали ее пока для просмотра таблицы маршрутизации. Сейчас мы научимся ее использовать для изменения таблицы маршрутов.

Маршрутизация осуществляется на сетевом уровне модели OSI. Когда маршрутизатор получает пакет, предназначенный для другого узла, его IP-адрес получателя сравнивается с записями в таблице маршрутизации. Если есть хотя бы частичное совпадение с каким-то маршрутом из таблицы, пакет отправляется по IP-адресу шлюза, связанного с данным маршрутом.

Если совпадений не найдено (то есть вообще нет маршрута, по которому можно было бы отправить пакет), тогда пакет отправляется на шлюз по умолчанию, если таковой задан в таблице маршрутизации. Если шлюза по умолчанию нет, отправителю пакета посылается ICMP-сообщение "сеть недоступна" (network unreachable).

Команда `route` за один вызов может добавить или удалить только один маршрут. Другими словами, вы не можете сразу добавить или удалить несколько маршрутов. Формат вызова `route` следующий:

```
# route [операция] [тип] адресат gw шлюз [метрика] [dev интерфейс]
```

> ***Примечание***
>
> Команды добавления/удаления маршрута нужно вводить от имени root. В современных системах непосредственно под именем пользователя root входить не обязательно — для получения root-доступа можно использовать команды `sudo` или `su`.

Параметр `операция` может принимать два значения: `add` (добавить маршрут) и `del` (удалить маршрут). Параметр `тип` необязательный, он задает тип маршрута: `-net` (маршрут к сети), `-host` (маршрут к узлу) или `default` (маршрут по умолчанию). Параметр `адресат` содержит адрес сети (если задается маршрут к сети), адрес узла (при добавлении маршрута к сети) или вообще не указывается, если задается маршрут по умолчанию.

Параметр `шлюз` задает IP-адрес (или доменное имя) шлюза. Последние два параметра — `метрика` и `dev` необязательны. Параметр `метрика` задает максимальное число переходов (через маршрутизаторы) на пути к адресату (именно в Linux он необязательный, в отличие от других ОС). Параметр `dev` имеет смысл указывать, если в системе установлено несколько сетевых интерфейсов и нужно указать, через какой именно сетевой интерфейс следует отправить пакеты по указанному маршруту.

Команда удаления маршрута выглядит так:

```
# route del адрес
```

В других UNIX-системах имеется параметр `-f`, удаляющий все маршруты (`route -f`), но в Linux такого параметра нет. Следовательно, для очистки всей таблицы маршрутизации вам нужно будет ввести серию команд `route del`. Изменять таблицу маршрутизации нужно только, зарегистрировавшись на компьютере локально. При удаленной регистрации (например, по ssh) легко ошибочно удалить маршрут, по которому вы вошли в систему. О последствиях такого действия, думаю, говорить не нужно.

Примеры использования команды route:

- `route add -net 192.76.16.0 netmask 255.255.255.0 dev eth0`

  Добавляет маршрут к сети 192.76.16.0 (сеть класса C, о чем свидетельствует сетевая маска, заданная параметром `netmask`) через устройство eth0. Шлюз не указан, просто все пакеты, адресованные сети 192.76.16.0, будут отправлены на интерфейс eth0.

- `route add -net 192.16.16.0 netmask 255.255.255.0 gw 192.76.16.1`

  Добавляет маршрут к сети 192.16.16.0 через маршрутизатор 192.76.16.1. Сетевой интерфейс указывать не обязательно, но можно и указать при особом желании.

- `route add default gw gate1`

  Добавляет маршрут по умолчанию. Все пакеты будут отправлены компьютеру с именем `gate1`. Обратите внимание — мы указываем доменное имя узла вместо IP-адреса.

- `route add -net 10.1.0.0 netmask 255.0.0.0 reject`

  Добавляет запрещающий маршрут. Отправка пакетов по этому маршруту (в сеть 10.1.0.0) запрещена.

Итак, мы добавили необходимые маршруты, пропинговали удаленные узлы — все работает. Теперь нужно сохранить установленные маршруты, чтобы они были доступны при следующей загрузке системы. Для этого в openSUSE следует отредактировать файл /etc/sysconfig/network/routes (/etc/route.conf — в старых версиях). Мы уже рассматривали этот файл (см. листинг 14.2), поэтому переходим сразу к другим дистрибутивам.

В Fedora/CentOS/ASP Linux (и других Red Hat-совместимых дистрибутивах) статические маршруты хранятся в файле /etc/sysconfig/static-routes. Строки в этом файле имеют вид:

```
any net адрес_сети netmask маска gw адрес_шлюза
```

Здесь `any` означает любой интерфейс. Можно указать конкретный интерфейс, например:

```
eth0 net 192.168.2.0 netmask 255.255.255.0 gw 192.168.1.1
```

Файл /etc/sysconfig/static-routes по умолчанию отсутствует, при необходимости его нужно создать самостоятельно.

В Debian/Ubuntu статические маршруты прописываются вместе с конфигурацией сетевого интерфейса в файле /etc/network/interfaces. С помощью параметров `up` и `down` этого файла можно задать команды, которые будут выполняться при "поднятии"

(up) и "закрытии" (down) интерфейса. После параметров `up` и `down` может следовать любая Linux-команда. Обычно это команда `route`. Например, при запуске интерфейса eth0 будет добавлен статический маршрут к сети 192.168.3.0 через шлюз 192.168.1.2:

```
up route add -net 192.168.3.0 netmask 255.255.255.0 gw 192.168.1.2
```

Можно также добавить маршрут по умолчанию:

```
up route add default gw 192.168.1.2
```

При "закрытии" интерфейса следует удалить маршруты, которые использовали этот интерфейс, для чего используется параметр `down`:

```
down route del default gw 192.168.1.2
down route del -net 192.168.3.0
```

Подробное описание файла /etc/network/interfaces вы найдете по адресу: **http://www.dkws.org.ua/index.php?page=show&file=a/ubuntu/network-interfaces**.

14.4. Включение IPv4-переадресации, или превращение компьютера в шлюз

Основное предназначение шлюза (маршрутизатора) — это пересылка (forwarding) пакетов. Чтобы включить пересылку пакетов протокола IPv4 (IPv4 forwarding), нужно записать значение 1 в файл /proc/sys/net/ipv4/ip_forward:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

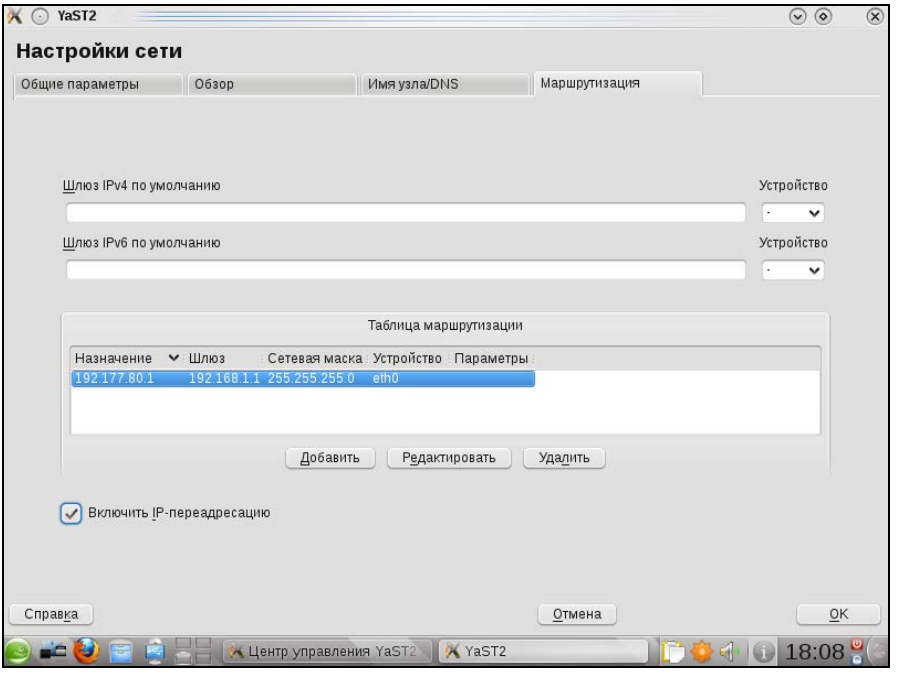


Рис. 14.2. Включение IP-переадресации в openSUSE 11.3

Но включить пересылку мало — нужно еще сохранить это значение, иначе при перезагрузке будет восстановлено значение по умолчанию (0). Для этого нужно в файл /etc/sysctl.conf добавить строку:

```
net.ipv4.ip_forward=0
```

В некоторых дистрибутивах, например, в openSUSE, можно воспользоваться конфигуратором, вызываемым командой меню **YaST | Сетевые настройки | Маршрутизация** (рис. 14.2). В некоторых других переадресацию можно включить путем редактирования конфигурационных файлов сети — например, в Fedora (см. листинг 14.1).

Подробно о настройке системы с помощью псевдофайловой системы /proc было рассказано в *главе 9*.

14.5. Протоколы маршрутизации

Вся следующая часть главы подается больше "для общего развития". Практически все, что на 90% необходимо знать администратору, ранее уже изложено. Но в некоторых сложных случаях необходимо применение специфического протокола маршрутизации.

14.5.1. Routing Information Protocol

RIP (Routing Information Protocol, протокол маршрутной информации) — довольно старый протокол, разработанный компанией Xerox и адаптированный для IP-сетей. Данный протокол используется демоном routed.

Протокол RIP разрабатывался в те времена, когда компьютеры были размерами с холодильник и стоили сотни тысяч долларов. Понятно, что позволить себе компьютер мог не каждый пользователь, да и не было такой необходимости. А необходимость была только у крупных компаний, но, учитывая стоимость компьютеров, в большом количестве их никто все равно не покупал. Поэтому сети тех времен были небольшими. Протокол RIP считает все узлы, находящиеся на расстоянии пятнадцати и более переходов, недоступными. Следовательно, для крупных сетей, где между двумя компьютерами могут находиться 15 и более маршрутизаторов, применять RIP нельзя.

Обычно RIP используется в локальных сетях, а вот для глобальной маршрутизации применяются более продвинутые протоколы. Протокол RIP поддерживается по умолчанию операционными системами UNIX (точнее, всеми UNIX-подобными системами) и Linux.

14.5.2. RIP-2

RIP-2 — это усовершенствованная версия протокола RIP. Самое важное усовершенствование — передача вместе с адресом следующего перехода сетевой маски. А это упрощает управление сетями, где есть подсети и используется протокол (Classless InterDomain Routing, бесклассовая адресация).

Демон для протокола RIP-2 поддерживает протокол RIP. Вообще, RIP-2 предпочтительнее, чем первая версия, но в Linux по умолчанию отсутствует поддержка этого протокола.

14.5.3. Open Shortest Path First

Протокол OSPF (протокол обнаружения кратчайшего маршрута) — протокол динамической маршрутизации, основанный на технологии отслеживания состояния канала. Используется для нахождения кратчайшего пути (маршрута).

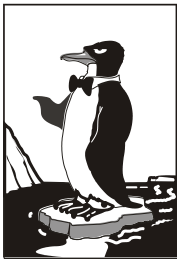
Протокол был разработан в 1988 году. Используется в крупных сетях со сложной топологией. По сравнению с RIP-2 имеет много преимуществ. В частности, OSPF работает быстрее, поскольку нет необходимости в выжидании многократных тайм-аутов по 30 секунд каждый, как в случае с RIP-2. Также быстро обнаруживаются отказавшие маршрутизаторы, и протокол изменяет топологию сети так, чтобы пакеты пошли по другому маршруту — "в обход".

Кроме перечисленных протоколов, возможно, вам придется столкнуться еще с "пограничными протоколами":

- IGRP (Interior Gateway Routing Protocol) — протокол маршрутизации, разработанный фирмой Cisco для своих многопротокольных маршрутизаторов;
- EIGRP (Enhanced Interior Gateway Routing Protocol) — расширенная версия протокола IGRP;
- BGP (Border Gateway Protocol, протокол граничного шлюза) — основной протокол динамической маршрутизации в Интернете. Предназначен для обмена информацией не между отдельными маршрутизаторами, а между целыми автономными системами (AS).

В книге эти протоколы мы рассматривать не будем, поскольку первые два относятся сугубо к продукции Cisco, а последний слишком сложный, и его рассмотрение выходит за рамки этой книги.

Глава 15



Брандмауэры iptables и ebtables

15.1. Что такое брандмауэр

Брандмауэр (он же firewall, бастион, межсетевой экран) предназначен для защиты внутренней сети (или даже одного компьютера, напрямую подключенного к Интернету) от вторжения извне. С помощью брандмауэра вы можете контролировать доступ пользователей Интернета к узлам вашей внутренней сети. Также можно контролировать доступ локальных пользователей к ресурсам Интернета — например, вы можете запретить им посещать определенные узлы с целью экономии трафика.

Прежде чем перейти к настройке межсетевого экрана, определимся с терминологией и, в частности, с понятием *шлюз*. Шлюзом называется компьютер, предоставляющий компьютерам локальной сети доступ к Интернету. Шлюз выполняет как бы маршрутизацию пакетов. Но не нужно путать шлюз с обычным маршрутизатором. Маршрутизатор осуществляет простую пересылку пакетов, поэтому его можно использовать для соединения сетей одного типа, например, локальной и локальной, глобальной и глобальной. А шлюз служит для соединения сетей разных типов, например, локальной и глобальной, как в нашем случае. Конечно, сейчас можно встретить маршрутизаторы с функцией шлюза, но это уже, скорее, аппаратные шлюзы, чем простые маршрутизаторы. Поэтому часто термины маршрутизатор и шлюз употребляются как синонимы, хотя это не совсем так.

Сложность в соединении сетей разных типов заключается в различной адресации. Как мы знаем, в локальной сети обычно используются локальные адреса, которые не допустимы в Интернете, например: 192.168.*.* (сеть класса C), 10.*.*.* (сеть класса A) и 172.16.*.*–172.31.*.* (класс B). Поэтому шлюз должен выполнить преобразование сетевого адреса (Network Address Translation, NAT). Суть такого преобразования в следующем. Предположим, у нас есть шлюз и локальная сеть с адресами 192.168.*.*. Реальный IP-адрес (который можно использовать в Интернете) есть только у шлюза, пусть это 193.254.219.1. У всех остальных компьютеров — локальные адреса, поэтому при всем своем желании они не могут обратиться к интернет-узлам. У нашего шлюза два сетевых интерфейса. Один из них, пусть ppp0, используется для подключения к Интернету. Его IP-адрес, как уже было отмечено, 193.254.219.1. Для подключения к локальной сети используется другой сетевой интерфейс — eth0 (сетевая плата) с IP-адресом 192.168.1.1. Таким образом, все узлы нашей локальной сети используют в качестве шлюза компьютер с адресом

192.168.1.1, то есть все запросы из сети будут переданы на узел 192.168.1.1. Запросы передаются в виде:

```
Назначение: IP-адрес узла Интернета
Источник: адрес компьютера локальной сети, пусть 192.168.1.10
```

Наш шлюз принимает запрос и перезаписывает его так:

```
Назначение: IP-адрес узла Интернета
Источник: 193.254.219.1
```

То есть шлюз подменяет адрес источника, устанавливая в качестве этого адреса свой реальный IP-адрес, иначе бы любой интернет-узел не принял бы запрос с локального адреса. Получив ответ от узла, он направляет его нашему узлу:

```
Назначение: 192.168.1.10
Источник: IP-адрес узла Интернета
```

Нашему локальному узлу "кажется", что он получил ответ непосредственно от узла Интернета, а на самом деле ответ приходит от шлюза.

Теперь, когда мы разобрались с теорией, самое время перейти к практике.

15.2. Цепочки и правила

Основная задача брандмауэра — это фильтрация пакетов, которые проходят через сетевой интерфейс. При поступлении пакета брандмауэр анализирует его и затем принимает решение: принять пакет (ACCEPT) или избавиться от него (DROP). Брандмауэр может выполнять и более сложные действия, но часто ограничивается именно этими двумя действиями.

Прежде чем брандмауэр примет решение относительно пакета, пакет должен пройти по цепочке правил. Каждое правило состоит из условия и действия (цели). Если пакет соответствует условию правила, то выполняется указанное в правиле действие. Если пакет не соответствует условию правила, он передается следующему правилу. Если же пакет не соответствует ни одному из правил цепочки, выполняется действие по умолчанию.

Вроде бы все понятно, но чтобы лучше закрепить знания, рассмотрим табл. 15.1, демонстрирующую принцип работы цепочки правил.

Таблица 15.1. *Цепочка правил*

Номер правила	Условие	Действие (цель)
1	Пакет от 192.168.1.0	ACCEPT
2	Пакет от 192.168.0.0	DROP
3	Пакет для 192.168.2.0	ACCEPT
DEFAULT	*	DROP

Предположим, что пакет пришел из сети 192.168.4.0 для узла 192.168.1.7 (это наша сеть). Пакет не соответствует первому правилу (отправитель не из сети 192.168.1.0), поэтому он передается правилу 2. Пакет не соответствует и этому пра-

вилу. Пакет адресован компьютеру 192.168.1.7, а не компьютеру из сети 192.168.2.0, поэтому он не соответствует третьему правилу. Брандмауэру остается применить правило по умолчанию — пакет будет отброшен (действие DROP).

Цепочки правил собираются в три основные таблицы:

- filter — таблица фильтрации, основная таблица;
- nat — таблица NAT, используется при создании пакетом нового соединения;
- mangle — используется, когда нужно произвести специальные действия над пакетом.

> ***Примечание***
>
> Ранее брандмауэр в Linux поддерживал только цепочки правил и назывался ipchains, сейчас брандмауэр поддерживает и цепочки правил, и таблицы цепочек, и называется iptables. Это примечание сделано, чтобы вы понимали разницу между старым брандмауэром ipchains (ядра 2.2 и ниже) и новым iptables (ядра 2.4 и выше).

Если необходимо, вы можете создать собственные таблицы. В состав таблицы входят три цепочки:

- INPUT — для входящих пакетов;
- OUTPUT — для исходящих пакетов;
- FORWARD — для пересылаемых (транзитных) пакетов.

Над пакетом можно выполнить следующие действия:

- <имя цепочки> — пакет будет отправлен для обработки в цепочку с указанным именем;
- ACCEPT — принять пакет;
- DROP — отбросить пакет, после этого пакет удаляется, больше над ним не выполняется каких-либо действий;
- MASQUERADE — скрыть IP-адрес пакета.

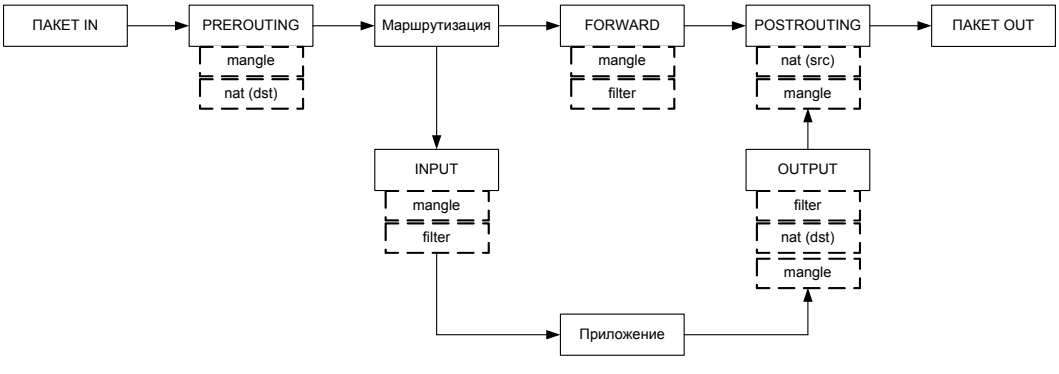


Рис. 15.1. Схема обработки пакета

Это не все действия, но пока нам больше знать не нужно. На рис. 15.1 изображена схема обработки пакета. Входящий пакет (на схеме ПАКЕТ IN) поступает

в цепочку PREROUTING таблицы mangle. После чего (если он не был отброшен правилами таблицы mangle) пакет обрабатывается правилами цепочки PREROUTING, но таблицы nat. На этом этапе проверяется, нужно ли модифицировать назначение пакета (этот вид NAT называется Destination NAT, DNAT).

Затем пакет может быть направлен либо в цепочку INPUT (если получателем пакета является этот компьютер), либо в цепочку FORWARD (если пакет нужно передать другому компьютеру).

Если получатель компьютера — сам шлюз (на нем может быть запущен, например, почтовый или Web-сервер), то пакет сначала обрабатывается правилами цепочки INPUT таблиц mangle и filter. Если пакет не был отброшен, он передается приложению (например, почтовому серверу). Приложение получило пакет, обработало его и отправляет ответный пакет. Этот пакет обрабатывается цепочкой OUTPUT таблиц mangle, nat и filter. Далее пакет отправляется на цепочку POSTROUTING и обрабатывается правилами таблиц mangle и nat.

Если пакет нужно передать другому компьютеру, то он обрабатывается правилами цепочки FORWARD таблиц mangle и filter, а после этого к нему применяются правила цепочки POSTROUTING. На этом этапе используется подмена источника пакета (этот вид NAT называется Source NAT, SNAT).

После всех правил пакет "выжил"? Тогда он становится исходящим пакетом (на схеме ПАКЕТ OUT) и отправляется в сеть.

15.3. Использование брандмауэра iptables

Теперь, когда мы разобрались с правилами и цепочками, самое время научиться использовать брандмауэр iptables. Для себя сразу определитесь, на что вы его настраиваете. Можно настраивать его просто как брандмауэр, защищающий локальный компьютер от всевозможных атак. А можно настраивать шлюз сети, предоставляющий всем остальным компьютерам сети доступ к Интернету. В последнем случае нужно включить IP-переадресацию (IPv4-forwarding). О том, как это сделать, было сказано в *главе 14*. В большинстве случаев хватит вот такой команды:

```
sudo sysctl -w net.ipv4.ip_forward="1"
sudo echo 1 > /proc/sys/net/ipv4/ip_forward
```

Примечание

Вот только не нужно думать, что она будет работать во всех дистрибутивах. Еще раз отмечу, что подробные инструкции вы найдете в *главе 14*. А будет ли работать данная команда у вас, зависит не только от вашего дистрибутива, но и от его настройки. Например, в Ubuntu данная команда выполнится превосходно, а вот в openSUSE, скорее всего, она не выполнится, поскольку ваша учетная запись не включена в файл /etc/sudoers, следовательно, вы не имеете права выполнять команду sudo.

Сейчас можно перейти к iptables. Для изменения правил брандмауэра нужны полномочия root, поэтому все команды iptables следует вводить или через команду `sudo` (для этого ваш пользователь должен иметь право использовать `sudo`), или с предварительно полученными полномочиями root (команда `su`).

Для добавления правила в цепочку используется команда:

```
sudo iptables -A цепочка правило
```

Например:

```
sudo iptables -A INPUT правило
```

Данная команда добавит правило в цепочку INPUT таблицы filter (это таблица по умолчанию). Если вы желаете добавить правило в другую таблицу, нужно указать ее в параметре `-t`:

```
sudo iptables -t таблица -A цепочка правило
```

Например:

```
sudo iptables -t nat -A INPUT правило
```

Действие по умолчанию задается ключом `-P`:

```
sudo iptables -P INPUT DROP
```

Обычно устанавливаются вот такие действия по умолчанию:

```
sudo iptables -P INPUT DROP
sudo iptables -P FORWARD ACCEPT
sudo iptables -P OUTPUT DROP
```

Параметры фильтрации пакетов приведены в табл. 15.2. Но прежде рассмотрим фазы установки TCP-соединения. Итак, соединение устанавливается в три этапа (фазы). Сначала первый компьютер отправляет второму компьютеру SYN-пакет, запрашивая открытие соединения. Второй компьютер отправляет ему подтверждение SYN-пакета — ACK-пакет. После этого соединение считается установленным (ESTABLISHED). Открытое, но не установленное соединение (когда компьютеры обмениваются пакетами SYN-ACK) называется новым (NEW). Слова в скобках я привел не просто так, а для понимания табл. 15.2. В таблице, при описании параметров, будут указываться не полные команды iptables, а только их фрагменты, имеющие отношение к тому или иному параметру.

Таблица 15.2. *Параметры фильтрации пакетов*

Параметр	Описание
`--source`	Позволяет указать источник пакета. Можно указывать как доменное имя компьютера (den.dkws.org.ua), так и его IP-адрес (192.156.1.1) и даже набор адресов (192.168.1.0/255.255.255.0). Пример: `iptables -A FORWARD --source 192.168.1.11 …`
`--destination`	Задает назначение (адрес получателя) пакета. Синтаксис такой же, как и у `--source`

`-protocol` (или `-p`)	Задает протокол. Чаще всего работают с tcp, icmp или udp, но можно указать любой протокол, определенный в файле /etc/protocols. Также можно указать all, что означает все протоколы. Примеры:
	`iptables -A FORWARD -protocol tcp …`
	`iptables -A FORWARD -p tcp …`

`--source-port` (или `--sport`)	Определяет порт отправителя. Данная опция может использоваться только вместе с параметром `-p`. Например: `iptables -A FORWARD -p tcp -source-port 23 …`
`--destination-port` (или `--dport`)	Задает порт-назначение. Опция возможна только с параметром `-p`. Синтаксис такой же, как и в случае с `-source-port`

Таблица 15.2 *(окончание)*

Параметр	Описание
`-state`	Позволяет отфильтровать пакеты по состоянию. Параметр `-state` доступен только при загрузке модуля state с помощью другого параметра `-m state`. Состояния пакета: • NEW — новое соединение (еще не установленное); • ESTABLISHED — установленное соединение; • RELATED — пакеты, которые не принадлежат соединению, но связаны с ним; • INVALID — неопознанные пакеты. Пример: `iptables -A FORWARD -m state -state RELATED,INVALID`
`-in-interface` (или `-i`)	Определяет интерфейс, по которому прибыл пакет. Пример: `iptables -A FORWARD -i eth1`
`-out-interface` (или `-o`)	Определяет интерфейс, по которому будет отправлен пакет: `iptables -A FORWARD -o ppp0`
`-tcp-flags`	Производит фильтрацию по TCP-флагам (man iptables)

Ранее мы познакомились с основными действиями iptables. В табл. 15.3 представлены все действия iptables (цели iptables). Действие задается параметром `-j`.

Таблица 15.3. *Цели iptables*

Действие	Описание
ACCEPT	Принять пакет. При этом пакет уходит из этой цепочки и передается дальше
DROP	Уничтожить пакет
REJECT	Уничтожает пакет и сообщает об этом отравителю с помощью ICMP-сообщения. Параметр `-reject-with` позволяет уточнить тип ICMP-сообщения: • icmp-host-unreachable — узел недоступен; • icmp-net-unreachable — сеть недоступна; • icmp-port-unreachable — порт недоступен; • icmp-proto-unreachable — протокол недоступен. По умолчанию отправляет сообщение о недоступности порта. Но, используя сообщение icmp-host-unreachable, можно сбить злоумышленника с толку. Предположим, что вы просто решили отбрасывать неугодные вам пакеты (действие DROP). Но злоумышленник будет посылать и посылать вам эти пакеты, чтобы брандмауэр только и делал, что занимался фильтрацией и удалением этих пакетов (один из видов атаки на отказ). А если вы ответите сообщением icmp-host-unreachable, то злоумышленник будет думать, что узел недоступен, то есть что компьютер выключен либо он уже достиг своей цели — добился отказа компьютера. С другой стороны, помните, что данное действие порождает ответный ICMP-пакет, что нагружает исходящий канал, который в некоторых случаях (например, одностороннее спутниковое соединение) очень "узкий". Если злоумышленник пришлет вам 1 миллион пакетов, то вы должны будете отправить 1 миллион сообщений в ответ. Подумайте, готовы ли вы к такой нагрузке на исходящий канал

Таблица 15.3 *(окончание)*

Действие	Описание
LOG	Заносит информацию о пакете в протокол. Полезно использовать для протоколирования возможных атак — если вы подозреваете, что ваш узел атакуется кем-то. Также полезно при отладке настроек брандмауэра
RETURN	Возвращает пакет в цепочку, откуда он прибыл. Действие возможно, но лучше его не использовать, так как легко ошибиться и создать непрерывный цикл: вы отправляете пакет обратно, а он опять следует на правило, содержащее цель RETURN
SNAT	Выполняет подмену IP-адреса отправителя (Source NAT). Используется в цепочках POSTROUTING и OUTPUT таблицы nat
DNAT	Выполняет подмену адреса получателя (Destination NAT). Используется только в цепочке POSTROUTING таблицы nat
MASQUERADE	Похож на SNAT, но "забывает" про все активные соединения при потере интерфейса. Используется при работе с динамическими IP-адресами, когда происходит "потеря" интерфейса при изменении IP-адреса. Применяется в цепочке POSTROUTING таблицы nat

15.4. Шлюз своими руками

Создать шлюз в Linux очень просто. Гораздо сложнее правильно его настроить, чтобы шлюз не только выполнял свою непосредственную функцию (то есть передачу пакетов из локальной сети в Интернет и обратно), но и защищал сеть.

В последнее время очень популярны DSL-соединения, поэтому будем считать, что для подключения к Интернету используется именно DSL-соединение (хотя вся разница только в названии интерфейса — ppp0). Вполне может быть, что у вас иная конфигурация. Например, у вас может быть два сетевых интерфейса: eth0 и eth1. Первый "смотрит" в локальную сеть, а второй — подключен к Интернету. Тогда и правила будете формировать, исходя из того, что соединение с Интернетом происходит по интерфейсу eth1.

В случае с DSL-соединением у нас тоже будет два сетевых адаптера. Первый (eth0) будет подключен к локальной сети, а к второму (eth1) будет подключен DSL-модем. Перед настройкой шлюза проверьте, действительно ли это так. Вполне может оказаться, что сетевая плата, к которой подключен DSL-модем, — это интерфейс eth0, а не eth1. В этом случае вам нужно или изменить названия интерфейсов при формировании правил или просто подключить модем к другому сетевому адаптеру.

IP-адрес DSL-соединения будет динамическим (обычно так оно и есть), а вот IP-адрес сетевого адаптера, обращенного к локальной сети, пусть будет 192.168.1.1. Вы можете использовать и другой адрес (адрес должен быть локальным, если только у вас нет подсети с реальными IP-адресами).

Итак, мы настроили локальную сеть, узнали имена сетевых адаптеров, включили IP-переадресацию. Осталось только ввести команду:

```
sudo iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

Установите на всех компьютерах вашей сети IP-адрес 192.168.1.1 в качестве шлюза по умолчанию (можно настроить DHCP-сервер, чтобы не настраивать все компьютеры вручную) и попробуйте пропинговать с любого узла какой-то сайт. Оказывается, вы прочитали всю эту главу ради одной строчки. Так и есть. Но, сами понимаете, на этом настройка шлюза не заканчивается. Нужно еще защитить вашу сеть. Как минимум, вам нужно установить следующие действия по умолчанию:

```
sudo iptables -P INPUT DROP
sudo iptables -P FORWARD ACCEPT
sudo iptables -P OUTPUT DROP
```

Разрешим входящие соединения на шлюз только от узлов нашей внутренней сети 192.168.1.0:

```
sudo iptables –A INPUT –i eth0 --source 192.168.1.0/24 --match state --
state NEW,ESTABLISHED –j ACCEPT
```

Надо также установить правило для цепочки OUTPUT — оно разрешает шлюзу отвечать компьютерам нашей локальной сети:

```
sudo iptables –A OUTPUT –o eth0 --destination 192.168.1.0/24 --match state
--state NEW,ESTABLISHED –j ACCEPT
```

Будьте внимательны при указании имен интерфейсов и IP-адресов. Очень легко запутаться, а потом полчаса разбираться, почему шлюз не работает.

Нам осталось только запретить соединения из Интернета (компьютеры нашей сети смогут устанавливать соединения с серверами Интернета, зато интернет-пользователи не смогут установить соединения с компьютерами нашей сети):

```
sudo iptables –A FORWARD –i eth0 --destination 192.168.1.0/24 --match state
--state ESTABLISHED –j ACCEPT
```

У нас получилась простенькая конфигурация. Компьютеры нашей сети могут выступать инициатором соединения, а интернет-узлы могут передавать данные в нашу сеть только в том случае, если инициатором соединения выступил локальный компьютер.

Но и это еще не все. Как вы уже догадались, поскольку мы не сохранили правила брандмауэра, при перезагрузке компьютера его придется настраивать заново. Поскольку мне не с руки описывать настройку брандмауэра (сохранение и восстановление правил) в каждом дистрибутиве (пусть это будет ваше домашнее задание), рассмотрим универсальный способ. Способ заключается в создании bash-сценария, вызывающего необходимые нам команды настройки iptables. После написания сценария вам останется вызвать его при загрузке системы. А для этого нужно изучить строение системы инициализации в вашем дистрибутиве *(см. главу 17)*.

Вместо того чтобы объяснять вам, как вызвать сценарий, загружающий правила брандмауэра (с этим вы и сами разберетесь), я лучше приведу сценарий (понятно, с комментариями), реализующий более сложную конфигурацию iptables. Данный сценарий будет не только выполнять все функции шлюза, но и защищать сеть от разного рода атак (листинг 15.1). Сценарий лучше сразу поместить в каталог /etc/init.d (это моя вам подсказка) и сделать исполняемым:

```
# touch /etc/init.d/firewall_start
# chmod +x /etc/init.d/firewall_start
```

Листинг 15.1. Сценарий firewall_start

```
# Путь к iptables
IPT="/sbin/iptables"

# Сетевой интерфейс, подключенный к Интернету
INET="ppp0"

# Номера непривилегированных портов
UPORTS="1024:65535"

# Включаем IPv4-forwarding (чтобы не думать, почему шлюз не работает)
echo 1 > /proc/sys/net/ipv4/ip_forward

# Удаляем все цепочки и правила
$IPT -F
$IPT -X

# Действия по умолчанию.
$IPT -P INPUT DROP
$IPT -P FORWARD ACCEPT
$IPT -P OUTPUT DROP

# Разрешаем все пакеты по интерфейсу lo (обратная петля)
$IPT -A INPUT -i lo -j ACCEPT
$IPT -A OUTPUT -o lo -j ACCEPT

# Запрещаем любые новые соединения с любых интерфейсов, кроме lo,
# с нашим компьютером
$IPT -A INPUT -m state ! -i lo --state NEW -j DROP
$IPT -A INPUT -s 127.0.0.1/255.0.0.0 ! -i lo -j DROP

# Отбрасываем все пакеты со статусом INVALID
$IPT -A INPUT   -m state --state INVALID -j DROP
$IPT -A FORWARD -m state --state INVALID -j DROP

# Принимаем все пакеты из уже установленного соединения
# Состояние ESTABLISHED
$IPT -A INPUT   -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# Мой провайдер использует IP-адреса из сети 10.0.0.0 для
# доступа к своим локальным ресурсам. Ничего не поделаешь,
# нужно разрешить эти адреса, иначе вы даже не сможете войти в
```

```
# билинговую систему. В вашем случае, может, и не нужно будет
# добавлять следующее правило, а, может, у вас будет такая же
# ситуация, но адрес подсети будет другим
$IPT -t nat -I PREROUTING -i $INET -s 10.0.0.1/32 -j ACCEPT

# Защищаемся от SYN-наводнения (довольно популярный вид атаки)
$IPT -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
$IPT -A OUTPUT -p tcp ! --syn -m state --state NEW -j DROP

# Защищаемся от UDP-наводнения
$IPT -A INPUT -p UDP -s 0/0 --dport 138 -j DROP
$IPT -A INPUT -p UDP -s 0/0 --dport 113 -j REJECT
$IPT -A INPUT -p UDP -s 0/0 --sport 67 --dport 68 -j ACCEPT
$IPT -A INPUT -p UDP -j RETURN
$IPT -A OUTPUT -p UDP -s 0/0 -j ACCEPT

# Защищаемся от ICMP-перенаправления
# Данный вид атаки может использоваться злоумышленником для
# перенаправления своего трафика через вашу машину
$IPT -A INPUT --fragment -p ICMP -j DROP
$IPT -A OUTPUT --fragment -p ICMP -j DROP

# Но обычные ICMP-сообщения мы разрешаем
$IPT -A INPUT -p icmp -m icmp -i $INET --icmp-type source-quench -j ACCEPT
$IPT -A OUTPUT -p icmp -m icmp -o $INET --icmp-type source-quench -j ACCEPT

# Разрешаем себе пинговать интернет-узлы
$IPT -A INPUT -p icmp -m icmp -i $INET --icmp-type echo-reply -j ACCEPT
$IPT -A OUTPUT -p icmp -m icmp -o $INET --icmp-type echo-request -j ACCEPT

# Разрешаем передачу ICMP-сообщения "неверный параметр"
$IPT -A INPUT -p icmp -m icmp -i $INET --icmp-type parameter-problem -j ACCEPT
$IPT -A OUTPUT -p icmp -m icmp -o $INET --icmp-type parameter-problem -j
ACCEPT

# Запрещаем подключение к X.Org через сетевые интерфейсы.
$IPT -A INPUT -p tcp -m tcp -i $INET --dport 6000:6063 -j DROP --syn

# Указываем порты, открытые в системе, но которые должны быть
# закрыты на сетевых интерфейсах. Я пропишу только порт 5501:
$IPT -A INPUT -p tcp -m tcp -m multiport -i $INET -j DROP --dports 5501

# Разрешаем DNS
$IPT -A OUTPUT -p udp -m udp -o $INET --dport 53 --sport $UPORTS -j ACCEPT
```

```
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 53 --sport $UPORTS -j ACCEPT
$IPT -A INPUT -p udp -m udp -i $INET --dport $UPORTS --sport 53 -j ACCEPT
$IPT -A INPUT -p tcp -m tcp -i $INET --dport $UPORTS --sport 53 -j ACCEPT

# Разрешаем AUTH-запросы к удаленным серверам, но запрещаем такие
# запросы к своему компьютеру
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 113 --sport $UPORTS -j ACCEPT
$IPT -A INPUT -p tcp -m tcp -i $INET --dport $UPORTS --sport 113 -j ACCEPT ! --
syn
$IPT -A INPUT -p tcp -m tcp -i $INET --dport 113 -j DROP

# Далее мы открываем некоторые порты, необходимые для
# функционирования сетевых служб.

# FTP-клиент (порт 21)
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 21 --sport $UPORTS -j ACCEPT
$IPT -A INPUT -p tcp -m tcp -i $INET --dport $UPORTS --sport 21 -j ACCEPT ! --
syn

# SSH-клиент (порт 22)
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 22 --sport $UPORTS -j ACCEPT
$IPT -A INPUT -p tcp -m tcp -i $INET --dport $UPORTS --sport 22 -j ACCEPT ! --
syn
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 22 --sport 1020:1023 -j ACCEPT
$IPT -A INPUT -p tcp -m tcp -i $INET --dport 1020:1023 --sport 22 -j ACCEPT !
--syn

# SMTP-клиент (порт 25)
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 25 --sport $UPORTS -j ACCEPT
$IPT -A INPUT -p tcp -m tcp -i $INET --dport $UPORTS --sport 25 -j ACCEPT ! --
syn

# HTTP/HTTPS-клиент (порты 80, 443)
$IPT -A OUTPUT -p tcp -m tcp -m multiport -o $INET --sport $UPORTS -j ACCEPT -
-dports 80,443
$IPT -A INPUT -p tcp -m tcp -m multiport -i $INET --dport $UPORTS -j ACCEPT --
sports 80,443 ! --syn

# POP-клиент (порт 110)
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 110 --sport $UPORTS -j ACCEPT
$IPT -A INPUT -p tcp -m tcp -i $INET --dport $UPORTS --sport 110 -j ACCEPT ! -
-syn

# Разрешаем прохождение DCHP-запросов через iptables
# Необходимо, если IP-адрес динамический
$IPT -A OUTPUT -p udp -m udp -o $INET --dport 67 --sport 68 -j ACCEPT
$IPT -A INPUT -p udp -m udp -i $INET --dport 68 --sport 67 -j ACCEPT
```

Вот, практически, и все. Приведенное здесь описание iptables нельзя назвать полным. Но если описать iptables полностью, то можно смело издавать отдельную книгу под названием "Брандмауэр в Linux". В Интернете я нашел одно из наиболее полных руководств по iptables на русском языке. Так вот, если его распечатать, то оно займет 121 страницу формата A4. Лист книги обычно меньше A4, поэтому смело можно говорить, что объем нашей книги составил бы около 200 страниц. Адрес этого руководства: **http://www.opennet.ru/docs/RUS/iptables/**.

Вот еще одна очень хорошая статья по iptables: **http://ru.wikipedia.org/wiki/Iptables**.

А для пользователей Debian и Ubuntu будет полезным следующее руководство: **http://www.linux.by/wiki/index.php/Debian_Firewall**.

15.5. Брандмауэр ebtables

Мы только что рассмотрели брандмауэр iptables. Но, кроме него, в Linux есть еще один актуальный на сегодняшний день брандмауэр — ebtables. Этот брандмауэр похож на iptables, но, в отличие от него, работает не на третьем, а на втором уровне сетевого стека.

Фильтрация в ebtables осуществляется на основании значений полей в заголовках Ethernet-кадров (а это канальный уровень модели OSI). Кроме фильтрации пакетов, ebtables может выполнять функции моста-маршрутизатора и может даже изменять MAC-адреса в кадрах Ethernet. Понятно, что подмена MAC-адреса актуальна только в локальной сети.

Если вам нужен брандмауэр для Ethernet-кадров, тогда перекомпилируйте ядро и включите опцию **Ethernet Bridge tables (ebtables) support**. Данная опция есть в ядрах ветки 2.6, но ее нет в старых ветках — 2.4 и 2.2.

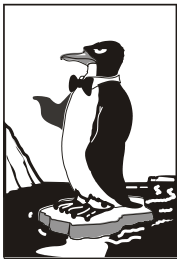
Необходимость в ebtables возникает редко. Но если вам действительно нужен брандмауэр, работающий на канальном уровне, тогда ознакомьтесь со следующими ссылками:

- **http://ebtables.sourceforge.net** — официальный сайт ebtables;
- **http://ebtables.sourceforge.net/examples.html#easy** — примеры использования ebtables (простые);
- **http://ebtables.sourceforge.net/examples.html#real** — сложные примеры из реальной жизни;
- **http://ebtables.sourceforge.net/ebtables-hacking/ebtables-hacking-HOWTO.html** — а вот эта ссылка пригодится, если вы действительно решили, что ebtables — то, что вам нужно.

ЧАСТЬ IV

Операционная система Linux

В третьей части книги мы настроили доступ нашего сервера к Интернету. Теперь можно приступить к дальнейшему изучению операционной системы Linux — ведь сейчас без подключения к Интернету даже программное обеспечение не установить. Да, некоторые дистрибутивы (openSUSE, Mandriva и др.) до сих пор позволяют устанавливать его с дистрибутивных дисков, но в последнее время наблюдается тенденция установки пакетов из интернет-репозиториев.



Глава 16

Загрузчики Linux

16.1. Базовые загрузчики

Основное назначение загрузчика — запуск выбранной пользователем операционной системы. Наиболее популярным загрузчиком сегодня является GRUB, который мы здесь подробно рассмотрим. В более старых дистрибутивах по умолчанию применялся загрузчик LILO. Списывать со счета LILO пока нельзя, поскольку еще много систем используют именно его, да и в современных дистрибутивах присутствует возможность установить старый добрый LILO. Многие администраторы по привычке ставят LILO вместо более современного GRUB. Однако в этой книге загрузчик LILO рассмотрен не будет[1].

Кроме LILO и GRUB некоторые дистрибутивы могут включать собственные загрузчики — например, в ASPLinux таковым является ASPLoader. Подобные загрузчики мы также рассматривать не будем, поскольку в большинстве случаев в дистрибутивах, использующих собственные загрузчики, все равно имеется возможность установки GRUB или LILO.

Загрузчик GRUB (GRand Unified Bootloader) считается более гибким и современным, чем LILO. Благодаря иной схеме загрузки операционных систем GRUB "понимает" больше файловых систем, нежели LILO, а именно: FAT/FAT32, ext2, ext3, ReiserFS, XFS, BSDFS и др.

Время не стоит на месте. Когда-то загрузчик GRUB пришел на смену LILO, поскольку последний не поддерживал загрузки с разделов, начинающихся после 1024-го цилиндра. Об этой проблеме знает, наверное, каждый Linux-пользователь — ведь всего несколько лет назад она была актуальной (пока все дистрибутивы не перешли на GRUB). Точно такая же участь постигла и GRUB — на его место пришел GRUB2, умеющий загружаться с файловой системы ext4. А загрузка с ext4-разделов просто необходима современному дистрибутиву.

GRUB2 — это не просто набор патчей для GRUB, а полностью новая разработка, созданная с "нуля". Именно поэтому у GRUB2 совершенно другой формат конфигурационного файла.

Разработка "обычного" GRUB полностью прекращена, к нему выпускаются лишь патчи. Да, можно скачать патч, добавляющий к GRUB загрузку с разделов ext4. Так, в Ubuntu 9.10, где по умолчанию впервые был установлен GRUB2, я его удалил

[1] Если вам до сих пор нужен LILO, рекомендую прочитать мою книгу "Linux. От новичка к профессионалу. 2-е изд.": **http://bhv.ru/books/book.php?id=186944**.

(с сохранением конфигурационных файлов), затем установил GRUB (имеющаяся в составе версии 9.10 версия GRUB поддерживает ext4), создал вручную его конфигурационный файл и перезагрузил систему — она загрузилась без ошибок. Но, учитывая, что будущее все-таки за GRUB2, я вернул его обратно на заслуженное место.

В Ubuntu GRUB2 используется, начиная с версии 9.10 — не зря я упомянул ее только что. И в этой версии Ubuntu, и в новой — 10.04 — имеется один небольшой "глюк", связанный с установкой тайм-аута выбора операционной системы. Чуть позже мы решим эту проблему, а пока приступим к рассмотрению конфигурационных файлов GRUB2.

Любопытно

На самом деле то, что называется GRUB2, — это GRUB v1.98. То есть — "почти вторая" версия, а когда выйдет реально вторая (в смысле 2.0), пока никто не знает. Хотя ведущие разработчики дистрибутивов уже включили этот GRUB2 в состав дистрибутивов, что говорит о его надежности.

16.2. Конфигурационные файлы GRUB и GRUB2

16.2.1. Конфигурационный файл GRUB

Конфигурационным файлом GRUB служит файл /boot/grub/grub.conf (в старых версиях — /boot/grub/menu.lst. Впрочем, menu.lst в новых версиях — это ссылка на grub.conf). Рассмотрим пример этого файла (листинг 16.1).

Листинг 16.1. Файл /boot/grub/grub.conf

```
# Следующие параметры будут описаны далее:
boot=/dev/sda
default=0
timeout=10
fallback=1
splashimage=(sd0,1)/grub/mysplash.xpm.gz

# по умолчанию скрывает меню (для того, чтобы увидеть меню
# нужно нажать ESC)
#hiddenmenu

# Главное загрузочное устройство GRUB (можно не указывать)
#groot=(sd0,1)

# Опции загрузчика по умолчанию (более подробно см. man menu.lst)
# defoptions=quiet splash

# опции ядра по умолчанию
```

```
# kopt=root=/dev/sda2 ro

# Предпочитаемые цвета
#color cyan/blue white/blue

title MDK
       root (sd0,1)
       kernel /vmlinuz-2.6.14-1.1263 ro root=/dev/sda2
       initrd /initrd-2.6.14-1.1263.img

title WinXP
       rootnoverify (sd0,0)
       makeactive
       chainloader+1
```

Параметр `boot` указывает загрузочное устройство, а параметр `default` — загрузочную метку по умолчанию. Метка начинается параметром `title` и продолжается до следующего `title`. Нумерация меток начинается с 0. Параметр `timeout` задает количество секунд, по истечении которых будет загружена операционная система по умолчанию.

Параметр `default` полезно использовать с параметром `fallback`. Первый задает операционную систему по умолчанию, а второй — операционную систему, которая будет загружена в случае, если с загрузкой операционной системы по умолчанию произошла ошибка.

Задать графическое изображение позволяет параметр `splashimage`. Чуть позже мы разберемся, как самостоятельно создать такое изображение.

> ***Внимание!***
>
> При работе с GRUB вам поначалу будет трудно разобраться с именами разделов. GRUB вместо привычных /dev/sd* использует свои собственные имена. Перевести имя /dev/sd* в имя в формате GRUB просто. Во-первых, опускается фрагмент /dev/. Во-вторых, устройства отсчитываются не с буквы "a", как принято в Linux, а с нуля. Разделы на дисках отсчитываются не с единицы, а тоже с нуля, причем номер раздела указывается через запятую. Потом все имя берется в скобки. Так, раздел /dev/sda1 в GRUB будет выглядеть как (sd0,0), а раздел /dev/sdb2 как (sd1,1). Впрочем, об именах разделов в GRUB мы еще поговорим, но чуть позже.

Параметр `rootnoverify` указывается для Windows (точнее, для всех операционных систем не типа Linux). Параметр `chainloader` указывается для операционных систем, поддерживающих цепочечную загрузку. Если Windows на вашем компьютере установлена в неактивном разделе, с которого Windows загружаться не может, перед параметром `chainloader` нужно указать параметр `makeactive`.

16.2.2. Конфигурационный файл GRUB2

В листинге 16.2 приведен основной конфигурационный файл GRUB2 — /boot/grub/grub.cfg. Этот конфигурационный файл не редактируется вручную. Для его создания используется утилита /usr/sbin/grub-mkconfig, которая генерирует этот

конфигурационный файл на основе шаблонов, хранящихся в каталоге /etc/grub.d, и настроек из файла /etc/default/grub.

Листинг 16.2. Конфигурационный файл grub.cfg

```
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by /usr/sbin/grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#

### BEGIN /etc/grub.d/00_header ###
if [ -s /boot/grub/grubenv ]; then
  have_grubenv=true
  load_env
fi
set default="0"
if [ ${prev_saved_entry} ]; then
  saved_entry=${prev_saved_entry}
  save_env saved_entry
  prev_saved_entry=
  save_env prev_saved_entry
fi
insmod ext2
set root=(hd0,1)
search --no-floppy --fs-uuid --set 34eaa635-ef0e-4d5c-8b61-3c22c767834b
if loadfont /usr/share/grub/unicode.pf2 ; then
  set gfxmode=640x480
  insmod gfxterm
  insmod vbe
  if terminal_output gfxterm ; then true ; else
    # For backward compatibility with versions of terminal.mod that don't
    # understand terminal_output
    terminal gfxterm
  fi
fi
if [ ${recordfail} = 1 ]; then
  set timeout=-1
else
  set timeout=10
fi
### END /etc/grub.d/00_header ###

### BEGIN /etc/grub.d/05_debian_theme ###
```

```
set menu_color_normal=white/black
set menu_color_highlight=black/white
### END /etc/grub.d/05_debian_theme ###

### BEGIN /etc/grub.d/10_linux ###
menuentry "Denix, Linux 2.6.31-14-generic" {
        recordfail=1
        if [ -n ${have_grubenv} ]; then save_env recordfail; fi
       set quiet=1
       insmod ext2
       set root=(hd0,1)
       search --no-floppy --fs-uuid --set 34eaa635-ef0e-4d5c-8b61-3c22c767834b
       linux   /boot/vmlinuz-2.6.31-14-generic root=UUID=34eaa635-ef0e-4d5c-
8b61-3c22c767834b ro   quiet splash
       initrd /boot/initrd.img-2.6.31-14-generic
}
menuentry "Denix, Linux 2.6.31-14-generic (recovery mode)" {
        recordfail=1
        if [ -n ${have_grubenv} ]; then save_env recordfail; fi
       insmod ext2
       set root=(hd0,1)
       search --no-floppy --fs-uuid --set 34eaa635-ef0e-4d5c-8b61-3c22c767834b
       linux   /boot/vmlinuz-2.6.31-14-generic root=UUID=34eaa635-ef0e-4d5c-
8b61-3c22c767834b ro single
       initrd /boot/initrd.img-2.6.31-14-generic
}
### END /etc/grub.d/10_linux ###

### BEGIN /etc/grub.d/20_memtest86+ ###
menuentry "Memory test (memtest86+)" {
       linux16/boot/memtest86+.bin
}
menuentry "Memory test (memtest86+, serial console 115200)" {
       linux16/boot/memtest86+.bin console=ttyS0,115200n8
}
### END /etc/grub.d/20_memtest86+ ###

### BEGIN /etc/grub.d/30_os-prober ###
if [ ${timeout} != -1 ]; then
  if keystatus; then
    if keystatus --shift; then
      set timeout=-1
    else
      set timeout=0
    fi
```

```
  else
    if sleep --interruptible 3 ; then
      set timeout=0
    fi
  fi
fi
### END /etc/grub.d/30_os-prober ###

### BEGIN /etc/grub.d/40_custom ###
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
### END /etc/grub.d/40_custom ###
```

Вы наверняка заметили, что синтаксис grub.cfg весьма напоминает синтаксис bash-сценариев. Параметры GRUB2 задаются в файле /etc/default/grub, а в файле grub.cfg описываются элементы меню загрузчика.

Рассмотрим описание элемента меню:

```
menuentry "Denix, Linux 2.6.31-14-generic" {
        recordfail=1
        if [ -n ${have_grubenv} ]; then save_env recordfail; fi
    set quiet=1
    insmod ext2
    set root=(hd0,1)
    search --no-floppy --fs-uuid --set 34eaa635-ef0e-4d5c-8b61-3c22c767834b
    linux  /boot/vmlinuz-2.6.31-14-generic root=UUID=34eaa635-ef0e-4d5c-
8b61-3c22c767834b ro   quiet splash
    initrd /boot/initrd.img-2.6.31-14-generic
}
```

В кавычках после `menuentry` находится описание элемента меню — можете заменить этот текст на все, что вам больше нравится. Далее следуют команды GRUB. Например, команда `insmod ext2` загружает модуль ext2. Это не модуль ядра Linux! Это модуль GRUB2 — файл ext2.mod, находящийся в каталоге /boot/grub.

Команда `set root` устанавливает загрузочное устройство. Формат имени устройства такой же, как в случае с GRUB.

> ***Внимание!***
>
> Мы знаем, что даже ATA-диски в новых дистрибутивах имеют имена вида /dev/sda*. Но команда `set root` загрузчика GRUB2 содержит имя hd. Это не опечатка! Это внутреннее имя устройства GRUB, а не имя системного устройства.

После служебного слова `linux` задается ядро (файл ядра) и параметры, которые будут переданы ядру. Служебное слово `initrd` указывает на файл initrd.

Теперь рассмотрим файл /etc/default/grub, содержащий параметры GRUB2 (листинг 16.3). Поскольку этот файл вы будете редактировать чаще, чем grub.cfg, то комментарии для большего удобства я перевел на русский язык.

Листинг 16.3. Файл /etc/default/grub

```
# Если вы измените этот файл, введите команду 'update-grub'
# для обновления вашего файла /boot/grub/grub.cfg.

# Элемент по умолчанию, нумерация начинается с 0
GRUB_DEFAULT=0
# Чтобы увидеть меню GRUB, нужно или закомментировать следующую
# опцию, или установить значение больше 0, но в этом случае
# нужно изменить значение GRUB_HIDDEN_TIMEOUT_QUIET на false
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
# Тайм-аут (в секундах)
GRUB_TIMEOUT="10"
# Название дистрибутива – вывод команды lsb_release или просто Debian
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
# Параметры ядра по умолчанию
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Раскомментируйте для отключения графического терминала
# (только для grub-pc)
#GRUB_TERMINAL=console

# Разрешение графического терминала
#GRUB_GFXMODE=640x480

# Раскомментируйте следующую опцию, если вы не хотите передавать
# параметр "root=UUID=xxx" ядру Linux
#GRUB_DISABLE_LINUX_UUID=true

# Раскомментируйте, если нужно отключить генерацию элемента меню
# режима восстановления
#GRUB_DISABLE_LINUX_RECOVERY="true"
```

После изменения файла /etc/default/grub не забудьте выполнить команду `update-grub` для обновления вашего /boot/grub/grub.cfg.

При редактировании конфигурации GRUB2 следует придерживаться одной стратегии из двух возможных. Первая заключается в ручном редактировании файла grub.cfg — вы редактируете его вручную и не используете других программ вроде grub-mkconfig или update-grub. Вторая стратегия заключается в использовании вспомогательных программ, но тогда не нужно редактировать файл grub.cfg вручную, иначе при последующем изменении файла grub.cfg программами grub-mkconfig или update-grub все изменения, внесенные вручную, будут уничтожены.

По умолчанию команда grub-mkconfig генерирует конфигурационный файл на консоль, поэтому вызывать ее нужно так:

```
sudo grub-mkconfig > /boot/grub/grub.cfg
```

16.3. Команды установки загрузчиков

Установить GRUB/GRUB2, если вы это еще не сделали, можно следующей командой:

```
/sbin/grub-install <устройство>
```

Например:

```
/sbin/grub-install /dev/sda
```

После изменения конфигурационного файла переустанавливать загрузчик, как в случае с устаревшим LILO, не нужно.

16.4. Установка тайм-аута выбора операционной системы. Редактирование параметров ядра Linux

По умолчанию GRUB2 не отображает меню выбора операционной системы. Следовательно, вы не можете ни выбрать другую операционную систему (в том числе и Windows), ни изменить параметры ядра Linux, ни выбрать режим восстановления или режим тестирования памяти. Одним словом, такое поведение загрузчика создает определенные неудобства.

Чуть ранее было сказано, что для установки тайм-аута загрузчика нужно отредактировать следующие параметры:

```
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
# Таймаут (в секундах)
GRUB_TIMEOUT="10"
```

Все правильно, но только для того случая, когда бы GRUB2 в Ubuntu не содержал "глюка".

> ***Примечание***
>
> Вообще, "глюки" — это хорошо. Чем корявее будет Canonical делать свои дистрибутивы, тем больше будет работы у авторов книг и дистрибутивов на базе Ubuntu. Вы думаете, почему я создал свой дистрибутив Denix (**denix.dkws.org.ua**)? Нет, не для того, чтобы гордо ткнуть себя в грудь — мол, я тоже могу сделать свой дистрибутив! А чтобы после каждой установки Ubuntu пользователи могли не тратить свое личное время, часами настраивая операционную систему.

И чтобы побороть неадекватное поведение загрузчика (а каким его еще назвать, если программа не реагирует на установку параметров из конфигурационного файла), мне пришлось потратить минут 15–20. К своему решению я пришел экспериментальным путем, поэтому не удивлюсь, если на каком-то форуме в Интернете вы найдете другое решение (не исключаю, может быть, даже лучше моего).

Итак, откройте ваш файл /etc/grub.d/30_os-prober:

```
sudo nano /etc/grub.d/30_os-prober
```

Найдите в нем строку:

```
if [ "x${GRUB_HIDDEN_TIMEOUT}" = "x0" ]
```

Далее все значения `-1` во фрагменте кода, представленном в листинге 16.4, замените на 1. Строки, которые нуждаются в редактировании, выделены полужирным. Изменять значение `-1` в остальном коде, выходящем за рамки листинга 16.4, не нужно!

Листинг 16.4. Фрагмент файла /etc/grub.d/30_os-prober

```
if [ "x${GRUB_HIDDEN_TIMEOUT}" = "x0" ] ; then
    cat <
if [ \${timeout} != 1 ]; then
  if keystatus; then
    if keystatus --shift; then
      set timeout=1
    else
      set timeout=0
    fi
  else
    if sleep$verbose --interruptible 3 ; then
      set timeout=0
    fi
  fi
fi
EOF
      else
    cat << EOF
if [ \${timeout} != 1 ]; then
  if sleep$verbose --interruptible ${GRUB_HIDDEN_TIMEOUT} ; then
    set timeout=0
  fi
fi
EOF
```

После этого сохраните файл и введите команды:

```
sudo grub-mkconfig > /boot/grub/grub.cfg
sudo update-grub
sudo reboot
```

Теперь после перезагрузки вы увидите меню GRUB2 (рис. 16.1). Для редактирования параметров ядра *(см. главу 7)*, которые передаются Linux, выделите загрузочную метку Linux и нажмите клавишу <e>. Если вы защитили GRUB от редактирования параметров ядра (как это сделать, будет показано в *главе 25*), вы увидите требование ввести имя пользователя и пароль (рис. 16.2). Если они правильные, вы

сможете отредактировать загрузочную метку (рис. 16.3). В данном случае дополнительные параметры нужно вводить после параметра `splash` (строка параметров начинается после служебного слова `linux`). Кстати, если у вас проблемы с запуском Linux, то, чтобы увидеть больше диагностических сообщений, параметры `quiet` и `splash` лучше вообще удалить. Для возврата обратно в меню GRUB2, нажмите клавишу <Esc>, а для загрузки выбранной операционной системы — комбинацию клавиш <Ctrl>+<X>.

Рис. 16.1. Меню загрузчика

Рис. 16.2. Ввод имени пользователя и пароля

Рис. 16.3. Редактирование загрузочной метки

16.5. Установка собственного фона загрузчика GRUB и GRUB2

Вы хотите создать собственный фон для загрузчика GRUB? Это очень просто. Создайте или найдите в Интернете понравившуюся вам картинку. Уменьшите ее до размера 640×480 пикселов и конвертируйте в формат XPM. Все это можно сделать одной командой:

```
# convert image.jpg -colors 14 -resize 640x480 image.xpm
```

Затем сожмите картинку с помощью команды `gzip`:

```
# gzip image.xpm
```

Скопируйте сжатую картинку в каталог /boot/grub и пропишите в конфигурационном файле /boot/grub/grub.conf:

```
splashimage=(hd0,1)/grub/image.xpm.gz
```

Теперь разберемся, как установить графический фон в GRUB2. Убедитесь, что установлен пакет grub2-splashimages. Этот пакет содержит графические заставки для GRUB2, которые будут установлены в каталог /usr/share/images/grub. Если вам не нравятся стандартные картинки, тогда множество фонов для GRUB2 вы можете скачать с сайта **http://www.gnome-look.org/** или создать вручную, как было здесь показано. Вот только GRUB2 уже поддерживает форматы PNG и TGA, поэтому можно не конвертировать файл фона в формат XPM.

Итак, будем считать, что картинка у нас уже выбрана. Осталось только установить ее как фон. Откройте файл темы GRUB2. Он находится в каталоге /etc/grub.d. В Ubuntu и Debian этот каталог называется /etc/grub.d/05_debian_theme.

Примечание

В другие дистрибутивах он может называться иначе — учитывая, что далеко не все современные дистрибутивы перешли на GRUB2, точное его название для каждого дистрибутива указать сложно.

Найдите в файле темы строку:

```
for i in {/boot/grub,/usr/share/images/desktop-base}/moreblue-orbit-
grub.{png,tga} ; do
```

Замените ее на следующую строку:

```
for i in {/boot/grub,/usr/share/images/desktop-
base,/usr/share/images/grub}/имя_файла.{png,tga} ; do
```

Как видите, мы просто прописали выбранную вами картинку. Далее нужно обновить GRUB2:

```
sudo update-grub
```

16.6. Постоянные имена и GRUB

Как было отмечено в *главе 9*, все современные дистрибутивы перешли на так называемые *постоянные* ("длинные") имена. Раньше, когда еще никто не знал о длинных именах, запись в файле grub.conf могла выглядеть так:

```
kernel /boot/vmlinuz26 root=/dev/sda1 vga=0x318 ro
```

Эта запись указывает имя ядра (`/boot/vmlinuz26`). Все, что после него — параметры, которые будут переданы ядру. Один из них (параметр `root`) — указывает имя корневой файловой системы. Здесь оно приведено еще в старом формате. Сейчас вы такие имена в grub.conf не увидите (если, конечно, сами не пропишете). Варианты указания длинных имен выглядят так:

```
root=/dev/disk/by-uuid/2d781b26-0285-421a-b9d0-d4a0d3b55680
root=/dev/disk/by-id/scsi-SATA_WDC_WD1600JB-00_WD-WCANM7959048-part5
root=LABEL=/
```

Какой вариант будет использоваться у вас, зависит от дистрибутива. Например, в Fedora применяют третий способ, а в openSUSE — второй.

16.7. Две и более ОС Linux на одном компьютере

Рассмотрим другую ситуацию, часто возникающую на практике. Вы решили установить на свой компьютер (на котором уже была установлена Windows) операционную систему Linux. Все прошло гладко, и теперь вы с помощью GRUB можете запустить обе системы: Windows и Linux. Но потом вы решили установить еще один дистрибутив Linux, однако старый удалять пока не хотите. Поэтому вы создали еще один Linux-раздел, установили в него новый дистрибутив, но после перезагрузки обнаружили небольшую проблему, описываемую следующими вариантами событий:

- в меню GRUB отображаются только Windows и последний установленный дистрибутив — то есть вы не можете загрузить первый дистрибутив. Так, Fedora,

например, напрочь игнорирует все установленные до нее дистрибутивы, и поэтому после установки этого дистрибутива вы можете запустить только ее и Windows;

- или в меню GRUB отображаются Windows и оба дистрибутива, но запустить вы можете только последний установленный дистрибутив (и, понятно, Windows). Такую картину я наблюдал после установки openSUSE — в моем загрузочном меню появилась метка для загрузки ранее установленного дистрибутива Fedora, но загрузить его не получалось.

Понятно, что восстановить загрузчик первого дистрибутива, воспользовавшись рекомендациями из предыдущего раздела, мы не можем, поскольку после этого мы сможем запустить только первый дистрибутив и Windows (на момент формирования файла grub.conf первого дистрибутива еще ничего не было известно о втором дистрибутиве, который вы недавно установили).

Наши действия будут зависеть от конкретной ситуации. Для большей определенности предположим, что первый дистрибутив был установлен в раздел /dev/sda5, а второй — в раздел /dev/sda6.

Если у вас проблема по первому варианту (когда ранее установленного дистрибутива вообще нет в загрузочном меню), тогда вам нужно примонтировать раздел первого дистрибутива (у нас это /dev/sda5) к каталогу /mnt (или к любому другому):

```
# mount /dev/sda5 /mnt
```

Затем надо открыть файл /mnt/boot/grub/grub.conf (/mnt/boot/grub/menu.lst).

> ***Внимание!***
>
> Исходя из приведенного здесь пути к файлу, мы понимаем, что открываем файл grub.conf первого дистрибутива.

Скопируйте из него метку загрузки первого дистрибутива. У меня сначала был установлен openSUSE 11.2, а потом я установил Fedora 12, поэтому загрузочная метка в моем случае выглядела так:

```
title openSUSE 11.2
    root (sd0,4)
    kernel /boot/vmlinuz-2.6.31-14-default root=/dev/disk/by-id/scsi-
SATA_WDC_WD1600JB-00_WD-WCANM7959048-part5 vga=0x317    resume=/dev/sda7
splash=silent showopts
    initrd /boot/initrd-2.6.31-14-default
```

> ***Совет***
>
> Обратите внимание — параметр `root` содержит постоянное (длинное) имя, поэтому его не придется изменять. Если же в вашем варианте параметр `root` содержит короткое имя вида /dev/sd*, его желательно заменить постоянным именем *(см. главу 9)*.

Скопированную загрузочную метку нужно вставить в файл /boot/grub/grub.conf — это файл конфигурации GRUB, используемый в настоящий момент. Файл сохраните, но пока не закрывайте и не перезагружайте компьютер. Обратите внимание — для загрузки нашего первого дистрибутива требуются файлы vmlinuz-2.6.22.5-31-

default и initrd-2.6.22.5-31-default. Их нужно скопировать из каталога /mnt/boot в каталог /boot:

```
cp /mnt/boot/vmlinuz* /boot
cp /mnt/boot/initrd* /boot
```

Теперь можно перезагрузить компьютер. Первый дистрибутив, установленный в /dev/sda5, будет загружен.

Перейдем ко второму варианту. Он проще тем, что нам не нужно редактировать grub.conf, поскольку за нас это уже сделала программа установки второго дистрибутива. Вам нужно только подмонтировать каталог /dev/sda5 к каталогу /mnt и скопировать файлы vmlinuz* и initrd* из каталога /mnt/boot в каталог /boot. Вот и все.

Напоследок рекомендую прочитать тему форума, непосредственно относящуюся к рассматриваемому вопросу: **http://www.dkws.org.ua/phpbb2/viewtopic.php?t=3085**.

16.8. Загрузка с ISO-образов

Предположим, вы скачали ISO-образ новой версии Ubuntu, но у вас нет "болванки", чтобы записать на нее образ и загрузиться с полученного диска. Могу вас обрадовать: "болванка" вам для этого не понадобится — GRUB2 умеет использовать ISO-образы в качестве загрузочных устройств. Просто пропишите ISO-образ в конфигурационном файле GRUB2 и перезагрузите компьютер. Новая загрузочная метка появится в меню GRUB2, и, если ее выбрать, система загрузится с ISO-образа.

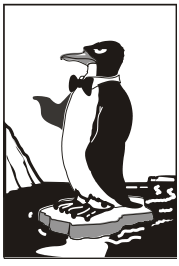
Итак, создайте в каталоге /boot подкаталог iso (название, сами понимаете, может быть любым), загрузите в него ISO-образ дистрибутива. Теперь вам осталось лишь отредактировать конфигурационный файл /boot/grub/grub.cfg, добавив в него вот такую загрузочную запись (выделенный полужирным шрифтом текст нужно записать в одну строку):

```
menuentry "Ubuntu LiveCD" {
    loopback loop /boot/iso/ubuntu.iso
    linux (loop)/casper/vmlinuz boot=casper iso-
scan/filename=/boot/iso/ubuntu.iso noeject noprompt --
    initrd (loop)/casper/initrd.lz
   }
```

Перезагружаемся и выбираем пункт меню **Ubuntu LiveCD**.

> ***Примечание***
>
> В *главе 25* мы подробно рассмотрим процесс установки пароля загрузчиков GRUB и GRUB2. Там также будет показано, как восстановить "упавший" загрузчик.

Глава 17



Системы инициализации Linux

17.1. Начальная загрузка Linux

Давайте разберемся, как загружается Linux. В этой книге мы уже упоминали о начальной загрузке компьютера, поэтому сейчас начнем с того момента, когда загрузчик BIOS нашел загрузочное устройство, например, жесткий диск. Далее загрузчик BIOS считывает первый (нулевой) сектор и передает ему управление. На этом работа загрузчика BIOS заканчивается.

В первом секторе находится главная загрузочная запись (Master Boot Record, MBR), состоящая из трех частей: первичного загрузчика, таблицы разделов диска (partition table) и флага загрузки.

Итак, из первой части MBR вызывается первичный загрузчик. Действия этого загрузчика зависят только от него самого. Работу загрузчика мы будем рассматривать на примере двух загрузчиков: LILO и GRUB.

> ***Пояснение***
>
> Загрузчик LILO хоть и безнадежно устарел, очень прост и поэтому идеально подходит для использования в "академических целях" — для изучения процесса загрузки. А GRUB — это более современный загрузчик, без понимания принципов работы которого сегодня просто нельзя. GRUB2 мы здесь рассматривать пока не будем. Во-первых, он довольно сложен (для демонстрации процесса загрузки вполне хватит обычного GRUB), а, во-вторых, окончательная версия 2.0 пока еще не вышла.

Загрузчик LILO состоит из двух частей: первая содержится в MBR, а вторая находится на диске в виде файла /boot/boot.b. По аналогии GRUB тоже состоит из двух частей: stage1 и stage2. Первая часть (stage1) помещается в MBR, а вторая хранится на диске в каталоге /boot/grub. Фактическое расположение stage2 указывается при установке GRUB примерно вот такой командой:

```
grub> install (hd0,4)/boot/grub/stage1 (hd0) (hd0,4)/boot/grub/stage2 p
(hd0,4) /boot/grub/menu.conf
```

Кроме stage1 и stage2 у загрузчика GRUB есть еще несколько промежуточных частей — *stage1_5 — помогающих загрузчику найти stage2 и выполняющих другие подготовительные действия, в частности, обеспечивающих поддержку разных файловых систем.

Задача первой части — запуск вторичного загрузчика (второй части), который и производит дальнейшую загрузку системы. Первая часть ничего не знает о файловых

системах, поэтому местонахождение второй части записано в "физических координатах", то есть явно указаны цилиндр, головка, сектор жесткого диска.

Вторая часть загрузчика более интеллектуальна. Она уже "знает", что такое файловая система, а карта размещения файлов записана в файле /boot/map. Аналогично, в GRUB тоже имеется карта устройств — файл /boot/grub/device.map. Эти файлы используются для поиска ядра и образа виртуального диска. Для чего нужен виртуальный диск? Представим, что мы еще не установили Linux, а только собираемся это сделать. Вставляем загрузочный диск, и загрузчик запускает не просто инсталлятор — на самом деле запускается операционная система Linux, ясно виден процесс загрузки ядра, и потом уже запускается программа установки. Но ядру нужно же откуда-то прочитать модули поддержки устройств и файловой системы — ведь корневая файловая система еще не создана. Вот все эти модули и находятся на виртуальном диске. Виртуальный диск загружается в память, ядро монтирует его, как обычную файловую систему, и загружает с него все необходимые модули. После этого виртуальный диск размонтируется и — в случае нормальной загрузки, а не установки Linux, — вместо него монтируется обычная корневая файловая система.

Для работы с виртуальным диском используется технология initrd (INITial Ram Disk). Файл образа виртуального диска находится в каталоге /boot и называется initrd-<версия ядра>.

17.2. Система инициализации init

В процессе запуска ядра монтируется корневая файловая система и запускается программа init, которая и выполняет дальнейшую инициализацию системы. Программа init — часть самой надежной и распространенной системы инициализации Linux, которая используется многими дистрибутивами: Fedora, ASPLinux, Mandriva, openSUSE и др.

Кроме системы инициализации init, существуют и другие системы, например, initng и upstart:

- система initng позволяет существенно ускорить запуск Linux, но она почему-то не прижилась в мире Linux, и ни один дистрибутив не использует ее по умолчанию, очевидно, из-за сложной настройки данной системы. Прочитать об этой системе можно в моей статье:
  http://www.dkws.org.ua/index.php?page=show&file=a/system/initng/initng;
- система upstart *(см. разд. 17.3)* была специально разработана для дистрибутива Ubuntu Linux, но ее при желании можно установить в любом дистрибутиве.

17.2.1. Файл /etc/inittab

Итак, программа init читает конфигурационный файл /etc/inittab и запускает другие процессы, согласно инструкциям этого файла (листинг 17.1).

Листинг 17.1. Файл /etc/inittab

```
id:5:initdefault:

# Инициализация системы
```

```
si::sysinit:/etc/rc.d/rc.sysinit
l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Что делать при нажатии CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# От UPS была получена команда, что пропало питание.
# Немного ждем и выключаем компьютер
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# От UPS получена команда, что питание возобновилось
# Отменяем shutdown
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

# Запуск gettys
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Однопользовательский режим
~~:S:wait:/bin/sh
```

Одна из главных инструкций файла /etc/inittab выглядит так:

```
id:<число>:initdefault:
```

Эта инструкция задает уровень запуска по умолчанию. Уровень запуска определяет, какие действия будут выполнены программой init (какие процессы будут запущены). Всего предусмотрено шесть уровней запуска:

- 0 — останов системы (ясно, что в качестве уровня по умолчанию этот уровень быть не может);
- 1 — однопользовательский режим (в него можно перейти сразу при загрузке, передав ядру параметр `single`);
- 2 — многопользовательский режим без поддержки сети;
- 3 — многопользовательский режим с поддержкой сети;
- 4 — не используется;

- 5 — многопользовательский графический режим с загрузкой X11 и поддержкой сети;
- 6 — перезагрузка системы.

В большинстве случаев в качестве уровня запуска по умолчанию устанавливается 3 или 5.

17.2.2. Команда *init*

Перейти на тот или иной уровень можно и после загрузки системы. Для этого используется команда:

```
# /sbin/init <уровень_запуска>
```

> ***Примечание***
>
> Напомню, что решетка (#) перед командой означает, что команда должна быть выполнена от имени пользователя root.

"Вычислив" уровень запуска, init поочередно запускает сценарии из каталога /etc/rc.d/rcX.d, где X — это номер уровня запуска. Если зайти в один из этих каталогов, например, в /etc/rc.d/rc3.d, то можно увидеть ссылки формата:

S<номер><имя>

Параметр <номер> определяет порядок запуска сценария (например, S10network запустится раньше, чем S11internet), а параметр <имя> — задает имя сценария. Сами сценарии находятся в каталоге /etc/rc.d/init.d.

Ссылки, начинающиеся на символ S, — это ссылки запуска (от S, start), при запуске соответствующих сценариев им будет передан аргумент `start`. Например, если init обнаружила в /etc/rc.d/rc3.d файл S10network, то она выполнит команду:

```
/etc/rc.d/init.d/network start
```

Если имя ссылки начинается на букву K (от kill, убить), то это ссылка останова сервиса, например, K01service. Данная ссылка указывает на команду:

```
/etc/rc.d/init.d/service stop
```

Вы можете запустить любой сценарий из каталога init.d непосредственно, передав ему параметры `start` (запуск), `stop` (останов) и другие (зависит от сервиса).

17.2.3. Команда service

А можете воспользоваться командой `service`:

```
# service <имя_сервиса> <start|stop|…>
```

Здесь <имя_сервиса> — это имя файла в каталоге /etc/rc.d/init.d.

17.2.4. Редакторы уровней запуска

Редактировать уровни запуска можно вручную, а можно и с помощью программ-конфигураторов. В Fedora (и ASPLinux) для редактирования уровней запуска используется конфигуратор system-config-services (рис. 17.1), а в Mandriva можно воспользоваться командой `ntsysv` с параметром `--level <номер сервиса>` (рис. 17.2).

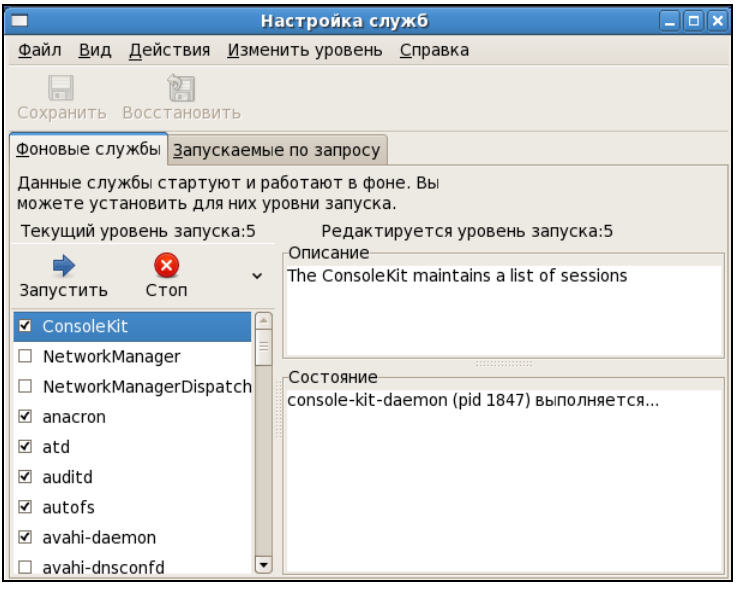


Рис. 17.1. Конфигуратор system-config-services

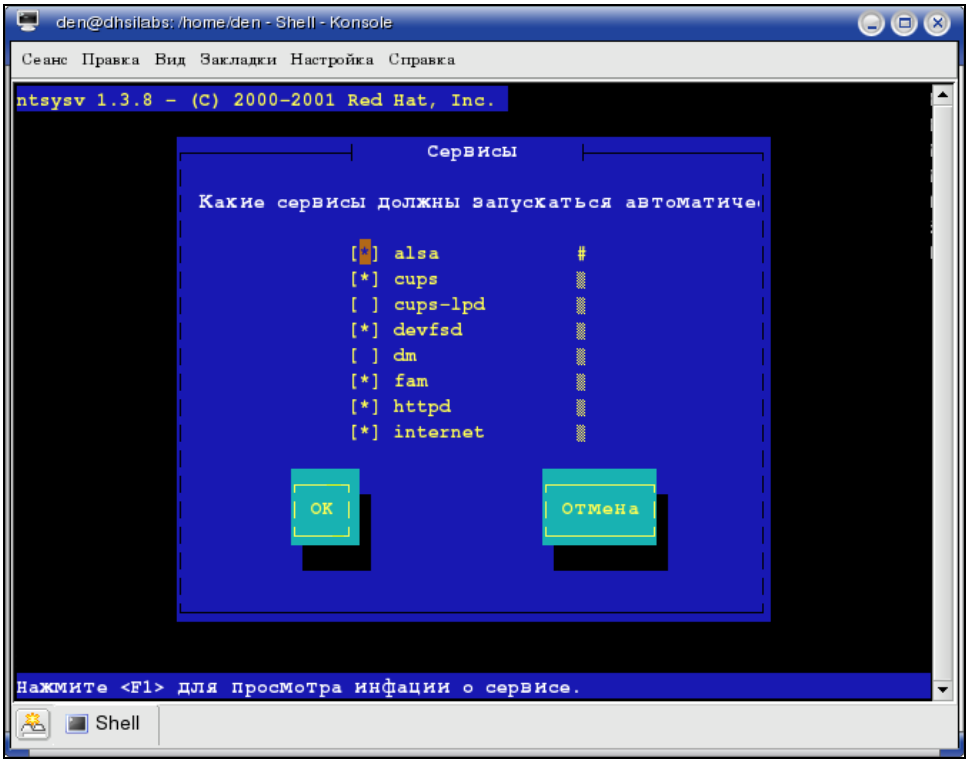


Рис. 17.2. Конфигуратор ntsysv

Конфигуратор в Fedora более удобен. Выбрать редактируемый уровень запуска можно с помощью меню **Изменить уровень**.

Конфигуратор drakboot (рис. 17.3), имеющийся в Linux Mandriva, позволяет указать, в каком режиме будет запускаться система: в графическом или в режиме консоли. По сути, конфигуратор позволяет выбрать уровень запуска (3 — консоль, 5 — графический режим).

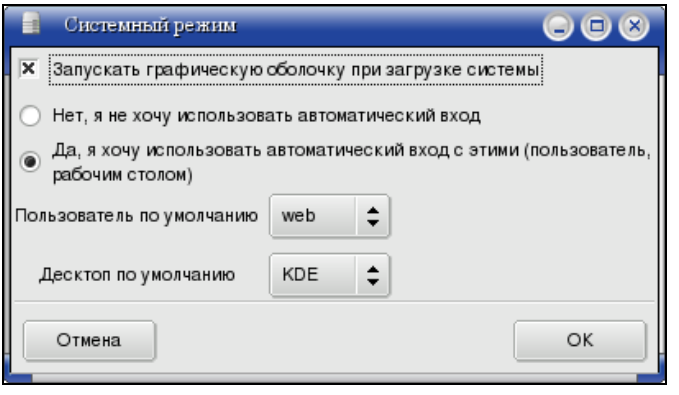


Рис. 17.3. Конфигуратор drakboot

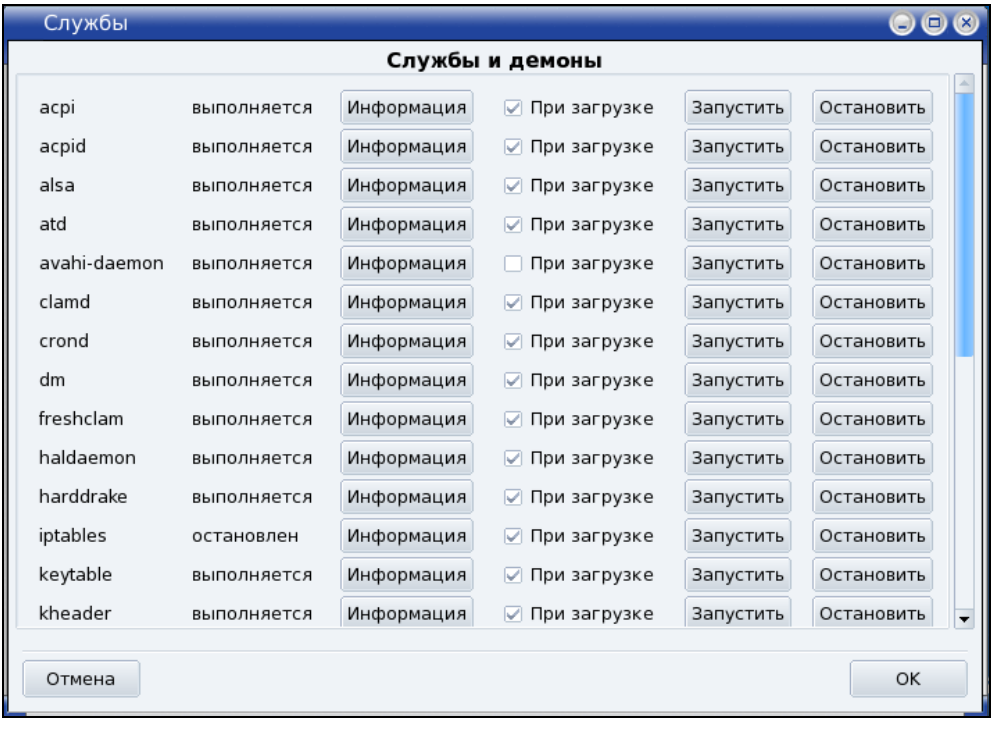


Рис. 17.4. Конфигуратор drakxservices

В случае если система будет запускаться в графическом режиме, данный конфигуратор позволяет включить автовход. Для функции автовхода нужно указать два параметра: имя пользователя и графическую среду.

Если автовход выключен, то при запуске X.Org система запросит у вас имя пользователя и пароль. Также у вас будет возможность выбрать графическую среду, с которой вы хотите работать. Если автовход включен, то будет выполнена автоматическая регистрация в системе выбранного пользователя с запуском выбранной графической среды. После этого вы можете работать в системе от имени этого пользователя. Из соображений безопасности конфигуратор не позволяет выбрать пользователя root.

Есть в Mandriva также конфигуратор drakxservices (рис. 17.4), позволяющий редактировать сервисы, запускаемые на пятом уровне запуска (обычно этот уровень запуска и используется).

17.3. Система инициализации upstart

Система инициализации upstart была разработана Скотом Джеймсом Ремнантом (Scott James Remnant) для дистрибутива Ubuntu, однако upstart, если она вам понравилась, можно с успехом использовать в других дистрибутивах. Мы не будем рассматривать установку upstart на другой дистрибутив, а разберемся, как с ней работать в Ubuntu.

17.3.1. Как работает upstart?

Upstart заменяет инициализирующие сценарии для поддержки событийно-ориентированного режима действий. Проще говоря, в upstart есть собственный процесс init, который запускается при запуске системы (аналогично программам init и initng). При запуске генерируется событие `startup`, при завершении работы — `shutdown`, при нажатии клавиатурной комбинации <Ctrl>+<Alt>+<Del> — событие `ctrl-alt-delete`.

Вы можете создавать собственные события. Вот небольшой пример создания события `my_event`:

```
on my_event
exec echo event received
console output
```

При получении этого события на консоль будет выведено сообщение:

event received

Файлы событий хранятся в каталоге /etc/event.d. Создайте в этом каталоге файл с именем my_event и поместите в него приведенный код. После этого вызвать событие вы можете командой:

```
initctl emit my_event
```

Подробнее об этой команде вы сможете прочитать на странице руководства (в Ubuntu оно на русском): `man initctl`.

17.3.2. Конфигурационные файлы upstart

Исследуйте содержимое каталога /etc/event.d. В нем вы найдете файлы событий перехода на определенный запуск. В листинге 17.2 представлен файл события перехода на пятый уровень запуска — /etc/event.d/rc5.

Листинг 17.2. Файл события /etc/event.d/rc5

```
start on runlevel 5

stop on runlevel [!5]

console output
script
       set $(runlevel --set 5 || true)
       if [ "$1" != "unknown" ]; then
           PREVLEVEL=$1
           RUNLEVEL=$2
           export PREVLEVEL RUNLEVEL
       fi

       exec /etc/init.d/rc 5
end script
```

Не нужно быть гуру в программировании, чтобы понять, что делает этот сценарий — он выполняет сценарий /etc/init.d/rc, передав ему значение 5 (номер уровня запуска). Сценарий /etc/init.d/rc занимается запуском/остановкой служб на определенном уровне, который ему передается в качестве параметра.

Но самое интересное в upstart, что уровни запуска здесь — виртуальные. На самом деле, номера уровней запуска остались только ради совместимости с init, чтобы человеку, который впервые увидел upstart (точнее, дистрибутив с установленной системой инициализации upstart), было проще с ней разобраться. В upstart, благодаря событийно-ориентированному режиму, вообще отпадает необходимость в уровнях запуска, подобных тем, которые использовались в init. Загрузка того или иного сервиса происходит при наличии нужного аппаратного обеспечения: нет устройства — не будет загружен и сервис, требующий его.

Кстати, в последних версиях Ubuntu имеется команда `service`, что очень удобно, особенно, если вы до этого привыкли к init. Формат вызова команды `service` такой же.

Upstart можно использовать в режиме "горячей замены" — если вы в процессе работы системы подключите какое-то устройство, например, PCMCIA-карту или USB-устройство, будет сгенерировано соответствующее событие. После этого будут запущены все необходимые для обеспечения работы этого устройства процессы. Так, при подключении сетевой карты PCMCIA будет сгенерировано событие `network-interface-added`, которое запустит процесс настройки сетевой карты по DHCP, при этом будет сгенерировано новое событие — `network-interface-up` и т. д. Понятно, что если нет сетевых устройств, то и соответствующие им события не будут генерироваться.

17.4. Система инициализации Slackware

Система инициализации Slackware отличается от привычной системы init, используемой в SysV-системах. Она больше похожа на систему инициализации BSD-систем, хотя некоторые сходства с SysV все же есть.

Пояснение

Если вы совсем незнакомы с историей UNIX, то вам неизвестны и термины SysV (System V) и BSD. Считается, что UNIX "родилась" в 1969 году. В то время над проектом работали сотрудники компании Bell Labs (это подразделение AT&T) Руд Кенедей (Rudd Canaday), Дуг Мак-Илрой (Doug McIlroy), Дэннис Ричи (Dennis Ritchie) и Кен Томпсон (Ken Thompson). Позже UNIX заинтересовались другие организации, в частности, институт Беркли (Калифорния, США). В 1975 году появилась слегка модифицированная версия UNIX от института Беркли, которая получила название BSD (Berkeley Software Distribution), а версия от AT&T (Bell Labs) стала называться System V (SysV). Обе системы были очень похожи друг на друга, но в то же время имели свои особенности. Например, BSD имела собственную систему инициализации, которая очень напоминает ту, что сейчас используется в Slackware Linux.

Если говорить о сходстве систем инициализации в стиле SysV и в стиле BSD, то у обеих систем присутствуют уровни запуска, имеется файл /etc/inittab — таблица инициализации (см. ранее). Однако имена файлов системы инициализации BSD-стиля немного отличаются от имен файлов SysV-стиля.

Система инициализации Slackware построена таким образом, что вне зависимости от уровня запуска первым всегда запускается сценарий /etc/rc.d/rc.S. Он монтирует псевдофайловые системы /proc, sysfs и devfs, запускает систему hotplug (драйвер устройств, обеспечивающий их "горячее" подключение, то есть подключение без выключения компьютера — например, USB-устройств), подключает разделы свопинга, монтирует и проверяет корневую файловую систему, монтирует другие файловые системы и т. д. Как видите, сценарий /etc/rc.d/rc.S выполняет бóльшую часть действий по инициализации системы. Обычно данный файл не требует изменения. Но иногда его приходится редактировать. Например, если вы создали файл подкачки и хотите, чтобы он подключался при загрузке системы, то команду `swapon <имя_файла>` нужно добавить в файл /etc/rc.d/rc.S после команды `/sbin/swapon -a`.

Сценарий /etc/rc.d/rc.S проверяет наличие файла /etc/rc.d/rc.modules.local, обеспечивающего загрузку модулей при старте системы. При условии, что файл rc.modules.local существует, он запускается. В противном случае происходит поиск файла /etc/rc.d/rc.modules<-версия.ядра>, а если и его нет, тогда сценарий /etc/rc.d/rc.S пытается запустить файл /etc/rc.d/rc.modules. Один из этих файлов должен существовать, иначе система будет загружена без модулей, а это означает, что не будут работать некоторые устройства и поддерживаться некоторые файловые системы.

Кроме файла /etc/rc.d/rc.modules.local (или другого файла загрузки модулей, см. ранее), также используется файл /etc/rc.d/rc.netdevice. Он служит для загрузки модулей сетевых карт (точнее сетевых интерфейсов).

Как уже было отмечено, файл /etc/rc.d/rc.S запускается вне зависимости от уровня запуска. Кроме этого файла в каталоге etc/rc.d вы найдете серию файлов rc.*N*, где *N* — номер уровня запуска. Данные файлы запускаются в зависимости от выбранного уровня запуска — например, на третьем уровне запуска будет запущен файл /etc/rc.d/rc.3. Каждый такой файл подготавливает систему к работе

на выбранном уровне запуска. Уровень запуска по умолчанию, как и в случае с системой инициализации в стиле SysV, задается в файле /etc/inittab.

Сценарий /etc/rc.d/rc.inet1 отвечает за инициализацию сетевых интерфейсов и построение таблицы маршрутизации. Конфигурация сетевых интерфейсов хранится в файле /etc/rc.d/rc.inet1.conf. Вот фрагмент этого файла:

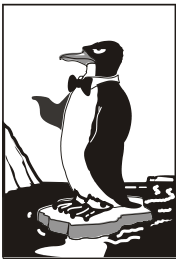
```
IPADDR[0]="192.168.1.1"
NETMASK[0]="255.255.255.0"
USE_DHCP[0]=""
DHCP_HOSTNAME[0]=""
```

Сценарий /etc/rc.d/r.inet2 управляет запуском сетевых служб и подключением сетевых файловых систем. Именно в этом файле происходит попытка монтирования файловых систем NFS и smbfs. Также из этого файла происходит запуск сетевых служб. Сценарии для запуска сетевых служб называются /etc/rc.d/rc.<название службы>, например, /etc/rc.d/rc.sshd — сценарий запуска SSH-сервера. Однако некоторые сетевые сервисы, например, sendmail и samba, в силу своих особенностей запускаются из файлов rc.*N*.

Иногда нужно обеспечить запуск сетевой службы, для которой нет собственного rc-файла. Тогда ее запуск можно или описать в файле /etc/rc.d/rc.local (что довольно просто), или создать собственный rc-файл и добавить его вызов в один из файлов rc.*N*. Шаблон собственного rc-файла приведен в листинге 17.3.

Листинг 17.3. Шаблон rc-файла для запуска сетевой службы

```
#! /bin/bash
start()
{
  echo "Service started"
  service_start
}
stop()
{
  echo "Service stoped"
  killall service
}

 case $1 in
  start)
    start ;;
  stop)
    stop ;;
  restart)
    stop
    sleep 2
    start ;;
  *)
    echo "Usage: service start|stop|restart"
esac
```



Глава 18

Пакеты и управление пакетами

18.1. Что такое пакет?

В Windows программное обеспечение устанавливается с помощью мастера установки — программы setup.exe или install.exe. Мастер установки свой для каждой программы, то есть программа setup.exe, предназначенная для установки MS Office, не установит Photoshop.

В Linux все иначе. Здесь используются два основных способа установки программного обеспечения:

- с помощью пакетов;
- из исходных кодов.

Пакет содержит все необходимое для установки программы. Существуют два основных типа пакетов:

- RPM-пакеты — применяются во всех Red Hat-совместимых дистрибутивах (Red Hat, Fedora, Mandriva, ALT Linux, ASPLinux и др.);
- DEB-пакеты — применяются в дистрибутиве Debian и в дистрибутивах, основанных на Debian (Ubuntu, Kubuntu, Edubuntu и др.).

> ***Примечание***
>
> В Slackware Linux используется собственный формат пакетов, не совместимый ни с RPM, ни с DEB. Об установке пакетов в Slackware мы поговорим отдельно.

Если в вашем дистрибутиве нет нужной вам программы, попробуйте найти ее пакет на следующих сайтах: **http://rpmfind.net** и **http://rpm.pbone.net** (для RPM-пакетов) или на **http://www.debian.org/distrib/packages** и **http://packages.ubuntu.com/** (для DEB-пакетов).

Если же вы не можете найти пакет программы в Интернете, тогда придется компилировать программу самому (при условии, что вы нашли архив с исходным кодом программы). Да, в Linux некоторые программы распространяются только в исходных кодах. Для установки такой программы нужно распаковать архив с исходными кодами (желательно, в каталог /usr/src), затем перейти в только что созданный каталог (содержащий исходные коды устанавливаемой программы) и выполнить следующие команды:

```
./configure
make
make install
```

Сценарий `configure` проверит, содержит ли ваша система необходимые библиотеки или программы, после чего, если все нормально, будет создан файл `Makefile`. Если вы увидели сообщение об ошибке, внимательно прочитайте его и попытайтесь устранить причину ошибки, например, установите недостающую библиотеку. Ясно, что в случае ошибки вводить последние две команды не нужно.

Вторая команда (`make`) на основании созданного файла `Makefile` компилирует программу. А последняя команда (`make install`) устанавливает программу и дополнительные файлы в дерево файловой системы (программы обычно в каталог /usr/bin, документацию — в /usr/share/doc, конфигурационные файлы — в /etc и т. д.).

> ***Совет***
>
> Для получения подробных инструкций по установке и удалению таких программ лучше всего просмотреть файл README, который обычно присутствует в архиве.

Устанавливаемая программа, как правило, состоит из набора файлов, например, исполнимого и конфигурационного файлов, файла справки. В зависимости от организации программы установки все эти файлы могут быть:

- заархивированы каждый отдельно — в этом случае мы получаем набор из $N + 1$ файлов (N — это файлы программы) плюс программа установки);
- заархивированы в один общий архив — у нас будет 2 файла: архив и программа установки;
- заархивированы в саму программу установки — самый удобный случай, когда у нас всего один файл — программа установки.

Как уже было отмечено, в современных дистрибутивах Linux все файлы, относящиеся к той или иной программе, помещаются в один файл — пакет. Пакет — это не просто архив, содержащий файлы программы. В пакете, кроме файлов программы, хранится служебная информация, описывающая процесс установки программы:

- пути — ведь один файл нужно скопировать, например, в каталог /usr/bin, а другой — в /usr/share/doc;
- дополнительные действия — например, создание каталога, установка тех или иных прав доступа к файлам и каталогам программы;
- зависимости — одна программа для своей работы может требовать какую-то библиотеку (без которой она не будет запускаться, поскольку использует функции этой библиотеки). Тогда в пакете указывается, что он *зависит* от другого пакета, содержащего библиотеку. При установке менеджер пакетов проверяет зависимости: если установлены не все пакеты, от которых зависит устанавливаемый пакет, установка будет прервана — пока вы не установите все необходимое. Правда, имеется возможность установки программы без удовлетворения зависимостей (тогда информация о зависимостях будет просто проигнорирована), но в большинстве случаев установленная таким образом программа работать не будет;
- конфликты — аналогично, одна программа может в системе конфликтовать с другой программой. Например, программы sendmail и postfix являются MTA-агентами (MTA, Mail Transfer Agent). Поскольку в системе может быть только

один MTA-агент, установить можно или sendmail, или postfix, то есть пакет sendmail конфликтует с пакетом postfix и наоборот.

Пакеты называются также RPM-файлами (или DEB-файлами — для дистрибутивов на основе Debian). С Debian все просто: пакеты были так названы, потому что последние три символа имени у файлов пакетов — deb (сокращение от Debian). Название RPM-файлов берет начало с разработок компании Red Hat, которая впервые предложила технологию RPM. Тогда в дистрибутиве Red Hat появился менеджер пакетов rpm (Red Hat Package Manager), откуда и название пакетов.

В имени пакета зашифрована некоторая информация о программе. Сделано это исключительно для удобства — можно узнать версию и другую информацию о программе, только лишь взглянув на название пакета, например:

```
program-1.5-14.i586.rpm
```

Здесь `program` — название программы, `1.5` — ее версия, `14` — выпуск пакета, `i586` — архитектура, на которую рассчитана программа. Не нужно пытаться устанавливать программы для архитектур i586/686 на компьютер с процессором Intel 386 или 486. Если программа независима от архитектуры, то указывается параметр `noarch` (обычно так делают для документации, примеров конфигурационных файлов, то есть для пакетов, содержащих информацию, которая не зависит от архитектуры).

18.2. Репозитории пакетов

Репозиторий — это хранилище пакетов. Репозиторий может быть локальным, например, каталогом на жестком диске или DVD, или же сетевым — сервером в Интернете или в локальной сети, содержащем RPM-пакеты. Для чего создаются репозитории? Для централизованного управления обновлением пакетов. Представьте, что у нас нет репозиториев. Тогда, чтобы узнать, вышла ли новая версия нужной вам программы, вам пришлось бы посещать сайт ее разработчика или по крайней мере сайт разработчика дистрибутива Linux. А это не очень удобно. Один раз вы можете забыть проверить наличие обновлений, а потом вам вообще надоест это делать. Проще дождаться выхода новой версии дистрибутива и обновить все программы за один раз.

Так и было раньше. Вот вышла программа, ее включили в состав дистрибутива, но полностью не протестировали (протестировать все невозможно). Оказалось, что программа работает неправильно, но только при определенных условиях, например, с определенным форматом файла. Или же Linux была установлена на сервер и организованы сетевые службы, например, Web-сервер. Через некоторое время оказалось, что в этой версии Web-сервера имеется "дыра", поэтому вскоре выпустили новую версию. Пользователь, установивший программу, ничего не подозревая о том, что вышла новая ее версия, мог бы мучаться минимум полгода или даже год — до выхода следующей версии дистрибутива. А его сервер могли бы взломать уже на следующий день после обнаружения "дыры". Но не тут-то было. Разработчики Linux, заботясь о нас с вами, создали репозитории. И с помощью репозиториев можно быстро и удобно отслеживать обновления тех

или иных пакетов. Причем это делает сам менеджер пакетов, а вам лишь остается указать, какие обновления нужно загружать, а какие — нет.

Практически все системы управления пакетами современных дистрибутивов поддерживают хранилища пакетов. В следующем разделе мы рассмотрим программы управления пакетами, использующиеся в современных дистрибутивах.

18.3. Программы для управления пакетами

Для управления пакетами в различных дистрибутивах используются разные программы. В табл. 18.1 приведены программы управления пакетами, которые можно встретить в современных дистрибутивах.

Таблица 18.1. *Программы управления пакетами*

Программа	Дистрибутив	Описание
rpm	Red Hat-совместимые дистрибутивы (Fedora, Mandriva, ALT Linux, ASPLinux, openSUSE и др.)	Простой менеджер пакетов. Работает в текстовом режиме. Не умеет разрешать зависимости пакетов
rpmdrake	Дистрибутивы, основанные на Mandrake (Mandriva)	Графический менеджер пакетов. Умеет разрешать зависимости и управлять источниками пакетов
urpmi	Дистрибутивы, основанные на Mandriva	Текстовый менеджер пакетов, поддерживающий источники пакетов и автоматически разрешающий зависимости
dpkg	Дистрибутивы, основанные на Debian (Ubuntu, Kubuntu, Denix и др.)	Простой менеджер пакетов. Работает в текстовом режиме. Не умеет разрешать зависимости пакетов
apt	Debian, Ubuntu (и клоны), ALT Linux и др.	Мощный менеджер пакетов, работающий в текстовом режиме. Умеет разрешать зависимости пакетов и поддерживает репозитории (источники пакетов)
yum	Fedora и др.	Мощный менеджер пакетов, работающий в текстовом режиме. Умеет разрешать зависимости пакетов и поддерживает репозитории (источники пакетов)
gpk-application pirut или system-config-packages	Fedora и дистрибутивы, основанные на нем (ASPLinux)	Графический менеджер пакетов. Впервые появился в одной из последних версий дистрибутива Red Hat, затем "перекочевал" в Fedora. По функциям похож на rpmdrake, хотя последний, все же, удобнее. В любом случае в Fedora вам придется довольствоваться только этим менеджером (если не считать yum). В последних версиях Fedora используется программа gpk-application
pkgtool	Slackware	Менеджер пакетов Slackware, заслуживающий отдельного разговора
zypper	openSUSE	Менеджер пакетов SUSE. Работает в текстовом режиме. Умеет разрешать зависимости пакетов

Примечание

Наверное, в таблице вы обратили внимание на фразу "умеет разрешать зависимости пакетов". Это означает следующее: если при установке пакета будет обнаружено, что для корректной его установки ему нужны дополнительные пакеты, то менеджер пакетов установит их. Если же менеджер пакетов не умеет разрешать зависимости, то он только сообщит, что установить пакет невозможно, и выведет лишь список файлов (файлов, а не пакетов!), которые нужны для установки данного пакета. А уж какой файл в каком пакете находится, вам придется догадываться самостоятельно.

18.4. Программы rpm и rpmbuild (все Red Hat-совместимые дистрибутивы)

Если вы хотите установить пакет, который не входит в состав дистрибутива (например, загруженный из Интернета), вам следует использовать программу rpm.

Совет

Для установки пакетов, которые входят в состав дистрибутива, намного удобнее использовать графический менеджер пакетов rpmdrake.

Программа rpm — полноценный текстовый менеджер пакетов, позволяющий устанавливать, удалять пакеты, просматривать информацию об уже установленных и новых пакетах, обновлять пакеты.

Чтобы установить пакет с помощью rpm, выполните команду:

```
# rpm -ihv <имя_пакета>
```

Удалить пакет так же просто:

```
# rpm -e <имя_пакета>
```

Для обновления пакета используется команда:

```
# rpm -U <имя_пакета>
```

Просмотреть, установлен ли тот или иной пакет, можно с помощью команды:

```
# rpm -qa | grep <имя_пакета>
```

Если вы хотите просмотреть информацию о пакете, то введите команду:

```
# rpm -qi <имя_пакета>
```

Просмотреть список файлов, входящих в состав пакета, можно командой:

```
# rpm -ql <имя_пакета>
```

Наконец, вывести все пакеты можно командой:

```
$ rpm -qa | grep more
```

Примечание

Программа rpm может также использоваться и для сборки собственных пакетов, но данная операция выходит за рамки этой книги. Вы можете прочитать мою статью о сборке собственных RPM-пакетов на сайте **http://www.dkws.org.ua/index.php?page=show&file=a/system/rpm_create**.

Обычно пакеты содержат уже откомпилированные версии программ. Но встречаются также и пакеты, содержащие исходный код. У таких пакетов "двойное рас-

ширение", то есть не просто .rpm, а .src.rpm. При установке такого пакета будет установлен исходный код программы, который можно будет потом откомпилировать. Но если вам нужна сама программа, а не ее исходный код, нет необходимости устанавливать пакет, а потом компилировать исходный код. Проще сразу вызвать программу rpmbuild так:

```
# rpmbuild --rebuild package.src.rpm
```

Если в процессе выполнения команды не возникнет ошибка, собранный RPM-пакет, содержащий уже откомпилированную версию программы, вы найдете в каталоге /usr/src/redhat/RPMS. Все, что вам останется — это установить пакет с помощью команды rpm.

18.5. Графический менеджер пакетов rpmdrake (Mandriva)

Для установки пакетов в Mandriva выполните команду главного меню **Установка и удаление программ**. Программа попросит ввести пароль root для продолжения работы.

> ***Примечание***
>
> Лично я предпочитаю открыть терминал и ввести команду `rpmdrake`, а не бродить по меню KDE/GNOME.

Программа rpmdrake (она же drakrpm) имеет несколько режимов отображения списка пакетов (выбор Mandriva, все пакеты по алфавиту, пакеты по группе) и два режима отображения информации о пакете (стандартная, максимальная информация). Если вы знаете, как называется пакет (хотя бы приблизительно), лучше просматривать список пакетов в режиме **Все**. Первый выпадающий список окна менеджера пакетов позволяет выбрать категорию пакетов (например, **Все**, **Пакеты с графическим интерфейсом** и т. д.). Дополнительные параметры списка пакетов можно найти в меню **Вид**. Второй выпадающий список позволяет отфильтровывать уже установленные пакеты и пакеты, доступные для установки. Если вы даже и приблизительно не знаете, что именно хотите установить, оптимальным является просмотр списка пакетов в сортировке по группам. Можно также ввести начальные буквы названия пакета в поле поиска и нажать клавишу <Enter> (рис. 18.1).

Искать можно в названиях пакетов, в описаниях и в именах файлов (способ поиска задается в меню **Вид**). Первый режим (**Все пакеты, по алфавиту**) удобен, если вы знаете приблизительное название пакета. Второй (**Все пакеты, по группам**) — если вы хотите найти сами не знаете что. Например, вы ищете игрушку, но не знаете, какую именно, — просто вам захотелось во что-то поиграть. Тогда в поле поиска введите слово `game`, выберите режим **в описаниях** и нажмите кнопку **Поиск**.

Чтобы установить пакеты, отметьте их (возле каждого пакета выводится флажок) и нажмите кнопку **Применить**. Напротив уже уставленных пакетов выводится зеленая пиктограмма со стрелкой вниз (справа от описания пакета). Если rpmdrake обнаружит, что для установки вашего пакета нужно удовлетворить зависимости (то есть установить дополнительные пакеты), то задаст вам соответст-

вующий вопрос. Если вы согласитесь, установка будет продолжена, в противном случае — прервана.

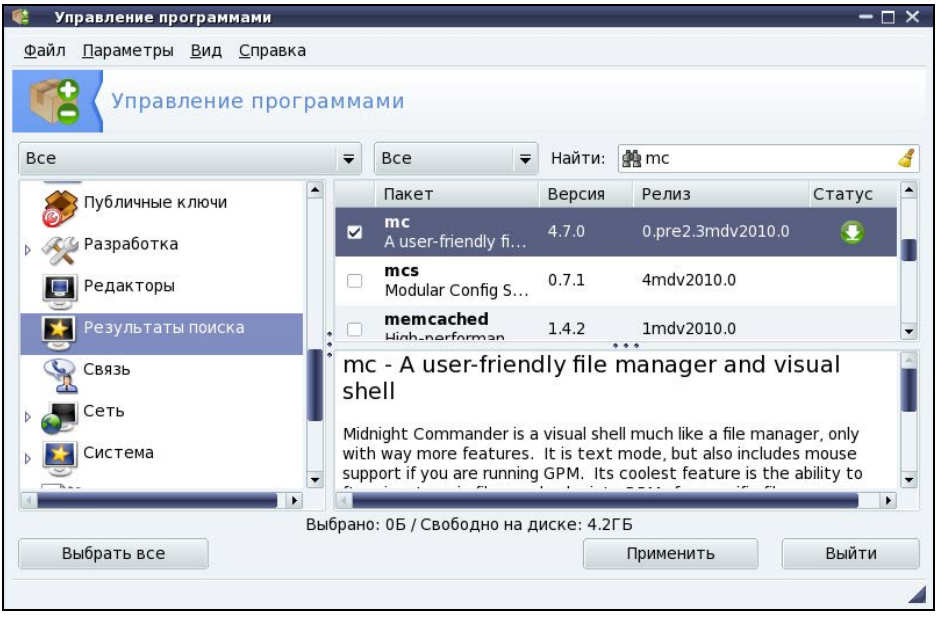


Рис. 18.1. Поиск пакетов

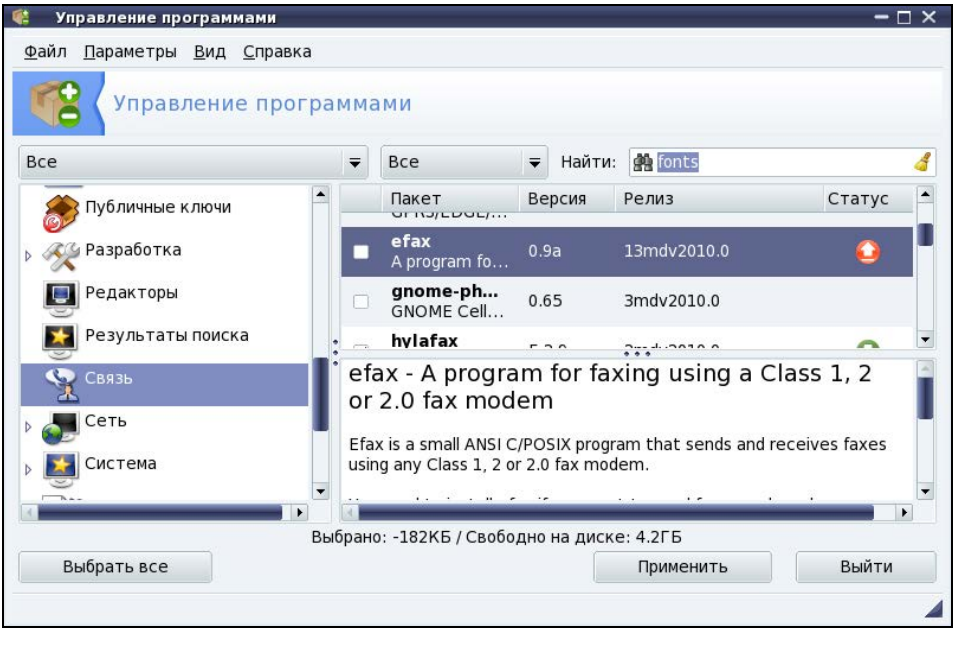


Рис. 18.2. Пакет efax помечен для удаления

Ранее для удаления пакетов использовался отдельный конфигуратор. Сейчас достаточно снять флажок, выводящийся слева от имени пакета. При этом значок статуса пакета будет изменен — пакеты, помеченные для удаления, отмечаются красным значком со стрелкой вверх (рис. 18.2). Для применения изменения (то есть для удаления пакетов) нужно нажать кнопку **Применить**.

Осуществляя поиск, программа rpmdrake просматривает список еще не установленных пакетов, который формируется в результате исключения уже установленных пакетов из общей базы пакетов. Общая база пакетов — это совокупность дистрибутивных дисков, которые называются *источниками пакетов*. При желании вы можете добавить в список источники пакетов с Web- и FTP-серверов. Делать это нужно только, если у вас высокоскоростной (и дешевый) доступ к Интернету. В противном случае проще через некоторое время купить следующую версию дистрибутива.

Для редактирования источников пакетов выполните команду **Параметры | Менеджер источников** (рис. 18.3). Как видно из рис. 18.3, по умолчанию Mandriva настроена на установочный диск, а не на репозиторий Интернета.

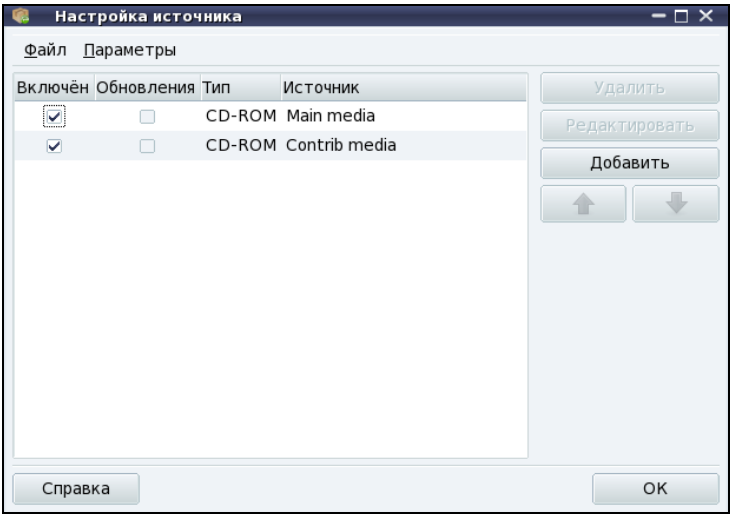


Рис. 18.3. Менеджер источников программ

18.6. Программа urpmi

Программа urpmi представляет собой систему управления пакетами, использующуюся в Mandriva. Как уже было отмечено в табл. 18.1, urpmi поддерживает зависимости пакетов. Конечно, обычным пользователям намного проще использовать программу rpmdrake для установки/удаления пакетов и управления источниками пакетов. Но rpmdrake — это всего лишь оболочка для системы urpmi, поэтому настоящий линуксоид должен знать, как работает urpmi.

Не нужно расценивать urpmi как замену rpm — система urpmi просто делает управление пакетами проще (хотя желающие могут использовать утилиту rpm, если сочтут ее более удобной).

Примечание

Я, например, предпочитаю использовать rpm для локальной установки пакетов (когда пакет из какого-либо источника уже закачан на мой компьютер).

18.6.1. Установка пакетов. Управления источниками пакетов

Для установки пакета служит команда:

```
# urpmi <имя пакета>
```

Так, для установки пакета mc (файловый менеджер Midnight Commander) следует ввести команду:

```
# urpmi mc
```

Программа просматривает список источников пакетов, хранящийся в файле /etc/urpmi/urpmi.conf. Если она находит пакет в одном из источников, то устанавливает его вместе со всеми необходимыми для его работы пакетами (при этом urpmi автоматически разрешает зависимости пакетов).

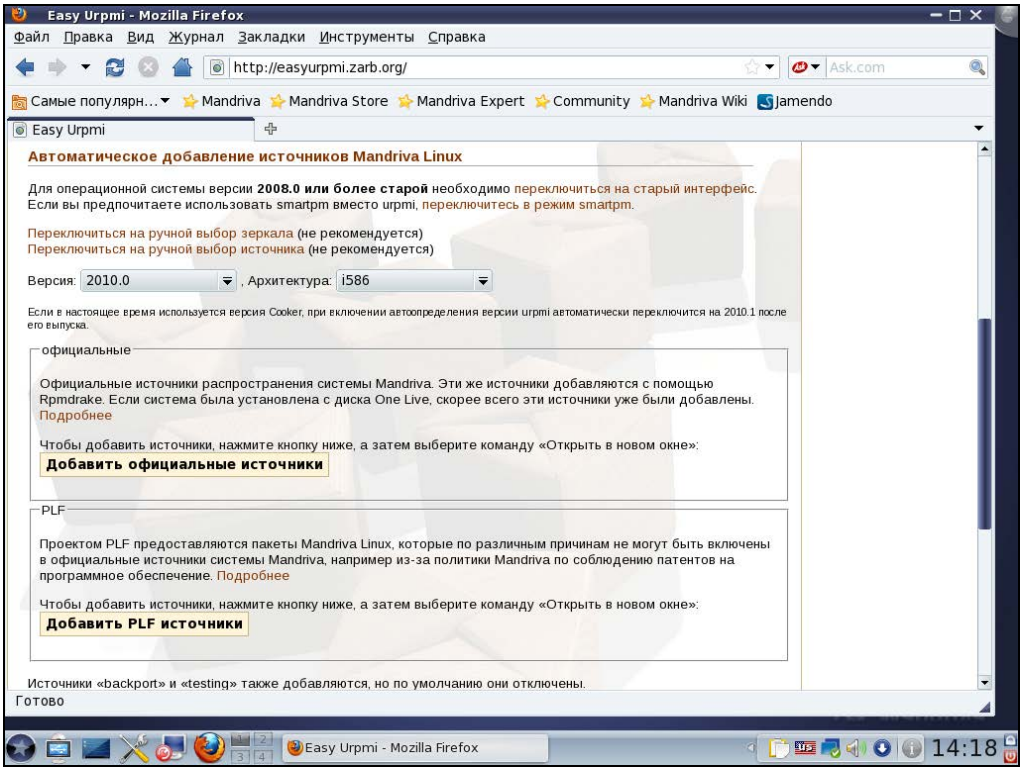


Рис. 18.4. Сайт **easyurpmi.zarb.org**

Существуют три вида репозиториев, поддерживаемых urpmi:

- хранилища на съемных носителях (removable) — репозитории на компакт-дисках, DVD, ZIP-носителях, Flash-дисках и т. д.;
- локальные (local) — находятся в каталоге на жестком диске;
- удаленные (distant server) — пакеты находятся на удаленном FTP- или HTTP-сервере.

Просмотреть список источников пакетов можно с помощью команды:

```
# urpmq --list-media
```

Добавить источники пакетов можно с помощью команды:

```
# urpmi.addmedia <источник>
```

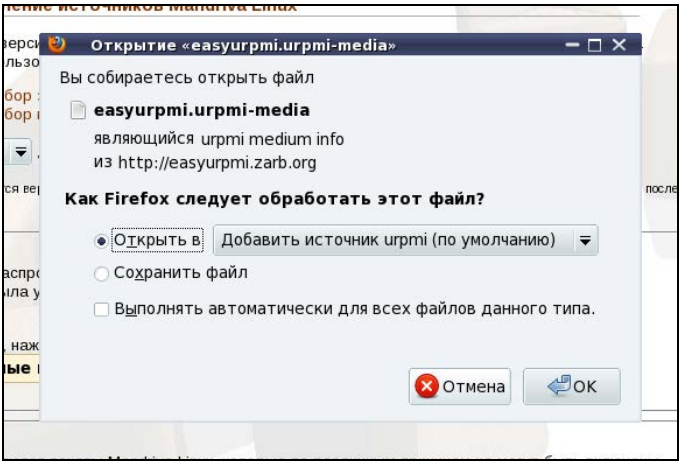


Рис. 18.5. Установка источника пакетов

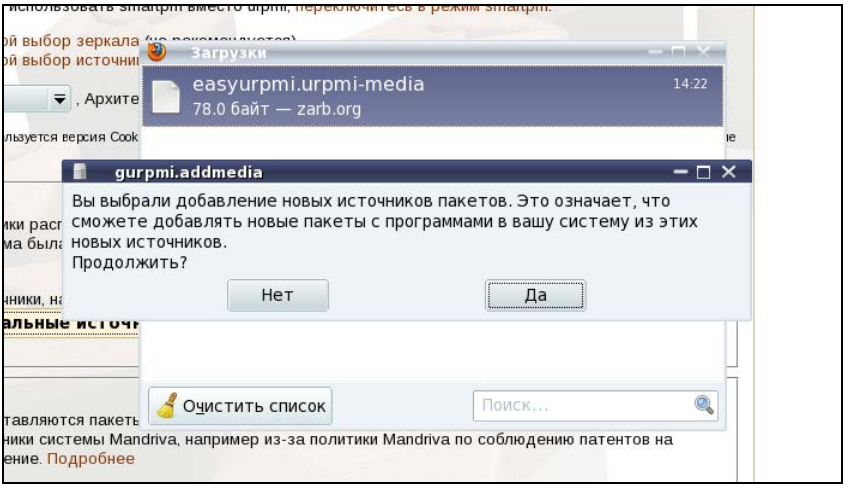


Рис. 18.6. Нажмите **Да** для установки источника пакетов

Получить список источников можно на сайте **http://easyurpmi.zarb.org**. Зайдите на этот сайт, выберите версию вашего дистрибутива (2010), архитектуру и нажмите кнопку **Добавить официальные источники** (рис. 18.4). В появившемся окне нажмите кнопку **OK** (рис. 18.5). Браузер скачает файл источника пакетов, запустит средство добавления источника, которое запросит у вас пароль root, после этого нужно нажать **Да** (рис. 18.6) для установки источника пакетов. После установки официальных источников установите PLF-источники.

После добавления источников пакетов мой файл конфигурации /etc/urpmi/urpmi.cfg (Madnriva 2010, платформа i586) стал выглядеть так, как показано в листинге 18.1. Листинг я несколько сократил, потому что в противном случае он бы растянулся на 4 страницы.

Листинг 18.1. Фрагмент файла /etc/urpmi/urpmi.cfg

```
{
}

Main\ media cdrom://i586/media/main {
key-ids: 70771ff3
}

Contrib\ media cdrom://i586/media/contrib {
key-ids: 78d019f5
}

Main {
key-ids: 70771ff3
mirrorlist: http://api.mandriva.com/mirrors/basic.2010.0.i586.list
with-dir: media/main/release
}

Main\ Updates {
key-ids: 22458a98
mirrorlist: http://api.mandriva.com/mirrors/basic.2010.0.i586.list
update
with-dir: media/main/updates
}

...
Contrib {
key-ids: 78d019f5
mirrorlist: http://api.mandriva.com/mirrors/basic.2010.0.i586.list
with-dir: media/contrib/release
}

...
```

```
Non-free {
key-ids: 70771ff3
mirrorlist: http://api.mandriva.com/mirrors/basic.2010.0.i586.list
with-dir: media/non-free/release
}

...

debug_non-free_release {
ignore
key-ids: 70771ff3
mirrorlist: http://api.mandriva.com/mirrors/basic.2010.0.i586.list
with-dir: media/debug_non-free/release
}

...
PLF\ Free {
key-ids: caba22ae
mirrorlist: http://plf.zarb.org/mirrors/2010.0.i586.list
update
with-dir: media/../../../../2010.0/free/release/binary/i586
}

PLF\ Free\ debug {
ignore
key-ids: caba22ae
mirrorlist: http://plf.zarb.org/mirrors/2010.0.i586.list
with-dir: media/../../../../2010.0/free/release/debug/i586
}

...
```

Для обновления репозитория (списка пакетов) используется команда:

```
# urpmi.update <имя источника>
```

Удалить источник пакетов можно или путем удаления информации о нем из файла urpmi.cfg, или с помощью команды:

```
# urpmi.removemedia <имя источника>
```

Для обновления всего списка пакетов используется команда:

```
# urpmi.update -a
```

Если вам от редактирования конфигурационных файлов вручную становится не по себе, вы можете использовать один из графических менеджеров управления источниками пакетов — тот, который вам больше понравится. Для этого выполните команду меню **Параметры | Менеджер источников** (см. рис. 18.3). На рис. 18.7 изображен **Менеджер источников** после установки дополнительных источников пакетов.

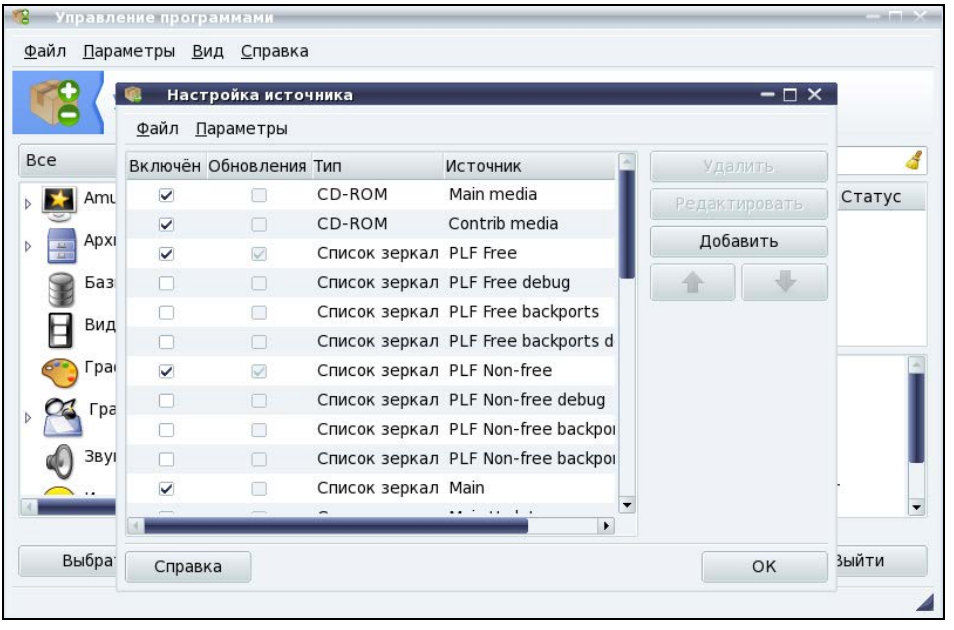


Рис. 18.7. Менеджер источников программ (после установки дополнительных источников)

18.6.2. Обновление и удаление пакетов

Для удаления пакета нужно ввести команду:

```
# urpme <пакет>
```

Если пакет нужен для работы других пакетов, то программа спросит у вас, хотите ли вы удалить и эти пакеты, иначе придется отказаться от удаления выбранного пакета.

Для обновления всей системы, то есть получения списка новых версий пакетов, используется команда:

```
# urpmi --auto-select
```

18.6.3. Поиск пакета. Получение информации о пакете

Найти пакеты, содержащие в названии определенную строку, можно с помощью команды:

```
# urpmq <строка>
```

Команда `urpmf` позволяет получить различную информацию о пакете, например:

- `urpmf <файл>` — выводит пакеты, содержащие указанный файл;
- `urpmf --group <группа>` — выводит пакеты, входящие в указанную группу;
- `urpmf --size <пакет>` — выводит размер указанного пакета;
- `urpmf --summary <пакет>` — выводит общую информацию о пакете.

18.7. Программа yum

Программа yum (Yellow dog Updater Modified) используется во многих дистрибутивах, в том числе и в Fedora.

Yum работает аналогично другим подобным программам (urpmi, apt) — когда вы устанавливаете пакет, yum производит поиск пакета в репозиториях, перечисленных в конфигурационном файле, загружает пакет и устанавливает его. В качестве репозитория могут выступать как дистрибутивные диски, так и серверы Интернета.

18.7.1. Использование yum

Общий формат вызова yum выглядит так:

```
yum команда [пакет(ы)]
```

Команды yum приведены в табл. 18.2.

Таблица 18.2. *Использование yum*

Команда	Описание
`yum install пакет`	Установить пакета из репозитория (также устанавливаются пакеты, необходимые для работы устанавливаемого пакета, то есть разрешаются зависимости)
`yum remove пакет`	Удалить пакет, а также все пакеты, которые зависят от данного
`yum update`	Проверить наличие обновлений всех пакетов. Если обновления есть, то они будут установлены
`yum update пакет`	Проверить обновления конкретного пакета. Если есть свежая версия, то она будет установлена
`yum check-update`	Только проверка наличия обновлений (обновления не устанавливаются)
`yum check-update пакет`	Проверка наличия обновлений конкретного пакета (обновления не устанавливаются)
`yum info пакет`	Вывести информацию о пакете
`yum list`	Выводит список всех пакетов. Выводятся как установленные, так и доступные для установки (в репозиториях) пакеты
`yum list a*`	Вывести список всех пакетов, которые начинаются на букву "a"
`yum search строка`	Найти все пакеты, в описаниях которых есть указанная строка
`yum groupinstall "группа"`	Установить все пакеты из указанной группы
`yum grouplist`	Вывести список групп пакетов

При установке пакетов с помощью yum не нужно далеко отходить от компьютера. Довольно часто нужные пакеты находятся не на локальных источниках, а на серверах в Интернете, поэтому yum выведет общий объем пакетов, которые вы хотите установить, и спросит вас, хотите ли вы их установить или нет:

```
Total download size: 10.5 M
It this ok [Y/N]:
```

Если вы согласны для установки выбранных пакетов загрузить 10,5 Мбайт файлов, нажмите клавишу <Y>, если передумали — нажмите <N>. Довольно удобно, иначе (с учетом того, что при разрешении зависимостей будут установлены дополнительные пакеты) можно при установке одного небольшого, на первый взгляд, пакета превысить месячную норму по трафику.

Получить информацию о пакете, как было показано в табл. 18.2, можно с помощью команды:

```
yum info пакет
```

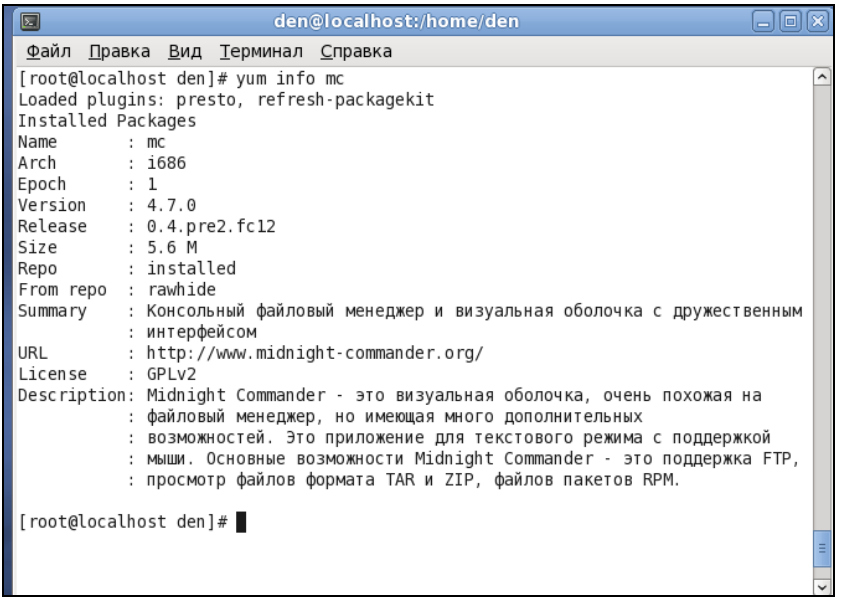


Рис. 18.8. Вывод информации о пакете

При этом на экран выводится следующая информация (рис. 18.8):

- **Name** — имя пакета;
- **Arch** — архитектура компьютера;
- **Epoch** — как бы подверсия пакета, поле Epoch используется, когда требуется уменьшить версию или релиз пакета по сравнению с имеющимся в репозитории;
- **Version** — версия пакета;
- **Release** — релиз пакета (можете считать это подверсией пакета);
- **Size** — размер занимаемого места на диске;
- **Repo** — хранилище пакета или значение **installed**, если пакет уже установлен;
- **Summary** — общая информация о пакете;
- **URL** — Web-страничка разработчика программы;
- **License** — лицензия, по которой распространяется программа;
- **Description** — описание пакета.

Для вывода всех пакетов можно использовать команду `yum list`, но пакетов слишком много, поэтому использовать ее неудобно. Удобнее задать маску имени пакета, например, `yum list a*` — в этом случае будут выведены все пакеты, начинающиеся на букву "a".

18.7.2. Управление источниками пакетов

Источники пакетов yum описываются в файле конфигурации /etc/yum.conf. Откройте этот файл (листинг 18.2).

> ***Совет***
>
> Обычно файл /etc/yum.conf приходится редактировать редко. Но помните, что делать это можно только от имени пользователя root. Если вы привыкли к графическому режиму, тогда в терминале для редактирования этого файла нужно ввести команду:
>
> `su -c <редактор> /etc/yum.conf`
>
> В качестве редактора могут выступать программы gedit (если у вас GNOME), kwrite или kate (если у вас KDE). Если открыть данный файл в редакторе без прав root, то просмотреть его вы сможете, но не сможете сохранить изменения.

Листинг 18.2. Конфигурационный файл yum.conf

```
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
metadata_expire=1800

# PUT YOUR    REPOS HERE OR IN separate files named file.repo
# in /etc/yum.repos.d
```

Ранее репозитории описывались непосредственно в файле yum.conf (как в случае с urpmi.cfg). Но потом было принято решение хранить описания репозиториев в отдельных файлах (REPO-файлах) в каталоге /etc/yum.repos.d. Каждый файл в этом каталоге называется так: `<имя репозитория>.repo`.

В листинге 18.3 приведен пример описания источника пакетов Fedora, взятый из файла fedora.repo.

Листинг 18.3. Пример описания источника пакетов

```
[fedora]
name=Fedora $releasever – $basearch
baseurl=http://download.fedora.redhat.com/pub/fedora/linus/releases/
$releasever/Everything/$basearch/os/
```

```
mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=fedora-
$releasever&arch=$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora file:///etc/pki/rpm-gpg/
RPM-GPG-KEY
```

Теперь разберемся, что здесь что. В квадратных скобках указывается сокращенное имя репозитория. Параметр `name` задает полное имя источника пакетов. Интернет-адрес (URL) источника пакетов указан параметром `baseurl`, а параметр `mirrorlist` задает список зеркал — копий репозитория, которые будут использоваться, если URL источника, указанный в `baseurl`, недоступен.

Параметр `enabled`, установленный в `1`, указывает на то, что данный источник активный, и yum использует его при установке пакетов. Следующий параметр, `gpgcheck`, указывает на то, что yum должен проверить подпись источника (если `gpgcheck=1`), а ключ, используемый для проверки подписи, задан параметром `gpgkey`.

Добавление источника производится путем добавления соответствующего ему REPO-файла в каталог /etc/yum.repos.d. Где этот файл взять? Обычно такие файлы представлены в виде RPM-пакетов на Web-серверах репозиториев. Поэтому нужно просто скачать RPM-пакет и установить его. Например, для установки REPO-файла популярного репозитория RPM Fusion нужно выполнить команду:

```
su -c 'rpm -Uvh http://download1.rpmfusion.org/free/fedora/rpmfusion-free-
release-stable.noarch.rpm
http://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-
stable.noarch.rpm'
```

Что делает данная команда, ясно и без комментариев. Если вы не можете найти соответствующий источнику REPO-файл, его можно написать вручную по формату листинга 18.3. При этом нужно еще знать базовый URL источника пакетов.

Удалять файлы источников пакетов, если сам источник уже не нужен, совсем не обязательно. Достаточно установить параметр `enabled` для источника в `0`. Тогда этот источник не будет использоваться.

18.7.3. Установка пакетов через прокси-сервер

По умолчанию yum полагает, что наш компьютер напрямую подключен к Интернету (не через прокси-сервер). Если вы подключаетесь к Интернету по локальной сети, то есть через прокси-сервер, данный факт нужно отразить в файле yum.conf, иначе вы не сможете устанавливать пакеты.

Узнайте у администратора сети параметры подключения к прокси-серверу (адрес, порт, имя пользователя и пароль) и пропишите их в файле yum.conf таким вот образом:

```
# Адрес прокси и его порт
proxy=http://proxy.company.ru:8080
# Имя пользователя и его пароль
proxy_username=dhsilabs
proxy_password=secret
```

18.7.4. Плагины для yum

Для yum доступно множество плагинов. Мы установим два из них: fastestmirror и presto. Первый плагин позволяет найти самый быстрый источник пакетов, что существенно сокращает время их установки. А второй пытается загружать только обновленные части пакетов вместо полной загрузки пакетов при обновлении, что сокращает трафик и уменьшает время обновления.

Для установки этих плагинов введите команды:

```
# yum install yum-plugin-fastestmirror
# yum install yum-presto
```

18.8. Графический менеджер пакетов в Fedora — gpk-application

В последних версиях Fedora используется графический менеджер пакетов (рис. 18.9), запустить который можно командой gpk-application или с помощью меню **Система | Администрирование | Add/Remove Software**. В старых версиях Fedora использовались конфигураторы pirut и system-config-packages.

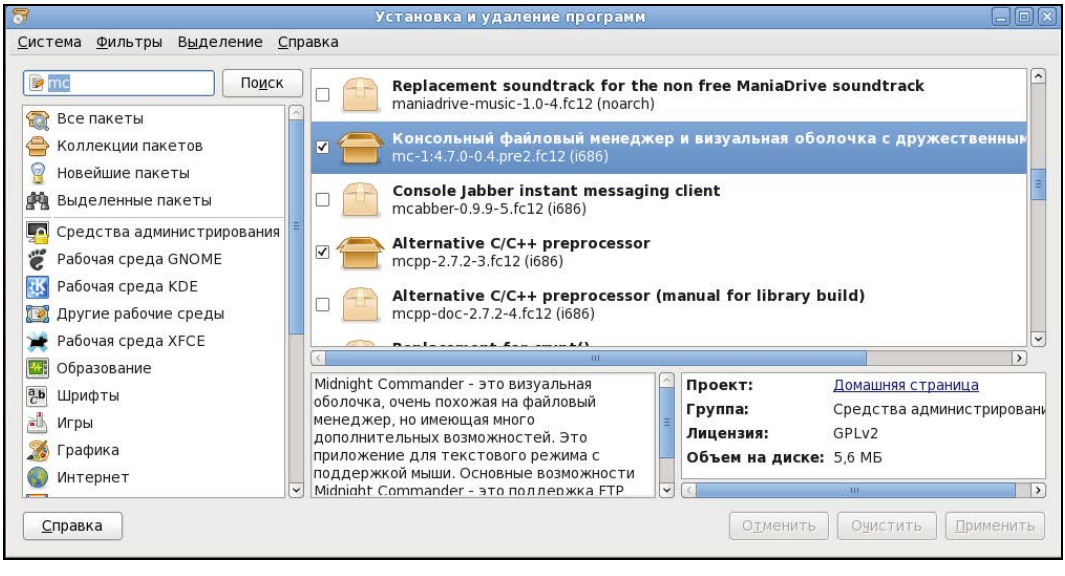


Рис. 18.9. Менеджер пакетов gpk-application

Использовать этот графический менеджер не сложнее, чем любой графический менеджер пакетов (тот же rpmdrake). Слева от имени пакета выводится переключатель: включив его, вы помечаете пакет для установки, а выключив — для обновления. Нажав кнопку **Применить**, вы примените изменения, то есть удалите или установите пакеты.

Как и во всех предыдущих версиях, менеджер пакетов Fedora по умолчанию настроен на использование интернет-репозитория для установки и обновления пакетов (а не на установочный диск, как Mandriva).

Примечание

Ранее в моих книгах (как правило, изданных до 2010 года) описывалось, как заставить менеджер пакетов Fedora устанавливать пакеты с дистрибутивного диска. В этой книге подобного материала для Fedora 12/13 не будет. Во-первых, скорость Интернета выросла, стоимость доступа снизилась, и высокоскоростной Интернет теперь доступен почти каждому. А при установке пакетов из Интернета у вас будут всегда самые новые версии пакетов. Во-вторых, чуть ранее в этой главе мы установили два плагина, уменьшающих время загрузки пакетов и экономящих ваш трафик, поэтому не вижу более смысла использовать устаревшие пакеты с установочного DVD. Если у вас медленное соединение или вы принципиально желаете устанавливать пакеты с установочного диска, а не из интернет-репозитория, тогда посетите следующую страничку: **http://www.dkws.org.ua/phpbb2/viewtopic.php?p=23984**. На ней, хотя и описывается настройка менеджера пакетов Fedora 9, вам не составит большого труда настроить "по образу и подобию" Fedora 12/13.

18.9. Программы dpkg и apt-get: установка пакетов в Debian/Ubuntu

18.9.1. Программа dpkg

Программа dpkg используется для установки, удаления и управления пакетами Debian/Ubuntu/Denix и вызывается из командной строки. Формат вызова следующий:

```
dpkg [ключи] действие
```

Для запуска dpkg нужно обладать полномочиями root, получить которые можно с помощью команды `sudo`. Рассмотрим, как правильно работать с программой dpkg.

Предположим, у нас есть пакет package.deb. Для его установки откройте терминал (**Приложения | Стандартные | Терминал**) и введите команду:

```
sudo dpkg -i /<путь>/package.deb
```

Как видите, в установке пакета нет ничего сложного. Процесс установки состоит из следующих шагов:

1. Извлечение управляющих файлов из пакета.
2. Если уже была установлена старая версия этого пакета, тогда из старого пакета запускается сценарий prerm (он подготавливает систему к удалению старой версии пакета). Другими словами, если нужно, то обновление пакета выполняется автоматически.
3. Выполняется сценарий preinst, если он есть в данном пакете.
4. Распаковываются остальные файлы из пакета (если был установлен старый пакет, то его файлы не удаляются, а сохраняются в другом месте, чтобы их можно было восстановить, если что-то пойдет не так).
5. Если была установлена старая версия пакета, то выполняется сценарий postrm (действия после удаления) из старого пакета. Сценарий запускается сразу после выполнения сценария preinst нового пакета, поскольку старые файлы удаляются во время записи новых файлов.

6. Выполняется настройка пакета:
   - распаковываются новые конфигурационные файлы, а старые сохраняются, если нужно будет их восстановить в случае ошибки во время установки нового пакета;
   - запускается сценарий postinst, если он есть в данном пакете.

Удалить пакет тоже просто:

```
sudo dpkg -r <package>
```

При удалении пакета не нужно указывать путь к пакету и "расширение" пакета, то есть символы `.deb` в конце имени файла.

Но установка и удаление пакетов — это далеко не все, что можно выполнить с помощью программы dpkg. Другие действия программы dpkg, которые могут быть интересны каждому пользователю Ubuntu, представлены в табл. 18.3.

Таблица 18.3. *Вспомогательные действия программы dpkg*

Ключ	Описание
`-l [образец]`	Выводит все установленные пакеты, имена которых соответствуют образцу. Образец задается с помощью масок `*` и `?`, например, образец `a*` соответствует любому имени пакета, начинающемуся на букву "a". Если образец не задан, выводятся все пакеты
`-L <имя_пакета>`	Выводит имена файлов из указанного пакета (пакет должен быть установлен)
`-p <имя_пакета>`	Выводит информацию об установленном пакете
`-s <имя_пакета>`	Выводит информацию о статусе пакета
`--unpack <имя_пакета.deb>`	Распаковывает, но не устанавливает пакет (полезно, если устанавливать пакет не требуется, а нужно лишь достать из него один или несколько файлов)

Если вы хотите получить более подробную информацию о программе dpkg, введите команду: `man dpkg` — страница руководства будет выведена на русском языке.

18.9.2. Программа apt-get

Программа apt-get применяется не только в Debian/Ubuntu, но и в других дистрибутивах, причем даже в Red Hat-совместимых (например, в ALT Linux), но там она используется для установки RPM-пакетов, а не DEB. Вообще, выбор менеджера пакетов зависит от разработчиков дистрибутива. В одной версии дистрибутива может использоваться apt-get, в другой — yum, а в третьей — какой-то новый и перспективный менеджер пакетов.

Предположим, что у нас есть пакет package.deb. При его установке обнаружилось, что он требует пакет lib.deb, который не установлен. Вы находите в Интернете нужный пакет, устанавливаете его, а затем устанавливаете пакет package.deb. Не очень удобно, правда?

Намного проще выполнить команду:

```
sudo apt-get install package
```

Программа apt-get просматривает файл /etc/apt/sources.list — в этом файле перечислены источники (репозитории) DEB-пакетов. В качестве источника может выступать как компакт-диск, содержащий пакеты, так и сервер в Интернете. Программа находит указанный пакет, читает служебную информацию о нем, затем разрешает зависимости (то есть устанавливает все другие пакеты, нужные для работы программ устанавливаемого пакета), а после устанавливает нужный нам пакет. Все загруженные программой apt-get и менеджером Synaptic (о нем — далее) пакеты записываются в каталог /var/cache/apt/archives.

Взглянем на файл /etc/apt/sources.list:

```
sudo gedit /etc/apt/sources.list
```

> ***Пояснение***
>
> В Ubuntu стандартный текстовый редактор называется gedit. В Kubuntu его нет, поэтому для правки файла нужно использовать текстовый редактор Kate. А в Xubuntu текстовый редактор называется mousepad.

Наверное, вам интересно, какие программы находятся в том или ином репозитории Ubuntu? В репозитории main находятся основные программы, они распространяются свободно и регулярно поддерживаются (обновляются). В репозитории restricted содержатся программы, которые распространяются по несвободным лицензиям, а также имеют ограниченную поддержку. Репозиторий universe содержит программы с открытыми лицензиями, поддержка программ из этого репозитория не гарантируется, но вполне возможна, все зависит от разработчика программы. В репозитории multiverse содержатся программы, которые распространяются несвободно и без всякой поддержки и гарантий. Репозиторий security содержит исправления пакетов из репозиториев main и restricted. Наконец, в репозитории backports есть неофициальные пакеты свежих версий программ, собранные из исходных текстов энтузиастами Ubuntu (а не разработчиками программ).

Чтобы настроить менеджер пакетов на русские репозитории (соответственно скорость загрузки пакетов будет выше), замените во всех строках файла /etc/apt/sources.list адрес `archive.ubuntu.com` на `ru.archive.ubuntu.com`.

Понятно, что программа apt-get может использоваться не только для установки пакетов. Общий формат вызова этой программы следующий:

```
apt-get [опции] команды [пакет]
```

Основные команды apt-get представлены в табл. 18.4.

Таблица 18.4. *Основные команды apt-get*

Команда	Описание
`update`	Синхронизирует файлы описаний пакетов (внутреннюю базу данных о пакетах) с источниками пакетов, которые указаны в файле /etc/apt/sources.list
`upgrade`	Обновляет указанный пакет. Может использоваться для обновления всех установленных пакетов. При этом установка новых пакетов не производится, а загружаются и устанавливаются только новые версии уже установленных пакетов

Таблица 18.4 *(окончание)*

Команда	Описание
`dist-upgrade`	Обновление дистрибутива. Для обновления всех пакетов рекомендуется использовать именно эту команду
`install`	Установка одного или нескольких пакетов
`remove`	Удаление одного или нескольких пакетов
`check`	Используется для поиска нарушенных зависимостей
`clean`	Используется для очистки локального хранилища полученных пакетов (перед установкой пакет загружается в локальное хранилище, а затем устанавливается оттуда; данная команда может очистить хранилище для экономии дискового пространства)

18.9.3. Установка RPM-пакетов в Debian/Ubuntu

Если у вас есть RPM-файл, его можно преобразовать в формат DEB с помощью команды `alien`. Сразу хочу заметить, что установка таких — преобразованных — пакетов не желательна, поскольку нет никакой гарантии, что установленная программа будет работать, но если другого выхода нет, можно попробовать:

```
sudo alien package_file.rpm
```

Если система сообщит вам, что команда `alien` не найдена, тогда нужно подключиться к Интернету и установить ее с помощью команды:

```
sudo apt-get install alien
```

18.9.4. Подключение репозитория Medibuntu

Репозиторий Medibuntu содержит мультимедиакодеки и различные мультимедиапроигрыватели. Для его установки введите команды:

```
sudo wget http://www.medibuntu.org/sources.list.d/$(lsb_release -cs).list \
 --output-document=/etc/apt/sources.list.d/medibuntu.list &&
sudo apt-get -q update &&
sudo apt-get --yes -q --allow-unauthenticated install medibuntu-keyring &&
sudo apt-get -q update
```

Нужно отметить, что репозиторий Medibuntu автоматически отключается утилитой обновления Ubuntu. Поэтому после очередного обновления не забудьте заново ввести указанные выше команды.

18.9.5. Графический менеджер Synaptic в Debian/Ubuntu

Дистрибутивы Debian/Ubuntu включают удобный графический менеджер пакетов Synaptic (рис. 18.10), запустить который можно с помощью команды меню **Система | Администрирование | Менеджер пакетов Synaptic**. На самом деле Synaptic — просто оболочка для apt-get, но оболочка очень удобная. Рассматривать

Synaptic подробно мы здесь не будем — он очень прост, и вы разберетесь с ним без моих комментариев.

При инсталляции Debian следует иметь в виду, что дистрибутив Debian 4 поставляется на трех DVD, но по умолчанию в качестве репозитория прописывается только первый DVD, остальные два диска не задействуются. Понятно, что хочется использовать размещенные на них пакеты. Запустите Synaptic (**Система | Администрирование | Программа управления пакетами Synaptic**). Выполните команду меню **Настройки | Репозитории**, а в открывшемся окне нажмите кнопку **Добавить Cdrom**. Вставьте второй диск и нажмите **OK**. Программа добавит второй DVD в список репозиториев и выведет сообщение о необходимости обновления источников пакетов. Для этого нажмите кнопку **Получить сведения** на панели Synaptic. Повторите все сказанное и для третьего DVD.

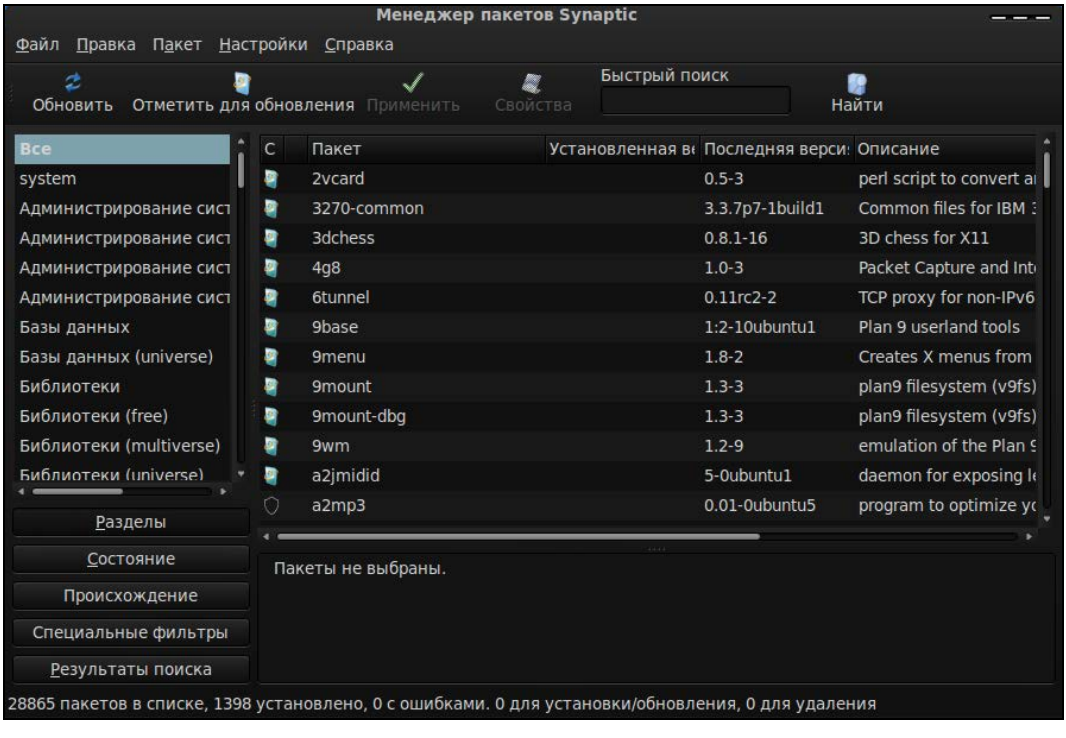


Рис. 18.10. Менеджер пакетов Synaptic

18.10. Установка пакетов в Slackware

Slackware в плане установки пакетов — довольно специфический дистрибутив. Мне частенько приходилось слышать мифы о сложности установки и управления пакетами в Slackware. Но все эти мифы, как оказалось, от незнания. Просто пользователям, привыкшим к Red Hat-совместимым дистрибутивам, трудно привыкнуть к особенностям Slackware. Возможно, "коренным" пользователям Slackware трудно

привыкнуть к обращению с RPM-пакетами… Тут утверждать не буду, потому что сам начинал свой путь линуксоида с дистрибутива Red Hat.

Но однажды я не выдержал и установил на свой компьютер Slackware. Цель была одна — разобраться с установкой пакетов. Неужели все так сложно? Как оказалось, ничего сложного нет, если понять особенности Slackware, не известные пользователям Red Hat.

Прежде чем приступить к рассмотрению системы управления пакетами, приведу ряд мифов, которые мне удалось разрушить:

- *в Slackware нет системы управления пакетами* — очевидно, данный миф сотворили пользователи, которые никогда не устанавливали Slackware, потому что такая система есть. Другое дело, что она не поддерживает RPM/DEB-пакеты. Пакеты Slackware выполнены в виде обычных TGZ-архивов. Но и формат пакетов RPM — это тоже слегка модифицированный архивный формат, просто его назвали иначе, а в Slackware используются обычные архивы. Хорошо это или плохо, решать вам. Но учитывая, что Slackware появился намного раньше, чем Red Hat с его системой RPM, использование архивов TGZ вполне закономерно;
- *в Slackware нет зависимостей пакетов* — это тоже миф, правда, в нем есть доля правды. Зависимости есть, но программы для установки пакетов их не обрабатывают — обработка зависимостей возложена на пользователя. Хорошо это или плохо? С одной стороны, есть вероятность недоустановить какой-то пакет или же удалить пакет, необходимый другим пакетам, что нарушит зависимости пакетов. Можно также установить пакет, который будет конфликтовать с уже установленными пакетами. Одним словом, при установке программного обеспечения нужно четко себе представлять, что вы делаете, а то очень легко превратить свою систему в мусорку, для наведения полного порядка в которой поможет только переустановка системы. Если в дистрибутивах, основанных на RPM/DEB, можно положиться на менеджер пакетов, то в Slackware нужно рассчитывать только на себя, поэтому перед установкой пакета поможет прочтение соответствующей пакету документации. С другой стороны, пакеты в Slackware достаточно объемные и содержат практически все необходимое для работы конкретного программного продукта. Например, чтобы установить PHP в Mandriva, вам понадобится 21 пакет, причем каждый из этих пакетов каким-то образом зависит от других пакетов группы. А вот для установки PHP в Slackware нужно установить всего один пакет, который включает все необходимое. Поэтому можно сказать, что разрешение зависимостей в Slackware совсем необязательно;
- *в Slackware отсутствует механизм обновления системы* — комментарии здесь примерно такие же, как и в предыдущем случае. Такой механизм есть, и его достаточно просто использовать, нужно только знать как;
- *в Slackware неудобно устанавливать программы, не входящие в состав дистрибутива*, — вот тут огромная доля правды. Можно даже сказать, что это не миф… С самой установкой ничего сложного нет, есть сложности с поиском необходимых пакетов. Но об этом мы поговорим чуть позже.

Вот теперь можно приступить к рассмотрению системы управления пакетами Slackware.

18.10.1. Управление пакетами

Для управления пакетами в Slackware используются четыре основные программы:

- pkgtool — псевдографический (использует текстовые меню) менеджер пакетов, позволяющий устанавливать, удалять и обновлять пакеты (рис. 18.11).

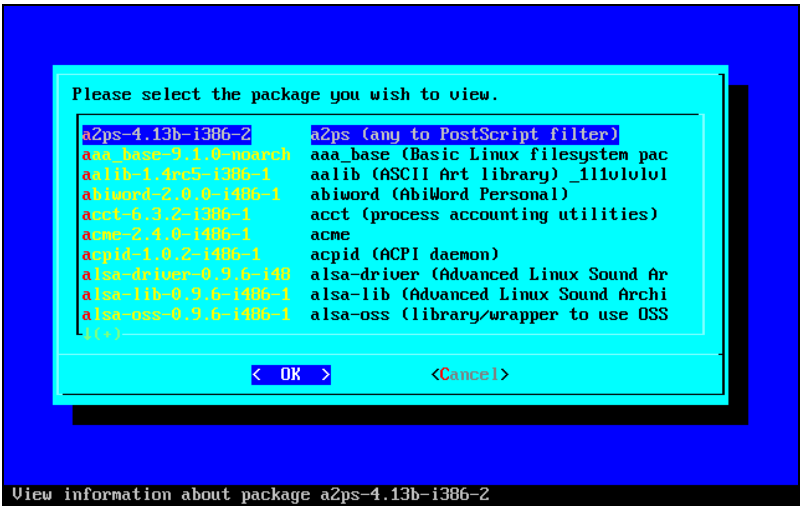


Рис. 18.11. Программа pkgtool

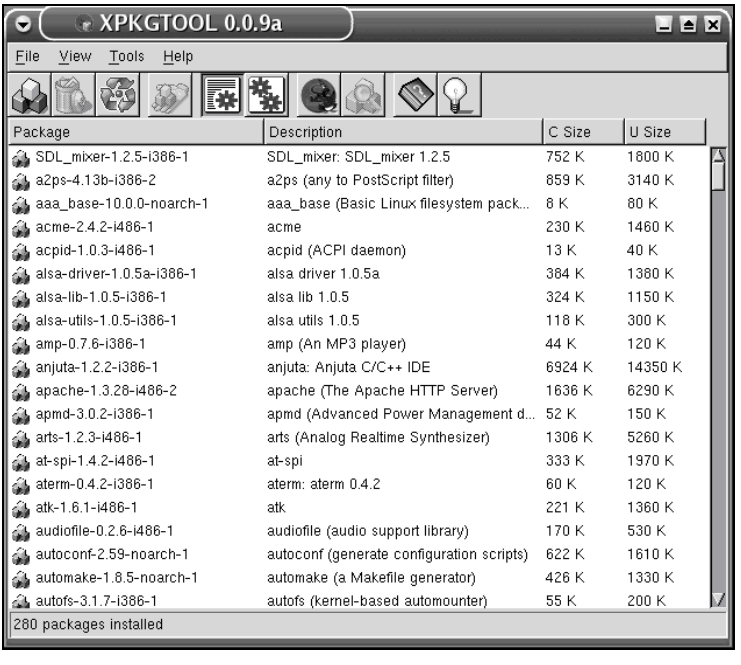


Рис. 18.12. Программа XPKGTOOL

В его работе несложно разобраться, поэтому мы подробно его рассматривать не будем. А любителям графических конфигураторов наверняка понравится графическая версия этой программы — XPKGTOOL (рис. 18.12);

- installpkg — программа для установки пакетов;
- removepkg — программа удаления пакетов;
- upgradepkg — программа обновления пакетов.

Программа установки пакетов installpkg

Перед рассмотрением программы installpkg определимся со структурой пакета. Как уже было отмечено, пакет с программным обеспечением в Slackware — это обычный TGZ-архив, предназначенный для распаковки в корневой каталог файловой системы. Вот пример структуры каталогов вымышленного пакета, содержащего всего одну программу — program:

```
./
usr/
usr/bin/
usr/bin/program
usr/man/
usr/man/man1
usr/man/man1/program.1.gz
install/
install/doinst.sh
```

Обратите внимание на каталог `install` — в нем находится сценарий `doinst.sh`, запускающийся после установки пакета.

Синтаксис команды для установки пакета:

```
# installpkg <опция> <имя пакета>
```

Вы можете задать одну из трех опций программы:

- `-m` — используется для сборки пакета (действие `makepkg`) в текущем каталоге;
- `-warn` — режим предупреждений: установка пакета не производится, однако выводится список планируемых действий. Если вы устанавливаете пакет на критически важной системе или просто не уверены в своих действиях, перед установкой пакета рекомендуется использовать режим предупреждений;
- `-r` — рекурсивно устанавливает все пакеты из текущего каталога и всех его подкаталогов.

Информация об установленных пакетах хранится в файле /var/log/packages. При установке пакетов вы можете указывать сразу несколько пакетов, а также использовать маски имен (типа `gnome*`).

Как уже было отмечено, при установке пакетов не проверяются зависимости пакетов, поэтому желательно первую установку производить в режиме `warn`. Также installpkg не сообщит, если вы попытаетесь установить уже установленный пакет. Программа просто перезапишет старые файлы новыми версиями (из устанавливаемого пакета). Вы думаете, что это недостаток? Может и так, зато

легко производить обновление пакета — можно просто использовать программу installpkg, хотя для более безопасного обновления рекомендуется использовать программу upgradepkg.

Пример вызова программы:

```
# installpkg bash-2.04b-i386-2.tgz
```

Программа удаления пакетов removepkg

Формат вызова программы removepkg такой же, как и в предыдущем случае:

```
# removepkg <опция> <имя пакета>
```

Опций у removepkg немного больше — четыре:

- `-copy` — копирует пакет в резервный каталог, но не удаляет его (см. опцию `preserve`);
- `-keep` — сохраняет временные файлы, которые программа создает при удалении пакета. Полезно при тестировании созданных вами пакетов (если вы разработчик/сборщик пакета);
- `-preserve` — удаляет пакет, но перед удалением копирует его в резервный каталог. Место на диске с этой опцией не сэкономишь, зато пакеты можно не удалять полностью из системы;
- `-warn` — режим предупреждения: не удаляет пакет, а просто показывает список действий, которые будут выполнены при удалении пакета.

Пример вызова программы:

```
# removepkg bash
```

Программа обновления пакетов upgradepkg

Использовать программу обновления пакетов очень просто:

```
# upgradepkg <имя пакета>
```

Программа сначала устанавливает новую версию пакета, а затем — удаляет старую, дабы в системе не остались старые версии файлов.

18.10.2. Нет нужного пакета: вам поможет программа rpm2tgz

Иногда просто невозможно найти программу, распространяющуюся в пакете Slackware, — большинство пакетов распространяется в формате RPM. В этом случае можно попробовать использовать программу rpm2tgz, преобразующую пакет формата RPM в формат Slackware. При этом следует понимать, что данная программа преобразует лишь формат пакетов, она не занимается разрешением зависимостей и т. п., то есть нет никакой гарантии, что после такого преобразования установленная программа будет работать.

> ***Совет***
>
> Вы думаете, что для вашей программы нет Slackware-пакета? А может, вы не там искали? Попробуйте посетить сайт **http://linuxpackages.net/** — там есть очень много Slackware-пакетов.

18.10.3. Программа slackpkg: установка пакетов из Интернета

Наверное, вы заметили, что программа installpkg занимается установкой пакетов из локального каталога. А что делать, если пакет находится в Интернете? Понятно, что его нужно скачать и установить программой installpkg, но если вы привыкли к программам вроде yum, Slackware вам может показаться несколько ущербным и малофункциональным дистрибутивом.

На помощь приходит программа slackpkg, позволяющая несколько автоматизировать установку пакетов из сетевых источников — в Slackware сетевые источники называются *зеркалами* (от англ. *mirrors*). Программа slackpkg может скачать и установить пакет, находящийся на одном из серверов-зеркал. Но эта программа не занимается разрешением зависимостей, а только несколько упрощает установку и обновление пакетов. Не нужно думать, что slackpkg — это замена installpkg, она всего лишь ее полезное дополнение, позволяющее немного облегчить установку пакетов.

Программа slackpkg находится в каталоге extra. После установки программы slackpkg нужно подготовить ее к работе. Первым делом откройте ее главный конфигурационный файл /etc/slackpkg/mirrors и раскомментируйте географически ближайшее к вам зеркало. Зеркало — это просто адрес FTP-сервера, содержащего Slackware-пакеты:

```
ftp://ftp.nluug.nl/pub/os/Linux/distr/slackware/slackware-12.0/
```

> ***Внимание!***
>
> Помните, что slackpkg позволяет использовать только одно зеркало. Если вы раскомментируете несколько зеркал, будет использоваться первое раскомментированное зеркало.

После редактирования файла зеркал нужно подготовить программу для работы с GPG-ключами.

Для этого введите команды:

```
# mkdir ~/.gnupg
# gpg --keyserver pgp.mit.edu --search security@slackware.com
```

При выполнении второй команды на экран будет выведено следующее сообщение:

```
gpg: searching for "security@slackware.com" from HKP server pgp.mit.edu
Keys 1-2 of 2 for "security@slackware.com"
(1)    Slackware Linux Project <security@slackware.com>
       1024 bit DSA key 40102233, created 2003-02-25
(2)    Slackware Linux Project <security@slackware.com>
       1024 bit DSA key 40102233, created 2003-02-25
Enter number(s), N)ext, or Q)uit >
```

Как видите, вас просят выбрать номер GPG-ключа. Введите номер одного из доступных GPG-ключей (список ключей перед вами, обычно можно ввести `1`).

Теперь вам осталось ввести еще одну команду:

```
gpg --fingerprint security@slackware.com
```

Все, программа slackpkg готова к использованию.

Перед установкой пакетов не помешает обновить список пакетов активного зеркала. Для этого используется команда:

```
# slackpkg upgrade
```

Чтобы иметь постоянно свежие сведения о пакетах, рекомендуется регулярно выполнять эту команду.

Для установки пакета введите команду:

```
# slackpkg install <пакет>
```

Для обновления пакета используется команда:

```
# slackpkg upgrade <пакет>
```

18.11. Установка программ в openSUSE

18.11.1. Менеджер пакетов zypper

Менеджер пакетов zypper работает по уже знакомому нам сценарию. Имеется список источников пакетов (каталог /etc/zypp/repos.d), который просматривается перед установкой пакета с целью определения хранилища, в котором находится устанавливаемый пакет. Затем менеджер пакетов загружает необходимый пакет (или пакеты) и устанавливает его.

Зайдите в каталог /etc/zypp/repos.d. В нем вы обнаружите несколько REPO-файлов, в каждом из которых прописан один репозиторий. В листинге 18.4 представлен репозиторий установочного DVD.

Листинг 18.4. Репозиторий установочного DVD (локальный репозиторий)

```
[openSUSE-11.2-DVD 11.2]
name=openSUSE-11.2-DVD 11.2
enabled=1
autorefresh=0
baseurl=cd:///?devices=/dev/sr0
path=/
type=yast2
gpgcheck=1
```

Параметр `baseurl` задает путь к источнику пакетов, а параметр `enabled`, установленный в `1`, говорит о том, что этот репозиторий активный.

Пример сетевого источника пакетов Main Repository (OSS) приведен в листинге 18.5.

Листинг 18.5. Пример сетевого репозитория

```
[Main Repository (OSS)]
name=Main Repository (OSS)
```

```
baseurl=http://download.opensuse.org/repositories/openSUSE:10.3/standard/
type=NONE
enabled=1
autorefresh=1
gpgcheck=1
keeppackages=0
```

Как видите, параметр `baseurl` указывает не на локальное устройство, а на сервер в Интернете. Также обратите внимание на опцию `autorefresh` (автоматическое обновление) — для сетевого репозитория она установлена в `1`, поскольку пакеты в хранилище могут меняться (например, там появляются новые версии пакетов). А для локального репозитория автоматическое обновление отключено, потому что пакеты в нем будут одни и те же.

Если установить опцию `keeppackages` в `1`, то для этого репозитория менеджер пакетов будет сохранять все загруженные пакеты. Если `keeppackages=0`, то после установки загруженный пакет удаляется.

Основной файл конфигурации менеджера пакетов называется /etc/zypp/zypp.conf, но в нем нет ничего интересного — обычно все опции там закомментированы, поскольку параметры по умолчанию устраивают всех, и их редко приходится менять.

Файлы репозиториев обычно не нужно подключать вручную — вы скачиваете из Интернета YMP-файл, в котором описаны все необходимые репозитории и пакеты, которые нужно установить (хотя могут быть прописаны только репозитории — без пакетов). Данный файл представлен в формате XML (eXtended Markup Language). В секции `<repository>` описывается один репозиторий. Если репозиториев несколько, то и секций `<repository>` будет несколько. В листинге 18.6 представлена секция `<repository>` YMP-файла для главного сетевого репозитория — Main Repository (OSS).

Листинг 18.6. Секция <repository> YMP-файла для главного сетевого репозитория

```
<repository recommended="true">
       <name>Main Repository (OSS)</name>
       <summary>Main OSS Repository</summary>
       <description>The largest and main repository from openSUSE for open
source software</description>
       <url>http://download.opensuse.org/repositories/openSUSE:11.2/standard/<
/url>
      </repository>
```

Каждый пакет, который нужно установить, прописывается в отдельной секции YMP-файла: `<item>` (листинг 18.7).

Листинг 18.7. Секция <item> YMP-файла для установки пакета w32codec-all

```
<item>
        <name>w32codec-all</name>
        <summary>Win 32 Codecs</summary>
```

```
        <description>This packages contains the media player windows codec
dlls for several multimedia formats.</description>
      </item>
```

Понятно, что если нужно установить несколько пакетов, то и секций `<item>` будет несколько.

Пояснение

В листингах 18.6 и 18.7 приведены фрагменты файла codecs-gnome.ymp, благодаря которому в openSUSE устанавливается поддержка мультимедиаформатов.

Примечание

Приведенная здесь информация нужна лишь для общего развития — вам никогда не придется изменять YMP-файлы (хотя кто знает, что нас ждет в этой жизни?), а установка таких файлов производится автоматически, практически без вмешательства пользователя.

Теперь перейдем непосредственно к использованию менеджера пакета zypper. Формат вызова zypper следующий:

```
zypper <команда> [пакеты]
```

Основные команды zypper приведены в табл. 18.5.

Таблица 18.5. *Основные команды zypper*

Команда	Описание
`sl`	Выводит список используемых репозиториев
`sa URL имя`	Добавляет репозиторий (URL — адрес репозитория, а имя — имя, под которым он будет отображаться). Пример: `zypper sa http://ftp.uni-kl.de/pub/linux/suse/update/10.3 SUSE-Linux-10.3-Updates`
`sd URL|имя`	Удаляет репозиторий. При удалении вы можете указать URL или имя репозитория
`install пакеты`	Устанавливает пакеты. Пример: `zypper install mc` Если нужно установить несколько пакетов, то имена пакетов разделяются пробелами
`search маска`	Ищет пакеты по маске. Маска — это часть имени (или полное имя) пакета. Пример: `zypper search mc*`
`list-updates`	Отображает доступные обновления
`update пакет`	Обновляет пакет. Если пакет не задан, обновляет всю систему
`info пакет`	Выводит информацию о пакете
`remove пакет`	Удаляет пакет

18.11.2. Графический менеджер пакетов openSUSE

Устанавливать RPM-пакеты в openSUSE можно с помощью трех программ: zypper, ее графической оболочкой и программой rpm. Программой zypper (*см. разд. 18.11.1*) пользоваться неудобно — она работает в командной строке. Программу rpm (*см. разд. 18.4*) удобно использовать, если есть уже скачанный собственными силами RPM-пакет и его нужно установить — то есть для локальной установки RPM-пакета. Для установки пакетов из любого репозитория, будь то DVD или сервер Интернета, намного удобнее использовать графическую оболочку программы zypper — ввел название пакета, отметил его для установки и установил.

Для запуска графического менеджера пакетов выполните команду **Компьютер | Установка программ**. В окне менеджера пакетов (рис. 18.13) четыре кнопки:

- **Доступно** — показывает пакеты, доступные для установки (эти пакеты еще не установлены);
- **Обновления** — показывает пакеты, для которых имеются обновления (эти пакеты уже установлены в вашей системе, но для них имеются обновления);
- **Установлено** — показывает список установленных пакетов;
- **Все** — показывает список всех пакетов.

Вы можете просмотреть списки пакетов и выбрать необходимые вам пакеты (под названием пакета выводится краткое описание программы, что помогает понять, какую программу устанавливает пакет) или же ввести в поле **Поиск** название пакета (хотя бы примерное) — менеджер отобразит пакеты, соответствующие введенной строке.

Предположим, мы хотим установить пакет clamav (антивирусная программа ClamAV). Введите `clamav` в поле **Поиск**, и вы увидите список доступных пакетов этой тематики (рис. 18.14).

Выберите интересующие вас пакеты (в нашем случае — это clamav) и нажмите кнопку **Установить**. Выбранные вами пакеты появятся в области **Изменения** (рис. 18.15). Кроме имени пакета в области **Изменения** будет показано и действие, которое нужно совершить над пакетом.

На самом деле пакеты еще не установлены. Для их установки нужно нажать кнопку **Применить**. Все — вам остается подождать, пока будут установлены выбранные вами пакеты (рис. 18.16).

Для удаления пакета нажмите кнопку **Установлено**, найдите пакет, который нужно удалить, выделите его и нажмите кнопку **Удалить** (рис. 18.17). Как обычно, менеджер пакетов сначала поставит этот пакет в очередь на удаление (об этом появится информация в области **Изменения**), а для окончательного удаления пакета нужно нажать кнопку **Применить**.

У менеджера пакетов openSUSE есть одна фирменная особенность — кнопка, позволяющая просмотреть, сколько свободного места осталось (рис. 18.18).

На этом обзор систем управления пакетов можно считать завершенным. Надеюсь, я ничего не забыл!

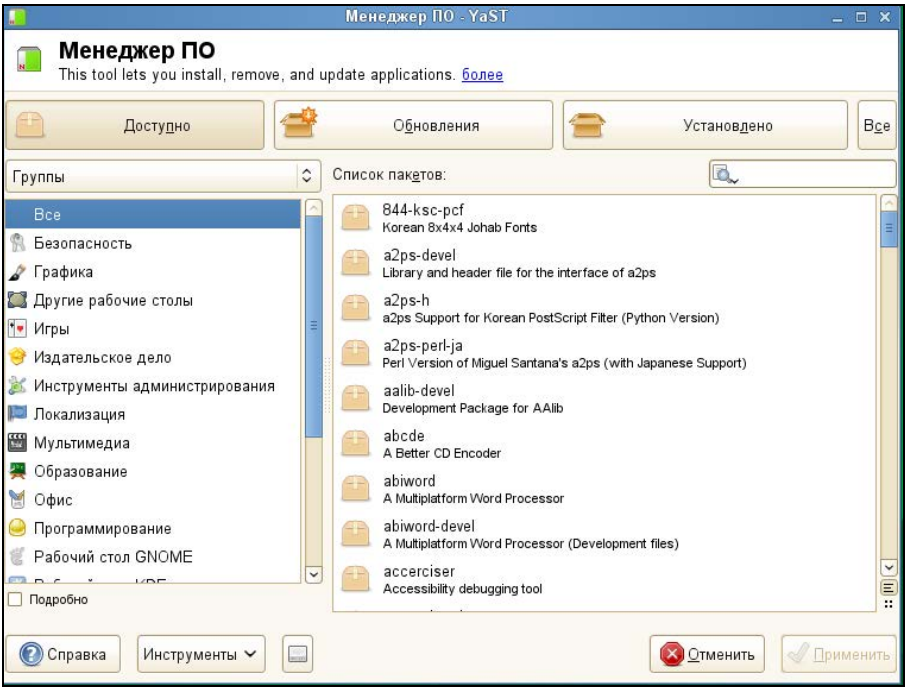


Рис. 18.13. Менеджер пакетов openSUSE

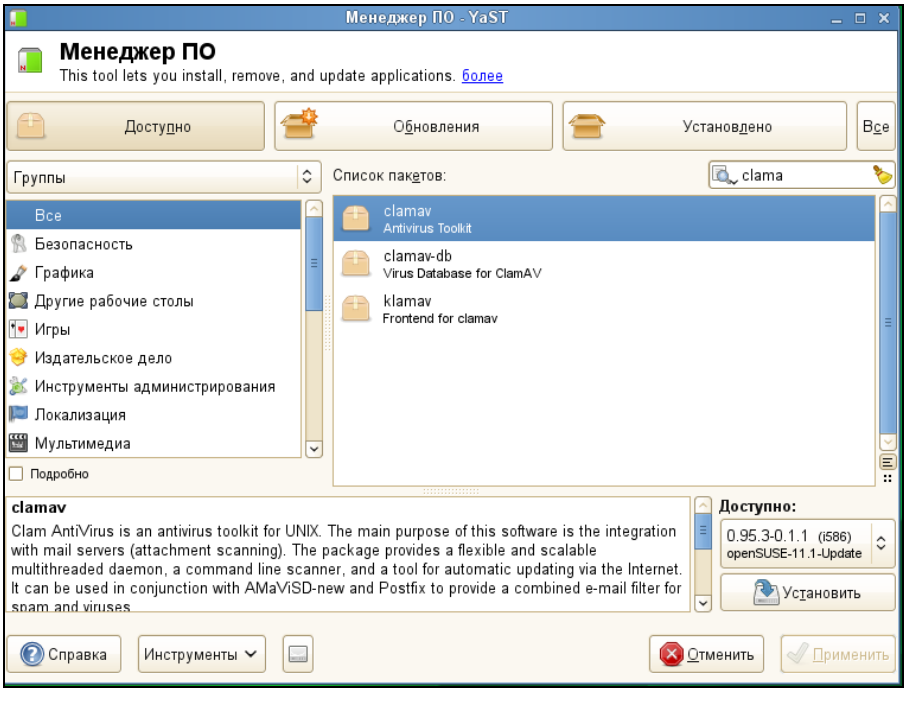


Рис. 18.14. Список доступных пакетов

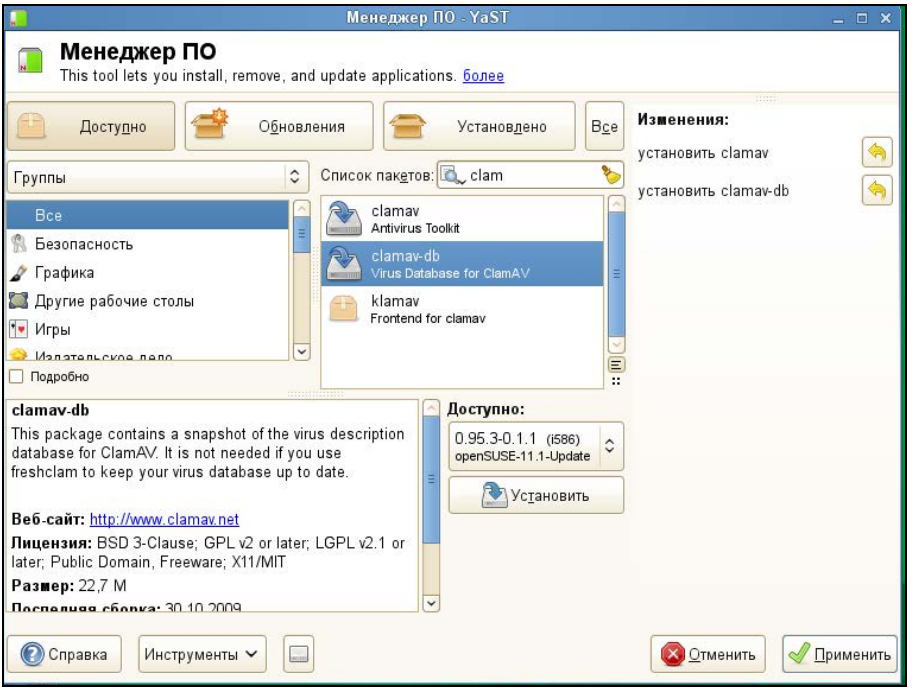


Рис. 18.15. Область **Изменения**

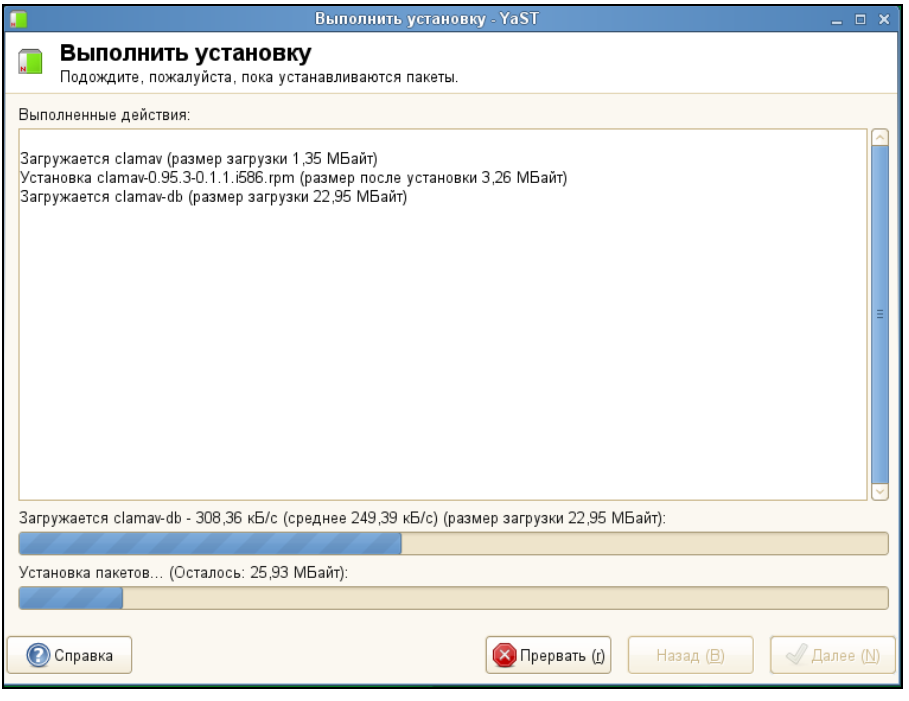


Рис. 18.16. Установка пакетов

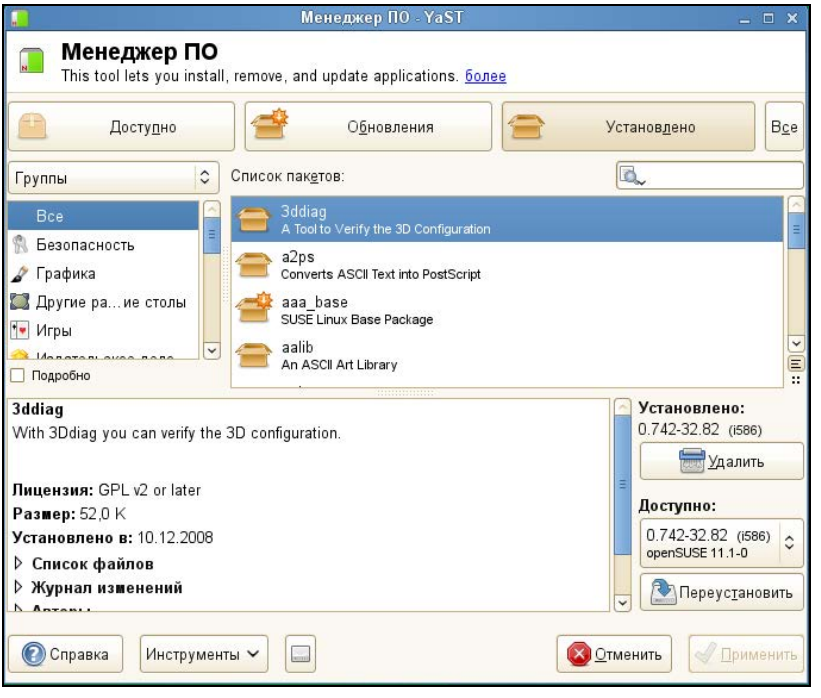


Рис. 18.17. Удаление пакета

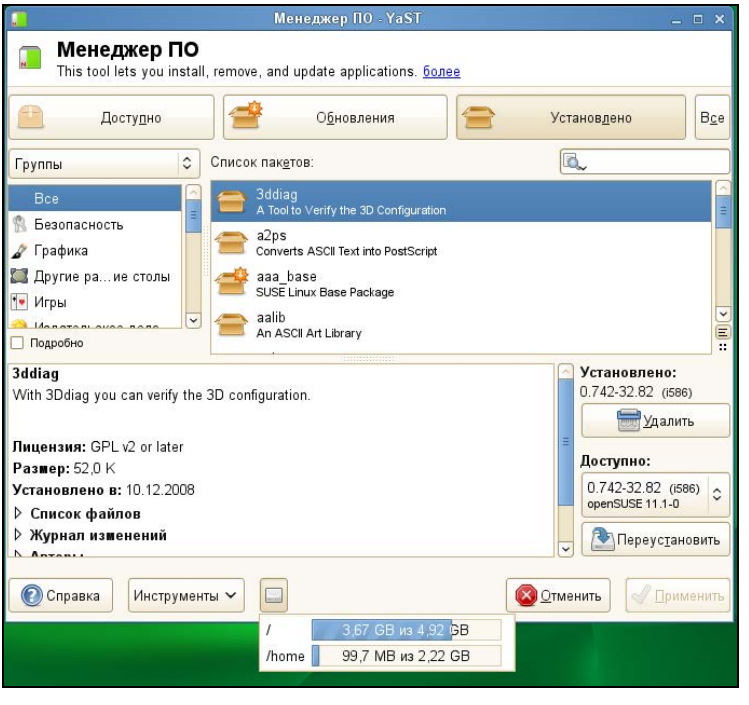
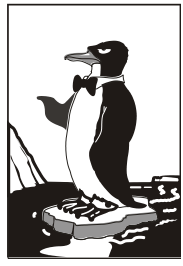


Рис. 18.18. Использование диска



Глава 19

Процессы

19.1. Аварийное завершение процесса

Каждому процессу в Linux присваивается уникальный номер — идентификатор процесса (PID, Process ID). Зная ID процесса, вы можете управлять процессом, а именно — завершить процесс или изменить приоритет процесса. Принудительное завершение процесса необходимо, если процесс завис, и его нельзя завершить обычным образом. А изменение приоритета может понадобиться, если вы хотите, чтобы процесс доделал свою работу быстрее.

Предположим, у вас зависла какая-то программа, например, пусть это будет файловый менеджер mc. Хоть это и маловероятно (не помню, чтобы он когда-нибудь зависал), но для примера пусть будет так. Принудительно завершить ("убить") процесс можно с помощью команды `kill`. Формат ее вызова следующий:

```
kill [параметры] PID
```

PID (Process ID) — это идентификатор процесса, который присваивается процессу системой и уникален для каждого процесса. Но мы знаем только имя процесса (имя команды), но не знаем идентификатор процесса. Узнать идентификатор процесса позволяет программа `ps`. Предположим, что mc находится на первой консоли. Поскольку он завис, вы не можете более использовать эту консоль, и вам нужно переключиться на вторую консоль (клавиатурной комбинацией <Alt>+<F2>). Зарегистрировавшись на второй консоли, введите команду `ps`. Она выведет список процессов, запущенных на второй консоли, — это будет bash и сам ps (рис. 19.1).

Чтобы добраться до нужного нам процесса (mc), который запущен на первой консоли, введите команду `ps -a` или `ps -U root`. В первом случае вы получите список процессов, запущенных вами, а во втором — список процессов, запущенных от вашего имени (я предполагаю, что вы работаете под именем root).

Обратите внимание — вы сами запустили процессы mc и ps (рис. 19.2), а от вашего имени (root) система запустила множество процессов. Следует заметить, что программа ps выводит также имя терминала (tty1), на котором запущен процесс. Это очень важно — если на разных консолях у вас запущены одинаковые процессы, можно легко ошибиться и завершить не тот процесс.

Теперь, когда мы знаем PID нашего процесса, мы можем его "убить":

```
# kill 2484
```

Перейдите на первую консоль после выполнения этой команды — mc на ней уже не будет. Если выполнить команду `ps -a`, то в списке процессов mc тоже не окажется.

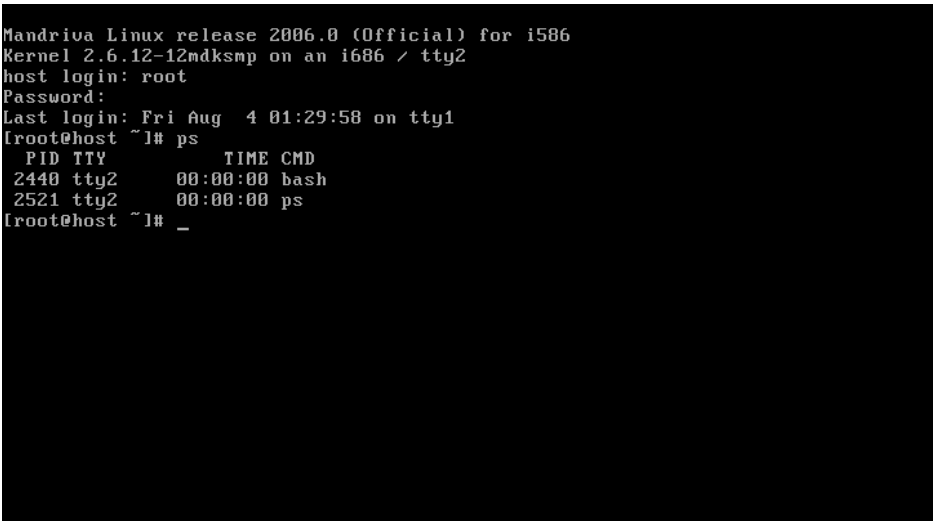


Рис. 19.1. Список процессов на текущей консоли

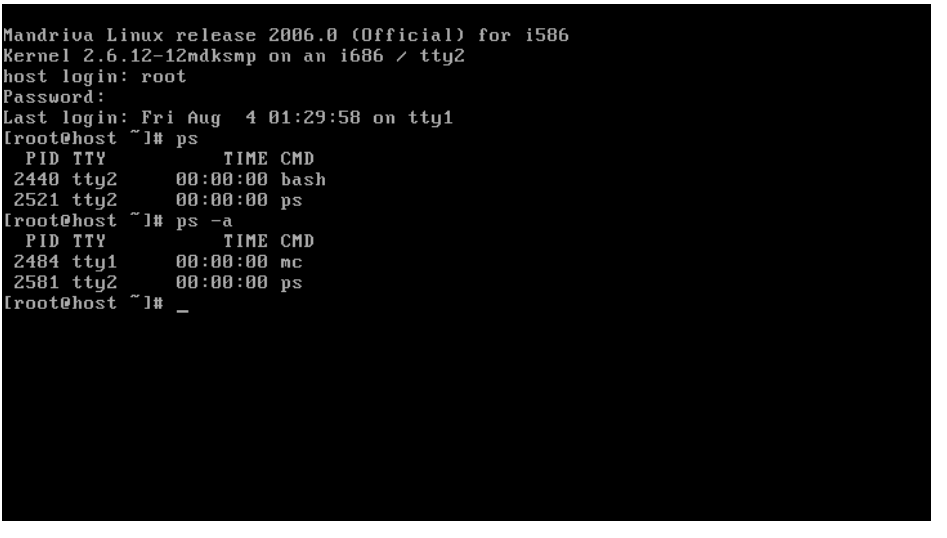


Рис. 19.2. Определение PID программы mc

Проще всего вычислить PID процесса с помощью следующий команды:

```
# ps -ax | grep <имя>
```

Например, `# ps -ax | grep firefox`.

Вообще-то все эти действия, связанные с вычислением PID процесса, мы рассмотрели только для того, чтобы познакомиться с командой `ps`.

Так что, если вы знаете только имя процесса, гораздо удобнее использовать команду:

```
# killall <имя процесса>
```

Но имейте в виду, что данная команда завершит все экземпляры данного процесса. А вполне может быть, что у нас на одной консоли находится mc, который нужно "убить", а на другой — нормально работающий mc. Команда `killall` "убьет" оба процесса.

При выполнении команд `kill` и `killall` нужно помнить, что если вы работаете от имени обычного пользователя, они могут завершить только те процессы, которые принадлежат вам. А если вы работаете от имени пользователя root, то можете завершить любой процесс в системе.

19.2. Программа *top* — кто больше всех расходует процессорное время?

Иногда бывает, что система ужасно тормозит — весь день работала нормально, а вдруг начала притормаживать.

```
top - 01:39:31 up 10 min,  3 users,  load average: 0.00, 0.00, 0.00
Tasks:  58 total,   1 running,  57 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.0% us,  0.3% sy,  0.0% ni, 99.7% id,  0.0% wa,  0.0% hi,  0.0% si
Mem:    189720k total,    68224k used,   121496k free,     5088k buffers
Swap:   128984k total,        0k used,   128984k free,    38072k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 2599 root      16   0  1996 1012  804 R  0.3  0.5   0:00.06 top
    1 root      16   0  1564  540  472 S  0.0  0.3   0:00.55 init
    2 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
    3 root      34  19     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/0
    4 root      10  -5     0    0    0 S  0.0  0.0   0:00.02 events/0
    5 root      16  -5     0    0    0 S  0.0  0.0   0:00.08 khelper
    6 root      11  -5     0    0    0 S  0.0  0.0   0:00.00 kthread
    8 root      20  -5     0    0    0 S  0.0  0.0   0:00.00 kacpid
   61 root      10  -5     0    0    0 S  0.0  0.0   0:00.03 kblockd/0
   93 root      20   0     0    0    0 S  0.0  0.0   0:00.00 pdflush
   94 root      15   0     0    0    0 S  0.0  0.0   0:00.05 pdflush
   96 root      16  -5     0    0    0 S  0.0  0.0   0:00.00 aio/0
   95 root      25   0     0    0    0 S  0.0  0.0   0:00.00 kswapd0
  684 root      16   0     0    0    0 S  0.0  0.0   0:00.00 kseriod
  766 root      13  -5     0    0    0 S  0.0  0.0   0:00.00 ata/0
  775 root      18   0     0    0    0 S  0.0  0.0   0:00.00 scsi_eh_0
  784 root      16   0     0    0    0 S  0.0  0.0   0:00.02 kjournald
  924 root      15  -4  1564  496  420 S  0.0  0.3   0:00.08 udevd
```

Рис. 19.3. Программа top

Если вы даже не догадываетесь, из-за чего это случилось, вам нужно использовать программу top (рис. 19.3) — она выводит список процессов с сортировкой по процессорному времени. То есть на вершине списка будет процесс, который занимает больше процессорного времени, чем сама система. Вероятно, из-за него и происходит эффект "торможения".

На рис. 19.3 показано, что больше всего процессорного времени (0,3%) занимает программа top. Конечно, в реальных условиях все будет иначе. Выйти из программы top можно, нажав клавишу <Q>. Кроме команды <Q> действуют следующие клавиши:

- <U> — показывает только пользовательские процессы (то есть те процессы, которые запустил пользователь, под именем которого вы работаете в системе);

- <D> — изменяет интервал обновления;
- <F> — изменяет столбец, по которому сортируются задачи. По умолчанию задачи сортируются по столбцу %CPU, то есть по процессорному времени, занимаемому процессом;
- <H> — получить справку по остальным командам программы top.

Назначение столбцов программы top указано в табл. 19.1.

Таблица 19.1. *Назначение столбцов программы top*

Столбец	Описание
PID	Идентификатор процесса
USER	Имя пользователя, запустившего процесс
PR	Приоритет процесса
NI	Показатель nice (*см. разд. 19.3*)
VIRT	Виртуальная память, использованная процессом (в Кбайт)
RES	Размер процесса, не перемещенный в область подкачки (в Кбайт). Этот размер равен размерам сегментов кода и данных, то есть RES = CODE + DATA
S	Состояние процесса: • **R** — выполняется; • **S** — "спит" (режим ожидания), в этом состоянии процесс выгружен из оперативной памяти в область подкачки; • **D** — "непрерываемый сон" (uninterruptible sleep), из такого состояния процесс может вывести только прямым сигналом от оборудования;
	• **T** — процесс в состоянии трассировки или остановлен; • **Z** (зомби) — специальное состояние процесса, когда сам процесс **уже** завершен, но его структура еще осталась в памяти
%CPU	Занимаемое процессом процессорное время
%MEM	Использование памяти процессом
TIME+	Процессорное время, израсходованное с момента запуска процесса
COMMAND	Команда, которая использовалась для запуска процесса (обычно имя исполнимого файла процесса)

Самые верхние пять строк вывода top тоже очень интересны. Они содержат полезную информацию о работе всей системы, а не отдельно взятого процесса. Так, в первой строке выводится время, которое работает система с момента загрузки (**up**), количество пользователей (**users**), общая нагрузка системы (**load average**) — за одну минуту, пять минут и пятнадцать минут соответственно (три числа в выводе load avevrage).

Во второй строке выводится информация и процессах: общее количество процессов (**total**), количество запущенных (**running**), "спящих" (**sleeping**), остановленных (**stopped**) процессов и процессов-зомби (**zombie**). Здесь "зомби" — это уже "мертвый" (завершенный процесс), информация о котором еще не удалена из таблицы процессов.

Примечание

Если вы обладаете минимальными навыками программирования на C, вам будет интересна следующая статья: **http://www.dkws.org.ua/index.php?page=show&file=a/dev/process2**. В ней я показываю, как можно создать зомби. Заодно поймете, как так получается, что система не успевает удалять информацию о процессе из служебной таблицы.

Третья строка выводит информацию о распределении процессорного времени. Первые два числа в ней отражают работу центрального процессора по обработке процессов. Если первые два числа у вас стабильно высокие (99–100%), это означает, что какой-то процесс (или группа процессов) очень сильно нагружают CPU или же CPU очень слабый. Остальные параметры в строке CPU не очень важны, хотя на параметр **wa** тоже следует обратить внимание — он сообщает нам о простое во время ввода/вывода. Если он показывает постоянно больше 80%, это указывает на то, что процессор проводит много времени в ожидании ввода/вывода. Косвенно это свидетельствует о проблемах с жестким диском — возможно, он скоро выйдет из строя. Если же с жестким диском все нормально, а процессор далеко не слабый, проблему нужно искать на программном уровне. Скорее всего, причина кроется в некоторых процессах. Для начала нужно определить эти процессы, а затем разбираться, что с ними делать.

Определить процессы, потребляющие много оперативной памяти и процессорного времени, поможет команда `ps axfu`. Команда помимо всего выводит состояние процесса:

- R — запущен;
- D — ожидание (например, ввода или вывода);
- S — процесс спит.

Две последние верхние строки вывода програмы top показывают информацию об использовании оперативной памяти (**memory**) и подкачки (**swap**): всего (**total**), использовано (**used**), свободно (**free**), буферизировано (**buffres**), прокэшировано (**cached**).

19.3. Изменение приоритета процесса

Предположим, что вы работаете с видео, и вам нужно перекодировать файл из одного видеоформата в другой. Конвертирование видео занимает много процессорного времени, а хотелось бы все сделать как можно быстрее и уйти домой пораньше. Тогда вам поможет программа nice — она позволяет запустить любую программу с указанным приоритетом. Ясно — чем выше приоритет, тем быстрее будет выполняться программа. Формат вызова команды следующий:

```
nice -n <приоритет> команда аргументы
```

Максимальный приоритет задается числом `-20`, а минимальный — числом `19`. Приоритет по умолчанию равен 10.

Если процесс уже запущен, тогда для изменения его приоритета можно использовать команду `renice`:

```
renice -n <приоритет> -p PID
```

19.4. Перенаправление ввода/вывода

С помощью перенаправления ввода/вывода мы можем перенаправить вывод одной программы в файл или на стандартный ввод другой программы. Например, у вас не получается настроить сеть, и вы хотите перенаправить вывод команды `ifconfig` в файл, а затем разместить этот файл на форуме, где вам помогут разобраться с этой проблемой. А можно перенаправить список всех процессов (командой `ps -ax`) команде `grep`, которая найдет в списке интересующий вас процесс.

Рассмотрим следующую команду:

```
echo "some text" > file.txt
```

Символ > означает, что вывод команды, находящейся слева от этого символа, будет записан в файл, находящийся справа от символа, при этом файл будет перезаписан.

Чуть ранее мы говорили о перенаправлении вывода программы `ifconfig` в файл. Команда будет выглядеть так:

```
ifconfig > ifconfig.txt
```

Если вместо > указано >>, то исходный файл не будет перезаписан, а вывод команды добавится в конец файла:

```
echo "some text" > file.txt
echo "more text" >> file.txt
cat file.txt
some text
more text
```

Кроме символов > и >> для перенаправления ввода/вывода часто употребляется вертикальная черта `|`. Предположим, что мы хотим вывести содержимое файла big_text:

```
cat big_text
```

Но в файле big_text много строк, поэтому мы ничего не успеем прочитать. Следовательно, целесообразно отправить вывод команды `cat` какой-то программе, которая будет выводить файл постранично, например,

```
cat big_text | more
```

Конечно, этот пример не очень убедительный, потому что для постраничного вывода гораздо удобнее команда `less`:

```
less big_text
```

Вот еще один интересный пример. Допустим, мы хотим удалить файл file.txt без запроса — для этого можно указать команду:

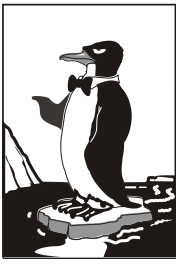
```
echo y | rm file.txt
```

Команда `rm` запросит подтверждение удаления (нужно нажать клавишу <Y>), но за нас это сделает команда `echo`.

И еще один пример. Пусть имеется большой файл, и нам нужно найти в нем все строки, содержащие подстроку 555-555. Чтобы не делать это вручную, можно воспользоваться командой:

```
cat file.txt | grep "555-555"
```

В следующей главе мы поговорим о журналировании событий системы, рассмотрим демоны syslog и syslog-ng, а также основные журналы (протоколы) системы.



Глава 20

Протоколирование системы. Журналы

20.1. Демоны протоколирования системы

В любой UNIX-системе, коей является и Linux, есть *демоны протоколирования* (далее просто "демоны"). Демоны записывают в протоколы (журналы) сообщения, генерируемые ядром, сервисами, пользовательскими программами. В большинстве случаев файлы протоколов размещаются в каталоге /var/log.

Основным демоном протоколирования является syslogd. Он имеется практически на всех UNIX-системах — от самых старых до самых новых. Правда, в современных дистрибутивах применяются модифицированные версии syslogd: rsyslogd или syslog-ng. Первый из них получил большее распространение, поэтому мы его и рассмотрим. Секрет популярности rsyslogd в файле конфигурации, синтаксис которого идентичен синтаксису файла настроек демона syslogd. Это очень удобно. Во-первых, не нужно изучать новый синтаксис, во-вторых, подобие формата файла упрощает миграцию на rsyslogd — достаточно просто переименовать файл конфигурации и запустить новый демон протоколирования.

Некоторые пользователи отключают сервис syslogd (или rsyslogd). Настоятельно рекомендуется не делать этого. Ведь у Linux довольно развита функция самодиагностики, и в случае возникновения сбоя по содержимому журналов вы можете понять, в чем причина сбоя, и устранить ее. Во всяком случае с записями в журнале это будет проще сделать, чем без них.

Основной файл конфигурации демона syslogd называется /etc/syslog.conf, а файл конфигурации демона rsyslogd — rsyslog.conf. Формат этих двух файлов следующий:

```
селектор[;селектор]        действие
```

Параметр `селектор` определяет, какие сообщения должны быть запротоколированы. Вот список наиболее часто использующихся селекторов:

- `auth`, `security` — все, что связано с регистрацией пользователя в системе;
- `authpriv` — отслеживает программы, изменяющие привилегии пользователей, например, программа su;
- `cron` — сообщения планировщиков заданий;
- `kern` — сообщения ядра;
- `mail` — сообщения почтовых программ;
- `news` — сообщения новостного демона;

- `uucp` — сообщения службы Unix-to-Unix-CoPy, уже давно не используется, но файл конфигурации демона все еще содержит упоминание о ней;
- `syslog` — сообщения самого демона syslogd;
- `user` — сообщения пользовательских программ;
- `daemon` — сообщения различных сервисов;
- `*` — все сообщения.

При указании селектора можно определить, какие сообщения нужно протоколировать:

- `debug` — отладочные сообщения;
- `info` — информационные сообщения;
- `err` — ошибки;
- `warning` — предупреждения (некритические ошибки);
- `crit` — критические ошибки;
- `alert` — "тревожные" сообщения, требующие вмешательства администратора;
- `emerg` — очень важные сообщения (произошло что-то такое, что мешает нормальной работе системы);
- `notice` — замечания.

Впрочем, обычно селекторы указываются так:

```
название_селектора.*
```

Это означает, что будут протоколироваться все сообщения селектора. Вот еще несколько примеров:

- `daemon.*` — протоколируются все сообщения сервисов;
- `daemon.err` — регистрировать только сообщения об ошибках сервисов.

Теперь перейдем к параметру `действие` — это второе поле файла конфигурации. В большинстве случаев `действие` — это имя файла журнала, в который нужно записать сообщение селектора. Если перед именем файла стоит "минус" (–), то после каждой записи в журнал демон не будет выполнять синхронизацию файла, то есть осуществлять системный вызов `fsync()`. Это повышает производительность системы, поскольку сообщений обычно много, и если после каждого выполнять синхронизацию журнала, система будет работать медленно.

Пример конфигурационного файла syslog.conf (rsyslog.conf) приведен в листинге 20.1.

Листинг 20.1. Пример файла конфигурации /etc/rsyslog.conf (дистрибутив Fedora)

```
# Сообщения ядра протоколируются на консоль
#kern.*                                                 /dev/console

# Протоколировать все сообщения (кроме почты) в /var/log/messages
*.info;mail.none;authpriv.none;cron.none                /var/log/messages

# Сообщения селектора authpriv записываются в файл /var/log/secure
```

```
authpriv.*                                                  /var/log/secure

# Сообщения почты (их будет много, если запущен агент MTA вроде postfix)
# записываются в файл maillog
mail.*                                                      -/var/log/maillog

# Сообщения планировщиков заданий записываются в cron
cron.*                                                      /var/log/cron

# Особо критичные сообщения выводятся на экран всех работающих в данный момент
# пользователей (вместо имени файла указана звездочка)
*.emerg                                                     *

# Сообщения UUCP и сообщения сервера новостей записываются в /var/log/spooler
uucp,news.crit                                              /var/log/spooler

# Загрузочные сообщения записываются в boot.log
local7.*                                                    /var/log/boot.log
```

20.2. Изучаем файлы журналов

Исследовав файл конфигурации rsyslog.conf, можно понять, для чего используется тот или иной журнал. Но некоторые сервисы, например, Apache, ведут свои журналы, минуя демон протоколирования — именно этим объясняется, что в каталоге /var/log есть дополнительные файлы и каталоги, не упомянутые в rsyslog.conf. Все эти журналы тоже хранятся в каталоге /var/log. Вот примеры некоторых файлов (каталогов) журналов:

- /httpd/ — журналы Web-сервера Apache;
- /cups/ — журналы системы CUPS (в вашей системе может быть установлена одна из этих систем печати);
- auth.log — журнал аутентификации: кто и когда входил в систему;
- boot.log — журнал загрузки системы;
- dmesg — загрузочные сообщения ядра (до запуска системы инициализации);
- explanations — в этот журнал некоторые программы записывают свои действия, объясняя вам, какие именно изменения они произвели в вашей системе. Данный файл есть только в дистрибутиве Mandriva;
- syslog — журнал демона syslog;
- XFree86.0.log — журнал системы XFree86.

В каком же журнале искать ошибку? Тут нужно исходить из принципа взаимоисключения: если у вас не работает Web-сервер Apache, то искать причину нужно в каталоге /var/log/httpd/, но никак не в файле /var/log/mail.

Если ошибка происходит во время загрузки системы, просмотрите файл boot.log:

```
# less boot.log
```

Эту команду нужно вводить от имени root, поскольку журналы системы просматривать может только он.

Для просмотра журналов удобно использовать команду `tac`, выводящую файл в обратном порядке: то есть сначала выводятся последние строки, а потом — первые. Например, если нужно вывести последние 15 строк, можно использовать команду:

```
# tac –n 15 <файл>
```

Сообщения различных программ пользовательского уровня, то есть обычных программ, возможно, запущенных с привилегиями root, протоколируются в файле /var/log/user.log. В некоторых системах этого файла нет — например, он есть в Linux Mandriva, но его нет в Fedora (и в ASPLinux). Если данный файл у вас отсутствует, значит, сообщения пользовательских программ протоколируются в другой файл, обычно в /var/log/messages. Чтобы узнать точно имя файла, просмотрите файл rsyslog.conf. Рассмотрим фрагмент данного файла (листинг 20.2).

Листинг 20.2. Фрагмент файла /var/log/user.log (/var/log/messages)

```
Jun 20 14:18:28 localhost rpmdrake[2573]: [RPM] libcroco1-0.4.0-1mdk installed
Jun 20 14:18:31 localhost rpmdrake[2573]: [RPM] librsvg2_2-2.4.0-1mdk installed
Jun 20 14:18:32 localhost rpmdrake[2573]: [RPM] libpanel-applet-2_0-2.4.2-6mdk
installed
Jun 20 14:18:32 localhost rpmdrake[2573]: [RPM] libmetacity-private0-2.6.5-
2mdk installed
```

Здесь программа rpmdrake сообщает, что она установила указанные в листинге пакеты. Если вас интересуют более подробные сведения об этом процессе, откройте файл /var/log/explanations (листинг 20.3).

Листинг 20.3. Фрагмент файла /var/log/explanations

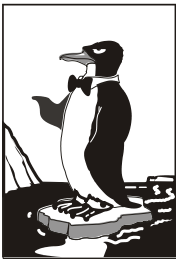
```
Jun 20 14:15:19 localhost rpmdrake[2573]: Extracting header of gnome2-2.4.0-
3mdk.noarch from /var/lib/urpmi/hdlist.Installation CD 3 (x86) (cdrom3).cz
Jun 20 14:15:36 localhost rpmdrake[2573]: Extracting header of gnome-audio-
2.0.0-1mdk.noarch from /var/lib/urpmi/hdlist.Installation CD 2 (x86)
(cdrom2).cz
Jun 20 14:15:41 localhost rpmdrake[2573]: Extracting header of gnome-common-
2.4.0-1mdk.noarch from /var/lib/urpmi/hdlist.Installation CD 4 (x86)
(cdrom4).cz
Jun 20 14:15:57 localhost rpmdrake[2573]: Installing package remova-
ble://mnt/cdrom/Mandrake/RPMS/eel-2.4.2-1mdk.i586.rpm
Jun 20 14:15:57 localhost rpmdrake[2573]: Installing package remova-
ble://mnt/cdrom/Mandrake/RPMS/eog-2.4.1-1mdk.i586.rpm
Jun 20 14:15:57 localhost rpmdrake[2573]: Installing package remova-
ble://mnt/cdrom/Mandrake/RPMS/libapm1-3.1.0-6mdk.i586.rpm
Jun 20 14:15:57 localhost rpmdrake[2573]: Installing package remova-
ble://mnt/cdrom/Mandrake/RPMS/libcroco1-0.4.0-1mdk.i586.
...
```

В этом файле вы найдете полный отчет о действиях программы — видно даже, откуда был установлен тот или иной пакет. Нужно отметить, что файл explanations существует только в Linux Mandriva — в других дистрибутивах вы его не найдете.

Когда вы определите причину сбоя (она будет записана в один из файлов протокола), вы сможете ее устранить.

> ***Совет***
>
> При поиске ошибок вряд ли стоит заглядывать в файл auth.log и secure — разве что в последнюю очередь.

Следующая глава посвящена очень важному действию системного администратора — резервному копированию. Резервное копирование важно выполнять регулярно — так и только так вы сможете уберечь свою систему от потери данных.



Глава 21

Резервное копирование

21.1. Зачем нужно делать резервные копии

К сожалению, даже самые новые компьютеры не совершенны. Они иногда ломаются. Причиной сбоя может быть все, что угодно, например, банальный перепад напряжения, из-за которого выходит из строя жесткий диск. Или же программный сбой — обычный вирус, который уничтожил таблицу разделов жесткого диска. Да, вирусов под Linux очень мало. Но в большинстве случаев на компьютере с Linux установлена еще и система Windows, а таблица разделов, как и винчестер, общая для обеих операционных систем. Поэтому нет никакой гарантии, что вирус, уничтожающий данные на винчестере, оставит данные с Linux-разделов в целости и сохранности.

Делать резервные копии полезно не только на сервере, но и на обычной рабочей станции (или домашнем компьютере). Представьте, что вы нечаянно удалили какой-то важный файл или изменили его не так, как нужно, а после этого выполнили команду **Файл | Сохранить**. В первом случае (удаление) файл еще можно восстановить, но только, если вы обнаружили пропажу файла сразу после удаления. Если прошло некоторое время, скажем, неделя, то удастся восстановить только часть файла или вообще ничего не получится восстановить, поскольку блоки, в которых размещался файл, могли быть физически перезаписаны другими данными. А восстановление части файла имеет смысл только в случае с текстовыми файлами (не двоичными). Однако все современные текстовые процессоры хранят данные не в текстовом формате, а в двоичном. Это связано, в первую очередь, с тем, что в документы часто внедряются двоичные данные — те же рисунки. Поэтому восстановление части файла ничего вам не даст — следовательно, можете считать, что файл потерян навсегда.

В случае же, если у вас есть резервная копия, восстановить файл не составляет большого труда — вы просто скопируете его из резерва. Правда, это хороший выход из положения?

Если же вы изменили файл и сохранили изменения, вам поможет только резервная копия. Ведь оригинальный файл уже не вернуть — как правило, после сохранения изменений в большинстве программ функция отмены последнего действия не работает. Конечно, можно сделать все заново (скажем, перенабрать несколько страниц), но намного быстрее и удобнее восстановить файл из резервной копии.

21.2. Выбор носителя для резервной копии

Раньше для создания резервных копий использовались *стримеры* — устройства, записывающие данные на магнитную ленту. Конечно, не на обычную магнитофонную — в стример устанавливалась специальная кассета с высококачественной магнитной лентой. Преимущество такого решения заключалось в его дешевизне. На кассету можно было записать 2–3 гигабайта информации, что для того времени было рекордным значением, а сами стримеры работали надежно и представляли собой проверенное решение. Но недостатков тоже хватало — процесс создания резервной копии из-за того, что стримеры были устройствами довольно медленными, мог длиться часами. Кассеты с резервными копиями нужно было очень бережно хранить, поскольку они имели свойство, как и обычные магнитофонные кассеты, размагничиваться. Поэтому со временем актуальные резервные копии приходилось перезаписывать — обновлять. Это уже не говоря о "диверсии" — испортить весь архив резервных копий мог один небольшой магнит.

С появлением лазерных компакт-дисков все изменилось. Конечно, сначала приводы CD-RW стоили довольно дорого, поэтому для создания резервных копий использовались стримеры или (в домашних условиях) обычные дискеты. Да, на дискету много не запишешь, но ведь и объемы данных были не такими, как сейчас. Скажем, в 1995 году на несколько дискет в сжатом виде можно было записать практически все документы небольшой фирмы — текстовая информация очень хорошо сжимается.

Спустя несколько лет приводы CD-RW, как и сами диски ("болванки"), существенно подешевели и стали доступны обычным пользователям. На CD можно записать до 700 Мбайт информации в несжатом виде. Если же ее сжать, то можно было записать до 1 Гбайт (все зависит от архиватора и от сжимаемой информации). Домашние пользователи, которые вообще раньше не делали резервных копий, начали активно использовать CD, хотя некоторые организации и тогда использовали стримеры — им так было проще.

С появлением и удешевлением DVD стримеры вымерли как вид. Может, они где-то и используются, но сейчас намного проще сделать резервную копию на DVD. Преимущество такого решения заключается в следующем:

- на обычный DVD можно записать до 4,5 Гбайт информации, на двухслойный и двухсторонний — до 17 Гбайт;
- скорость записи и чтения DVD не сравнится со скоростью чтения/записи стримера;
- DVD намного надежнее кассет стримера.

Совершенно нет смысла делать резервную копию на другом жестком диске (или в другом разделе жесткого диска). В случае выхода жесткого диска из строя вы не сможете прочитать не только свои данные, но и резервную копию. Поэтому резервные копии нужно хранить на съемных носителях. Идеально подходят CD и DVD. Конечно, лучше записывать на DVD — на них больше помещается данных. Можно использовать и другие съемные носители, которые есть под рукой — Flash-диски, магнитооптические носители. Хотя по емкости я пока не знаю ни одного доступного обычному пользователю съемного носителя, который смог бы превзойти DVD.

21.3. Правила хранения носителей с резервными копиями

Ваша резервная копия "проживет долго", если вы будете придерживаться следующих простых правил.

- На носитель с резервной копией не нужно записывать посторонние данные.

  Предположим, вы решили записать на DVD резервную копию своих документов общим объемом 1 Гбайт, и при этом оказалось, что на диске осталось свободными 3,5 Гбайт Существует соблазн использовать свободное место по прямому назначению и дописать диск до конца — записать еще, например, фильм или музыку. Но делать этого не стоит. Рано или поздно вы захотите послушать музыку или посмотреть фильм или, возможно, одолжите диск приятелю, чтобы фильм посмотрел и он. В результате диск может быть утерян или поврежден. Помните, диском с резервной копией нужно пользоваться только тогда, когда эта копия вам необходима.

- Не нужно дописывать на диск вторую резервную копию.

  Опять-таки на диске осталось еще много свободного места, и вы хотите дописать на него следующую резервную копию (спустя некоторое время после записи первой). Не нужно этого делать — чем меньше мы используем диск, тем меньше он изнашивается, следовательно, тем лучше (правило 1). Хотя из этого правила есть исключения, диктуемые здравым смыслом и стратегией копирования. Об этом мы еще поговорим.

- Никогда не доверяйте диски с важными данными посторонним людям — во-первых, это не желательно с точки зрения конфиденциальности данных, во-вторых, важные резервные копии могут быть просто утеряны.

- Хранить диски нужно в темном сухом помещении и обязательно в отдельном боксе — на CD/DVD, как и на магнитооптические носители (не говоря уже о лентах стримеров), негативно влияют солнечные лучи. Поэтому диски с резервными копиями следует убрать подальше от прямого попадания солнечных лучей и взглядов посторонних. В случае с магнитооптическими носителями их полагается держать подальше от источников магнитного излучения, чтобы избежать размагничивания диска. Каждый диск нужно хранить в отдельном боксе — не храните диски в круглых коробках, в которых диски нанизываются на шкив, размещенный по центру коробки.

- Подписывайте ваши носители — указывайте дату и время копирования, а также, по возможности, что находится на диске (это нужно писать на бумажной обложке бокса). Для надписи на поверхности диска можно использовать маркер. Все это облегчит поиск нужной резервной копии.

21.4. Стратегии создания резервной копии

Существует несколько стратегий создания резервных копий. Одна из них предполагает создание копии всего жесткого диска, а потом на отдельные носители записываются только изменившиеся данные. Спустя некоторое время (зависит

от количества новых данных — от недели до месяца) опять делается копия всего жесткого диска. Такая стратегия идеально подходит для сервера предприятия, но не для обычного домашнего пользователя. К сожалению, хороших стратегий мало, поэтому пришлось разрабатывать собственную.

Начнем с первого варианта — копирование всего винчестера. Делать так просто нет смысла. Сейчас никого не удивишь винчестером в 200 Гбайт. Даже при условии, что на DVD помещается до 17 Гбайт информации, для создания полной копии вам понадобится от 1 до 12 чистых дисков. А теперь посчитаем, сколько времени понадобится на запись такой резервной копии? Учитывая даже восьмикратную скорость записи, на данное мероприятие уйдет целый день. Оно того не стоит. Да, на сервере предприятия все эти 200 Гбайт могут быть "забиты" важной информацией, например, базой данных. Но на домашнем компьютере большая часть дискового пространства занята фильмами и музыкой. Даже если произойдет потеря информации, то большую часть фильмов и музыки можно будет заново получить или во внутренней сети, или в ближайшем прокате. Поэтому данный вид информации лучше вообще исключить из резервной копии.

Делать резервную копию всех программных файлов тоже не вижу смысла — все это можно легко восстановить с дистрибутивных дисков. В резервную копию следует поместить только RPM-пакеты, загруженные из Интернета (то есть пакеты, которых нет на других съемных носителях). Тогда, в случае потери данных, вам не придется заново загружать нужные вам пакеты из Интернета.

На носитель резервной копии следует записывать:

- конфигурационные файлы системы — просто каждый раз при создании резервной копии записывайте на диск весь каталог /etc;
- измененные пользовательские данные — если не помните, что вы изменяли (с какими документами работали), можно записать весь каталог /home (не забудьте о каталоге /root). Учитывая объем DVD, места вам хватит;
- RPM-пакеты, загруженные из Интернета, — чтобы потом не пришлось заново их загружать;
- каталог /var/www/html — это корневой каталог Web-сервера, если, конечно, вы его используете. Как правило, Web-программисты тестируют на домашнем компьютере свои сценарии, поэтому копию данного каталога нужно сделать обязательно;
- каталог /var/named — в этом каталоге хранятся настройки кэширующего сервера DNS, если, конечно, вы его используете;
- каталог /var/lib/mysql — содержит базы данных сервера MySQL, если вы его используете;
- файл /usr/src/linux-<версия_ядра>/.config — это конфигурационный файл вашего ядра. Его нужно записывать, если вы используете не стандартную версию ядра, а перекомпилировали ядро после установки системы.

Самое главное — создать первую резервную копию. Как правило, она будет самая большая. Потом нужно делать резервные копии, в среднем — раз в неделю. На диски следует записывать только изменившиеся данные. Если вы знаете, что не изменяли конфигурацию системы, записывать каталог /etc уже не нужно. Если вы только работали с документами, запишите просто свой домашний каталог.

Для данной схемы вам понадобится два диска. Первый диск назовем диском месяца. Он будет содержать полную копию (по приведенным пунктам). На диск недели вы будете записывать каждую неделю (возможно, чаще — все зависит от важности информации) изменившуюся информацию (в лучшем случае — просто каталог /home). На диск недели можно информацию дописывать, тогда в конце месяца у вас на этом диске будет минимум 4 каталога /home (названные по-разному, естественно, например, home-1, home-2, home-3 и home-4 — по номеру недели).

После этого в начале следующего месяца вы на новый диск записываете полную копию — снова создаете диск месяца. Ну, а потом схема повторяется. Здесь все просто, думаю, не запутаетесь.

21.5. Программа tar

Программу tar можно использовать для создания архива резервной копии — с одним архивом работать проще, чем тысячей файлов, которые нужно поместить в резервную копию, да и место на DVD сэкономим.

Мы не будем рассматривать все опции tar — их достаточно много (о них вы можете прочитать в руководстве по команде `man tar`), а рассмотрим только команду, позволяющую заархивировать нужный нам каталог:

```
tar -cvjf имя_архива.tar.bz2 каталог
```

Например,

```
tar -cvjf homes.tar.bz2 /home
```

На рис. 21.1 приведен процесс архивации.

```
[root@localhost etc]# tar -cvjf homes.tar.bz2 /home
tar: Удаляется начальный `/' из имен объектов
/home/
/home/den/
/home/den/.gnome/
/home/den/.gnome/gnome-vfs/
/home/den/.gnome/gnome-vfs/.trash_entry_cache
/home/den/.nautilus/
/home/den/.nautilus/metafiles/
/home/den/.nautilus/metafiles/file:%2F%2F%2Fmedia%2F2007.0-disc1.xml
/home/den/.nautilus/metafiles/x-nautilus-desktop:%2F%2F%2F.xml
/home/den/.nautilus/metafiles/file:%2F%2F%2Fmedia.xml
/home/den/.nautilus/metafiles/file:%2F%2F%2Fmedia%2F2007.0-disc1%2Fi586.xml
/home/den/.nautilus/metafiles/file:%2F%2F%2Fmedia%2F2007.0-disc1%2Fi586%2Fmedia.xml
/home/den/.nautilus/saved-session-S6PXXT
/home/den/.nautilus/saved-session-WBWRXT
/home/den/.nautilus/saved-session-LEGOXT
/home/den/Public/
/home/den/.gconfd/
/home/den/.gconfd/saved_state
/home/den/Music/
/home/den/.gnome2_private/
/home/den/.gstreamer-0.10/
/home/den/.gstreamer-0.10/registry.i686.xml
/home/den/.metacity/
/home/den/.metacity/sessions/
/home/den/.metacity/sessions/1188925986-2855-3760837421.ms
/home/den/.metacity/sessions/1181203896-2407-3925888756.ms
/home/den/.metacity/sessions/1188925847-2586-4103640012.ms
```

Рис. 21.1. Архивирование

Чтобы разархивировать архив, перейдите в каталог, в который вы хотите его распаковать, и введите команду:

```
tar -xvjf имя_архива.tar.bz2
```

Если вам нужно извлечь всего пару файлов, тогда проще использовать файловый менеджер mc (пакет тоже называется mc).

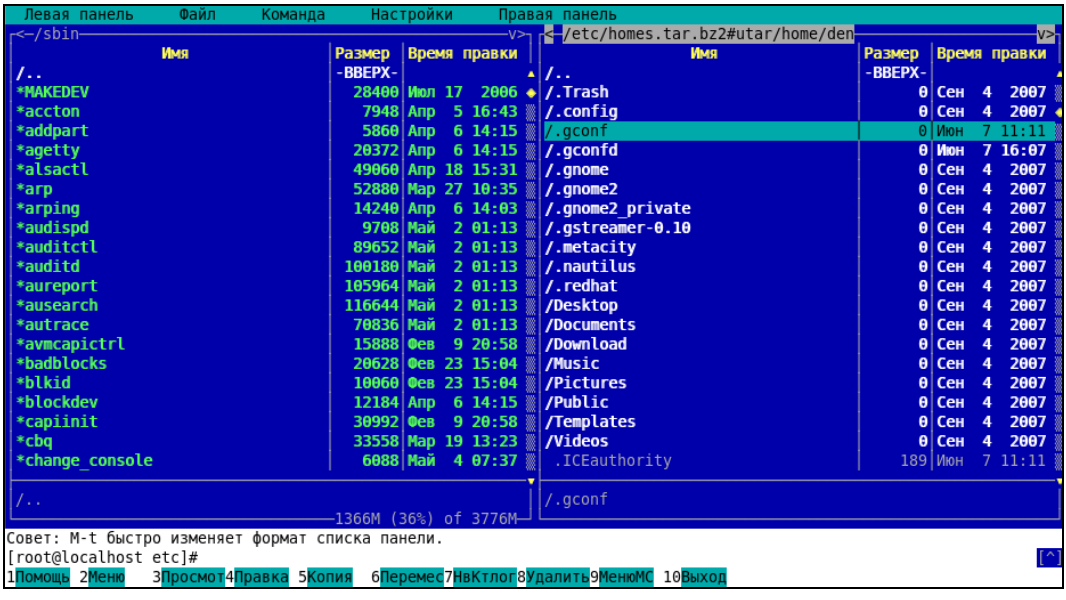


Рис. 21.2. Работа с архивом в mc

21.6. Сетевое резервное копирование

Задача проста: у вас в сети есть компьютеры, например, Web-сервер и почтовый сервер. Вам нужно сделать резервные копии данных, хранимых на этих компьютерах. Понятно, что отлучаться со своего рабочего места очень не хочется. Поэтому вы можете создать резервную копию по сети с помощью команды `scp`:

```
scp -r имя_каталога компьютер:каталог
```

Например,

```
scp -r web-cp web-server:/var/www
```

Команда `scp` (secure copy) используется для безопасного копирования файлов по сети. Для того чтобы она работала, нужно, чтобы на удаленном компьютере был установлен сервер sshd, о котором мы поговорим в *главе 33*.

Вернемся к нашей команде. Параметр `-r` означает, что нужно скопировать подкаталоги удаленного каталога, то есть произвести рекурсивное копирование. После него задается имя локального каталога, куда будут записаны скопированные файлы и каталоги. Параметр `web-sever` — это имя удаленного компьютера (можно задать

IP-адрес), а через двоеточие указан удаленный каталог, который вы хотите скопировать.

Вам осталось лишь заархивировать каталог web-cp:

```
tar -cvjf web-cp.tar.bz2 web-cp
```

21.7. Запись CD/DVD из консоли

21.7.1. Команда *dd* — создание образа диска

Довольно часто бывает нужно создать образ оптического диска (не знаю, как у вас, но у меня такая потребность возникает примерно один раз в неделю). Причина проста — или под рукой нет чистой болванки, или же нужно поработать с диском, который придется отдать, но при этом нет никакого желания записывать его на болванку.

В Windows для создания образа диска обычно используются посторонние программы, например Nero или WinImage. В Linux мы будем пользоваться только средствами операционной системы.

Образ CD/DVD-диска можно создать с помощью команды `dd`:

```
dd if=/dev/cdrom of=~/cd.iso
```

Вместо `/dev/cdrom` надо подставить имя файла устройства вашего привода CD/DVD (обычно этого делать не приходится, поскольку ссылка `/dev/cdrom` устанавливается самой системой на ваш привод CD/DVD).

Указанная команда создаст файл образа cd.iso, который будет записан в ваш домашний каталог. Аналогично с помощью этой команды можно создать и образ дискеты — только вместо `/dev/cdrom` нужно указать имя файла устройства `/dev/fd0`.

Что можно сделать с ISO-образом в Windows? Его можно записать на чистую болванку или же открыть в специальной программе (например, ISOpen или UltraISO) для изменения. В Linux открыть образ можно с помощью средств самой операционной системы. Для этого его надо просто подмонтировать к корневой файловой системе с помощью команды следующего формата:

```
# mount -o loop -t iso9660 образ точка_монтирования
```

- опция `-o loop` означает, что будет монтироваться не файл устройства, а образ диска, который записан на жесткий диск;
- параметр `-t 9660` задает тип файловой системы образа: iso9660 — стандартная файловая система для CD/DVD;
- после файловой системы указывается файл образа, например `~/cd.iso`;
- последний параметр — это точка монтирования, каталог, к которому будет подмонтирован образ (напомню, что каталог должен существовать).

> ***Примечание***
>
> В большинстве случаев команду `mount` нужно выполнять от имени пользователя root или с помощью команд `sudo` или `su`.

В нашем случае для монтирования образа ~/cd.iso к каталогу /mnt/image нужно выполнить команду:

```
# mount -o loop -t iso9660 ~/cd.iso /mnt/image
```

После этого можно обращаться к образу, как к обычному каталогу:

```
ls /mnt/image
```

21.7.2. Команды *cdrecord* и *dvdrecord* — запись образа на болванку

Предположим, у вас есть файл образа cd.iso, и нужно записать его на компакт-диск, но вы не хотите (или не имеете возможности) использовать графические программы вроде Nero или k3b. В этом случае вам нужно использовать программу cdrecord (пакет называется аналогично). Команда для записи образа на болванку CD-R очень проста и выглядит так:

```
# cdrecord dev=0,0,0 -dao speed=16 файл_образа
```

Для записи DVD-R используется аналогичная команда:

```
# dvdrecord dev=0,0,0 -dao speed=4 файл_образа
```

В этой команде вам нужно изменить параметр `dev` — идентификатор устройства CD/DVD. Если в вашей системе установлен только один привод CD/DVD, и он же является пишущим, тогда, скорее всего, у него будет идентификатор 0,0,0. Но если у вас несколько приводов CD/DVD (например, обычный и пишущий), вы должны ввести следующую команду:

```
# cdrecord -scanbus
```

Команда выведет список CD/DVD, установленных в вашей системе (рис. 21.3). Вам нужно запомнить идентификатор нужного привода и использовать его при записи образа диска.

```
[den@localhost ~]$ cdrecord -scanbus
scsibus1000:
        1000,0,0 100000) *
        1000,1,0 100001) 'HL-DT-ST' 'DVDRAM GSA-4167B' 'DL11' Removable CD-ROM
        1000,2,0 100002) *
        1000,3,0 100003) *
        1000,4,0 100004) *
        1000,5,0 100005) *
        1000,6,0 100006) *
        1000,7,0 100007) *
[den@localhost ~]$
```

Рис. 21.3. Идентификаторы приводов CD/DVD

21.7.3. Команды очистки перезаписываемых дисков

Для очистки DVD-RW-диска используется команда:

```
# dvd+rw-format -f имя_устройства_DVD-RW
```

Для быстрой очистки CD-RW введите команду:

```
# cdrecord -v blank=fast dev=0,0,0
```

Если нужно произвести полную, а не быструю очистку, замените `blank=fast` на `blank=all`.

21.7.4. Команда *mkisofs* — создание ISO-образа

Иногда нужно создать образ CD/DVD не с оригинального диска, а с каталогов файловой системы. Другими словами — у вас есть файлы и каталоги, которые вам нужно записать на CD/DVD. Технология CD/DVD не позволяет записывать файлы и каталоги непосредственно на носитель — вам нужно создать каталог, поместить в него все файлы и каталоги, которые вы хотите записать на оптический диск, затем создать по этому каталогу ISO-образ, а потом записать его на болванку.

Скопируйте все необходимые вам файлы в каталог ~/cd. Затем выполните команду:

```
mkisofs -r -jcharset koi8-r -o ~/cd.iso ~/cd
```

Эта команда создаст по каталогу ~/cd файл образа cd.iso и поместит его в ваш домашний каталог. Обратите внимание на кодировку локализованной версии — сейчас используется KOI8-R. Если у вас другая кодировка, например UTF-8, вы должны указать ее:

```
mkisofs -r -jcharset utf8 -o ~/cd.iso ~/cd
```

Указание кодировки необходимо для правильного отображения русскоязычных имен файлов и каталогов под управлением MS Windows.

После создания ISO-образа его нужно записать на носитель с помощью команды `cdrecord`, как было показано ранее. После записи не забудьте удалить образ, чтобы он не занимал места на диске.

Существует способ записи каталога на CD/DVD без создания промежуточного ISO-образа. Для этого служит команда:

```
mkisofs -jcharset кодировка /каталог | cdrecord -опции
```

21.7.5. Преобразование образов дисков

Иногда нужно записать созданный в другой программе образ диска, формат которого отличается от ISO9660. Чаще всего встречаются образы дисков в форматах IMG, BIN, CUE, NRG, CCD.

Если у файла образа "расширение" (в Linux нет понятия "расширение", поэтому данное слово взято в кавычки) img, то это еще не означает, что формат образа ISO9660. Одни программы, например k3b, действительно создают образ в формате ISO9660 и записывают его в файл с расширением img, а другие — могут записывать в файл с таким же расширением образы диска в собственных форматах.

Файлы .bin/.cue можно записать на диск с помощью программы `cdrdao` или преобразовать в ISO с помощью программы `bchunk`.

Nero записывает образы диска в формате NRG, который можно преобразовать в ISO с помощью программы nrg2iso. Если вам нужно открыть NRG-образ, чтобы просмотреть его содержимое, вы это можете сделать с помощью команды:

```
mount -t udf,iso9660 -o loop,ro,offset=307200 файл.nrg точка_монтирования
```

Образ в формате CloneCD (ccd) можно преобразовать в ISO с помощью программы ccd2iso.

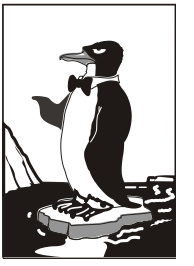
21.7.6. Создание и монтирование файлов с файловой системой

Иногда (например, для создания мини-дистрибутива) нужно создать файл, содержащий собственную файловую систему. Первым делом нужно создать пустой файл, потом создать в нем файловую систему, а затем подмонтировать этот файл к корневой файловой системе. Все это можно сделать с помощью трех команд:

```
# dd if=/dev/zero of=/file.fs bs=1k count=100000
# mkfs.ext2 -F /file.fs
# mount -t ext2 -o loop file.fs /mnt/disk
```

Первая команда создает пустой файл размером почти 100 Мбайт (100 000 Кбайт), вторая команда создает в этом файле файловую систему типа ext2, третья — монтирует файл к каталогу /mnt/disk.

Глава 22



Автоматизация выполнения задач. Планировщики задач crond, anacron, atd

22.1. Планировщик задач — зачем он нужен?

Очень часто приходится периодически выполнять одни и те же действия. Например, каждый день проверять обновление антивируса (или раз в неделю — в зависимости от того, как часто выходят для него обновления) или каждые 30 минут — почту. Можно выполнять эти действия самому, но это вариант не лучший. Представьте, что ваш рабочий день будет начинаться с команды запуска программы обновления антивируса, а каждые 30 минут вы будете вводить программу проверки почты. Во-первых, это не очень удобно, а во-вторых, можно легко забыть выполнить ту или иную команду. Например, в пятницу вечером вы можете забыть выполнить команду создания резервной копии, а в понедельник утром что-то случится с сервером, и вы не досчитаетесь всего пользовательского каталога. Не очень приятно, правда?

22.2. Планировщик crond

В Linux есть специальный демон crond, позволяющий выполнять программы по расписанию. Откройте конфигурационный файл демона crond — /etc/crontab (листинг 22.1).

Листинг 22.1. Пример файла /etc/crontab

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root nice -n 19 run-parts --report /etc/cron.hourly
02 4 * * * root nice -n 19 run-parts --report /etc/cron.daily
22 4 * * 0 root nice -n 19 run-parts --report /etc/cron.weekly
42 4 1 * * root nice -n 19 run-parts --report /etc/cron.monthly
```

Параметр SHELL задает имя программы-оболочки, параметр PATH — путь поиска программ, MAILTO — имя пользователя, которому будет отправлен отчет о выполнении расписания, а HOME — домашний каталог crond.

Но самое главное — не эти параметры, а сама таблица расписаний, занимающая в нашем случае последние четыре строки листинга. Согласно этой таблице, каждый час будут выполняться программы из каталога /etc/cron.hourly, каждый день — из каталога /etc/cron.daily, каждую неделю — из каталога /etc/cron.weekly, и раз в месяц — из каталога /etc/cron.monthly.

Предположим, вам нужно каждый день выполнять команду `update_av ftp://server.ru/bases/`. В каталоге /etc/cron.daily создайте файл update_av следующего содержания:

```
#!/bin/bash
update_av ftp://server.ru/bases/
```

Этот файл представляет собой небольшой bash-сценарий (сценарий командного интерпретатора). Теперь сделаем его исполнимым:

```
# chmod +x update_av
```

Правда, удобно?

Но иногда нам бывает нужно создать более гибкое расписание. Например, мы хотим, чтобы одна программа выполнялась в 7:00, а другая в 7:20. Тут простым добавлением сценария в каталог /etc/cron.daily уже не отделаешься. Чтобы создать такое расписание, вам придется изучить формат записей таблицы расписаний:

```
минуты (0–59) часы (0–23) день (1–31) месяц (1–12) день_недели (0–6, 0 — Вс) команда
```

Чтобы реализовать наше расписание, надо добавить в файл /etc/crontab следующие строки:

```
0   7      *      *      *      /usr/bin/command1 arguments
20  7      *      *      *      /usr/bin/command2 arguments
```

Первая команда будет запускаться каждый день в 7 часов утра, а вторая — тоже каждый день, но в 7:20.

Зная формат файла crontab, мы можем отредактировать стандартную таблицу расписаний (см. листинг 22.1). Обратите внимание — команды, выполняемые ежедневно, будут запускаться в 4 часа утра. Это, конечно, удобно, но они не будут выполнены, если вы выключаете сервер на ночь. Поэтому давайте установим другое время, например, 8 часов утра:

```
02 8 * * * root nice -n 19 run-parts --report /etc/cron.daily
```

Аналогичная ситуация и с еженедельным запуском. Программы будут запущены не только в 4:22 утра, но еще и в воскресенье. Однако на выходные вы точно выключаете свой сервер (впрочем, это зависит от политики организации — в некоторых организациях на выходные все компьютеры и не выключают). Поэтому целесообразно назначить запуск на понедельник в 8 часов 22 минуты:

```
22 8 * * 1 root nice -n 19 run-parts --report /etc/cron.weekly
```

С ежемесячным запуском вроде бы все нормально — программы будут выполняться в 4:42 первого числа каждого месяца. Хотя время лучше изменить на 8:42:

```
42 8 1 * * root nice -n 19 run-parts --report /etc/cron.monthly
```

22.3. Планировщик anacron

Планировщик anacron — непосредственный "родственник" crond, дальнейшее его развитие. Главное преимущество anacron заключается в том, что он, в отличие от crond, учитывает время, когда компьютер был выключен. Планировщик crond родом из UNIX, а эта операционная система устанавливалась только на серверах, которые всегда включены. Предположим, что вам нужно каждый понедельник в 7 часов утра рассылать некоторую информацию вашим сотрудникам. Вы настроили crond так, чтобы он запускал сценарий отправки сообщений каждый понедельник в 7 утра. Но вот беда — в 6 часов утра выключили электричество, а включили его, скажем, в 7:20. Но 7:20 — это не 7:00, следовательно, crond не выполнит задание по отправке сообщений, а ваши сотрудники не получат важную информацию.

Anacron работает не так. Если он обнаружил, что некоторые задания не выполнены по тем или иным причинам (выключение электричества, перезагрузка компьютера), он обязательно выполнит их. Поэтому ваши сотрудники получат информацию, но с небольшой задержкой. Все же лучше, чем получить важную информацию лишь в следующий понедельник.

Но и у anacron есть свои недостатки. В частности, пользователи не могут создавать свои собственные расписания, а файл /etc/anacrontab может редактировать только root. К тому же более старый crond является и более гибким в настройке — например, вы можете точно указать часы и минуты, а в случае с anacron можно указать только период, когда будет выполнена команда.

Формат файла /etc/anacrontab выглядит так:

```
Период          Задержка          ID          Команда
```

Например:

```
1   5           cron.daily        run-parts /etc/cron.daily
7   10          cron.weekly       run-parts /etc/cron.weekly
30  75          cron.monthly      run-parts /etc/cron.monthly
```

22.4. Разовое выполнение команд — демон atd

Иногда нужно просто выполнить определенные команды в определенное время (однократно), поэтому редактировать для этого таблицу crontab не совсем уместно. Такую задачу можно решить более рационально. Убедитесь, что у вас установлен и запущен демон atd. После этого введите команду:

```
at <время> [дата]
```

Затем просто вводите команды, которые вы хотите выполнить в указанное время. Для завершения ввода нажмите комбинацию клавиш <Ctrl>+<D>. Время указывается в AM/PM-формате — например, если вам нужно выполнить команды в 14:00, то вы должны ввести команду: `at 2pm`.

Просмотреть очередь заданий можно командой `atq`, а удалить какое-либо задание — командой `atrm`.

На рис. 22.1 изображено добавление команды в очередь atd, просмотр очереди, удаление задачи и повторный просмотр очереди.

В целях повышения безопасности в файл /etc/at.deny можно добавить команды, которые запрещены для выполнения планировщиком at.

```
[den@localhost ~]$ at 2pm
warning: commands will be executed using (in order) a) $SHELL b) login shell c)
/bin/sh
at> mc
at> <EOT>
job 1 at 2007-07-17 14:00
[den@localhost ~]$ atq
1       2007-07-17 14:00 a den
[den@localhost ~]$ atrm 1
[den@localhost ~]$ atq
[den@localhost ~]$
```

Рис. 22.1. Использование atd

ЧАСТЬ V

Локальная безопасность Linux-сервера

Настроить сервер — мало. Нужно также позаботиться о его безопасности. В пятой части книги будут рассмотрены основные уязвимости, а также способы их устранения. Все сказанное в этой части относится только к локальной безопасности сервера. А в седьмой части мы поговорим о безопасности сетевых служб, которые настроим в шестой части книги.

Глава 23

Основные уязвимости

23.1. От кого будем защищать сервер? Мотивация взлома

Взломать могут любую систему. Абсолютно любую. Даже самую защищенную. Если один человек защитил систему, то другой может ее взломать. Ведь человеку свойственно ошибаться. Кто-то совершает ошибки, а кто-то этим пользуется. От этого никуда не денешься. Все серверы, как и обычные компьютеры, подключенные к Интернету, являются потенциальными "жертвами".

В один прекрасный момент вы можете обнаружить, что ваш сервер взломан. О взломе вы рано или поздно узнаете. Например, обнаружится утечка данных из вашей компании или об этом будет красноречиво свидетельствовать главная страница сайта, оформленная в стиле "эТоТ сАЙт взлОмаЛ ВаСя". Но то, что уничтожена первая страница, — еще полбеды, ее легко восстановить, а вот если вы потеряете свою базу данных, то, конечно, это будет не очень хорошо.

Почему взламываются серверы? Кто-то делает это для удовлетворения собственного "Я". Как правило, это молодые люди в возрасте от 13 до 25 лет, которым хочется доказать кому-то (например, владельцу сайта) свое превосходство. Кто-то взламывает сайты по заказу.

Как вы думаете, каких "крекеров" больше — любителей, которые взламывают сайты от нечего делать (для них это вроде хобби), или профессионалов? Понятно, что любителей больше. Так что посмотрим на все это глазами взломщика-любителя. Какой бы сайт вы взломали: менее защищенный или хорошо защищенный? Правильно, менее защищенный. Ведь ваша цель — это сам факт взлома. Главное взломать, а какой именно сайт — это уже второстепенный вопрос.

Понятно, что кроме удовлетворения своего "эго", существуют и другие мотивы взлома:

- финансовый — самый значимый мотив, ради которого работают профессионалы своего дела. Как правило, вашу систему "заказывают" конкуренты. Это может быть обычный взлом с уничтожением данных сервера или же взлом с целью хищения информации;
- уменьшение пропускной полосы — для этого часто используют атаки типа DoS (Denial of Service, отказ в обслуживании). Тут все просто — ваш сервер загружают ненужной работой, пропускная полоса уменьшается, и сервер не может

больше обслуживать обращения реальных клиентов. Вы только представьте, что к вашему интернет-магазину не смогут обратиться настоящие покупатели — в результате вы потеряете клиентов, а это и потеря репутации, и финансовый ущерб;

- процессорное время — расшифровка некоторых зашифрованных данных требует много системных ресурсов. Злоумышленник может получить доступ к компьютерам вашей сети, чтобы использовать их ресурсы в своих целях;
- политический (религиозный) — ваш сервер могут взломать только из-за того, что он принадлежит какой-то политической или религиозной организации;
- месть — недовольный уволенный сотрудник (например, предыдущий системный администратор) вполне может взломать вашу сеть из-за мести. Поэтому в первый же день на новом месте рекомендуется изменить все значимые пароли и проверить систему на предмет неиспользуемых учетных записей. Предыдущий администратор может создать пользователя с административными правами, имя и пароль которого будет знать только он. Данная учетная запись впоследствии и будет использоваться для взлома сервера;
- анонимность — ваша система может использоваться как транзитная для взлома другой системы. Получается, что взлом той, другой, системы осуществляется как бы от вашего имени.

23.2. Ваша система взломана

Это довольно серьезная и одновременно неприятная ситуация. Взлом системы — это хуже, чем просто потеря данных. В случае потери данных (например, при выходе из строя жесткого диска) можно просто их восстановить из резервной копии. А вот при взломе системы важно понять, как она была взломана. После восстановления данных важно устранить уязвимость — "залатать дыру".

Если вы определили, что ваша система взломана, нужно первым делом отключить ее от сети (имеется в виду не сеть питания). После этого можно произвести анализ взлома — узнать, какие файлы были модифицированы. Может быть, злоумышленник не только взломал вашу систему, но и установил backdoor ("черный ход"), с помощью которого он в следующий раз проникнет в систему, или же, наоборот, проник в систему через ранее установленный backdoor.

> ***Пояснение***
>
> Под backdoor следует понимать как и установленную специально для этого программу, маскирующуюся под один из сетевых сервисов, так и учетную запись с административными правами, ранее созданную злоумышленником.

Также нужно выяснить, какие файлы "украл" злоумышленник и какая информация была в этих файлах. Все это поможет:

- устранить уязвимость;
- узнать мотивы злоумышленника;
- оценить ущерб взлома.

Самое сложное — определить, как злоумышленник проник в систему. Ведь если он "почистил" журналы, то сделать это будет слишком сложно. Тут все зависит

от уровня доступа злоумышленника (как "глубоко" он взломал систему) и от его квалификации. Самый простой пример — взлом "движка" сайта. В итоге вместо главной страницы отобразилась надпись злоумышленника. Кроме всеобщего позора, вам беспокоиться не от чем. Как правило, при таком взломе злоумышленник не может удалить протоколы сервера, поэтому достаточно их проанализировать, чтобы понять, как злоумышленник взломал вашу страничку. Затем следует устранить уязвимость, для чего нужно обладать минимальными навыками программирования на PHP (как правило, это будет PHP), или же обратиться к профессионалам за помощью, или вообще сменить движок на более безопасный.

В более сложных случаях после определения причины взлома надо восстановить резервную копию всей системы, устранить уязвимость и только после этого подключать сервер к Сети. Иначе все может повториться сначала.

Стоит понимать, что у взлома могут быть и юридические последствия. Простой сервера и финансовый ущерб — это только "цветочки", а "ягодки" — это возможный юридический ущерб, когда злоумышленник похитил конфиденциальные данные.

А теперь самое интересное — кто будет отвечать за взлом? С одной стороны, есть злоумышленник, который и произвел взлом. А с другой стороны, есть вы — администратор, который допустил взлом. Создать абсолютно безопасную систему невозможно — это что-то из области научной фантастики, но максимально защитить систему можно попытаться. Представьте ситуацию, что в вашем магазине кто-то бросил на пол кожуру от банана, и ваш покупатель на ней поскользнулся. С одной стороны, вы не виноваты, а с другой — виноваты, потому что вовремя не убрали ее. Точно так же и с сервером. Уязвимость может существовать довольно давно, но вы ее не заметили, а через некоторое время ваш сервер взломали.

Когда вы приходите на новое рабочее место, перед вами всегда стоит выбор: или настраивать сервер заново, или же оставить как есть. Многие выбирают второй вариант, мотивируя это нежелательным простоем сервера — ведь придется разбираться с настройками сервера: от настроек загрузчика до последнего параметра последнего сетевого сервиса. Однако рекомендую это сделать — возможно, где-то окажется уязвимость, специально оставленная предыдущим администратором.

Также следует понимать, что в корпоративных сетях взлом осуществляется часто изнутри сети. Не нужно искать мифического хакера, который взламывает ваш сервер, находясь на другой стороне Земного шара. Вашим "хакером" может вполне оказаться сотрудник, находящийся в соседнем кабинете. Вот так...

Следите за новинками безопасности, регулярно устанавливайте обновления безопасности (впрочем, рекомендуется устанавливать не все подряд, а хотя бы ознакомиться с сутью обновления перед его установкой), а также время от времени устраивайте аудит вашей сети.

Вот что нужно сделать в первый рабочий день, получив под ответственность уже настроенный сервер:

- изменить пароль к учетной записи администратора — это не только учетная запись root, но и, возможно, другая запись, которая используется для входа администратора. Вообще, если учетная запись root отключена, а для входа администратора используется учетная запись admin, рекомендуется ее отключить, а вместо нее создать другую учетную запись (чтобы никто не знал, как она называется,

особенно — предыдущий администратор). Возможно, придется изменить некоторые конфигурационные файлы, но ничего страшного — это полезно;

- проанализировать учетные записи пользователей — если пользователей всего 100 (100, 151, 200 и т. д.), а учетных записей больше, и некоторые из них активны, лучше отключить "лишние" записи — до выяснения. Потом, возможно, кто-то из пользователей пожалуется, что не может войти под отключенной записью. После "разбора полетов", можно включить эту учетную запись или предоставить пользователю новую. Возможно, никакого "разбора полетов" и не будет, а лишние записи окажутся принадлежащими уволенным сотрудникам (или же специально созданными уволенными сотрудниками, чтобы иметь возможность доступа к системе). Работа рутинная, но ее нужно сделать;
- отключить лишние сетевые сервисы — к слову, если ваш сервер используется только как внутренний FTP-сервер, то зачем на нем запущен SSH-демон? Если вы не занимаетесь удаленным администрированием сервера, то его лучше отключить. Во-первых, "лишний" сервис — это потребление лишних ресурсов. Во-вторых, это лишняя возможность входа в систему.

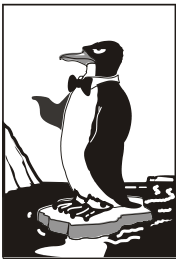
23.3. Основные уязвимости Linux-сервера

Linux уже давно избавилась от "детских болезней", и теперь это вполне безопасная система. Но не нужно забывать о человеческом факторе — например, о неправильной настройке сервера или слишком простом пароле. Вот характерные уязвимости современного Linux-сервера:

- **простой пароль** — рекомендуется раз в два месяца менять пароли для всех пользователей вашего сервера. Пароли лучше генерировать случайным образом, чтобы их было сложнее подобрать;
- **неправильные права доступа** — некоторые администраторы устанавливают биты SUID/SGID на некоторые административные программы, чтобы их могли запускать обычные пользователи. Этого делать не нужно, поскольку этим может воспользоваться злоумышленник;
- **неправильные настройки сетевых служб** — бывает, что служба настроена неправильно, и злоумышленник может получить доступ к файловой системе. Это часто касается служб ssh, ftpd и др. Но если вы будете запускать все службы в chroot-окружении *(см. главу 39)*, то ничего страшного не произойдет — злоумышленник получит доступ лишь к файловой системе chroot-окружения, а не всего сервера;
- **"лишние" учетные записи** — об этом уже было много сказано, поэтому не буду повторяться;
- **руткиты** — это наборы программ, таящие в себе скрытые возможности. Руткит может заменять набор стандартных команд (`ls`, `cat` и т. д.), но каждая из таких команд будет обладать скрытыми возможностями. Например, команда `ls` может не показывать "лишние" файлы — файлы скрытой службы (backdoor), предоставляющей злоумышленнику доступ к системе. А команда `who` может не пока-

зывать регистрацию в системе определенного пользователя (учетную запись взломщика). Как правило, руткиты устанавливаются предыдущими администраторами или же "крекерами" после взлома системы, чтобы замаскировать свое пребывание в системе. Обнаружить руткит бывает чрезвычайно сложно (особенно, если руткит работает на уровне ядра), поэтому рекомендуется настраивать свой сервер с нуля — тогда уж точно в нем не будет руткитов;

- **вирусы и черви** — для Linux тоже есть вирусы, и не нужно ими пренебрегать. Однако при соблюдении элементарных правил безопасности — например, если не работать постоянно в системе под пользователем root, а использовать команду `sudo` для запуска только определенных программ с полномочиями root, никакие вирусы и черви вам не страшны;
- **спам** — это не прямая уязвимость, а косвенная. От одного письма со спамом вашему серверу ничего не будет. Но спам может заполнить весь жесткий диск, после чего сервер перестанет принимать письма нормальных пользователей. Кстати, это один из вариантов атаки на отказ (DoS). Поэтому со спамом вам тоже придется бороться.



Глава 24

Обеспечение безопасности сервера

Гибкость Linux — это источник ее проблем. Она настолько гибка, что с одинаковой легкостью предоставляет свои возможности как законному администратору, так и злоумышленнику. Любой, кто получит физический доступ к серверу (то есть к его клавиатуре и монитору), может захватить root-доступ всего за несколько секунд, если будет знать, что делать. В этой главе мы поговорим о том, как защитить наш сервер от подобных вмешательств.

24.1. Защита от "восстановления пароля root"

Любой желающий может подойти к компьютеру, перезагрузить его и передать ядру Linux параметр `single`. В результате система будет загружена в однопользовательском режиме, а злоумышленник без лишних вопросов получит root-доступ. Время, необходимое на варварскую перезагрузку системы (нажатием кнопки Reset) — несколько миллисекунд, затем еще 15–30 секунд до появления загрузочного меню, еще несколько секунд на ввод параметров ядра, секунд 20–30 на загрузку Linux в однопользовательском режиме. Грубо говоря, через минуту злоумышленник сможет делать с вашей системой все, что захочет. Сможет даже с помощью команды `passwd root` изменить пароль root. Вот вам и одна из самых безопасных систем! С другой стороны, описанная тактика используется для восстановления пароля root в случае, если администратор системы страдает легкой формой склероза.

Однако делу можно помочь. Например, настроить систему так, чтобы она запрашивала пароль при загрузке в однопользовательском режиме. Но мы не будем этого делать. Почему? Да потому что это — не панацея. Злоумышленник может передать ядру другой параметр: `init=/bin/bash`. Параметр `init` задает программу инициализации системы. По умолчанию загружается программа /sbin/init, но, используя параметр `init`, можно запустить любую программу. В данном случае будет выполнена команда `/bin/bash` — запущен командный интерпретатор. Вы уже догадались, что программа эта будет запущена с правами root. По умолчанию корневая файловая система монтируется в режиме `ro` (только чтение), поэтому чтобы получить полный контроль над системой, злоумышленнику достаточно перемонтировать файловую систему в режиме `rw` (см. руководство `man mount`) — и снова можно творить с системой все, что заблагорассудится.

Защитить систему можно с помощью загрузчика Linux. Чтобы никто без вашего ведома не мог изменить параметры ядра Linux, нужно установить пароль на изменение параметров. После установки пароля любую операционную систему можно будет загрузить без пароля, а вот при попытке изменения параметров ядра Linux загрузчик запросит пароль. О том, как установить пароль в загрузчиках GRUB и GRUB2, мы поговорим в *главе 25*.

> ***Примечание***
>
> Вообще, загрузчик GRUB позволяет установить пароль также на загрузку той или иной ОС, но мы этого делать не будем — иначе вы не сможете удаленно перезагрузить ваш сервер (а это иногда может понадобиться), ведь тогда при загрузке GRUB запросит пароль, а ввести его будет некому.

Но это еще не все. Злоумышленник может загрузиться с LiveCD, после чего с помощью команды `chroot` заменить корневую файловую систему LiveCD файловой системой сервера и опять получить полный контроль над ним. В этом случае нужно не забыть установить пароль на вход в BIOS Setup, что не позволит злоумышленнику изменить порядок загрузки и загрузиться с LiveCD. Казалось бы все. Но корпус системного блока современного компьютера можно открыть даже без отвертки, причем за пару секунд, и еще за несколько секунд переустановить джампер, стирающий все настройки BIOS, в том числе и установленный пароль. В этом случае желательно использовать специальные корпуса с замками — такой замок тоже можно взломать, но тут факт взлома будет, как говорится, налицо. А в случае с обычным корпусом вы можете ничего не заметить, а злоумышленник тем временем внедрит backdoor-программу, позволяющую удаленно контролировать ваш сервер!

24.2. Защита от перезагрузки

От грубой перезагрузки защиты нет. Да, можно в BIOS Setup отключить кнопку Reset (а если такой опции там нет, то просто физически отключить разъем кнопки Reset в системном блоке). Но толку особого от этого не будет — если кто-то получит физический доступ к серверу, то ничего ему не помешает вытащить из розетки кабель питания. Вот и весь секрет.

Однако мы можем блокировать хотя бы программную перезагрузку, осуществляющуюся с помощью клавиатурной комбинации <Ctrl>+<Alt>+<Del>. Процедура блокирования <Ctrl>+<Alt>+<Del> зависит от используемой системы инициализации. Сначала мы рассмотрим процесс блокирования для системы инициализации init, которая используется в дистрибутивах Fedora, Mandriva, openSUSE и др. А затем разберемся, как заблокировать <Ctrl>+<Alt>+<Del> в Ubuntu.

Итак, откройте файл /etc/inittab и найдите в нем строчку:

```
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now
```

Эту строчку нужно закомментировать, а еще лучше — изменить так (рис. 24.1):

```
ca::ctrlaltdel:/bin/true
```

Как вы уже догадались, система не будет производить никаких действий при нажатии комбинации <Ctrl>+<Alt>+<Del>. А вы для перезагрузки компьютера будете использовать команды `reboot` или `shutdown`.

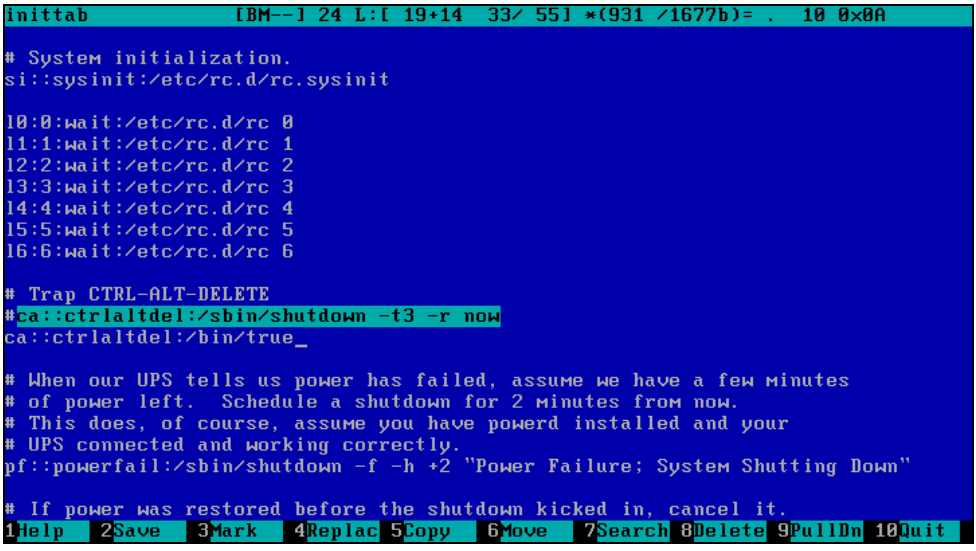


Рис. 24.1. Редактирование файла inittab:
реакция на <Ctrl>+<Alt>+<Del> отключена

Теперь рассмотрим процесс блокирования <Ctrl>+<Alt>+<Del> в дистрибутивах Ubuntu/Denix. Откройте файл /etc/event.d/control-alt-delete:

```
sudo nano /etc/event.d/control-alt-delete
```

Закомментируйте следующую строку:

```
exec /sbin/shutdown -r now "Control-Alt-Delete pressed"
```

Должно получиться так:

```
#exec /sbin/shutdown -r now "Control-Alt-Delete pressed"
```

После этого сохраните файл. Но это еще не все. Приведенная здесь последовательность действий правильна для старых версий Ubuntu. А вот в последних версиях (если мне не изменяет память, начиная с версии 9.10, хотя может быть и с версии 9.04) редактировать нужно файл /etc/init/ctrl-alt-delete.conf. В этом файле нужно найти подобную строку, а именно:

```
exec shutdown -r now "Control-Alt-Delete pressed"
```

Что делать с ней, вы уже знаете — ее нужно закомментировать. После этого ваша система не будет реагировать на нажатие комбинации <Ctrl>+<Alt>+<Delete>.

24.3. Отключение учетной записи root — нестандартный метод

Довольно часто сервер настраивается по принципу — настроил и забыл. Да, после правильной настройки о нем забудете вы, но только не злоумышленники, которым не будет давать покоя ваша учетная запись root. Именно она, как учетная запись с максимальными правами, представляет наибольший интерес для злоумышленника.

Сейчас мы рассмотрим один из методов отключения учетной записи root. Предлагаемый метод довольно-таки варварский, но не более варварский, чем команда `rm -rf /`, которую может ввести "доброжелатель", получив доступ к root. Метод неудобен тем, что для входа в систему как root придется перезагружать компьютер.

Первым делом разрешим одному обычному пользователю (это ваша учетная запись, под которой вы работаете каждый день) выполнять некоторые команды от имени root. Для этого откройте ваш файл /etc/sudoers и добавьте в него подобную строку:

```
den    localhost = NOPASSWD: /bin/kill, /sbin/reboot, /sbin/halt
```

Такая строка разрешает пользователю `den` выполнять команды `/bin/kill`, `/sbin/reboot`, `/sbin/halt` с привилегиями root на машине `localhost` без ввода пароля. Можете дополнительно обезопасить систему, изменив приведенную строчку так:

```
den    localhost = PASSWD: /bin/kill, /sbin/reboot, /sbin/halt
```

Тогда при вводе указанных команд у вас будет запрошен пароль пользователя `den` (а не root!). Команды `kill`, `reboot` и `halt` нужно вызывать через `sudo`:

```
sudo /bin/kill <PID>
sudo /sbin/reboot
sudo /sbin/halt
```

Еще раз отмечу, что сначала нужно разрешить обычному пользователю перезагружать компьютер, иначе после отключения учетной записи root и клавиатурной комбинации <Ctrl>+<Alt>+<Del> единственным способом перезагрузки останется кнопка Reset.

Вот теперь надо открыть файл /etc/passwd и заменить строку:

```
root:x:0:0:root:/root:/bin/bash
```

следующей строкой:

```
root:x:0:0:root:/root:/bin/true
```

После этого в качестве командной оболочки пользователя root будет использоваться программа /bin/true — она запускается, возвращает истинное значение (для сценариев) и завершает работу. Сохраните /etc/passwd, введите команду `exit` и попробуйте войти как root. У вас ничего не получится — даже если злоумышленник узнает пароль root, он не сможет им воспользоваться.

Теперь войдите в систему как обычный пользователь и введите команду `su`. Как обычно будет запрошен пароль root, но также будет запущена оболочка root — команда /bin/true, которая немедленно завершит сеанс root.

А сейчас — самое интересное. Поговорим, как получить обратно root-доступ, когда он нам понадобится. Для этого придется перезагрузить компьютер и передать ядру параметр `init=/bin/bash` (рис. 24.2) — поскольку мы знаем пароль загрузчика, для нас это не составит проблемы (рис. 24.3).

После загрузки системы нужно будет перемонтировать корневую файловую систему в режиме `rw` (чтение/запись). Для этого используется команда (рис. 24.4):

```
mount -w -o remount /dev/sda1/
```

Конечно, фрагмент `/dev/sda1` надо заменить на имя раздела, на который установлена Linux. Вуаля! Мы получили полный контроль над системой.

Рис. 24.2. Передача параметра init=/bin/bash ядру

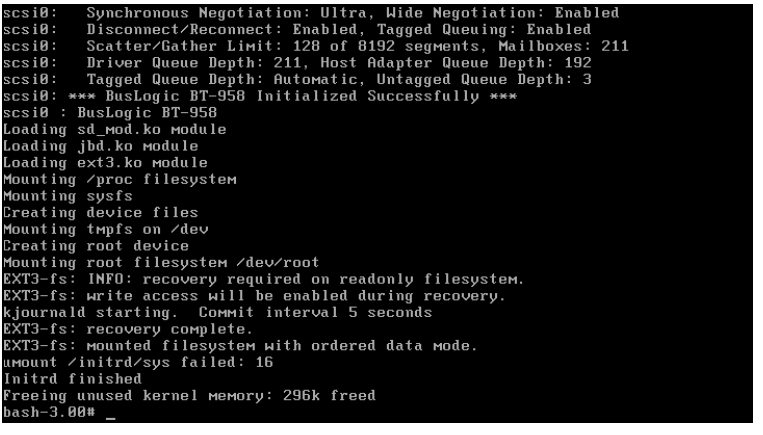


Рис. 24.3. Загрузка с параметром init=/bin/bash

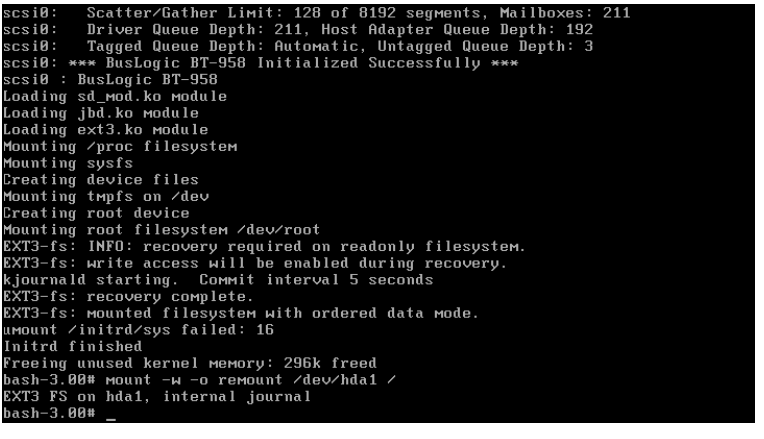


Рис. 24.4. Монтирование корневой файловой системы в режиме чтение/запись

Теперь запускаем наш любимый `mc`, открываем файл /etc/passwd и изменяем оболочку root — в этот раз на /bin/bash.

После этого для загрузки системы без X.Org (`init 3`) или в графическом режиме (`init 5`) вводим команду:

```
/sbin/init 3
```

или, соответственно:

```
/sbin/init 5
```

Теперь вы можете полноценно войти в систему как root. По завершении работы не забудьте вернуть все, как было.

24.4. Отключение учетной записи root средствами KDM

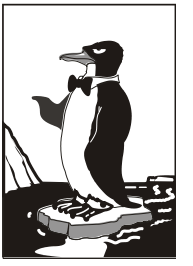
Довольно часто "доброжелатели" знают, что такое root, но не знают ни одной UNIX-команды — следовательно, не могут причинить системе особого вреда в консоли, но могут натворить невесть что, зарегистрировавшись в графическом режиме (если, конечно, сервер загружается на пятом уровне запуска).

Так вот, KDM (K Display Manager) позволяет запретить вход пользователю root в графическом режиме. Конечно, для того, чтобы приведенный далее совет работал, нужно, чтобы KDM был установлен как основной менеджер дисплея (обычно это так, если установлена KDE).

Итак, откройте файл `/etc/alternatives/kdm4-config`. Найдите в нем строку `AllowRootLogin=true` и замените ее строкой `AllowRootLogin=false`. Сохраните файл и завершите сеанс работы пользователя. После этого вы не сможете войти в систему как root в графическом режиме.

24.5. Система управления доступом

Все описанные здесь действия довольно полезны, особенно учитывая, что о них часто забывают. Но все же они не дают полной защиты сервера. Если вы заботитесь о безопасности сервера, настоятельно рекомендую установить и настроить одну из *систем управления доступом* (SELinux, Tomoyo, LIDS, GrSecurity). Самой защищенной считается система Tomoyo, которая будет рассмотрена в *главе 40*.



Глава 25

Параметры загрузчика Linux

25.1. Установка пароля

25.1.1. Загрузчик GRUB2

Как уже отмечалось, начиная с версии 9.10, в Ubuntu используется загрузчик GRUB2 вместо обычного GRUB. По сравнению с GRUB, новый загрузчик одновременно и проще в обращении, и сложнее в настройке. Настраивать GRUB2 придется реже, но к его сложной настройке надо будет привыкать, — практически все современные дистрибутивы перешли на GRUB2.

В GRUB можно было задать общий пароль для всех загрузочных меток, а также установить пароль только на некоторые загрузочные метки. В GRUB2 можно сделать то же самое, но, кроме самого пароля, понадобится указать еще и имя пользователя (логин), что усложняет злоумышленнику взлом системы, поскольку ему нужно будет знать не только пароль, но и логин. Защита отдельных загрузочных меток, как правило, используется редко, чаще устанавливается пароль на все метки сразу, что и будет продемонстрировано в этой главе.

Сначала установим простой (незашифрованный) пароль, а затем зашифруем его, чтобы никто не смог его прочитать, загрузившись с LiveCD. Прежде всего откройте файл /etc/grub.d/00_header:

```
sudo nano /etc/grub.d/00_header
```

В конец файла добавьте строки:

```
cat << EOF
set superusers="den"
password den 1234
EOF
```

Здесь имя пользователя `den`, пароль мы придумали для примера такой: `1234`.

Теперь обновите GRUB2:

```
sudo update-grub
```

Можно также напрямую редактировать grub.cfg — файл конфигурации GRUB2. В него следует добавить вот такие строки:

```
set superusers="user1"
password user1 password1
password user2 password2
```

Обратите внимание, что командами `password` заданы два пользователя: `user1` и `user2` с паролями `password1` и `password2` соответственно. Но пользователь `user1` является суперпользователем, то есть может редактировать загрузочные метки GRUB2, а обычный пользователь (`user2`) может только загружать метки. Таким образом, у пользователя `user1` получится передать ядру новые параметры, а пользователь `user2` сможет только загрузить Linux с параметрами по умолчанию.

Можно даже задать условие, что метку Windows будет загружать только пользователь `user2`:

```
menuentry "Windows" --users user2 {
        set root=(sd0,2)
        chainloader +1
}
```

Теперь разберемся с шифрованием пароля. Команда `password` поддерживает только незашифрованные пароли. Если вы хотите использовать зашифрованные пароли, то нужно применить команду `password_pbkdf2`. Например:

```
password_pbkdf2 den зашифрованный_пароль
```

Получить зашифрованный пароль можно командой:

```
grub-mkpasswd-pbkdf2
```

Программа запросит у вас пароль (придумайте и введите пароль в ответ на запрос), закодирует его и выведет на экран хэш (шифр) введенного вами пароля:

Your PBKDF2 is grub.pbkdf2.зашифрованный_пароль

Пример такого шифра:

```
grub.pbkdf2.sha512.10000.9290F727ED06C38BA4549EF7DE25CF5642659211B7FC076F2D
28FEFD71784BB8D8F6FB244A8CC5C06240631B97008565A120764C0EE9C2CB0073994D79080
136.887CFF169EA8335235D8004242AA7D6187A41E3187DF0CE14E256D85ED97A97357AAA8F
F0A3871AB9EEFF458392F462F495487387F685B7472FC6C29E293F0A0
```

Весь этот хэш нужно скопировать в конфигурационный файл GRUB2:

```
password_pbkdf2 den
grub.pbkdf2.sha512.10000.9290F727ED06C38BA4549EF7DE25CF5642659211B7FC076F2D
28FEFD71784BB8D8F6FB244A8CC5C06240631B97008565A120764C0EE9C2CB0073994D79080
136.887CFF169EA8335235D8004242AA7D6187A41E3187DF0CE14E256D85ED97A97357AAA8F
F0A3871AB9EEFF458392F462F495487387F685B7472FC6C29E293F0A0
```

Если вы не использовали файл 00_header, а редактировали непосредственно файл grub.cfg, то команду `update-grub` вводить не нужно!

Дополнительную информацию вы сможете получить по адресам:

http://ubuntuguide.net/how-to-setup-boot-password-for-grub2-entries и **http://grub.enbug.org/Authentication**.

25.1.2. Загрузчик GRUB

Теперь проделаем то же самое, но для загрузчика GRUB. Введите команду `grub` (с полномочиями root). Появится приглашение:

```
grub>
```

В ответ на приглашение введите команду:

```
md5crypt
```

Программа запросит у вас пароль (придумайте и введите пароль в ответ на запрос), закодирует его и выведет на экран шифр введенного вами пароля:

```
Password: ******
```

Итак, вы получили зашифрованный пароль. Перепишите этот шифр (а еще лучше выделите его и выполните команду меню терминала **Правка | Копировать**). После этого введите команду `Quit`.

На всякий случай сделайте копию конфигурационного файла загрузчика:

```
sudo cp /boot/grub/grub.conf /boot/grub/grub.conf_backup
```

Теперь откройте файл /boot/grub/grub.conf в любом текстовом редакторе:

```
gksudo gedit /boot/grub/grub.conf
```

Найдите секцию пароля:

```
## password ['--md5'] passwd
# If used in the first section of a menu file, disable all interactive editing
# control (menu entry editor and command-line)  and entries protected by the
# command 'lock'
# e.g. password topsecret
#      password --md5 $1$gLhU0/$aW78kHK1QfV3P2b2znUoe/
# password topsecret
```

После нее вставьте строку:

```
password --md5 ваш-шифр
```

Параметр `ваш-шифр` здесь — это тот шифр, который вы получили в ответ на введенный пароль.

Таким образом мы задали пароль, с помощью которого можно редактировать загрузочное меню GRUB. Пока не будет указан заданный пароль, GRUB не разрешит редактировать загрузочное меню. Другими словами, сервер загрузится (например, если он был выключен по сигналу от UPS, когда пропало питание), но никто не сможет изменить загрузочные параметры без ввода пароля.

25.2. Восстановление загрузчика GRUB/GRUB2

Ситуации бывают разные. Иногда "падает" загрузчик Linux. Такая ситуация чаще случается на домашних компьютерах, где еще установлена и Windows, — после очередной переустановки "окон" происходит перезапись загрузчика, и GRUB удаляется, а на его место записывается загрузчик NTLoader. На сервере же, как правило, установлена одна операционная система, и такая ситуация полностью исключена. Но мало ли чего бывает. Может, вы производили какие-то манипуляции с загрузочным сектором и повредили загрузчик. Тогда его нужно восстановить. И в самом деле — не переустанавливать же всю систему из-за такой мелочи?

Для восстановления загрузчика GRUB нужно загрузиться с LiveCD (подойдет LiveCD с любым дистрибутивом Linux) и ввести следующие команды:

```
mkdir /old
mkdir /old/dev
mount /dev/sdaN /old
```

Все команды нужно вводить от имени root. Для этого следует использовать команды `su` или `sudo`. В частности, в LiveCD Ubuntu нужно вводить все команды с использованием команды `sudo`, например, так:

```
sudo mkdir /old
sudo mkdir /old/dev
…
```

Разберемся, что означают эти команды:

- первая из них создает каталог /old, который будет использоваться в качестве точки монтирования;
- вторая — создает в этом каталоге подкаталог dev, который пригодится для монтирования devfs — псевдофайловой системы;
- третья — служит для монтирования корневой файловой системы дистрибутива Linux, установленного на жестком диске в разделе /dev/sda*N* (где *N* — номер раздела), к каталогу /old. Предположим, что на вашем компьютере дистрибутив Linux был установлен в раздел /dev/sda5. Тогда вам нужно ввести следующую команду:

  ```
  mount /dev/sda5 /old
  ```

После этого нужно подмонтировать каталог /dev к каталогу /old/dev. Это делается с помощью все той же команды `mount`, но с параметром `--bind`:

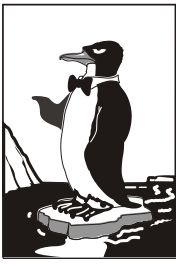
```
mount --bind /dev /old/dev
chroot /old
```

Команда `chroot` заменяет корневую систему нашего LiveCD на корневую систему дистрибутива, установленного на винчестере. Вам остается лишь ввести команду:

```
/sbin/grub-install /dev/sda
```

Эта команда установит загрузчик GRUB так, как он был установлен до переустановки Windows. После установки GRUB нужно перезагрузить компьютер командой `reboot`.

> ***Примечание***
>
> Дополнительную информацию о восстановлении загрузчика GRUB вы можете получить на моем форуме: **http://www.dkws.org.ua/phpbb2/viewtopic.php?t=3275**.



Глава 26

RAID-массивы

26.1. Что такое RAID?

Массивы RAID (Redundant Array of Independent Disk, матрица независимых дисков с избыточностью) обеспечивают более надежное хранение ваших данных. Так, при объединении двух винчестеров в один RAID-массив все, что будет записано на первый винчестер, автоматически продублируется на второй. Если с первым винчестером что-то случится (у жестких дисков есть свойство периодически выходить со строя, это может произойти раз в 5 лет, но, все равно, терять данные не хочется), мы сможем восстановить свои данные со второго винчестера. Приведенный пример является далеко не единственным способом организации RAID-массивов. Алгоритм работы RAID-массива зависит от уровня RAID. Всего существует 6 уровней, описанных в табл. 26.1.

Таблица 26.1. *Уровни RAID-массивов*

Уровень	Алгоритм работы
0	Предназначен не для обеспечения надежности, а для увеличения суммарного объема диска. Предположим, у нас есть два винчестера по 200 Гбайт. Объединив их в один винчестер, мы получим один диск на 400 Гбайт. Очень удобно, если мы работаем с видео (имеется в виду профессиональный видеомонтаж, а не просто просмотр фильмов)
1	Простое зеркальное копирование, как было описано ранее. Все, что записано на первый жесткий диск, будет продублировано на второй. Желательно, чтобы диски были одного размера. Если это не так, то размер RAID-массива будет равен размеру меньшего диска
2	Используется метод битового чередования блоков данных, при этом добавляются коды коррекции ошибок
3	Усовершенствованный уровень 2 — коды коррекции ошибок записываются на другой диск
4	Усовершенствованный уровень 3 — практически то же самое, но изменен метод записи контрольных кодов
5	Пока не появился уровень 6, этот уровень был самым надежным. Использует контрольные суммы, и данные записываются вместе с контрольными кодами на все диски. Если с одним из дисков что-то случилось, то данные можно восстановить с помощью контрольной суммы. Общий размер массива вычисляется по формуле M ×(N − 1), где N — количество дисков в массиве, а M — размер наименьшего диска. Минимальное значение N = 3

Таблица 26.1 *(окончание)*

Уровень	Алгоритм работы
6	Похож на RAID 5, но отличается тем, что в каждом ряду данных есть не один, а два блока контрольных сумм, причем эти контрольные суммы многомерные, то есть независимые друг от друга, что позволяет сохранить исходные данные даже при отказе двух дисков в массиве. RAID 6 является более надежным, чем RAID 5, поэтому по возможности нужно использовать RAID 6, но, все же, делать резервные копии

На практике обычно используются уровни 5, 1, 0. Некоторые материнские платы поддерживают RAID-массивы на аппаратном уровне. Раньше поддержкой RAID-массивов обладали только дорогие серверные материнские платы. Сейчас поддержку RAID можно встретить в относительно недорогих материнских платах среднего ценового диапазона. О создании и поддержке аппаратных RAID-массивов вы можете прочитать в документации по вашей материнской плате.

Кроме уровней RAID 1–6, описанных в стандарте, некоторые производители создают комбинированные уровни: RAID 10 (1+0), RAID 15 (1+5), RAID 50 (5+0) и т. д. Суть этих комбинаций заключается в следующем: RAID 10 — комбинация уровней 1 и 0, 15 — уровней 1 и 5 (то есть — это зеркало "пятерок") и т. д. Такие комбинированные уровни сочетают в себе преимущества и недостатки своих "родителей". Например, уровень RAID 50 — практически то же, что и RAID 5, но зато быстрее, чем RAID 5.

Кроме обычных уровней, есть еще и расширенные уровни RAID, тогда к наименованию уровня добавляется буква E, например, RAID 1E, RAID 5E и т. д. Это усовершенствованные версии базовых уровней. Чтобы не описывать каждый такой уровень, лучше рассмотрим таблицу с общими характеристиками самых часто используемых уровней RAID (табл. 26.2), то есть именно с характеристиками тех уровней, которые вы будете использовать на практике.

Таблица 26.2. *Характеристики уровней RAID*

Уровень	Избыточность	Мин.	Макс.	Чтение	Запись	Емкость
0	–	1	16	10	10	100
1	+	2	2	8	8	50
5	+	3	16	10	7	67–94
6	+	4	16	10	7	50–88
10 (1+0)	+	4	16	9	9	50
15	+	6	60	10	7	33–48
1E	+	3	16	8	8	50
1E0	+	2	60	8	8	50
50	+	6	60	10	7	67–94
5E	+	4	16	10	7	50–88
5EE	+	4	16	10	7	50–88
00	–	2	60	10	10	100

Пояснение

Здесь колонка "Избыточность" показывает, поддерживает ли уровень избыточность данных. Колонка "Мин." — минимальное количество дисков в массиве, колонка "Макс." — соответственно, максимальное количество дисков. Колонки "Чтение" и "Запись" — оценки производительности чтения и записи по 10-балльной шкале. Колонка "Емкость" — использование емкости дисков в процентном соотношении.

26.2. Программные RAID-массивы

В Linux можно создавать программные RAID-массивы, даже если ваша материнская плата не поддерживает их на аппаратном уровне. У программных массивов есть один маленький недостаток — они работают немного медленнее аппаратных. Впрочем, у них есть и одно неоспоримое преимущество — поскольку обработка данных происходит на программном уровне, совсем необязательно, чтобы жесткие диски, входящие в состав массива, были совместимы между собой. Например, можно создать массив уровня 5, который будет состоять из дисков EIDE, SATA и SCSI, — это три разных интерфейса, объединить которые в аппаратный массив нельзя никак.

Поддержка RAID-массивов встроена в ядро по умолчанию, поэтому вам даже не придется перекомпилировать ядро. При загрузке Linux вы должны увидеть следующие строки:

```
md: md driver 0.90.2 MAX_MD_DEVS=256, MD_SB_DISKS=27
md: bitmap version 3.39
…
md: Autodetecting RAID arrays.
md: autorun ...
md: ... autorun DONE.
```

Если вы их увидели (если при загрузке вы не успели их заметить, введите команду `dmesg`), значит, ваше ядро поддерживает RAID. Не поддерживать RAID могут компактные ядра некоторых дистрибутивов, которые мы в этой книге не рассматриваем. Fedora, ASP Linux, Mandriva, ALTLinux поддерживают RAID-массивы по умолчанию.

Если же поддержки RAID почему-то в вашем дистрибутиве не оказалось, то включить ее можно в разделе **Block device** конфигуратора `make menuconfig`. Затем придется перекомпилировать ядро. После загрузки с новым ядром следует установить пакет raidtools, содержащий необходимые нам программы `raidhotadd`, `raidhotremove`, `mkraid`. Последняя команда создает RAID-массив, первая добавляет в него диск, а вторая — удаляет диск из массива.

26.3. Создание программных массивов

В этом разделе мы создадим массивы уровней 1 и 5. Уровень 0 нам не нужен, поскольку обрабатывать видео в Linux мы не будем.

Начнем с массива уровня 1. Создайте два раздела типа **Linux raid autodetect**. Разделы можно создать как на одном, так и на разных дисках. Лучше, если вы соз-

дадите разделы на разных дисках, — так будет надежнее. После этого отредактируйте файл /etc/raidtab (листинг 26.1).

Листинг 26.1. Файл /etc/raidtab для уровня 1

```
# Имя устройства RAID-массива
raiddev /dev/md0
# Указываем уровень
raid-level 1
# Число дисков в RAID-массиве
nr-raid-disk 2
# Число дисков "на подхвате" — они будут использованы, если 1 из дисков
# выйдет из строя
nr-spare-disk 0
# Другие параметры
chunk-size 8
persistent-superblock 1

# Первый диск RAID
device /dev/sdc3
raid-disk 0
# Второй диск RAID
device /dev/sda7
raid-disk 1
```

Теперь нужно создать устройство /dev/md0, которое мы упомянули в конфигурационном файле. Для этого используем команду:

```
# mkraid /dev/md0
```

После этого вы можете использовать устройство /dev/md0 как самый обычный жесткий диск — создавать на нем разделы, монтировать разделы, создавать данные и т. д.

Конфигурационный файл для уровня 5 выглядит немного иначе (листинг 26.2).

Листинг 26.2. Файл /etc/raidtab для уровня 5

```
raiddev /dev/md0
raid-level 5
nr-raid-disk 3
nr-spare-disk 0
persistent-superblock 1
parity-algorithm left-symmetric
```

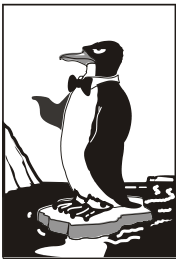
```
chunk-size 64
device /dev/sdc1
raid-disk 0
device /dev/sda7
raid-disk 1
device /dev/sdd3
raid-disk 2
```

Если один из дисков вышел из строя, то надо использовать команду `raidhotremove`, чтобы извлечь его из массива. Затем на другом жестком диске создать разделы для RAID-массива (размер и количество разделов должны быть такими же, как у извлеченного диска), а затем добавить новый диск командой `raidhotadd`.

ЧАСТЬ VI

Настройка сетевых служб

Итак, мы получили безопасно настроенный компьютер, работающий под управлением Linux. Сервером его не назовешь, поскольку никаких сервисов другим пользователям/компьютерам сети он не предоставляет. В шестой части книги происходит превращение хорошо настроенного компьютера в сервер.



Глава 27

DNS-сервер

27.1. Еще раз о том, что такое DNS

Система доменных имен (DNS, Domain Name System) используется для преобразования IP-адресов в доменные имена и обратно. Компьютеру намного проще работать с числами, человеку же проще запомнить символьное имя узла, чем его IP-адрес.

Система DNS имеет древовидную иерархическую структуру (рис. 27.1). Список корневых серверов DNS хранится на каждом DNS-сервере (позже мы узнаем, где именно и как его обновлять).

На рис. 27.1 изображен корень системы DNS, домены первого уровня (ru, com, org) и домен второго уровня (firma). Доменов первого уровня (их еще называют TLD, Top Level Domains) довольно много: com, biz, org, info, gov, net, ws, домены стран (ru, ua, uk, ...) и т. д. Понятно, что доменов второго уровня еще больше, не говоря уже о доменах третьего и последующих уровней.

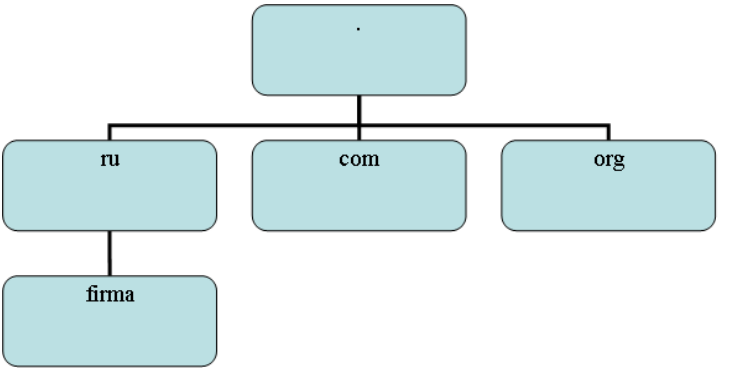


Рис. 27.1. Иерархическая структура DNS

Доменное имя компьютера имеет следующий формат:

```
[имя_компьютера].[домен_N]. ... [домен.TLD]
```

Например,

```
ftp.sales.firma.ru
```

При запросе к DNS-серверу доменное имя обрабатывается в доменном порядке. Сначала наш DNS-сервер посылает запрос к DNS-серверу домена ru — знает ли он что-нибудь о домене firma? DNS-сервер домена ru, если домен firma найден, сообщает IP-адрес сервера DNS домена firma. Потом наш DNS-сервер (или наш собственный сервер имен, или же это сервер имен провайдера) обращается к серверу имен домена firma.ru. Ему нужно узнать, знает ли он что-то о домене sales. Получив IP-адрес DNS-сервера домена sales.firma.ru, мы можем к нему обратиться, чтобы получить IP-адрес компьютера с именем ftp.sales.firma.ru (очевидно, это FTP-сервер отдела продаж какой-нибудь фирмы).

Приведенная схема разрешения доменного имени называется *рекурсивной*, а наш запрос — рекурсивным запросом. Конечно, саму схему я немного упростил, но общий смысл должен быть понятен. Понятно также, и что такой запрос занимает довольно много времени и ресурсов, поэтому целесообразно настроить кэширующий сервер DNS, даже если у вас нет собственного домена. Всю "грязную" работу (то есть рекурсивные запросы) будут делать серверы DNS провайдера, а наш сервер будет только кэшировать результаты запросов — так можно повысить скорость разрешения доменных имен и, следовательно, ускорить работу Интернета в целом. Поэтому кэширующий сервер можно установить не только на шлюзе, но и на домашнем компьютере, где он также будет с успехом выполнять свою функцию.

Настройку сервера DNS мы начнем именно с кэширующего сервера DNS. Во-первых, он настраивается проще, чем полноценный сервер DNS, но зато в процессе его настройки мы познакомимся с основными конфигурационными файлами, и тогда при настройке полноценного DNS-сервера нам будет проще. Во-вторых, не всегда есть необходимость настраивать полноценный DNS-сервер — у вас может быть локальная сеть с выходом в Интернет, но у нее не обязательно должен быть свой собственный домен.

27.2. Кэширующий сервер DNS

Наверняка все мы знакомы с так называемыми "ускорителями" Интернета — программами, якобы помогающими сделать Интернет намного быстрее. Второе название этих программ — оптимизаторы Интернета. Как правило, это Windows-программы, которые распространяются в Интернете за определенную плату. Впрочем, иногда их даже можно скачать бесплатно. В первом случае, если программа распространяется за деньги, "ускоритель" Интернета вообще ничего не делает. Он запускается, пользователь устанавливает параметры, но на самом деле никакого ускорения не происходит. Просто кто-то таким не очень честным образом зарабатывает деньги. Во втором случае, когда программа распространяется бесплатно, также не наблюдается никакого ускорения, а наоборот, заметны падение скорости и повышенный расход трафика. Почему? Да потому что "оптимизаторы" Интернета в большинстве случаев являются вирусами-троянами. Пользователи добровольно устанавливают программу, которая потом передаст секретную информацию (например, ключи от электронного кошелька) злоумышленнику. Помните, что бесплатный сыр — только в мышеловке.

Linux же позволяет организовать настоящий "ускоритель" Интернета. Впрочем, не нужно ожидать, что ваш Интернет будет работать на 70, а то и на все 100%

быстрее, как это обещают оптимизаторы-вирусы. Ускорение будет заключаться в установке кэширующего сервера DNS. Установка DNS-сервера позволяет:

- сократить время разрешения доменных имен, поскольку свой DNS-сервер будет в нашей сети — ответы на запросы о разрешении доменных имен будут приходить от локального сервера, а не от загруженного DNS-сервера провайдера;
- немного сэкономить трафик, поскольку локальный трафик не будет учитываться, чего не скажешь о трафике между вами (вашей сетью) и провайдером.

Итак, кэширующий DNS-сервер — дело нужное, поэтому не будем терять времени и приступим к настройке. Установите пакет bind9.

> ***Примечание***
>
> Обратите внимание, что пакет называется bind9 (Berkley Internet Nameserver Deamon), а сам сервер — named. В старых версиях дистрибутивов данный пакет называется просто bind — скорее всего, этот пакет содержит восьмую версию BIND. Отмечу также, что настройка сервера будет производиться на примере дистрибутива Debian, но в других дистрибутивах при условии использования девятой версии BIND процесс настройки должен быть аналогичен.

После установки пакета bind9 отредактируйте файл /etc/bind/named.conf — это основной файл конфигурации named (листинг 27.1). Комментарии в оригинальном файле написаны, понятно, на английском, но для книги я их перевел на русский язык.

Листинг 27.1. Файл конфигурации /etc/bind/named.conf

```
// Это основной конфигурационный файл DNS-сервера BIND
// См. файл /usr/share/doc/bind9/README.Debian.gz для
// получения информации о структуре конфигурационных файлов BIND
// *ДО* изменения этого файла

// Если вам нужно добавить зоны, сделайте это в
// файле /etc/bind/named.conf.local
include "/etc/bind/named.conf.options";

// Зона корневых серверов имен
zone "." {
       type hint;
       file "/etc/bind/db.root";
};

// Локальная зона localhost
zone "localhost" {
       type master;
       file "/etc/bind/db.local";
};
zone "127.in-addr.arpa" {
       type master;
       file "/etc/bind/db.127";
};
zone "0.in-addr.arpa" {
       type master;
```

```
        file "/etc/bind/db.0";
};
zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};
include "/etc/bind/named.conf.local";
```

В основном конфигурационном файле прописываются корневая и локальная зоны. Локальная зона служит для преобразования имени localhost в IP-адрес 127.0.0.1 и наоборот. И все. Корневая зона содержит список корневых серверов DNS.

Раньше все, что касалось настройки DNS-сервера: и описание зон, и настройки сервера, хранилось в файле named.conf. Сейчас принято в основном конфигурационном файле хранить только описание корневой и локальной зон. Описание опций выносится в файл /etc/bind/named.conf.options, а описание зон, обслуживаемых сервером, — в файл в /etc/bind/named.conf.local.

Вообще-то, собственные зоны вы можете описать в файле named.conf — особой разницы нет. Но если ваш DNS-сервер описывает много зон или одну большую зону (где много компьютеров), тогда целесообразно вынести описание этих зон в файл named.conf.local — вам так будет удобнее настраивать DNS-сервер.

Рассмотрим файл, содержащий опции DNS-сервера: /etc/bind/named.conf.options (листинг 27.2). Опять-таки оригинальный файл содержит комментарии на английском, а я привожу их в переводе на родной язык.

Листинг 27.2. Файл опций /etc/bind/named.conf.options

```
options {
        directory "/var/cache/bind";
// Если "между" вашим DNS-сервером и форвард-серверами находится
// брандмауэр, вы должны его настроить должным образом. О настройке
// брандмауэра можно прочитать по адресу:
// http://www.kb.cert.org/vuls/id/800113

// Здесь прописываются форвард-серверы
// forwarders {
//      0.0.0.0;
// };
        auth-nxdomain no;            # в соответствии с RFC1035
        listen-on-v6 { any; };
};
```

Основной рабочий каталог — /var/cache/bind (задается параметром `directory`). А далее начинается самое интересное. Напомню, что мы сейчас создаем кэширующий сервер, позволяющий ускорить процесс разрешения доменных имен. Но мы можем ускорить работу самого сервера, указав форвард-серверы. В обычном режиме наш сервер сам формирует кэш, но так как сеть у нас относительно небольшая, кэш будет формироваться долго, а насколько долго, зависит от количества запросов, поступаемых от клиентов сети. А если вы установили кэширующий сервер только для

обслуживания своего компьютера, то сначала вообще не почувствуете никакой разницы. Ведь серверу, прежде чем добавить IP-адрес в кэш, нужно его разрешить. Зато уже при втором обращении к доменному имени его IP-адрес будет получен из кэша. Так вот, мы можем использовать кэш от форвард-серверов. Как правило, форвард-серверами выступают серверы провайдера. Как правило, у провайдера уже сформирован довольно большой кэш, который мы можем использовать.

Все, что нужно для использования форвард-сервера, — это добавить его IP-адрес в блок `forwarders`:

```
forwarders {
        # все запросы будут переадресованы к DNS-серверу
      # провайдера 192.168.99.1
        # если с этим сервером что-то случится, то локальный сервер
        # попробует найти ответ в своем кэше или обратится к другим
        # DNS-серверам, которые указываются в /etc/resolv.conf
          192.168.99.1;
    };
```

Параметр `forwarders` задает заключенный в фигурные скобки список IP-адресов, соответствующих DNS-серверам, которым наш DNS-сервер будет переадресовывать запросы, вместо того, чтобы отвечать на них самому. IP-адреса перечисляются через точку с запятой.

Кроме параметра `forwarders` можно использовать параметр `forward`, который может принимать следующие значения:

- `only` — наш DNS-сервер никогда не должен предпринимать попыток обработать запрос самостоятельно;
- `first` — наш сервер должен пытаться сам обработать запрос, если указанные далее параметром `forwarders` серверы DNS не были найдены.

Использование параметра `forward` лишено смысла без использования параметра `forwarders`. Параметр `forward` обычно нужно указывать до параметра `forwarders`:

```
forward first;
    forwarders {
               192.168.99.1;
             192.168.99.2;
    };
```

Где взять адреса форвард-серверов? Обычно они находятся в файле /etc/resolv.conf.

Вот вроде бы и все. Можно приступить к запуску сервера. Но перед этим отмечу, что раз мы создавали кэширующий сервер, то отсутствует блок `controls {}`. Пустой или отсутствующий блок `controls{}` нужен для того, чтобы named не обращал внимания на отсутствие ключа `rndc.key`, который нужен для программы удаленного управления сервером — rndc. Это, правда, не вполне корректно, поскольку для останова сервера нужно будет использовать команду `killall named`, что для нас не существенно, поскольку мы не будем часто его останавливать.

Теперь можно запустить ваш сервер имен:

```
sudo /etc/init.d/bind9 start
```

Поскольку сервер может быть запущен (при установке пакета он запускается автоматически), то его нужно перезапустить:

```
sudo /etc/init.d/bind9 restart
```

Если ошибок в конфигурационных файлах нет, вы получите сообщение:

```
* Starting domain name service... bind9  [ OK ]
```

В Fedora/Mandriva/Ubuntu можно использовать команду `service` для управления сервером:

```
# service bind9 start
# service bind9 restart
# service bind9 stop
```

Если у вас восьмая версия BIND, то сервис, скорее всего, будет называться named, поэтому команды управления будут такими:

```
# service named start
# service named restart
# service named stop
```

Впрочем, имя сервиса зависит от используемого дистрибутива и его версии. Например, у меня был установлен BIND версии 9.2.3, но сервис по-прежнему назывался named — вот посмотрите сами вывод команды `tail /var/log/messages`:

```
# tail /var/log/messages

Aug 8 9:58:16 den named[3140]: starting BIND 9.2.3
Aug 8 9:58:16 den named[3140]: using 1 CPU
Aug 8 9:58:16 den named[3140]: loading configuration from
'/etc/bind/named.conf'
Aug 8 9:58:16 den named[3140]: listening on IPv4 interface lo, 127.0.0.1#53
Aug 8 9:58:16 den named[3140]: listening on IPv4 interface eth0,
192.168.0.1#53
Aug 8 9:58:16 den named[3140]: zone 0.0.127.in-addr.arpa/IN: loaded serial
1997022700
Aug 8 9:58:16 den named[3140]: running
```

Кстати, данный вывод я привел не просто так — последняя строка свидетельствует о том, что сервер запущен. Первая же запись сообщает нам версию BIND, вторая — то, что используется один (1) процессор, далее приводятся: используемый конфигурационный файл, прослушиваемые интерфейсы (lo и eth0) и порт — 53, а также загруженная локальная зона. Число в квадратных скобках (3140) — это PID процесса (идентификатор процесса), "убить" процесс в данном случае можно так:

```
# kill 3140
```

Проверить, работает ли сервер, можно и другим способом, например:

```
# ps -ax | grep named
# ps -ax | grep bind9
```

Теперь осталось в файле /etc/resolv.conf прописать IP-адрес собственного сервера DNS. То же самое нужно сделать на всех остальных компьютерах сети:

```
domain firma.ru
# IP адрес или 127.0.0.1
```

```
nameserver 127.0.0.1
# или IP-адрес DNS-сервера — для остальных компьютеров сети
nameserver 10.0.0.1
```

Протестировать настройки можно с помощью программы nslookup:

```
# nslookup yandex.ru
Server:  localhost.firma.ru
Address:  127.0.0.1
Non-authoritative answer:
Name:    yandex.ru
Address:  213.180.216.200
```

Если вы получили подобный ответ, то это означает, что наш сервер работает нормально. Обратите внимание, что ответ пришел не от DNS-сервера провайдера, а от нашего локального сервера.

В Ubuntu есть небольшая проблема с перезаписью файла resolv.conf. Как только вы его перезапишете, он будет возвращен в исходное состояние при установке соединения или при перезагрузке[1]. Вообще, можно было бы запретить изменение файла с помощью команды `chattr`, зато я докопался до истины. В книге весь процесс для экономии места мы рассматривать не будем, но все желающие смогут ознакомиться с ним по указанному адресу.

27.3. Полноценный DNS-сервер

Теперь можно перейти к настройке полноценного сервера DNS, если, конечно, он вам нужен. Но сначала нужно поговорить о том, что такое *зона*, поскольку полноценный DNS-сервер обслуживает одну или несколько зон. Ошибочно считать зоной обслуживаемый домен — это не так. Давайте разберемся, в чем разница. Домен — это группа компьютеров с одинаковой правой частью доменного имени. Пусть у нас есть домен firma.ru. Компания, которой принадлежит этот домен, довольно большая, поэтому для каждого подразделения пришлось организовать свой домен — sales.firma.ru, dev.firma.ru, orders.firma.ru и т. д. Для управления всем доменом firma.ru (и всеми поддоменами) мы можем использовать или один-единственный DNS-сервер, или же создать независимые серверы для каждого поддомена или только для некоторых поддоменов. Например, основной сервер будет обслуживать только домены firma.ru и sales.firma.ru, а дополнительный сервер — домены dev.firma.ru и orders.firma.ru. Домены firma.ru и sales.firma.ru образуют одну зону, а домены dev.firma.ru и orders.firma.ru — другую. Другими словами, зона — это часть домена, управляемая определенным DNS-сервером. Зона, которая содержит домены низшего уровня, называется *подчиненной зоной* (subordinate zone).

Вот теперь можно приступить к настройке сервера. Первым делом нам нужно настроить удаленное управление сервером, а именно — настроить секцию `controls`, которую мы оставили пустой в предыдущем примере. Выполните команду:

```
# /usr/sbin/rndc-confgen > rndc.conf
```

[1] О моей борьбе с Ubuntu можно прочитать по адресу:
http://www.dkws.org.ua/index.php?page=show&file=a/ubuntu/static-dns-ubuntu9.

Откройте файл rndc.conf в любом текстовом редакторе. Нам нужно выделить и скопировать две директивы: `controls` и `key`:

```
key "rndc-key" {
      algorithm hmac-md5;
      secret "ключ";
};
controls {
# разрешаем "удаленное" управление только с локального компьютера
      inet 127.0.0.1 port 953
      allow { 127.0.0.1; } keys { "rndc-key"; };
};
```

Скопированный блок текста нужно вставить в самое начало файла named.conf. Понятно, что из него нужно удалить пустую директиву `controls`, если она есть в файле.

При настройке кэширующего сервера DNS мы в его конфигурационном файле описали две зоны: корневую и локальную. Теперь нам нужно описать две зоны: прямого и обратного преобразования, которые и будут обслуживать наш домен. Добавьте в файл конфигурации named.conf строки:

```
zone "firma.ru"  {
    type master;
    file "firma.ru";
    notify no;
};

zone "1.0.0.10.in-addr.arpa" {
    type master;
    file "10.0.0.1";
    notify yes;
}
```

Файл firma.ru (он должен находиться в каталоге, заданном директивой `directory`) используется для прямого преобразования, то есть для преобразования доменных имен в IP-адреса. В листинге 27.3 представлен пример этого файла.

Листинг 27.3. Пример файла прямого преобразования

```
@  IN   SOA    server.firma.ru.      hostmaster.firma.ru. (
       20040603       ; серийный номер (можно узнать в
                      ; файлах с примерами)
       3600           ; обновление каждый час
       3600           ; повтор каждый час
       3600000        ; время хранения информации 1000 часов
       3600           ; TTL-записи
)
   IN NS              server.firma.ru.
```

```
   IN A       10.0.0.1
   IN MX              100    server.firma.ru.
www    IN CNAME      server.firma.ru.
ftp    IN CNAME      server.firma.ru.
mail   IN CNAME      server.firma.ru.
c2 IN A       10.0.0.2
c3 IN A       10.0.0.2
localhost.    IN A   127.0.0.1
```

Разберемся, что означают записи этого файла. Первым делом, обратите внимание на то, что в конце каждого доменного имени ставится точка, — это чтобы сервер не приписывал имя домена (firma.ru) к доменному имени. Если лень писать имя домена, тогда можно просто указывать имя компьютера (`server` вместо `server.firma.ru`), но тогда не нужно ставить точку в конце доменного имени.

Разберемся с записью `IN SOA`. Она описывает начало полномочий (Start Of Authority, SOA). Первое имя после SOA — это имя данного компьютера (на котором запущен DNS-сервер). В нашем случае это server.firma.ru. Затем следует e-mail администратора сервера, но поскольку символ @ зарезервирован, то вместо него используется точка. Остальные элементы записи SOA прокомментированы в листинге.

Запись NS (`IN NS`) задает имя сервера доменных имен, а запись `A` — его IP-адрес. Запись `MX` используется для задания почтового сервера. Как мы видим, в роли почтового сервера используется все тот же наш server.firma.ru. 100 — это приоритет почтового сервера. Приоритет используется, если указано два (или более почтовых сервера). Чем меньше число, тем выше приоритет:

```
   IN MX       100   mail1
   IN MX       150   mail2
```

Запись `CNAME` используется для определения канонических имен, то есть псевдонимов. Как мы видим, к нашему серверу server.firma.com можно обратиться по следующим именам: www.firma.ru, ftp.firma.ru, mail.firma.ru.

Далее описаны два компьютера: c2.firma.ru (мы не ставили точку после c2, поэтому firma.ru сервер "допишет" автоматически) и c3.firma.ru, с IP-адресами 10.0.0.2 и 10.0.0.3 соответственно.

Последняя запись — это определение имени localhost, желательно не забыть о нем.

Теперь пора приступить к рассмотрению файла обратного соответствия, который представлен в листинге 27.4. Напомню, что этот файл используется для преобразования IP-адресов в доменные имена.

Листинг 27.4. Пример файла обратного преобразования

```
@  IN   SOA    server.firma.ru.      hostmaster.firma.ru. (
        20040603       ; серийный номер (можно узнать в файлах с примерами)
        3600           ; обновление каждый час
        3600           ; повтор каждый час
        3600000        ; время хранения информации 1000 часов
```

```
        3600          ; TTL записи
)
@  IN   NS     server.firma.ru
1  IN   PTR    server.firmaru
2  IN   PTR    c2.firma.ru
3  IN   PTR    c3.firma.ru
```

В данном файле, если вы успели заметить, можно полностью не указывать IP-адрес, но требуется полностью указывать доменное имя (точки в конце доменного имени не нужны). Если же вам хочется указать IP-адрес полностью, тогда следует указывать его в обратном порядке, например:

```
2.0.0.10    IN      PTR     c2.firma.ru
```

Вот, практически, и все. Можно в целях защиты сервера добавить в блок `options` (конфигурационный файл named.conf.options) директиву `allow-query`:

```
allow-query {
10.0.0.0/24;
localhost;
}
```

Блок `allow-query` разрешает запросы к серверу только узлам подсети 10.0.0.0 и от узла localhost. Узлы других подсетей не смогут использовать наш сервер. Когда вы настраиваете DNS-сервер, который будет работать в локальной сети (обслуживать только клиентов нашей локальной сети), то, по большому счету, блок `allow-query` вам не нужен. Но когда вы настраиваете DNS-сервер провайдера или же сервер, работающий в сети с реальными IP-адресами, то директива `allow-query` просто необходима, чтобы "чужие" узлы не смогли использовать наш сервер.

Полный файл конфигурации полноценного DNS-сервера для домена firma.ru представлен в листинге 27.5. Описание зон и опций я не выносил в файлы named.conf.options и named.conf.local для наглядности — чтобы вы в одном листинге увидели все настройки сервера.

Листинг 27.5. Полная версия файла конфигурации named.conf

```
key "rndc-key" {
      algorithm hmac-md5;
      secret "ключ";
};
controls {
      inet 127.0.0.1 port 953
              allow { 127.0.0.1; } keys { "rndc-key"; };
};

options {
              directory "/etc/bind";

allow-query {
10.0.0.0/24;
```

```
localhost;
}

};

zone "." in {
        type hint;
        file "db.root";
};
zone "0.0.127.in-addr.arpa" in {
        type master;
        file "db.127";
};
zone "localhost" {
        type master;
        file "db.local";
};
zone "255.in-addr.arpa" {
        type master;
        file "db.255";
};

zone "firma.ru"  {
   type master;
   file "firma.ru";
   notify no;
};

zone "1.0.0.10.in-addr.arpa" {
   type master;
   file "10.0.0.1";
   notify yes;
}
```

После настройки сервер нужно перезапустить:

```
# service named restart
```

27.4. Вторичный DNS-сервер

В идеале для поддержки домена должно быть выделено два сервера: первичный и вторичный. Вторичный используется для подстраховки, если вдруг с первичным что-то случится (например, банальная перезагрузка администратором).

Вторичный сервер DNS описывается аналогично первичному, но несколько иначе указывается зона домена:

```
zone "firma.ru"  {
    type slave;
```

```
        file "firma.ru";
        masters { 10.0.0.1; };
};
```

Как видим, устанавливается тип сервера — подчиненный (`slave`), а в блоке `masters` описываются первичные серверы (у нас он один).

В файл конфигурации первичного сервера надо добавить директиву `allow-transfer`, в которой указать DNS-серверы, которым разрешен трансфер зоны, то есть все вторичные серверы:

```
options {
...
allow-transfer { 10.0.0.2; };
}
```

27.5. Обновление базы данных корневых серверов

Чтобы база данных корневых серверов всегда была актуальной, ее нужно регулярно обновлять. Получить ее можно по адресу ftp://ftp.internic.net/domain/named.root, а обновить — с помощью трех команд:

```
wget ftp://ftp.internic.net/domain/named.root
sudo cp named.root /etc/bind/db.root
sudo /etc/init.d/bind9 restart
```

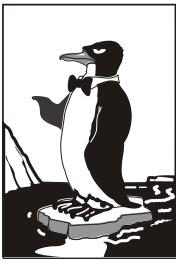
В листинге 27.6 содержится самая актуальная на момент написания этих строк версия файла named.root.

Листинг 27.6. Файл named.root (db.root)

```
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC
;       under anonymous FTP as
;           file                /domain/named.cache
;           on server           FTP.INTERNIC.NET
;       -OR-                    RS.INTERNIC.NET
;
;       last update:    Jun 17, 2010
;       related version of root zone:   2010061700
;
; formerly NS.INTERNIC.NET
;
.                       3600000  IN  NS    A.ROOT-SERVERS.NET.
```

```
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
A.ROOT-SERVERS.NET.      3600000      AAAA  2001:503:BA3E::2:30
;
; FORMERLY NS1.ISI.EDU
;
.                        3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     192.228.79.201
;
; FORMERLY C.PSI.NET
;
.                        3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.      3600000      A     192.33.4.12
;
; FORMERLY TERP.UMD.EDU
;
.                        3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.      3600000      A     128.8.10.90
;
; FORMERLY NS.NASA.GOV
;
.                        3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.      3600000      A     192.203.230.10
;
; FORMERLY NS.ISC.ORG
;
.                        3600000      NS    F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET.      3600000      A     192.5.5.241
F.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:2F::F
;
; FORMERLY NS.NIC.DDN.MIL
;
.                        3600000      NS    G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET.      3600000      A     192.112.36.4
;
; FORMERLY AOS.ARL.ARMY.MIL
;
.                        3600000      NS    H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET.      3600000      A     128.63.2.53
H.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:1::803F:235
;
; FORMERLY NIC.NORDU.NET
;
.                        3600000      NS    I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET.      3600000      A     192.36.148.17
```

```
I.ROOT-SERVERS.NET.      3600000      AAAA  2001:7FE::53
;
; OPERATED BY VERISIGN, INC.
;
.                        3600000      NS    J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET.      3600000      A     192.58.128.30
J.ROOT-SERVERS.NET.      3600000      AAAA  2001:503:C27::2:30
;
; OPERATED BY RIPE NCC
;
.                        3600000      NS    K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET.      3600000      A     193.0.14.129
K.ROOT-SERVERS.NET.      3600000      AAAA  2001:7FD::1
;
; OPERATED BY ICANN
;
.                        3600000      NS    L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET.      3600000      A     199.7.83.42
L.ROOT-SERVERS.NET.      3600000      AAAA  2001:500:3::42
;
; OPERATED BY WIDE
;
.                        3600000      NS    M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET.      3600000      A     202.12.27.33
M.ROOT-SERVERS.NET.      3600000      AAAA  2001:DC3::35
; End of File
```



Глава 28

DHCP-сервер

28.1. Протокол динамической конфигурации узла

DHCP (Dynamic Host Configuration Protocol, протокол динамической конфигурации узла) используется для автоматической настройки узлов сети. С помощью DHCP компьютер, подключенный к сети, в которой есть DHCP-сервер, может получить IP-адрес, маску сети, IP-адрес шлюза, адреса серверов DNS и другие сетевые параметры.

Особенно удобно использовать DHCP в средних и больших сетях. Вы только представьте, что у вас есть, скажем, 20 компьютеров. Если каждому компьютеру назначать IP-адрес статически, то вам придется подойти к каждому компьютеру и указать его IP-адрес. Заодно вам нужно будет ввести IP-адрес сети, IP-адрес шлюза и адреса серверов DNS. Понятно, что эту процедуру нужно выполнить разово — при настройке сети. Но если через некоторое время конфигурация сети изменится (например, вы поменяете провайдера), и надо будет изменить IP-адреса DNS-серверов, то вам придется все повторить заново — подойти к каждому компьютеру и прописать DNS-серверы.

Если же потратить полчаса на настройку DHCP-сервера, можно будет централизованно управлять конфигурацией сети. Вам стоит изменить IP-адрес DNS-сервера в конфигурационном файле DHCP-сервера — на остальных компьютерах сети новые IP-адреса DNS-серверов "пропишутся" автоматически. Удобно? Я тоже так думаю.

Для установки DHCP-сервера вам достаточно установить пакет dhcp (или dhcp3-server в Ubuntu). DHCP-клиенты входят в состав Linux и Windows, поэтому их устанавливать отдельно не нужно.

28.2. Конфигурационный файл DHCP-сервера

Конфигурационный файл DHCP-сервера называется /etc/dhcpd.conf. Пример этого файла вы можете найти в /usr/share/doc/dhcp-<версия>/dhcpd.conf.sample.

Относительно файла конфигурации нужно сделать два замечания:

- директивы не чувствительны к регистру символов, то есть вы можете написать как `option`, так и `OPTION`, но принято писать строчными буквами;
- комментарии начинаются с символа решетки `#`.

В начало файла конфигурации нужно поместить одну из директив:

```
ddns-update-style ad-hoc;
```

или

```
ddns-update-style interim;
```

Разберемся, о чем речь. Сейчас существуют две схемы обновления DNS: непосредственное обновление (ah-doc) и предварительное взаимодействие DHCP-DNS (interim). Вторая схема пока не утверждена комитетом по техническому развитию Интернета, но уже успешно используется, и разработчики DHCP рекомендуют применять именно ее. Тут выбирать вам: или использовать старую схему взаимодействия (первая директива), или выбрать более перспективную (вторая директива).

По сути, весь конфигурационный файл DHCP-сервера будет состоять из директивы `ddns-update-style` и блочной директивы `subnet`, описывающей вашу сеть.

Рассмотрим пример объявления сети 192.168.1.0 (листинг 28.1).

Листинг 28.1. Описание сети 192.168.1.0

```
subnet 192.168.1.0 netmask 255.255.255.0 {
# шлюз по умолчанию
        option routers                  192.168.1.1;
# маска сети — этот параметр будет передан всем компьютерам сети
        option subnet-mask              255.255.255.0;
# наш домен
        option domain-name              "example.ru";
# IP-адрес сервера DNS
        option domain-name-servers       192.168.1.1;

# диапазон IP-адресов: компьютерам нашей сети будут присваиваться
# IP-адреса из этого диапазона
range 192.168.1.10 192.168.1.100;
}
```

Если у нас есть большая сеть и есть несколько подсетей, то все подсети (директива `subnet`) должны быть описаны в одной директиве `shared-network`. При этом все общие для подсетей параметры: описание маршрутизаторов, DNS-серверов, доменное имя — выносятся за пределы директив `subnet` (листинг 28.2).

Листинг 28.2. Большая сеть и ее подсети

```
shared-network имя_нашей_сети {
# описываем глобальные для всех подсетей параметры

# домен
    option domain-name                  "example.ru";
# серверы DNS
    option domain-name-servers          ns1.isp.com, ns2.isp.com;
# шлюз по умолчанию
```

```
    option routers                          192.168.0.1;

# описываем подсети 192.168.1.0 и 192.168.2.0

    subnet 192.168.1.0 netmask 255.255.252.0 {
        range 192.168.1.10 192.168.1.254;
    }
    subnet 192.168.2.0 netmask 255.255.252.0 {
        range 192.168.2.10 192.168.2.254;
    }
}
# конец директивы shared-network
```

28.3. База данных аренды

DHCP-сервер назначает IP-адрес компьютеру не на все время, а только на некоторое, называемое *временем аренды*. По его истечении компьютеру будет назначен другой IP-адрес.

Время аренды регулируется директивами `default-leased-time` и `max-leased-time`, но обычно не нужно изменять значения этих директив, потому что значения по умолчанию вполне приемлемы.

База данных аренды, то есть информация, кому и какой IP-адрес был назначен, находится в файле /var/lib/dhcp/dhcpd.leases. В этом файле содержится следующая информация: уникальный MAC-адрес сетевого адаптера компьютера (аппаратный адрес), назначенный IP-адрес, дата и время окончания аренды и др.

Базу данных аренды нельзя редактировать вручную, ее можно только просматривать.

28.4. Полный листинг конфигурационного файла

Окончательный вариант конфигурационного файла для подсети 192.168.1.0 представлен в листинге. 28.3.

Листинг 28.3. Окончательный вариант конфигурационного файла DHCP-сервера

```
# схема взаимодействия с DNS
ddns-update-style ad-hoc;

subnet 192.168.1.0 netmask 255.255.255.0 {

# шлюз по умолчанию
        option routers                  192.168.1.1;
# маска сети — этот параметр будет передан всем компьютерам сети
        option subnet-mask              255.255.255.0;
# наш домен
        option domain-name              "example.ru";
# IP-адрес сервера DNS
```

```
        option domain-name-servers       192.168.1.1;

# диапазон IP-адресов: компьютерам нашей сети будут присваиваться
# IP-адреса из этого диапазона
range 192.168.1.10 192.168.1.100;
}
```

28.5. Управление сервером DHCP

Для запуска, перезапуска и останова сервера можно использовать команду `service`:

```
service dhcpd start
service dhcpd restart
service dhcpd stop
```

28.6. Настройка клиентов

Все клиенты вашей сети (разумеется, кроме серверов сети, у которых должны быть постоянные IP) должны быть настроены на автоматическое получение IP-адреса (рис. 28.1) и IP-адресов DNS-серверов. Хотя в большинстве случаев Windows и так настроена на автоматическое получение IP-адреса и другой сетевой информации, поэтому настраивать компьютеры вашей локальной сети вам не придется.

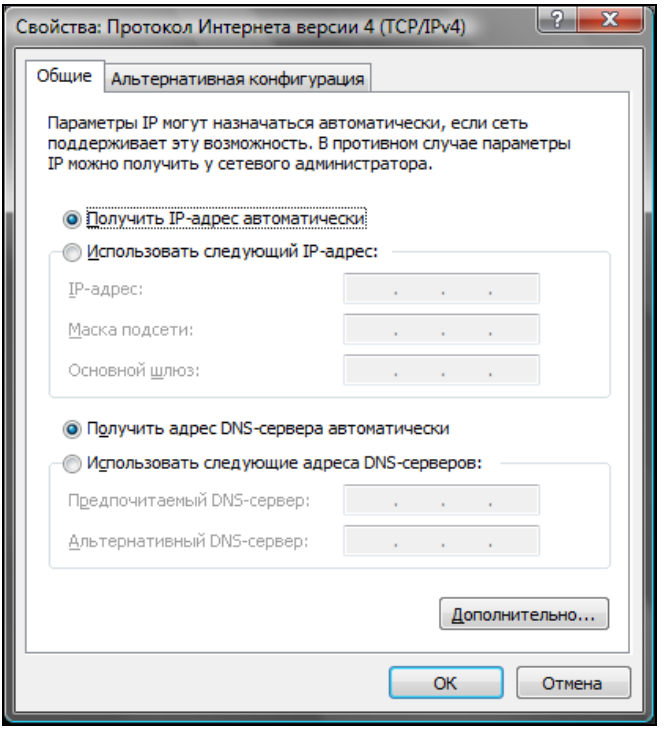
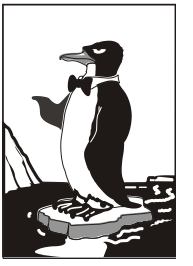


Рис. 28.1. Настройка Windows-клиента



Глава 29

Web-сервер. Связка Apache + PHP + MySQL

29.1. Самый популярный Web-сервер

Apache — это Web-сервер с открытым исходным кодом. История его развития началась в 1995 году — тогда Apache был всего лишь "заплаткой", устраняющей ошибки популярного в то время Web-сервера NCSA HTTPd 1.3. Считается, что отсюда произошло и название Apache (a patchy — заплатка). Сейчас Apache — самый популярный Web-сервер в Интернете: в апреле 2007 года было подсчитано, что он установлен на 58% Web-серверов Интернета.

Основные достоинства Apache — надежность, безопасность и гибкость настройки. Apache позволяет подключать различные модули, добавляющие в него новые возможности, — например, можно подключить модуль, обеспечивающий поддержку PHP или любого другого Web-ориентированного языка программирования.

Но есть и недостатки — без этого никак, всегда есть обратная сторона медали. Основной недостаток — отсутствие удобного графического интерфейса администратора. Да, настройка Apache осуществляется путем редактирования его конфигурационного файла. В Интернете можно найти простые конфигураторы Apache, но их возможностей явно не хватает для настройки всех функций Web-сервера.

29.2. Установка Web-сервера и интерпретатора PHP. Выбор версии

Вы можете установить одну из трех версий Apache: Apache 1.3.34, Apache 2 и Apache 2.2. С одной стороны, версия Apache 2.2 более новая и современная, Apache 2 — стабильная и уже проверенная. С другой, версии 1.3.*x* все еще поддерживаются и имеются в репозиториях некоторых дистрибутивов.

Однако я рекомендую установить Apache 2. И дело тут даже не в том, что эта версия более новая. Ради эксперимента я установил сначала версию 1.3.34, интерпретатор PHP4 и еще ряд дополнительных пакетов объемом 14 Мбайт. Оказалось, что в одной из библиотек или конфигурационном файле этого набора была ошибка — что я ни делал, поддержки PHP не было. А поддержка PHP очень нужна на современном Web-сервере! Поэтому я сразу установил вторую версию Apache и PHP5. Кроме того, вторая версия проще в настройке, поддерживает протокол IPv6, многопоточность и умеет выводить сообщения об ошибках на разных языках.

Что же касается версии 2.2, то она еще не достаточно "обкатанная". Если нужно настроить корпоративный сервер, вполне хватит версии 2.0, а если хочется поэкспериментировать с новой версией — попробуйте 2.2.

Запустите менеджер пакетов (например, Synaptic, используемый в Ubuntu). Произведите поиск пакета apache. Выберите пакет apache2. Менеджер пакетов сообщит вам, что нужно установить дополнительные пакеты (рис. 29.1).

Чтобы сразу "убить двух зайцев", выберите еще и пакет php5. Он устанавливает PHP5 и добавляет его поддержку в Apache. Опять менеджер предложит установить дополнительные пакеты, но на этот раз для PHP (рис. 29.2).

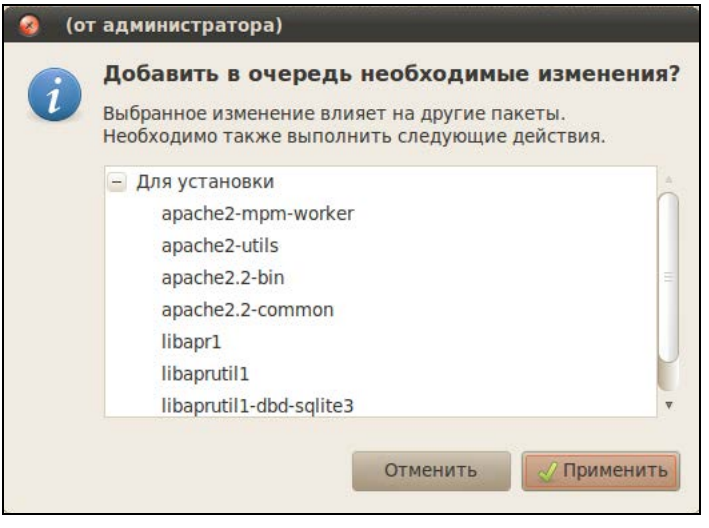


Рис. 29.1. Дополнительные пакеты для Apache

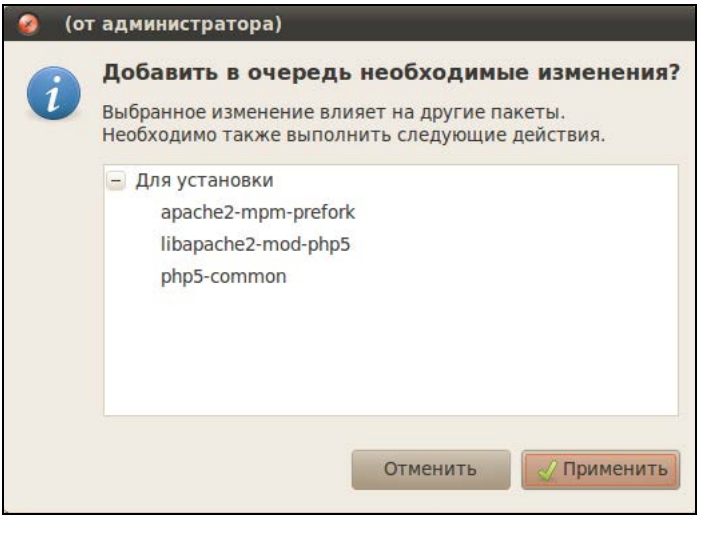


Рис. 29.2. Дополнительные пакеты для PHP

Нажмем кнопку **Применить**, и машина установит все выбранное. После этого рекомендуется установить и следующие пакеты (их можно найти по запросу `php`):

- ph5-cli — интерпретатор PHP, работающий в режиме командной строки (command-line interpreter);
- php5-imap — поддержка протоколов POP/IMAP для PHP;
- php5-gd — поддержка графических функций PHP;
- php5-mysql — поддержка функций для работы с базой данных MySQL.

Если вы выбрали PHP4, тогда вам нужно установить эти же пакеты, но для PHP4 (php4-*). Необходимые дополнительные пакеты будут установлены автоматически, об этом позаботится менеджер пакетов. Просмотрите весь список пакетов, возможно, нужные вам найдутся.

29.3. Тестирование настроек

Теперь протестируем Web-сервер. По идее, после установки сервер должен запуститься автоматически. Но в некоторых дистрибутивах его нужно запустить вручную (как это сделать, рассказано в *разд. 29.5*).

Запустите сервер или убедитесь, что он запущен *(см. разд. 29.5)*. Откройте браузер и введите адрес:

```
http://localhost
```

Должна открыться страница, изображенная на рис. 29.3.

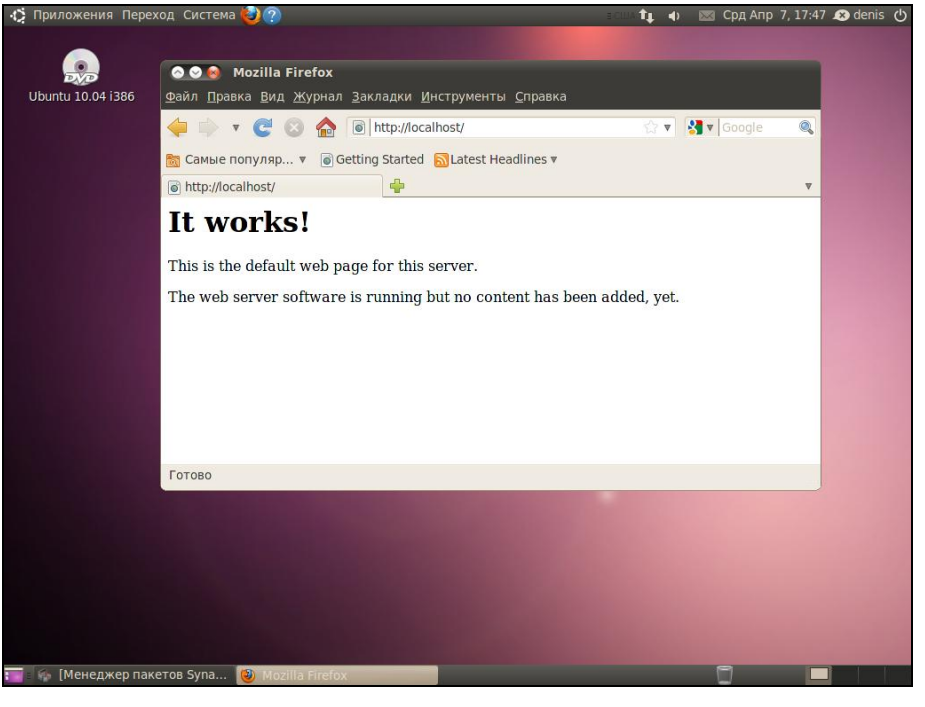


Рис. 29.3. Тестовая страница Apache

После этого протестируем поддержку PHP. Поместите в каталог /var/www/ файл test.php (листинг 29.1).

Листинг 29.1. Файл test.php

```
<?

phpinfo();

?>
```

Чтобы создать файл в этом каталоге, нужны права root. После того как файл будет создан, введите в строке браузера следующий адрес:

```
http://localhost/test.php
```

В окне браузера вы должны увидеть информацию о своем сервере и PHP (рис. 29.4).

Примечание

Если вместо отображения тестовой странички, изображенной на рис. 29.4, браузер предлагает вам сохранить файл test.php, перезапустите Web-сервер *(см. разд. 29.5)*.

Как вы уже догадались, каталог /var/www является корневым для вашего сервера. Если создать в нем файл test.html, то он будет доступен по адресу **http://localhost/test.html**.

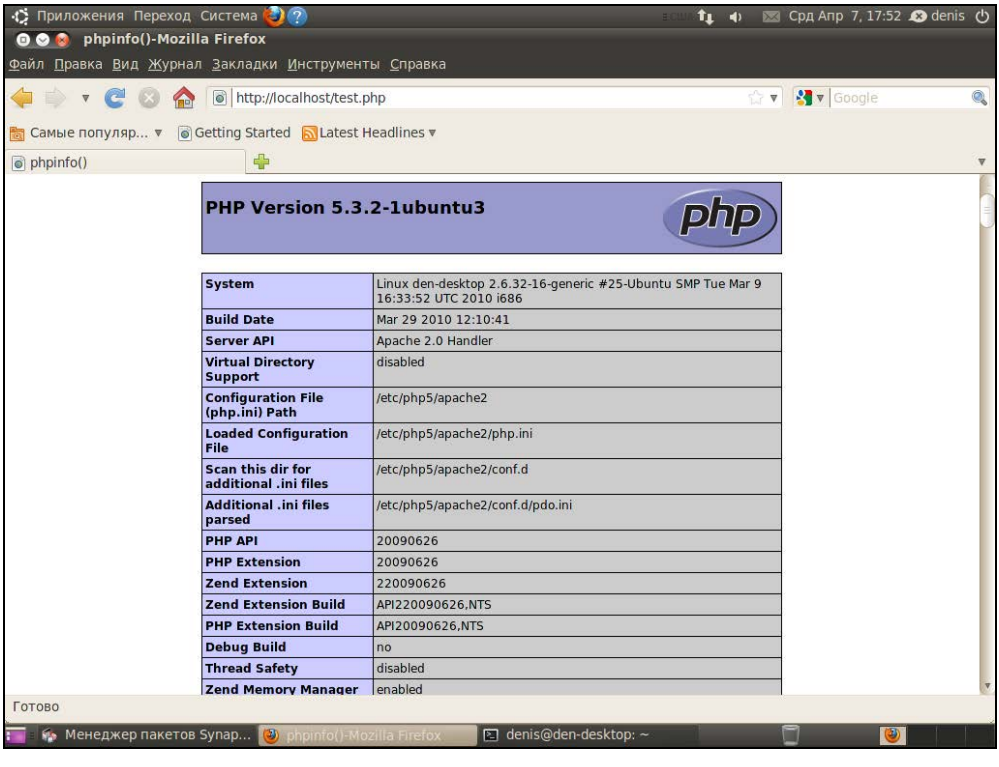


Рис. 29.4. Тестовый сценарий

29.4. Файл конфигурации Web-сервера

29.4.1. Базовая настройка

В зависимости от версии Apache и вашего дистрибутива, конфигурационные файлы Apache могут находиться в следующих каталогах: /etc/apache, /etc/apache2, /etc/httpd или /etc/httpd2. Основные конфигурационные файлы называются httpd.conf, httpd2.conf или apache.conf и apache2.conf. Название каталогов и файлов, содержащих слово "apache", характерно для дистрибутивов Debian и Ubuntu, а содержащих слово "httpd" — для Mandriva/Fedora. В любом случае найти конфигурационные файлы не сложно: ищите или apache, или httpd — и не промахнетесь!

> ***Внимание!***
>
> После каждого изменения конфигурационных файлов сервера его нужно перезапустить *(см. разд. 29.5)*!

Раньше все настройки хранились в одном огромном файле конфигурации. Сейчас этот файл чаще содержит Include-инструкции подключения других файлов конфигурации (более компактных). Все это сделано для удобства администраторов — проще работать с несколькими компактными файлами, чем с одним огромным.

Первым делом откройте конфигурационный файл (для определенности будем считать, что он называется httpd2.conf) и найдите директиву:

```
#ServerName new.host.name
```

Нужно ее раскомментировать и указать имя сервера, которое будут задавать пользователи в строке браузера. Данное имя должно быть зарегистрировано в DNS-сервере вашей сети (или указано в файле /etc/hosts каждого компьютера сети). Обычно здесь указывается имя компьютера, например:

```
ServerName user-desktop
```

После этого можно будет обращаться к серверу по адресу **http://user-desktop/**.

29.4.2. Самые полезные директивы файла конфигурации

Понятно, что для полноценной настройки сервера одной директивы `ServerName` недостаточно. В табл. 29.1 приведены самые полезные директивы файла конфигурации Apache. Нужно отметить, что в таблице не рассматриваются некоторые директивы (например, `Port`, `BindAddress`), которые не используются во второй версии Apache.

Таблица 29.1. *Директивы файла конфигурации*

Директива	Описание
`ServerName имя`	Задает имя Web-сервера, имя должно быть зарегистрированным на DNS-сервере, то есть обычно это доменное имя сервера
`ServerAdmin e-mail`	Задает e-mail администратора сервера
`ServerRoot каталог`	Определяет каталог с конфигурационными файлами сервера

Таблица 29.1 *(продолжение)*

Директива	Описание
`PidFile файл`	Определяет имя файла, в котором будет храниться PID исходного процесса Web-сервера. Обычно изменять эту директиву не нужно
`DocumentRoot каталог`	Позволяет задать каталог, в котором хранятся документы Web-сервера — это корневой каталог документов. Обычно это /var/www
`StartServers N, MaxSpareServers N, MinSpareServers N, MaxClients N`	Директивы, непосредственно влияющие на производительность сервера. Мы их рассмотрим отдельно в *разд. 29.6*
`KeepAlive On | Off, KeepAliveTimeout N`	Управляют постоянными соединениями, будут рассмотрены в *разд. 29.6*
`DirectoryIndex список`	Задает имена файлов, которые могут использоваться в качестве главной страницы (индекса). Значение по умолчанию `index.html index.cgi index.pl index.php index.xhtml`
`HostnameLookups On|Off`	Если директива включена (`On`), то IP-адрес клиента перед записью в журнал будет разрешен (то есть Web-сервер вычислит доменное имя клиента перед записью информации о попытке доступа в журнал). Выключение (`Off`) этой опции позволяет повысить производительность сервера, поскольку не нужно тратить время на разрешение IP-адресов в доменные имена
`ErrorLog файл`	Задает журнал ошибок
`TransferLog файл`	Задает журнал обращений к серверу
`Timeout N`	Тайм-аут в секундах (время, на протяжении которого сервер будет ждать возобновления прерванной попытки передачи данных)
`User пользователь` `Group группа`	Директивы `User` и `Group` задают имя пользователя и группы, от имени которых запускается Web-сервер
`FancyIndexing on | off`	Если пользователь в запросе не укажет имя документа, а только каталог, но в нем не окажется главной страницы, заданной директивой `DirectoryIndex`, сервер передаст пользователю оглавление каталога. Данная директива определяет, в каком виде будет передано оглавление каталога: в более красивом, со значками каталогов и описаниями файлов (значение `On`), или в более простом (`Off`)
`AddIcon картинка список`	Если `FancyIndexing` включена, то `AddIcon` позволяет связать графическую картинку с типом файла, например, `AddIcon /images/graphics.gif .gif, .jpeg, .bmp, .png, .tiff`
`DefaultIcon картинка`	Позволяет задать картинку по умолчанию (`AddIcon`, `FancyIndexing`)

Таблица 29.1 *(окончание)*

Директива	Описание
`ErrorDocument N файл`	Позволяет задать файл, содержащий сообщение об ошибке, для ошибки с номером N, например: ErrorDocument 404 /errors/file_not_found.html
`Directory, Limit, Location, Files`	Это так называемые *блочные* директивы, которые нельзя описать одной строкой, поэтому о них мы поговорим отдельно (*см. разд. 29.4.3*)

29.4.3. Директивы *Directory*, *Limit*, *Location*, *Files*

Рассмотрим сначала блочные директивы `Directory` и `Limit`.

- С помощью блочной директивы `Directory` можно установить параметры отдельного каталога. Внутри директивы `Directory` могут использоваться директивы `AllowOverride`, `Limit`, `Options`. Вот пример определения параметров корневого сервера:

```
<Directory />
AllowOverride None
Options None
</Directory>
```

Значения `None` для обеих директив (`AllowOverride` и `Options`) считаются самыми безопасными. `None` для `AllowOverride` запрещает использование файлов .htaccess, которые могут переопределять директивы конфигурационного файла Apache. К тому же, `AllowOverride None` позволяет повысить производительность сервера.

Допустимые опции каталога (значения директивы `Options`) указаны в табл. 29.2.

Таблица 29.2. *Опции каталога*

Опция	Описание
`None`	Запрещены все опции
`All`	Все опции разрешены
`Indexes`	Если указана эта опция, при отсутствии файла, заданного `DirectoryIndex`, будет выведено оглавление каталога. Если `Options` установлена в `None` (или `Indexes` не указана в списке опций), то оглавление каталога выводиться не будет
`Includes`	Разрешает использование SSI (Server Side Includes)
`IncludesNoExec`	Более безопасный режим SSI: разрешает SSI, но запрещает запускать из включений внешние программы
`ExecCGI`	Разрешает выполнение CGI-сценариев
`FollowSymLink`	Разрешает использование символических ссылок. Довольно опасная опция, поэтому лучше ее не использовать

- Блочная директива `Limit` позволяет ограничить доступ. Внутри этой директивы можно использовать директивы `order`, `deny` и `allow` (вообще есть еще и директива `require`, но она очень редко используется). Директива `order` задает порядок выполнения директив `deny` и `allow`:

```
# сначала запретить, потом разрешить
order deny, allow
# сначала разрешить, потом запретить
order allow, deny
```

  Директивы `allow` и `deny` нужно использовать так:

```
# запрещаем доступ всем
deny from all
# разрешаем доступ только нашей сети
allow from firma.ru
```

Пример использования директив `Directory` и `Limit` представлен в листинге 29.2.

Листинг 29.2. Фрагмент файла конфигурации Apache

```
<Directory />
AllowOverride None
Options None
<Limit>
       order deny, allow
       # запрещаем доступ всем
       deny from all
       # разрешаем доступ только нашей сети
       allow from firma.ru
</Limit>

</Directory>
```

В качестве параметра директиве `Limit` можно передать метод передачи данных (`GET`, `POST`), например:

```
<Limit GET>
<Limit POST>
```

Теперь обратимся к блочным директивам `Location` и `Files`.

- Директива `Location` очень похожа на директиву `Directory`. Только если `Directory` ограничивает доступ к каталогу, то `Location` предназначена для ограничения доступа к отдельным URL сервера:

```
<Location URL>
директивы ограничения доступа
</Location>
```

  К директивам ограничения доступа относятся `order`, `deny`, `allow`.

- Директива `Files` предназначена для ограничения доступа к отдельным файлам:

```
<Files файл>
директивы ограничения доступа
</Files>
```

Вы можете указать как отдельный файл, так и регулярное выражение, которому должны соответствовать файлы:

```
# запрещаем доступ к файлу privat.html всем, кроме нашей сети
<Files privat.html>
order deny, allow
deny from all
allow from firma.ru
</Files>

# запрещаем доступ к файлам .ht* всем
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>
```

Мы рассмотрели все самые полезные директивы конфигурационного файла Apache. Напомню, что директивы, непосредственно влияющие на производительность сервера, будут рассмотрены в *разд. 29.6*.

29.5. Управление запуском сервера Apache

Для управления Web-сервером можно использовать команду `service`:

`# service httpd start` — запуск сервера

`# service httpd stop` — останов сервера

`# service httpd restart` — перезапуск сервера

Понятно, что Web-сервер запускается автоматически, поэтому каждый день вам не придется вводить команду `service httpd start`.

В новых версиях Ubuntu есть команда `serice`, поэтому управлять Apache можно, как показано ранее. В старых версиях Ubuntu и Debian нет команды `service`, поэтому управлять Apache можно так:

```
sudo /etc/init.d/apache2 start
sudo /etc/init.d/apache2 stop
sudo /etc/init.d/apache2 restart
```

Примечание

Напомню, что в разных дистрибутивах сервис Apache называется по-разному: или apache2 или httpd.

29.6. Пользовательские каталоги

Если вы когда-нибудь настраивали сервер Apache, то наверняка знакомы с директивой `UserDir`. Я специально ее не описал в табл. 29.1, потому что она заслуживает отдельного разговора.

По умолчанию директива `UserDir` отключена:

```
UserDir disabled
```

Включить ее можно, указав вместо `disabled` любое другое значение, обычно указывается значение `public_html`:

```
UserDir public_html
```

Затем в пользовательском каталоге /home/<имя> создается каталог public_html, в него помещаются HTML/PHP-файлы персонального сайта пользователя. Обращение к сайту пользователя происходит по URL:

http://имя_сервера/~имя_пользователя

Например, если при включенной директиве `UserDir` вы поместили в каталог /home/den/public_html файл report.xml, то обратиться к нему можно по адресу:

http://server/~den/report.xml.

Недавно настраивая сервер на базе openSUSE, столкнулся с небольшой проблемой. Ранее, во времена огромного конфигурационного файла, достаточно было раскомментировать эту директиву в конфигурационном файле. Сейчас, когда конфигурация сервера состоит из нескольких небольших файлов, добавление этой опции в основной конфигурационный файл не привело ни к каким изменениям. Оказалось, опцию `UserDir` нужно добавить (точнее просто раскомментировать) в файл /etc/apache2/mod_userdir.conf. После этого нужно добавить следующую строку в самый конец файла /etc/apache2/default-server.conf:

```
Include /etc/apache2/mod_userdir.conf
```

После всего этого нужно перезапустить сервер.

29.7. Установка сервера баз данных MySQL

29.7.1. Установка сервера

Для организации связки Apache + PHP + MySQL нам осталось установить последний компонент — сервер баз данных MySQL. Для установки MySQL-сервера установите следующие пакеты:

- mysql-server-5.0;
- mysql-client-5.0;
- mysql-admin.

Первый пакет содержит последнюю версию MySQL-сервера (на данный момент это пятая версия), во втором пакете находится MySQL-клиент, то есть программа, которая будет подключаться к MySQL-серверу, передавать ему SQL-запросы и отобра-

жать результат их выполнения. Третий пакет содержит программу для администрирования MySQL-сервера. Все необходимые дополнительные пакеты будут установлены автоматически.

29.7.2. Изменение пароля root и добавление пользователей

Сразу после установки пакетов введите следующие команды:

```
# mysql_install_db
# mysqladmin -u root password ваш_пароль
```

Примечание

В процессе выполнения команды `mysql_install_db` вы можете получить сообщение:
[ERROR] /usr/libexec/mysqld: Can't find file: './mysql/help_relation.frm' (errno: 13)
Поможет команда `chown -R mysql /var/lib/mysql`. После ее выполнения нужно заново выполнить команду `mysql_install_db`.

Первая команда (`# mysql_install_db`) создаст необходимые таблицы привилегий, а вторая (`# mysqladmin -u root password ваш_пароль`) — задаст пароль пользователя root для сервера MySQL. Этот пароль вы будете использовать для администрирования сервера (данный пароль может и должен отличаться от того, который вы используете для входа в систему). Для обычной работы с сервером рекомендуется создать обычного пользователя. Для этого введите команду:

```
mysql -u root -p mysql
```

Программа mysql является клиентом MySQL-сервера. В данном случае она должна подключиться к базе данных `mysql` (служебная база данных), используя имя пользователя root (`-u root`). Поскольку вы только что указали пароль для пользователя root (до этого пароль для root не был задан), вам нужно указать параметр `-p`. После того как программа mysql подключится к серверу, вы увидите приглашение программы. В ответ на него нужно ввести следующий SQL-оператор:

```
insert into user(Host, User, Password, Select_priv, Insert_priv,
Update_priv, Delete_priv)
values ('%','username',password('123456'),'Y','Y','Y','Y');
```

Этим оператором мы создали пользователя с именем `username` и паролем `123456`. Данный пользователь имеет право использовать SQL-операторы `select` (выборка из таблицы), `insert` (добавление новой записи в таблицу), `update` (обновление записи), `delete` (удаление записи). Если вам нужно, чтобы ваш пользователь имел право создавать и удалять таблицы, тогда добавьте привилегии `Create_priv` и `Drop_priv`:

```
insert into user(Host, User, Password, Select_priv, Insert_priv, Up-
date_priv, Delete_priv, Create_priv, Drop_prov)
values ('%','username',password('123456'),'Y','Y','Y','Y','Y','Y');
```

Совет

Приведенный здесь SQL-оператор можно записать в одну строку, можно разбить на несколько строк — как вам будет удобно. Но в конце каждого SQL-оператора должна быть точка с запятой! Помните об этом.

Для выхода из программы mysql нужно ввести команду `quit`;

Кроме программы mysql, в состав MySQL-клиента входит одна очень полезная программа — mysqlshow, которая может вывести список таблиц, находящихся в той или иной базе данных. Кроме этого, она еще много чего может, но в данный момент нам нужен пока список таблиц — чтобы вы знали, какие таблицы есть в базе данных:

```
mysqlshow -p <база данных>
```

29.7.3. Запуск и останов сервера

Для управления MySQL-сервером используется программа /etc/init.d/mysql. Чтобы запустить сервер, нужно передать этой программе параметр `start`, для останова — `stop`, а для перезапуска — `restart`:

```
sudo /etc/init.d/mysql start
sudo /etc/init.d/mysql stop
sudo /etc/init.d/mysql restart
```

В Mandriva/Fedora можно воспользоваться командой `service`:

```
# service mysql start
# service mysql stop
# service mysql restart
```

Также для управления сервером можно использовать программу `mysqladmin`, узнать больше о ней можно с помощью команды:

```
man mysqladmin
```

29.7.4. Программа MySQL Administrator

При установке сервера мы установили программу MySQL Administrator (пакет mysql-admin). Запустите программу командой меню **Приложения | Программирование | MySQL Administrator**. Укажите адрес сервера `localhost`, имя пользователя `root` и пароль, который вы указали при установке сервера (рис. 29.5) и нажмите кнопку **Connect**. Далее управлять сервером будет существенно проще (рис. 29.6).

Пройдемся по основным разделам программы MySQL Administrator:

- **Server Information** — общая информация о сервере (см. рис. 29.6);
- **Service Control** — управление запуском сервиса MySQL (здесь вы можете перезапустить сервер);
- **Startup Parameters** — параметры, указываемые при запуске сервера;
- **User Administration** — здесь можно добавить новых пользователей MySQL и установить права пользователей;
- **Server Connections** — позволяет просмотреть текущие соединения с сервером;
- **Server Logs** — журналы сервера;
- **Backup** — создание резервной копии сервера;
- **Restore Backup** — восстановление из резервной копии;
- **Replication Status** — состояние репликации сервера;
- **Catalogs** — просмотр имеющихся баз данных и таблиц внутри них.

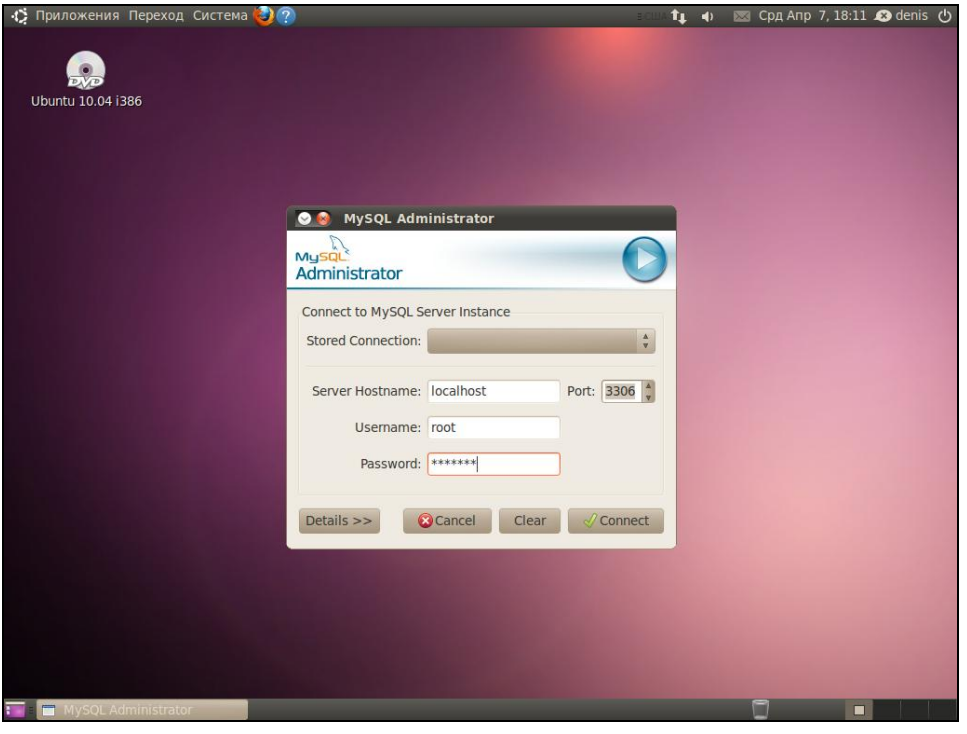


Рис. 29.5. Вход на сервер

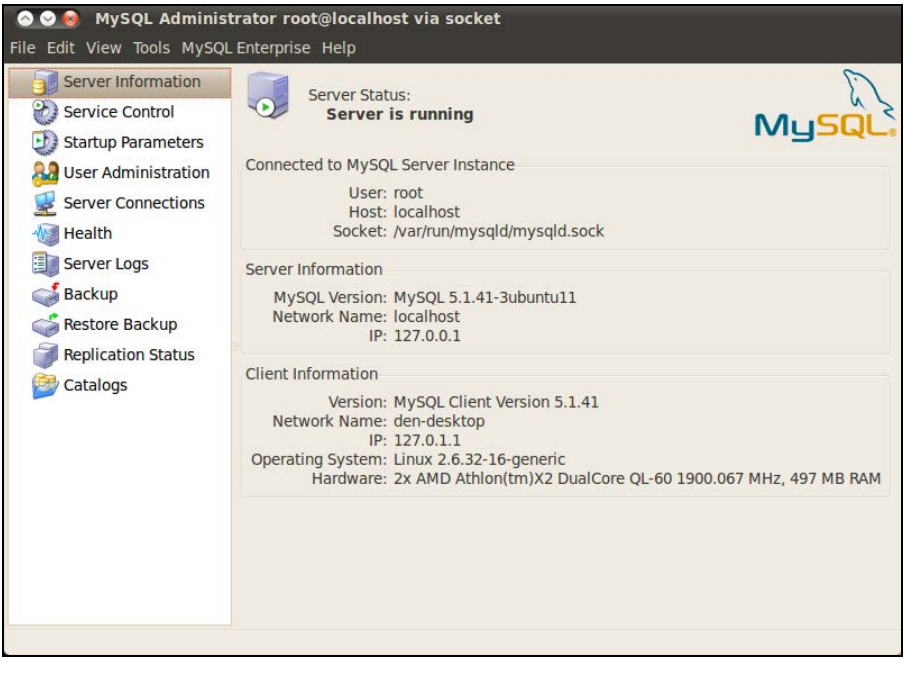
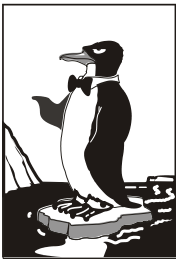


Рис. 29.6. Основное окно программы MySQL Administrator



Глава 30

FTP-сервер

30.1. Зачем нужен FTP

Сервер FTP (File Transfer Protocol) используется для обмена файлами между системами Интернета. Принцип работы FTP следующий: на FTP-сервере размещается какой-нибудь файл. Пользователи Интернета с помощью FTP-клиента (в любой операционной системе есть стандартный FTP-клиент — программа ftp) подключаются к FTP-серверу и скачивают данный файл.

Права FTP-пользователя определяются администратором FTP-сервера. Одни пользователи могут загружать на сервер файлы в свои личные каталоги, другие имеют полный доступ к FTP-серверу (могут загружать файлы в любые каталоги, как правило, это администраторы FTP-сервера), третьи могут только скачивать публично доступные файлы. Третья группа пользователей — самая большая. Это так называемые *анонимные* пользователи. Чтобы не создавать учетную запись для каждого анонимного пользователя, все они работают под так называемой *анонимной учетной записью*, когда вместо имени пользователя указывается имя anonymous, а вместо пароля — адрес электронной почты пользователя.

В локальной сети для обмена файлами можно использовать сервер Samba, имитирующий работу рабочей станции под управлением Windows, в Интернете же для обмена файлами нужно использовать только FTP-сервер. С другой стороны, ничего не мешает вам организовать FTP-сервер для обмена файлами внутри локальной сети — это дело вкуса и предпочтений администратора.

Все необходимое для организации FTP-сервера программное обеспечение входит в состав дистрибутива или же бесплатно доступно для скачивания в Интернете. В этой главе мы рассмотрим самый удобный, на мой взгляд, FTP-сервер ProFTPD. Это не единственный FTP-сервер для Linux, например, есть еще wu-ftpd, но ProFTPD является одним из самых защищенных и удобных в настройке.

30.2. Установка FTP-сервера

Для установки FTP-сервера нужно установить пакет proftpd. Можно также установить конфигуратор gproftpd, если он доступен в вашем дистрибутиве.

Для запуска и останова сервера можно использовать команду `service`:

```
service  protfpd start
service  protfpd stop
```

30.3. Конфигурационный файл

Основной конфигурационный файл сервера ProFTPD называется /etc/proftpd/proftpd.conf. В листинге 30.1 представлен его простейший пример.

Листинг 30.1. Пример файла конфигурации /etc/proftpd/proftpd.conf

```
# Подключаем файл с модулями
Include /etc/proftpd/modules.conf

ServerName              "My server"          # можно написать все,
       # что угодно
ServerType              standalone           # автономный
DeferWelcome            off                  # вывести приветствие до
       # аутентификации

MultilineRFC2228        on                   # поддержка RFC2228
DefaultServer           on                   # сервер по умолчанию
ShowSymlinks            on                   # показывать символические ссылки

# настройка тайм-аутов
TimeoutNoTransfer       600
TimeoutStalled          600
TimeoutIdle             1200

DisplayLogin            welcome.msg          # файл с приветствием
DisplayFirstChdir       .message             # отобразить этот файл при
       # каждой смене каталога

# запрещает использовать данное выражение в FTP-командах
# (все файлы (маска *.*) вы уже не сможете удалить, придется удалять по
одиночке!)
DenyFilter              \*.*/

Port                    21                   # стандартный порт

MaxInstances            30                   # количество копий proftpd

# пользователь и группа, от имени которых работает proftpd
User                    proftpd
Group                   nogroup

Umask                   022  022             # см. man umask
```

```
AllowOverwrite       on                    # можно перезаписывать файлы

# Журналы сервера
TransferLog /var/log/proftpd/xferlog
SystemLog   /var/log/proftpd/proftpd.log

# Параметры подключаемых модулей. Изменять не нужно
<IfModule mod_tls.c>
TLSEngine off
</IfModule>

<IfModule mod_quota.c>
QuotaEngine on
</IfModule>

<IfModule mod_ratio.c>
Ratios on
</IfModule>

<IfModule mod_delay.c>
DelayEngine on
</IfModule>

<IfModule mod_ctrls.c>
ControlsEngine        on
ControlsMaxClients    2
ControlsLog           /var/log/proftpd/controls.log
ControlsInterval      5
ControlsSocket        /var/run/proftpd/proftpd.sock
</IfModule>

<IfModule mod_ctrls_admin.c>
AdminControlsEngine on
</IfModule>
```

В конфигурационном файле profptd.conf вы можете использовать обычные директивы, задающие одиночные свойства, и блочные директивы, определяющие группы свойств (параметров). Например, директива `ServerName` — обычная, она задает одно свойство, а директива `Directory` — блочная, позволяющая задать несколько параметров для одного каталога.

С директивами файла конфигурации можно ознакомиться в табл. 30.1. В этой таблице указаны не все директивы, а только самые полезные. С остальными вы всегда можете ознакомиться, прочитав документацию по ProFTPD.

Таблица 30.1. *Директивы файла конфигурации profptd.conf*

Директива	Описание
`AccessGrantMsg "сообщение"`	Задает сообщение, которое будет отправлено пользователю при его регистрации на сервере. Можно задать грозное сообщение, напоминающее о том, что попытка несанкционированного доступа карается статьей такой-то уголовного кодекса
`Allow from all | узел | сеть [,узел | сеть[, ...]]`	Данная директива может использоваться только в блоке `Limit`. Директива разрешает доступ к серверу. По умолчанию используется значение `all`, которое разрешает доступ к серверу всем узлам со всех сетей
`AllowAll`	Разрешает доступ всем. Может использоваться в блоках `Directory`, `Anonymous`, `Limit`
`AllowForeignAddress on | off`	Разрешает узлу при подключении к серверу указывать адрес, не принадлежащий ему. По умолчанию используется значение `off` (то есть доступ запрещен), рекомендуется не изменять его. Директива может использоваться в блоках `Anonymous`, `<Global>`
`AllowGroup список групп`	Разрешает доступ к серверу указанным группам пользователей (группы должны быть зарегистрированы на этом сервере)
`AllowOverwrite on | off`	Разрешает (`on`) перезаписывать существующие файлы
`AllowUser список пользователей`	Разрешает доступ к серверу указанным группам пользователей (пользователи должны быть зарегистрированы на этом сервере)
`<Anonymous каталог>`	Разрешает анонимный доступ к указанному каталогу. Указанный каталог будет корневым каталогом анонимного FTP-сервера
`AuthGroupFile файл`	Задает альтернативный файл групп. По умолчанию /etc/group
`AuthUserFile файл`	Задает альтернативный файл паролей. По умолчанию /etc/passwd
`Bind IP-адрес`	Выполняет привязку дополнительного адреса к FTP-серверу
`DeferWelcome on | off`	Вывести приветствие после аутентификации (`on`) или до нее (`off`)
`Deny from all | узел | сеть`	Директива запрещает доступ к FTP-серверу. Используется в блоке Limit
`DenyAll`	Запрещает доступ всем к объектам, указанным в `Directory`, `Anonymous`, `Limit`
`DenyUser список пользователей`	Запрещает доступ указанным пользователям
`DefaultRoot каталог`	Определяет корневой каталог FTP-сервера. В качестве значения этого параметра полезно указать значение ~, тогда в качестве корневого каталога будет использоваться домашний каталог пользователя, который зашел на сервер

Таблица 30.1 *(окончание)*

Директива	Описание
`DisplayLogin файл`	Указанный текстовый файл будет отображен, когда пользователь зайдет на сервер
`DisplayFirstChdir файл`	Отобразить указанный файл при каждой смене каталога
`<Directory каталог>`	Задает параметры доступа к каталогу и его подкаталогам
`<Global>`	Задает глобальные параметры FTP-сервера
`<Limit команда>`	Накладывает ограничение на выполнение FTP-команд, например, READ, WRITE, STOR, LOGIN
`MaxClients число сообщение`	Максимальное количество одновременно работающих клиентов. Если указанное число будет превышено, FTP-сервер отобразит указанное сообщение
`MaxLoginAttempts`	Максимальное количество попыток регистрации на сервере. По умолчанию 3. Указывается в блоке `Global`
`MaxInstances`	Максимальное количество одновременно работающих экземпляров демона proftpd
`ServerType тип`	Задает тип запуска сервера. Значение по умолчанию — standalone (автономный запуск). Не нужно его изменять
`ServerName "имя"`	Задает имя сервера. Можете написать все, что угодно, например, `My server`
`ServerAdmin e-mail`	Позволяет указать адрес электронной почты администратора сервера
`ShowSymlinks on | off`	Показывать символические ссылки (`on`) или сразу результирующие файлы (`off`)
`Order allow, deny | deny, allow`	Задает порядок выполнения директив `Allow` и `Deny` в блоке `Limit`
`TimeoutIdle секунды`	Определяет тайм-аут простоя. Если пользователь не проявит активности за указанное время, соединение будет разорвано. По умолчанию используется значение 60
`TimeoutNoTransfer секунды`	Тайм-аут начала передачи. Сколько времени нужно ждать до разъединения, если пользователь вошел, но не начал передачу
`TimeoutStalled секунды`	"Замирание" во время передачи файла. Бывает так, что клиент начал передачу (или прием) файла, но связь оборвалась. Этот тайм-аут определяет, сколько нужно ждать до разъединения в такой ситуации. Данный тайм-аут нужен, потому что бывает другая ситуация —когда у пользователя очень медленный канал
`Umask маска`	Задает права доступа для созданного файла
`User имя_пользователя`	Пользователь, от имени которого работает демон ProFTPD

30.4. Настройка реального сервера

В этом разделе мы настроим реальный FTP-сервер, к которому смогут получить доступ как обычные (зарегистрированные) пользователи, так и анонимные.

Приведенная в листинге 30.1 конфигурация вполне работоспособная и может использоваться для создания обычного (не анонимного) FTP-сервера. Но в конфигурационный файл нужно добавить две директивы:

```
DefaultRoot       ~
MaxClients  20 "Server is full!!!"
```

- Директива `DefaultRoot` делает корневым домашний каталог пользователя (то есть пользователь не может выйти за пределы своего домашнего каталога, следовательно, не может навредить системе, если администратор неправильно установил права к каким-нибудь системным каталогам).
- Директива `MaxClients` ограничивает число одновременно работающих клиентов во избежание перегрузки сервера.

Остальные параметры вы можете задать по своему усмотрению. Рассмотрим несколько примеров использования блоков `Directory` и `Login`:

```
<Directory upload>
    <Limit READ>
           DenyAll
    </Limit>
    <Limit WRITE>
           AllowAll
    </Limit>
</Directory>
```

Директива `Directory` определяет две директивы `Limit` для каталога upload. Первая запрещает всем читать этот каталог, а вторая — разрешает всем записывать новые файлы в этот каталог. Каталог upload, таким образом, полностью оправдывает свое название — только для закачки файлов.

Рассмотрим еще один пример, запрещающий доступ к серверу всех узлов из подсети 192.168.1.0

```
<Limit LOGIN>
    DenyAll
    Deny from 192.168.1.
</Limit>
```

Если нужно, наоборот, разрешить доступ к серверу только пользователей из сети 192.168.1.0, то нужно использовать следующий блок `Limit`:

```
<Limit LOGIN>
Order deny, allow         # порядок действия deny-allow
    DenyAll               # запрещаем доступ всем
    Allow from 192.168.1. # разрешаем доступ только из сети 192.168.1.0
</Limit>
```

Теперь перейдем к анонимного доступу. Для организации анонимного доступа нужно добавить в файл конфигурации следующую директиву `Anonymous`:

```
<Anonymous ~ftp>

User                    ftp
Group                           nogroup

# Определяем псевдоним "anonymous" для пользователя "ftp"
# Клиенты смогут войти под обеими именами

UserAlias               anonymous ftp

# Все файлы принадлежат пользователю ftp
DirFakeUser on ftp
DirFakeGroup on ftp

# Не нужно требовать "правильную" оболочку
# "Правильной" считается оболочка, указанная в /etc/shells
RequireValidShell       off

# Максимальное число анонимных пользователей
MaxClients              10

# Файлы с сообщениями
DisplayLogin                    welcome.msg
DisplayFirstChdir       .message

# Ограничим WRITE для анонимных пользователей
   <Directory *>
     <Limit WRITE>
       DenyAll
     </Limit>
   </Directory>

</Anonymous>
```

30.5. Программы ftpwho и ftpcount

Вспомогательные программы ftpwho и ftpcount помогут администратору FTP-сервера определить, какие пользователи в данный момент зарегистрированы на сервере (ftpwho), и узнать общее число зарегистрированных на сервере в данный момент пользователей (ftpcount). Вывод обеих программ показан на рис. 30.1.

```
den@den-desktop:~$ ftpwho
standalone FTP daemon [5176], up for 37 min
 7378 den      [ 0m10s]   0m7s idle
Service class                      -   1 user
den@den-desktop:~$ ftpcount
Master proftpd process 5176:
Service class                      -   1 user
den@den-desktop:~$
```

Рис. 30.1. Программы ftpwho и ftpcount

30.6. Конфигуратор gproftpd

Графический конфигуратор gproftpd (рис. 30.2) позволяет быстро и комфортно настроить FTP-сервер. С этим конфигуратором вы разберетесь и без моих комментариев — там все очень просто, особенно сейчас, когда вы знаете назначение основных директив сервера. Но не следует забывать, что это всего лишь конфигуратор, который может помочь настроить только базовые возможности сервера.

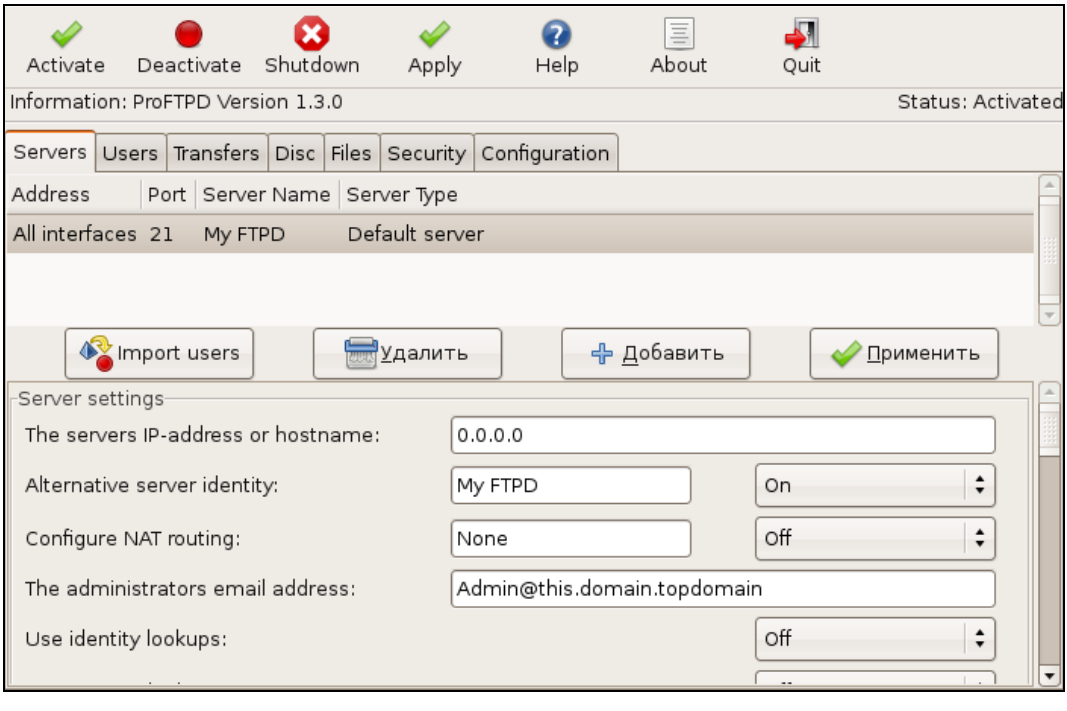
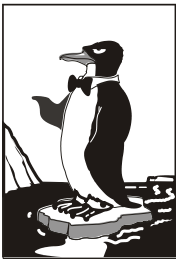


Рис. 30.2. Конфигуратор gproftpd

30.7. Альтернативные FTP-серверы

Сервер ProFTPD, наверное, самый лучший, но далеко не единственный. Кроме него есть еще и wu-ftpd, довольно популярный сервер в свое время. Но сегодня он устарел и используется только на старых машинах, где давно не обновлялось программное обеспечение (ну или где администратору лень разбираться с ProFTPD и он по привычке использует старый wu-ftpd).

Наш выбор — ProFTPD. Этот довольно безопасный сервер установлен на огромном количестве FTP-серверов Интернета. Но если вам хочется чего-то экзотического — чтобы было не как у всех, то попробуйте установить сервер VSFTPD (Very Secure FTPD). Да, его так и называли "очень безопасный FTP-демон". Это очень безопасный, компактный и быстрый FTP-сервер. К сожалению, учитывая его компактность, по возможностям он не дотягивает до ProFTPD, поэтому для некоторых серверов его функциональности будет маловато. Но для большинства серверов VSFTPD вполне подойдет. Официальный сайт проекта: **http://vsftpd.beasts.org/**, но VSFTPD входит в состав репозиториев большинства современных дистрибутивов, поэтому загружать его с этого сайта вам не придется.



Глава 31

Почтовый сервер

31.1. Что такое Qmail?

Почтовый сервис состоит из агента отправки почты (MTA, Mail Transfer Agent), реализующего протокол SMTP (Simple Mail Transfer Protocol), и сервиса POP3 (Post Office Protocol v3), который используется для приема почты. Вы можете установить любой POP3-сервис, например, cyrus-pop3d, и любой MTA-агент, например, sendmail или postfix, "пошаманить" над конфигурацией этих сервисов и получить работающий почтовый сервер. Один сервис будет принимать почту для ваших пользователей, а другой — отправлять ее.

Если же вы установите и настроите Qmail, то получите два средства в одном флаконе: Qmail заменит и MTA, и POP3-сервис. Это намного проще и удобнее, особенно при настройке почтового сервера. Судите сами: если у вас два разных сервиса, то для правильной их настройки нужно прочитать в два раза больше документации. Для правильной настройки Qmail вполне хватит документации на сайте производителя, ну и, конечно же, этой книги.

Qmail — это очень гибкий и очень мощный почтовый сервер, но у него есть одна особенность — он не совсем прост в настройке. Я бы не сказал, что он сложный, но и не нужно думать, что вы за пять минут его настроите. Если вам нужно поднять почтовый сервер за пять минут, тогда установите cyrus-pop3d и postfix. Правда, знайте, что и работать такой сервер будет, как "настроенный за 5 минут".

31.2. Подготовка к установке Qmail

Первым делом, поговорим о необходимом дисковом пространстве. В корневой файловой системе должно быть не менее 70 Мбайт свободного места, а в файловой системе, примонтированной к /var, — не менее 20 Мбайт доступного дискового пространства. Всего нужно чуть меньше 100 Мбайт, но помните, что дисковое пространство также понадобится для писем пользователей, а тут рекомендация одна: чем больше места, тем лучше.

Перед установкой Qmail нужно удалить все SMTP и POP3-сервисы: postfix, sendmail, cyrus, Qpopper и др. — чтобы не возникло конфликта между этими сервисами и Qmail. Отключите также SELinux — впрочем, обычно она по умолчанию выключена, поэтому никаких таких действий, очевидно, производить не придется.

Теперь рассмотрим список программного обеспечения, которое должно быть установлено до Qmail:

- Web-сервер Apache (версия сервера 1.3 или 2.x, роли не играет);
- интерпретатор PHP с поддержкой imap и mysql (нужно дополнительно установить пакеты php-imap и php-mysql);
- интерпретатор Perl (версия 5.x);
- сервер MySQL (подойдут версии 3.x и 4.x);
- библиотека OpenSSL, а также файлы для разработчиков — OpenSSL-devel (пакеты называются openssl и openssl-devel);
- программа wget (нужна для загрузки файлов из Интернета);
- пакеты patch и patchutils (нужны для накладывания патчей).

Кроме этого, вам нужно установить следующие модули Perl:

```
Digest::SHA1
Digest::HMAC
Net::DNS
Time::HiRes
HTML::Tagset
HTML::Parser
```

Давайте сделаем это прямо сейчас. Подключитесь к Интернету и введите команду:

```
perl -MCPAN -e shell
```

Затем команду:

```
install имя_модуля
```

Данную команду нужно ввести для каждого устанавливаемого модуля. Подробнее об установке модулей Perl можно прочитать по адресу: **http://www.dkws.org.ua/phpbb2//viewtopic.php?topic=3035**.

Вам также придется перенастроить свой брандмауэр. Нужны порты, указанные в табл. 31.1.

Таблица 31.1. *Порты, которые необходимо открыть*

Входящие TCP-соединения	Исходящие TCP-соединения
25 — SMTP	25 — SMTP
80 — HTTP	110 — POP
110 — POP	143 — IMAP
143 — IMAP	783 — Spamassassin
443 — HTTPS	993 — IMAPS
783 — Spamassassin (защита от спама)	
993 — IMAPS	

31.3. Установка Qmail и необходимых дополнений

31.3.1. Загрузка и установка Qmail

Сначала создадим каталог, в который мы загрузим исходные коды Qmail. Да, установку будем производить из исходных кодов, а не из RPM-пакета:

```
# mkdir /qmail
# cd /qmail
```

Скачайте файл qmailrocks.tar.gz с сайта **qmailrocks.ru**:

```
# wget http://www.qmailrocks.ru/downloads/qmailrocks.tar.gz
```

Затем распакуйте его:

```
# tar zxvf qmailrocks.tar.gz
```

После этого запустите сценарий qmr_install_linux-s1.script:

```
# /qmail/qmailrocks/scripts/install/qmr_install_linux-s1.script
```

Указанный сценарий создаст необходимых пользователей и группы, необходимые каталоги и установит права доступа к ним.

После этого нужно "пропатчить" Qmail. В исходные коды Qmail будет в общем внедрено 15 патчей, которые добавят новые возможности к Qmail. Особо вникать в эти патчи не будем, а просто введем команду:

```
/qmail/qmailrocks/scripts/util/qmail_big_patches.script
```

Этот сценарий добавит все 15 патчей, поэтому вам не нужно устанавливать каждый патч отдельно.

Следующий шаг — это компиляция Qmail. Для этого введите следующие команды:

```
# cd /usr/src/qmail/qmail-1.03
# make man && make setup check
# ./config-fast полное_имя_узла
```

В качестве параметра последней команды вы должны указать полное доменное имя почтового сервера, например,

```
# ./config-fast mail.firma.ru
```

Теперь у нас есть установленный Qmail. Сразу после этого нужно сгенерировать сертификаты, которые используются для шифрования SMTP-сессии:

```
# make cert
```

Эта команда задаст вам несколько вопросов, вроде вашего местоположения, названия фирмы и т. д. Сгенерированный сертификат будет помещен в каталог /var/qmail/control/. Файл сертификата называется servercert.pem. Файл clientcert.pem — это ссылка на файл servercert.pem.

Сразу после создания сертификатов нужно установить права доступа к нему:

```
# cd /var/qmail/control/
# chown -R vpopmail:qmail clientcert.pem servercert.pem
```

31.3.2. Установка ucspi-tcp и daemontools

Вот теперь можно приступить к сборке ucspi-tcp (используется для создания приложений клиент/сервер, работающих по протоколу TCP) и daemontools (не путать с эмулятором виртуального CD-ROM в Windows!):

```
# cd /usr/src/qmail/ucspi-tcp-0.88/
# patch < /qmail/qmailrocks/patches/ucspi-tcp-0.88.errno.patch
# make && make setup check
# cd /package/admin/daemontools-0.76/src
# patch < /qmail/qmailrocks/patches/daemontools-0.76.errno.patch
# cd /package/admin/daemontools-0.76
# package/install
```

Теперь в вашей системе появится работающий сервис svscanboot.

31.3.3. Установка EZmlm — средства для создания рассылки

Вы планируете создавать списки рассылки? Если да, тогда вам нужно установить дополнение EZmlm, которое будет интегрировано в утилиту управления Qmailadmin. Для установки EZmlm введите команды:

```
# cd /qmail/qmailrocks/
# tar zxvf ezmlm-0.53-idx-0.41.tar.gz
# cd ezmlm-0.53-idx-0.41
# make && make setup
```

Если в ответ: "тишина", значит, все прошло успешно. А вот если произошли ошибки, тогда вы увидите их описание.

31.3.4. Установка Autoresponder — автоответчика

Autoresponder позволяет настраивать автоответчики для почтовых ящиков. Для его установки введите команды:

```
# cd /qmail/qmailrocks
# tar zxvf autorespond-2.0.5.tar.gz
# cd autorespond-2.0.5
#make && make install
```

31.3.5. Установка MailDrop — фильтра для сообщений

MailDrop — фильтр сообщений, приходящий на почтовый сервер. Подробно мы его рассматривать не будем, выполним только его установку, а с документацией по MailDrop вы сможете ознакомиться по адресу: **http://www.courier-mta.org/maildrop/**.

Итак, для установки MailDrop нужно ввести следующие команды:

```
# cd /qmail/qmailrocks
# tar zxvf maildrop-1.6.3.tar.gz
```

```
# cd maildrop-1.6.3
# ./configure --prefix=/usr/local --exec-prefix=/usr/local --enable-
maildrop-uid=root --enable-maildrop-gid=vchkpw --enable-maildirquota
# make && make install-strip && make install-man
```

31.3.6. Установка QmailAdmin — Web-интерфейса для настройки Qmail

У Qmail нет обычного конфигуратора, зато есть конфигуратор с Web-интерфейсом, что еще удобнее, поскольку вы сможете настраивать Qmail с любого компьютера вашей сети (и не только с вашей — при должной настройке брандмауэра и Apache).

Для установки Web-интерфейса QmailAdmin введите следующие команды:

```
# cd /qmail/qmailrocks
# tar zxvf qmailadmin-1.2.9.tar.gz
# cd qmailadmin-1.2.9
# ./configure --enable-autoresponder-path=/usr/local/bin --enable-
cgibindir=/путь/к/cgi-bin --enable-htmldir=/путь/к/каталогу/html
# make && make install-strip
```

Обратите внимание — вы должны указать путь к каталогам cgi-bin и html вашего Web-сервера. Обычно это /var/www/cgi-bin и /var/www/html.

После этого откройте браузер и ведите следующий URL:

```
http://localhost/cgi-bin/qmailadmin
```

Вы увидите форму для ввода имени пользователя и пароля. Войдите под пользователем postmaster (соответственно, указав пароль этого пользователя, а не пользователя root). После этого вы можете настраивать Qmail с помощью QmailAdmin — это действительно очень легко.

31.4. Настройка после установки и запуск Qmail

Почти все готово к запуску вашего почтового сервера, осталось совсем немного. Вам нужно создать сценарий для запуска и управления qmail. Для этого запустите следующий сценарий, который автоматизирует эту задачу:

```
# /qmail/qmailrocks/scripts/finalize/linux/finalize_linux.script
```

После этого откройте в любом текстовом редакторе файл /var/qmail/supervise/qmail-pop3d/run. Найдите в нем строку `mail.example.com` и замените ее именем вашего почтового сервера, например, `mail.firma.ru`, после чего сохраните файл.

Теперь введите команду:

```
# qmailctl stop
```

Как вы уже догадались, данная команда завершает все запущенные процессы qmail. Нужно установить релей (relay) для локальной машины:

```
# echo '127.:allow,RELAYCLIENT=""' >> /etc/tcp.smtp
# qmailctl cdb
```

Теперь следует установить адрес электронной почты, на который должны приходить системные сообщения:

```
# echo postmaster@firma.ru > /var/qmail/alias/.qmail-root
# echo postmaster@firma.ru > /var/qmail/alias/.qmail-postmaster
# echo postmaster@firma.ru > /var/qmail/alias/.qmail-mailer-daemon
```

Понятно, что `firma.ru` нужно заменить именем вашего домена.

Вам осталось ввести всего две команды:

```
# ln -s /var/qmail/alias/.qmail-root /var/qmail/alias/.qmail-anonymous
# chmod 644 /var/qmail/alias/.qmail*
```

После этого у вас есть полноценный почтовый сервер. Перед запуском Qmail нужно убедиться, что мы все сделали правильно. Запустите следующий сценарий для проверки вашей конфигурации:

```
# /qmail/qmailrocks/scripts/util/qmr_inst_check
```

Если вы в ответ получите **congratulations**, то все сделали правильно, поздравляю вас и я! А вот если будут сообщения об ошибках, внимательно прочитайте их, исправьте и запустите сценарий проверки конфигурации снова.

Для управления Qmail используется программа qmailctl:

```
# qmailctl start   - запуск Qmail
# qmailctl stop    - запуск Qmail
# qmailctl stat    - вывод статистики
```

Запустите Qmail:

```
# qmailctl start
```

Сейчас проверим его работоспособность с помощью telnet:

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 somewhere.anywhere.com ESMTP
ehlo localhost
250-somewhere.anywhere.com
250-AUTH LOGIN CRAM-MD5 PLAIN
250-AUTH=LOGIN CRAM-MD5 PLAIN
250- STARTTLS
250-PIPELINING
250 8BITMIME
starttls
220 ready for tls
quit
quit
Connection closed by foreign host.
```

Полужирным шрифтом выделены команды, которые вы должны ввести в telnet-сессии. Если вы в ответ на команду `starttls` увидели сообщение **454 TLS not available: missing RSA private key (#4.3.0)**, проверьте, есть ли в каталоге /var/qmail/control/ файл servercert.pem. Если его нет, нужно сгенерировать сертификат заново.

Если же файл /var/qmail/control/servercert.pem существует, установите для него права:

```
chown vpopmail:qmail /var/qmail/control/servercert.pem
```

Теперь проверим работу POP3:

```
# telnet localhost 110
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
+OK
user den@firma.ru
+OK
pass den_password
+OK
quit
+OK
Connection closed by foreign host.
```

Как и в прошлом случае, команды, которые вы должны ввести, выделены. Команда `user` позволяет указать пользователя, к почтовому ящику которого вы хотите подключиться. Пользователь должен быть зарегистрирован на сервере. Команда `pass` задает пароль пользователя (без кавычек). Если ваша POP3-сессия похожа на приведенную, тогда вы все сделали правильно.

31.5. Настройка почтовых клиентов

При настройке почтовых клиентов в качестве имени пользователя нужно указывать полное его имя в формате **имя_пользователя@сервер**, например, **den@firma.ru** (рис. 31.1).

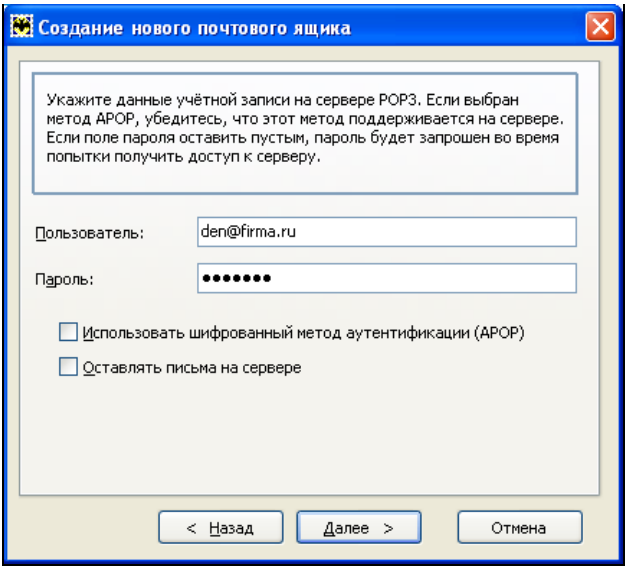


Рис. 31.1. Настройка программы The Bat!

Настроенный нами SMTP-сервер требует SMTP-аутентификации для отправки писем, поэтому не забудьте при настройке почтового клиента указать и это (рис. 31.2).

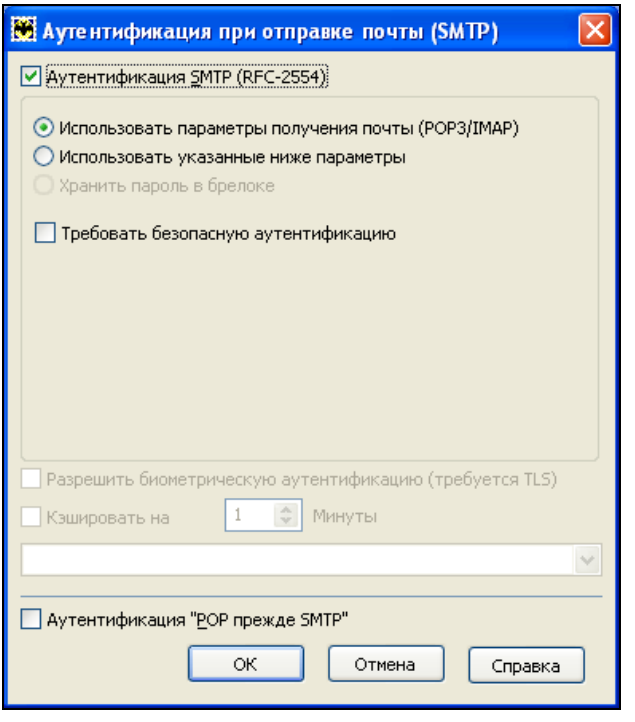
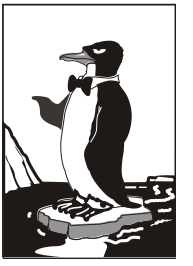


Рис. 31.2. Включение SMTP-аутентификации в The Bat!

31.6. Дополнительная информация

Вам нужно настроить Qmail на FreeBSD или установить фильтр спама? Тогда настоятельно рекомендую посетить сайт **http://www.qmailrocks.ru/**, где вы найдете много информации относительно настройки Qmail и антиспамовых фильтров.



Глава 32

Сервис Samba

32.1. Установка Samba

Linux — отличная операционная система, но от Windows нам не уйти. Windows будет окружать нас всегда, будь то домашняя, корпоративная сеть или интернет-кафе. Нам постоянно предстоит обмениваться документами с Windows-компьютерами — ведь далеко не все пользователи предпочитают работать в Linux. В этой книге особое внимание было уделено взаимодействию с Windows-компьютерами, и было бы нелогично не сказать о подключении Linux к сети Microsoft.

В Linux для взаимодействия с сетью Microsoft служит пакет samba-server. Если вы хотите использовать общие ресурсы Windows-сети, установите этот пакет. Он позволяет не только пользоваться общими ресурсами сети, но и предоставлять собственные ресурсы Windows-пользователям. Причем все происходит так, что Windows-пользователи даже не заметят разницы.

После установки этого пакета будет установлен сервис smb — это и есть основной сервис Samba. Запускать и останавливать его можно командами:

```
service smb start
service smb stop
```

32.2. Базовая настройка Samba

Основной конфигурационный файл Samba — /etc/samba/smb.conf. Откройте его. Сейчас мы изменим пару параметров. Первым делом измените параметр `WORKGROUP` — он задает имя рабочей группы или домена NT:

```
WORKGROUP = MSHOME
```

Конечно, имя группы у вас, скорее всего, будет другим. Можете также установить параметр `comment` — это описание вашего компьютера:

```
server string = My Linux computer
```

Установите параметр `security`. Если у вас сеть клиент/сервер, то нужно выбрать параметр `server`, а если у вас одноранговая сеть (то есть сеть без выделенного сервера), то нужно выбрать `user` или `share`:

```
security = share
```

Имя гостевой учетной записи установите так:

```
guest account = guest
```

Также нужно настроить кодировки:

```
unix charset = UTF-8
dos charset = UTF-8
display charset = UTF-8
```

Для того чтобы Samba работала быстрее, установите следующие опции:

```
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
dns proxy = no
```

Что они означают, мы разберемся чуть позже.

Параметр `interfaces` указывает интерфейсы, на которых должен работать сервис smb. Укажите те интерфейсы, которые связывают вашу машину с Windows-сетями:

```
interfaces = 192.168.0.22/24
```

А теперь позволю себе несколько комментариев для пользователей предыдущей версии Samba. В более ранних версиях Samba параметр `server string` назывался `comment`. Теперь вместо параметров `client code page` и `character set` используются параметры `unix charset`, `dos charset` и `display charset`. Текущая версия Samba полностью поддерживает UTF-8 (как и современные версии Linux и Windows), поэтому проблем с UTF-8 возникнуть не должно. Параметры `client code page` и `character set` больше не поддерживаются.

Параметр `unix charset` задает кодировку, в которой хранятся файлы конфигурации Samba, `dos charset` — кодировку для Windows-клиентов, а `display charset` — кодировку для Samba-клиентов.

32.3. Настройка общих ресурсов

Теперь осталось сконфигурировать ресурсы, которые вы хотите предоставить в общее пользование (листинг 32.1). Фрагмент, приведенный в листинге 32.1, нужно добавить в файл конфигурации Samba.

Листинг 32.1. Секция `[public]`

```
[public]
# общий каталог, комментарий для ресурса задается директивой comment
comment = Public Directory
# путь
path = /var/samba
# не только чтение
read only = no
# разрешить запись
writable = yes
# разрешить гостевой доступ
guest ok = yes
# разрешить просмотр содержимого каталога
browseable = yes
```

В этом случае общим ресурсом нашего компьютера будет каталог /var/samba. В него другие пользователи смогут записывать свои файлы (`read only = no`, `writeable = yes`), естественно, они смогут их и читать (`browseable = yes`). Проверка имени пользователя и пароля для доступа к ресурсу не нужна (`guest ok = yes`) — используется так называемый *гостевой* доступ. Комментарий `Public Directory` увидят другие пользователи Windows-сети при просмотре ресурсов вашего компьютера.

Рассмотрим еще один пример, позволяющий сделать общими домашние каталоги пользователей — секция `[homes]` (листинг 32.2).

Листинг 32.2. Секция `[homes]`

```
[homes]
 comment = Home Directories
 browseable = no
 valid users = %S

# запись запрещена, только просмотр
 writable = no

# маска при создании файлов, нужна, если writable=yes
create mask = 0600
# маска при создании каталогов, нужна, если writable=yes
directory mask = 0700
```

В листинге 32.3 приведен пример предоставления общего доступа к CD/DVD. Будем считать, что наш CD/DVD смонтирован в /cdrom.

Листинг 32.3. Пример общего доступа к CD/DVD

```
[cdrom]
   comment = Samba server's CD-ROM
   writable = no
   locking = no
# каталог /cdrom должен существовать и являться точкой монтирования CD/DVD
   path = /cdrom
   public = yes
# следующие два параметра нужны для автоматического монтирования CD/DVD
# они будут работать, если /etc/fstab содержит следующую строку:
# /dev/scd0   /cdrom  iso9660 defaults,noauto,ro,user   0 0
# /dev/scd0 — имя устройства CD/DVD
# /cdrom — точка монтирования (каталог должен существовать)
   preexec = /bin/mount /cdrom
   postexec = /bin/umount /cdrom
```

32.4. Просмотр ресурсов Windows-сети

Просмотреть ресурсы Windows-сети можно с помощью программы smbclient, но она работает в текстовом режиме, поэтому не совсем удобна. В современных дистрибутивах ресурсы Windows-сети можно просмотреть средствами графической среды. В KDE откройте файловый менеджер Dolphin, а в боковой панели выберите **Сеть**, а затем **Samba Shares** (рис. 32.1).

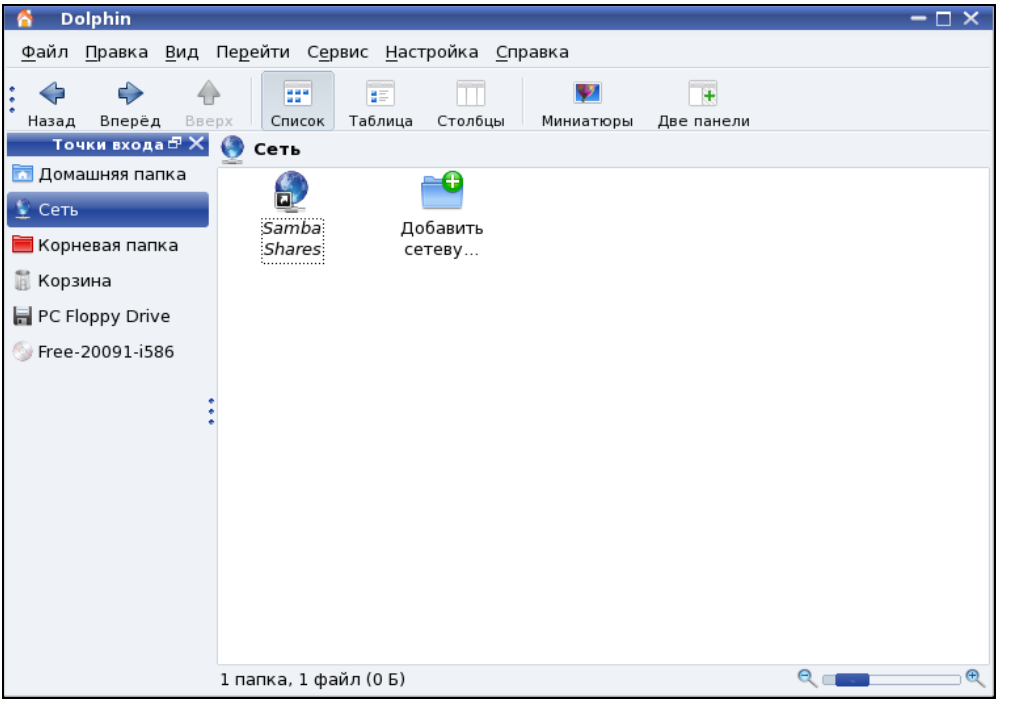
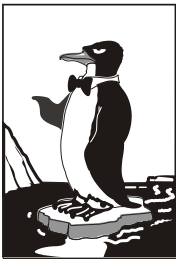


Рис. 32.1. Просмотр ресурсов сети с помощью Dolphin

Если вы используете GNOME, то для просмотра ресурсов сети можно использовать команду главного меню **Переход | Сеть**, что даже проще и удобнее, чем описанная здесь программа.

В *главе 38* мы поговорим об оптимизации Samba, а сейчас самое время перейти к следующей главе.



Глава 33

Настройка SSH-сервера

33.1. Протокол SSH и SSH-клиент

Для сервера очень важна возможность удаленного доступа. Ведь не всегда есть возможность получить физический доступ к серверу — вы можете находиться в другом конце города или даже в другой стране, а сервер предприятия будет требовать вашего оперативного вмешательства.

Раньше для организации удаленного доступа к консоли сервера использовался протокол Telnet. В каждой сетевой операционной системе, будь то FreeBSD или Windows 95 (которую, впрочем, сложно назвать сетевой), есть telnet-клиент. Данная программа так и называется — telnet (в Windows — telnet.exe) (рис. 33.1).

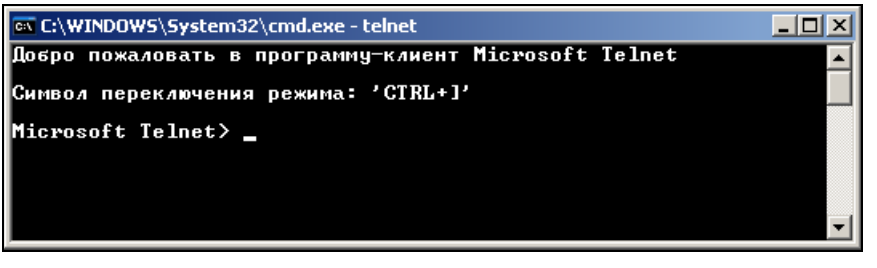


Рис. 33.1. Telnet-клиент в Windows XP Pro

После подключения с помощью telnet к удаленному компьютеру в окне telnet-клиента вы увидите как бы консоль удаленного компьютера: вы будете вводить команды и получать результат их выполнения — все так, как если бы вы работали непосредственно с клавиатурой и монитором удаленного компьютера.

Но технологии не стоят на месте, и протокол Telnet устарел. Сейчас им практически никто не пользуется. На смену ему пришел протокол SSH (Secure Shell) — как видно из названия, представляющий собой безопасную оболочку. Главное отличие SSH от telnet состоит в том, что в SSH все данные (включая пароли доступа к удаленному компьютеру и пересылаемые файлы) передаются в зашифрованном виде. Причиной создания SSH и стало, что во времена telnet участились случаи перехвата паролей и другой важной информации.

SSH использует для шифрования передаваемых данных алгоритмы BlowFish, 3DES (Data Encryption Standard), IDEA (International Data Encryption Algorithm) и RSA (Rivest-Shamir-Adelman algorithm). Самыми надежными являются алгоритмы IDEA и RSA. Поэтому, если вы передаете действительно конфиденциальные данные, лучше использовать один из этих алгоритмов.

В состав любого дистрибутива Linux входит ssh-сервер (программа, которая и обеспечивает удаленный доступ к компьютеру, на котором она установлена) и ssh-клиент (программа, позволяющая подключаться к ssh-серверу). Для установки ssh-сервера нужно установить пакет openssh (разновидность ssh-сервера), а для установки ssh-клиента — пакет openssh-clients.

Если на вашей рабочей станции установлена система Windows, и вам нужно подключиться к ssh-серверу, запущенному на Linux-машине, то по адресу **http://www.cs.hut.fi/ssh/** вы можете скачать Windows-клиент для SSH. Нужно отметить, что Windows-клиент, в отличие от Linux-клиента, не бесплатен.

Работать с ssh-клиентом очень просто. Для подключения к удаленному компьютеру введите команду:

```
ssh [опции] <адрес_удаленного_компьютера>
```

В качестве адреса можно указать как IP-адрес, так и доменное имя компьютера. В табл. 33.1 приведены часто используемые опции программы ssh.

Таблица 33.1. *Опции программы ssh*

Опция	Описание
`-c blowfish|3des|des`	Используется для выбора алгоритма шифрования, при условии, что используется первая версия протокола SSH (об этом позже). Можно указать `blowfish`, `des` или `3des`
`-c шифр`	Задает список шифров, разделенных запятыми в порядке предпочтения. Опция используется для второй версии SSH. Можно указать `blowfish`, `twofish`, `arcfour`, `cast`, `des` и `3des`
`-f`	Переводит ssh в фоновый режим после аутентификации пользователя. Рекомендуется использовать для запуска программы X11. Например: `ssh -f server xterm`
`-l имя_пользователя`	Указывает имя пользователя, от имени которого нужно зарегистрироваться на удаленном компьютере. Опцию использовать не обязательно, поскольку удаленный компьютер и так запросит имя пользователя и пароль
`-p порт`	Определяет порт ssh-сервера (по умолчанию используется порт 22)
`-q`	"Тихий режим". Будут отображаться только сообщения о фатальных ошибках. Все прочие предупреждающие сообщения в стандартный выходной поток выводиться не будут
`-x`	Отключает перенаправление X11
`-X`	Задействует перенаправление X11. Полезна при запуске X11-программ

Таблица 33.1 *(окончание)*

Опция	Описание
-1	Использовать только первую версию протокола SSH
-2	Использовать только вторую версию протокола SSH. Вторая версия протокола более безопасна, поэтому при настройке ssh-сервера нужно использовать именно ее

33.2. ssh-сервер

Теперь можно приступить к конфигурированию ssh-сервера. Если вы используете OpenSSH (в большинстве случаев так оно и есть), все настройки ssh-сервера хранятся в одном-единственном файле — /etc/sshd_config, а настройки программы-клиента — в файле /etc/ssh_config. Настройки программы-клиента обычно задавать не нужно, поскольку они приемлемы по умолчанию. На всякий случай, вы можете заглянуть в файл /etc/ssh_config — его формат, как и назначение опций (большая часть из них закомментирована), вы поймете без моих описаний.

В данный момент нас сейчас больше интересует файл sshd_config, содержащий конфигурацию ssh-сервера. Рассмотрим пример файла конфигурации ssh-сервера (листинг 33.1). Чтобы понять назначение директив, внимательно читайте комментарии, приведенные в листинге.

Листинг 33.1. Пример файла конфигурации /etc/sshd_config

```
#      $OpenBSD: sshd_config,v 1.72 2005/07/25 11:59:40 markus Exp $

# Задает порт, на котором будет работать ssh-сервер. Если директива
# не указана (закомментирована), то по умолчанию используется порт 22
#Port 22

# Директива Protocol позволяет выбрать версию протокола,
# рекомендуется использовать вторую версию
#Protocol 2,1
Protocol 2

# Директива AddressFamily задает семейство интерфейсов, которые должен
# прослушивать ssh-сервер
#AddressFamily any

# Локальный адрес, который должен прослушиваться ssh-сервером
#ListenAddress 0.0.0.0

# Ключевой файл для протокола SSH версии 1
# HostKey for protocol version 1
HostKey /etc/ssh/ssh_host_key
# Ключевые файлы для второй версии протокола SSH
HostKey /etc/ssh/ssh_host_rsa_key
```

```
HostKey /etc/ssh/ssh_host_dsa_key

# Время жизни ключа протокола первой версии. Время можно задавать в
# секундах или в часах (постфикс h, например, 1h — это 1 час или 3600
# секунд). По истечении указанного времени ключевой файл будет
# сгенерирован заново
#KeyRegenerationInterval 1h

# Разрядность ключа сервера в битах (только для первой версии протокола
# SSH)
#ServerKeyBits 768

# Директивы управления протоколированием (можно не изменять)
#SyslogFacility AUTH
#LogLevel INFO

# Директивы аутентификации

# Время, предоставляемое клиенту для аутентификации. Задается в секундах
# или минутах (1m = 60 секунд). Если за это время клиент не
# аутентифицировал себя, соединение будет прекращено
#LoginGraceTime 2m

# Директива разрешает (yes) удаленный доступ пользователя root
PermitRootLogin yes

# Максимальное количество попыток аутентификации
#MaxAuthTries 6

# Использование RSA (yes)
#RSAAuthentication yes
# Аутентификация с открытым ключом (при значении yes)
#PubkeyAuthentication yes
#AuthorizedKeysFile   .ssh/authorized_keys

# Использование .rhosts-аутентификации с поддержкой RSA.
# Rhosts-аутентификацию использовать не рекомендуется, поэтому по
# умолчанию для этой директивы указано значение no. Если вы все-таки
№ установите значение yes для этой директивы, то не
# забудьте указать в файле /etc/ssh/ssh_known_hosts IP-адреса
# компьютеров, которым разрешен доступ к ssh-серверу. Только для первой
# версии протокола
#RhostsRSAAuthentication no

# Если вы используете вторую версию протокола и хотите разрешить
# Rhosts-аутентификацию, то вам нужно включить директиву HostbasedAuthentica-
tion,
# а разрешенные узлы указываются в файле ~/.ssh/known_hosts
# HostbasedAuthentication no
```

```
# Если вы не доверяете пользовательским файлам ~/.ssh/known_hosts,
# установите значение yes для директивы IgnoreUserKnownHosts. Тогда будет
# использован только
# файл /etc/ssh/ssh_known_hosts
#IgnoreUserKnownHosts no

# Игнорировать файлы ~/.rhosts и ~/.shosts (рекомендуется установить yes)
#IgnoreRhosts yes

# Следующие директивы не рекомендуется изменять из соображений
# безопасности — они включают аутентификацию по паролю (а не IP-адресу
# компьютера, указанному в файле /etc/ssh/ssh_known_hosts)
# и запрещают использование пустых паролей
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Параметры протокола аутентификации Kerberos
# Рекомендуется использовать RSA-аутентификацию
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# Параметры GSSAPI
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes

# Использовать для аутентификации модули PAM (по умолчанию они не используются)
#UsePAM no

# Разрешить TCP-форвардинг
#AllowTcpForwarding yes

# Использовать порты шлюза
#GatewayPorts no

# Использовать X11-форвардинг (для запуска X11-приложений)
X11Forwarding yes

# Выводить сообщение дня (содержится в файле /etc/motd)
#PrintMotd yes

# Выводить время последней регистрации пользователя
#PrintLastLog yes

# Не обрывать TCP-соединения после выполнения команды по SSH
#TCPKeepAlive yes
```

```
# Отключение (значение no) этой опции позволяет немного ускорить работу
# SSH, поскольку DNS не будет использоваться для разрешения доменных имен
#UseDNS yes

# Остальные параметры рекомендуется оставить как есть
#UseLogin no
UsePrivilegeSeparation yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3

#PidFile /var/run/sshd.pid
#MaxStartups 10

#Banner /some/path

Subsystem     sftp    /usr/lib/ssh/sftp-server
```

После установки пакетов openssh и openssh-clients можно приступить к тестированию работы ssh-сервера. Для запуска сервера в Mandriva или Fedora Core можно использовать команду:

```
# service sshd start
```

А для останова (в Mandriva или Fedora Core) — ту же команду, но с параметром `stop`:

```
# service sshd stop
```

В Debian/Ubuntu для запуска/останова сервера используются команды (соответственно):

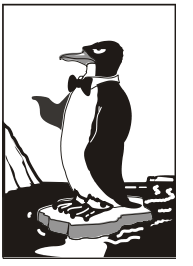
```
sudo /etc/init.d/ssh start
sudo /etc/init.d/ssh stop
```

Также запустите конфигуратор управления сервисами (drakxservices в Mandriva и system-config-services в Fedora) и убедитесь, что сервис sshd запускается при запуске системы. В Ubuntu конфигуратор, управляющий службами (сервисами), можно запустить с помощью команды меню **Система | Администрирование | Службы**.

После этого для подключения к локальному компьютеру можно ввести команду:

```
ssh 127.0.0.1
```

Можно также подключиться с удаленного компьютера. Если сеть на локальном и удаленном компьютере настроена правильно, проблем не должно возникнуть.



Глава 34

Сервер времени

34.1. Проблема синхронизации времени

Компьютерные таймеры работают с большой погрешностью — это вам не швейцарские часы. На домашнем компьютере проблема синхронизации времени не очень актуальна, но на производстве очень важно, чтобы на всех компьютерах было указано одно и то же время. Например, Иванов запустил на своем компьютере выполнение технологической операции в 11:45, а сервер запротоколировал, что операция началась в 11:47 или даже в 11:53. Когда нет никаких ЧП, то на данную погрешность времени никто не обратит внимание. А вот когда произойдет что-то неприятное и будет начато служебное расследование, все погрешности во времени только усложнят ситуацию.

В этой главе мы рассмотрим настройку собственного сервера времени. Работать все будет так — мы настроим сервер времени и запустим синхронизацию времени на всех рабочих станциях сети. Конечно, синхронизаторы будут направлены на наш сервер времени. Все, что вам остается — это следить за тем, чтобы время на сервере времени было точным, хотя это уже и не столь важно — ведь время на всех компьютерах сети будет одинаковым.

Спрашивается, зачем это нужно, если есть сервер **time.windows.com**, который можно использовать для нашей цели? Этот сервер использовать не всегда возможно. Ведь не у всех компьютеров есть доступ к Интернету, и перенастраивать ради этого брандмауэр не совсем корректно, тем более, что проблему очень легко и просто можно решить локально, своими силами, не прибегая к помощи Microsoft.

34.2. Настройка сервера и Linux-клиентов

Одним из самых удачных серверов времени для Linux является ntpd (пакет называется ntp). На каждой Linux-машине нужно установить это пакет. Одна Linux-машина будет эталонной, то есть на ней время будете устанавливать вы. Если вручную устанавливать время вам лень, тогда можно настроить синхронизацию с каким-то удаленным сервером (с тем же **time.windows.com**).

Итак, после установки ntpd приступим к его настройке. На всех клиентах конфигурационный файл /etc/ntp.conf будет выглядеть, как показано в листинге 34.1.

Листинг 34.1. Файл /etc/ntp.conf для клиента

```
restrict default ignore
restrict 127.0.0.1

# это IP-адрес эталонной машины
server 192.168.1.1

driftfile /etc/ntp/drift
broadcastdelay 0.008
authenticate yes
keys /etc/ntp/keys
```

Данный конфигурационный файл вполне приемлем. Единственное, что нужно изменить — это IP-адрес эталонной машины (сервера). У вас он будет, скорее всего, другим.

Теперь перейдем к конфигурационному файлу сервера (листинг 34.2).

Листинг 34.2. Файл /etc/ntp.conf для сервера

```
restrict default ignore
restrict 127.0.0.1

# можно указать IP-адрес какого-то удаленного сервера
# если не хочется вручную устанавливать время
server 127.127.1.0          # локальное время
fludge 127.127.1.0 stratum 10

driftfile /etc/ntp/drift
broadcastdelay 0.008
authenticate yes
keys /etc/ntp/keys
```

После того как сервер и клиенты настроены, на всех компьютерах нужно запустить сервер ntpd:

```
# service ntpd start
```

или

```
# /etc/init.d/ntpd start
```

34.3. Настройка Windows-клиентов

Тут все просто — двойным щелчком щелкните правой кнопкой на времени в области уведомлений Windows. Из появившегося меню выберите команду **Настройка даты/времени**. Перейдите на вкладку **Время Интернета** (рис. 34.1).

Нажмите кнопку **Изменить параметры**. В открывшемся окне (рис. 34.2) установите флажок **Синхронизация с сервером времени в Интернете** и введите адрес сервера времени (в нашем случае 192.168.1.1). Обратите внимание: на рис. 34.2 показано, что служба времени Windows не запущена (не выполняется). Для ее запуска нажмите кнопку **Пуск**, введите команду `services.msc`, запустится окно **Службы**, в котором и нужно включить службы времени Windows (рис. 34.3). Для этого двойным щелчком щелкните на службе, а в открывшемся окне (рис. 34.4) выберите тип запуска **Авто** и нажмите кнопку **Применить**. После чего станет активной кнопка **Запустить**. Для запуска службы вам осталось нажать эту кнопку.

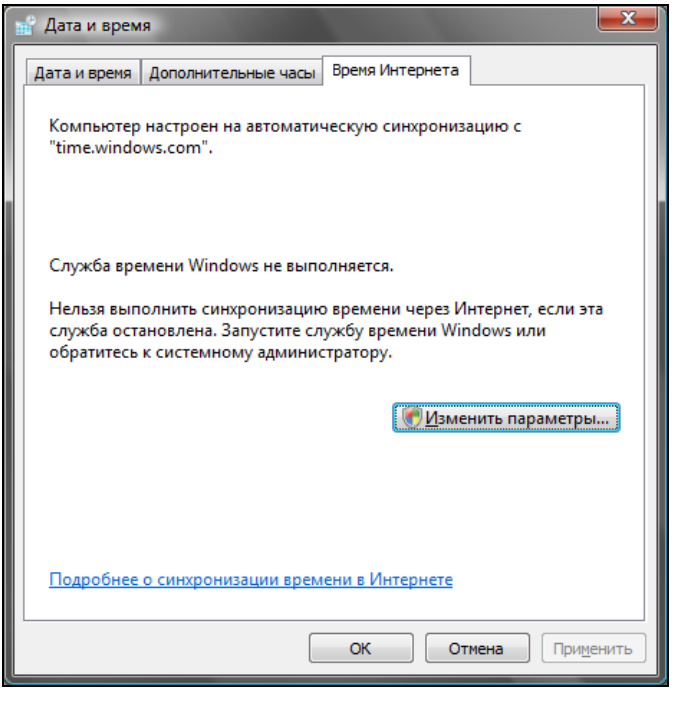


Рис. 34.1. Параметры даты и времени

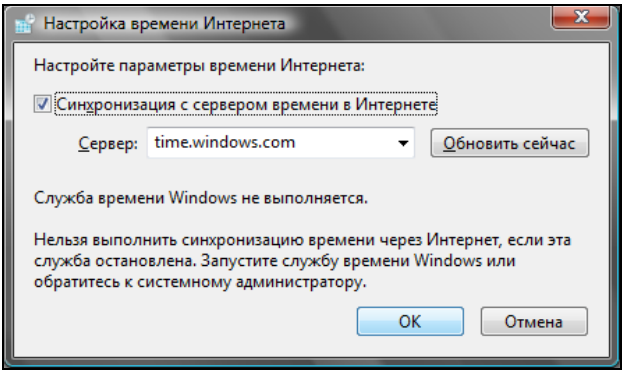


Рис. 34.2. Параметры синхронизации времени

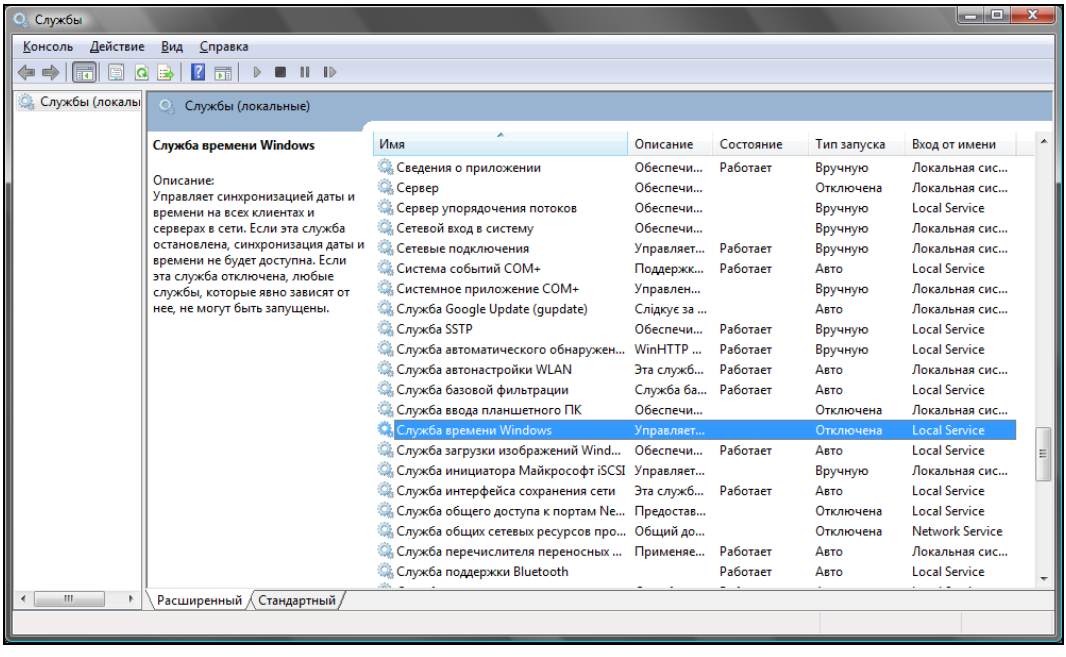


Рис. 34.3. Службы

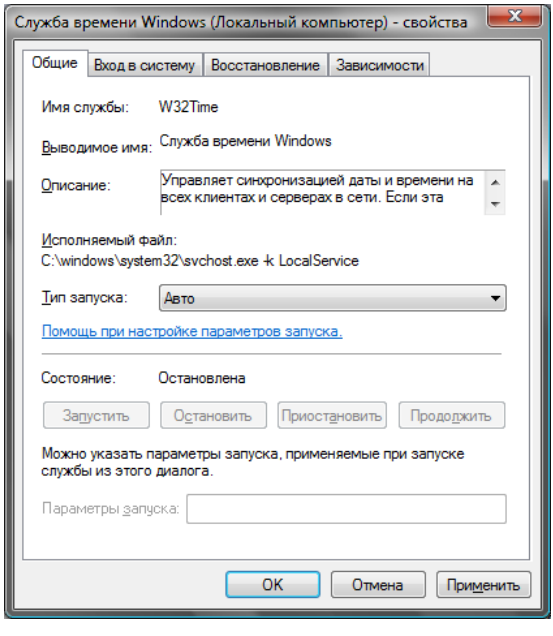
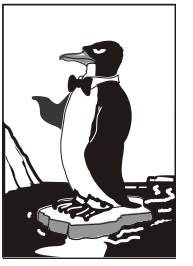


Рис. 34.4. Настройка службы



Глава 35

Сетевая файловая система NFS

35.1. Установка сервера и клиента

Сетевая файловая система NFS (Network File System) позволяет монтировать файловые системы, физически расположенные на удаленных компьютерах локальной сети. При этом работа с такой файловой системой осуществляется совершенно прозрачно, то есть создается ощущение, что файловая система локальная, а не удаленная. Конечно, скорость доступа будет меньше — ведь данные нужно еще передать по сети, да и команда монтирования не совсем простая. Но все это нюансы.

Сетевая файловая система по принципу своей работы чем-то напоминает общие файлы и папки в Windows — там тоже можно предоставить свои ресурсы другим пользователям. Конечно, реализация другая, но общий принцип почти такой же. Не нужно путать NFS с Samba, которая применяется для использования ресурсов сети Microsoft. Компьютеры, работающие под управлением Windows, не могут использовать NFS, равно как и с помощью NFS-клиента нельзя подключить общий ресурс Windows-станции. Поэтому первое, что нужно вам знать об NFS, — эта служба может работать только между UNIX-компьютерами.

Архитектура NFS ничем не отличается от обычной архитектуры клиент-сервер: в сети есть один (или несколько) NFS-серверов, к которым подключаются NFS-клиенты с целью монтирования сетевых файловых систем (примонтировать можно не все файловые системы сервера, а лишь те, которые разрешил администратор).

Для инсталляции сервера в Ubuntu/Debian нужно установить пакеты nfs-common и nfs-user-server, для инсталляции клиента хватит одного пакета nfs-common. В Mandriva/Fedora надо установить пакет nfs-utils — он содержит как NFS-сервер, так и NFS-клиент.

35.2. Настройка сервера

В файле /etc/exports прописываются экспортируемые файловые системы (которые могут монтировать удаленные пользователи). В листинге 35.1 приведен небольшой пример этого файла (по умолчанию файл пуст).

Листинг 35.1. Пример файла /etc/exports

```
/mnt/disk1 (ro, all_squash)
/mnt/upload admin.firma.ru(rw)
```

Формат этого файла следующий:

```
файловая_система [компьютер] (опции)
```

- первое поле файла — это экспортируемая файловая система. Она может экспортироваться на все компьютеры или же на один;
- поле `компьютер` не обязательное — его надо указывать, если требуется предоставить доступ только определенному компьютеру или же назначить специальные параметры доступа для определенного компьютера. Например, одна и та же файловая система может быть доступна всем компьютерам сети для чтения, а одному компьютеру сети — и для записи;
- третье поле `(опции)` — позволяет задать параметры доступа к файловой системе.

Проанализируем листинг 35.1. Файловая система /mnt/disk1 доступна всем компьютерам только для чтения. Файловую систему /mnt/upload может использовать только пользователь root компьютера admin.firma.ru. Доступ полный — чтение/запись.

Опции, которые можно использовать в файле exports, приведены в табл. 35.1.

Таблица 35.1. *Опции NFS*

Опция	Описание
`secure`	Запросы на монтирование файловой системы могут поступать от портов с номерами меньше 1024. Такие порты может создавать только root, поэтому соединение считается безопасным (его не могут создать обычные пользователи). Используется по умолчанию
`insecure`	Запросы могут поступать с любых портов
`ro`	Монтирование экспортируемой файловой системы возможно в режиме "только чтение"
`rw`	К экспортируемой файловой системе разрешен полный доступ. Используйте с осторожностью!
`noaccess`	Запрещает доступ к файловой системе. Может использоваться для запрещения доступа конкретному компьютеру: `/mnt/public comp.firma.ru (noaceess)`
`link_absolute`	Не изменяет символические ссылки. Используется по умолчанию
`link_relative`	Преобразует абсолютные ссылки в относительные
`all_squash`	Идентификаторы групп и пользователей будут преобразованы в анонимные
`no_all_squash`	Противоположна предыдущей опции. Используется по умолчанию
`root_squash`	Используется для преобразования всех запросов от root в запросы от анонимного пользователя. Используется по умолчанию
`no_root_squash`	Разрешает доступ к файловой системе от имени root. Противоположна опции `root_squash`

35.3. Монтирование удаленных файловых систем

Подмонтировать удаленную файловую систему можно с помощью все той же команды `mount`. Формат команды следующий:

```
mount -t nfs сервер:ФС точка_монтирования
```

Например,

```
mount -t nfs 192.168.1.1:/mnt/disk1 /mnt/remote
```

В нашем случае файловая система /mnt/disk1 экспортируется сервером 192.168.1.1. Она будет примонтирована к каталогу /mnt/remote. Параметр `-t` задает тип файловой системы — nfs.

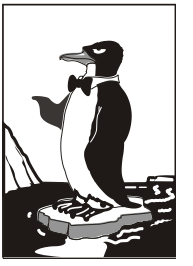
Если нужно, чтобы данная файловая система монтировалась автоматически при загрузке системы, в файл /etc/fstab нужно добавить следующую запись:

```
192.168.1.1:/mnt/disk1 /mnt/remote nfs bg,hard,rw 0 0
```

ЧАСТЬ VII

Безопасность в сети

Очень часто сетевые сервисы в Linux настраиваются по принципу "лишь бы работало". Да, практически любая сетевая служба будет работать сразу после установки и изменения базовых параметров (вроде имени компьютера и его IP-адреса). Но насколько безопасно она будет работать? Вся седьмая часть книги посвящена безопасности Linux-сервера при работе в сети.



Глава 36

Аудит сети с помощью nmap

36.1. Что такое nmap?

Программа nmap предназначена для сканирования как отдельных узлов, так и целых сетей с любым количеством узлов, определения состояния узлов сканируемой сети, а также открытых портов на этих узлах. Сетевой сканер nmap использует множество самых разных методов сканирования: UDP, TCP connect, TCP SYN, ICMP (ping) и т. д. Подробно о различиях в методах сканирования можно прочитать в справочной системе (команда `man nmap`) или по адресу: **http://cherepovets-city.ru/insecure/runmap/nmap_manpage-ru.htm**.

Сразу хочу отметить, что мы рассмотрим только практическое применение nmap, а теорию вы и сами сможете прочитать на русском языке по указанному здесь адресу (оригинальная страница руководства `man` на английском). Просто не вижу смысла переписывать страницу руководства, которую можно и так найти бесплатно в Интернете.

Сканер nmap поддерживает множество дополнительных возможностей: определение операционной системы узла, "невидимое" сканирование, вычисление времени задержки, параллельное сканирование, определение неактивных узлов с помощью параллельного ping-опроса, определение наличия брандмауэров, прямое RPC-сканирование, произвольное указание IP-адресов и номеров портов сканируемых сетей.

Результат работы сканера: список отсканированных портов удаленного узла с указанием номера и состояния порта, используемого протокола, а также названия службы, использующей этот порт. У порта может быть три состояния: открыт, фильтруемый, нефильтруемый. Первое состояние означает, что удаленная машина прослушивает данный порт. Второе состояние означает, что брандмауэр блокирует доступ к этому порту, и nmap не может определить его состояние. Третье состояние означает, что порт просто закрыт.

Примечание

Сканер nmap очень популярен. Он популярен до такой степени, что его можно увидеть даже в фильме "Матрица". В большинстве случаев — смотришь фильмы и видишь операционные системы какого-то непонятного происхождения. Создается впечатление, что все эти ОС создавались специально для фильма, чтобы не делать рекламы Microsoft или Apple. А тут старый добрый знакомый — nmap. Не верите? Значит, вы невнимательно смотрели "Матрицу", вот тот самый эпизод из фильма: **http://www.youtube.com/watch?v=0TJuipCrjZQ**.

А вот интервью с создателем nmap: **http://www.xakep.ru/post/20972/default.asp**.

36.2. Где мне взять nmap?

Сканер nmap — это не нечто мистическое. Это вполне реальная программа, причем абсолютно бесплатная и входит в состав многих дистрибутивов. Например, в Ubuntu, чтобы установить nmap, нужно ввести команду:

```
sudo apt-get install nmap
```

Если в вашем дистрибутиве не оказалось nmap, его можно скачать с официального сайта (там же вы найдете и Windows-версию): **http://nmap.org/download.html**.

36.3. Основы использования nmap

Теперь рассмотрим основы использования nmap. Синтаксис вызова nmap такой (запускать nmap нужно с привилегиями root):

```
# nmap параметры цель
```

Цель — это узел или список узлов для сканирования. Все опции мы рассматривать не будем — для этого есть страница руководства (см. ранее). Рассмотрим лишь самые интересные.

Предположим, что мы хотим знать, какая операционная система запущена на удаленном узле. Для этого нужно запустить nmap с опцией `-O`:

```
# nmap -O узел
```

Вот результат сканирования узла с запущенной Ubuntu 10.04:

```
Starting Nmap 5.21 ( nmap.org ) at 2010-08-23 10:28 EST
Nmap scan report for 192.168.1.1
Host is up (0.0040s latency).
Not shown: 999 closed ports
PORT STATE SERVICE
22/tcp open ssh
MAC Address: XX:XX:XX:XX:XX:XX
Device type: general purpose
Running: Linux 2.6.X
OS details: Linux 2.6.19 — 2.6.32
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at
nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.20 seconds
```

Как видно из вывода, узел работает под управлением Linux с ядром 2.6.19 — 2.6.32. Как раз, в Ubuntu 10.04 ядро 2.6.32. Также видно, что открыт один порт — 22 (ssh), все остальные порты закрыты.

Если nmap запущен для сканирования узла локальной сети, то он также сообщает и MAC-адрес узла (свой MAC-адрес здесь я скрыл).

Относительно любого сканирования нужно отметить, что администраторы не любят, когда несанкционированно сканируют их узлы. Во-первых, это не этично. Во-вторых, у вас могут быть проблемы — все зависит от организации, которую вы

сканируете. Само сканирование не противозаконно, но все мы понимаем, что просто так никто ничего не сканирует. Если вам хочется поэкспериментировать, можно использовать тестовый сервер nmap: **scanme.nmap.org**.

Сканер позволяет просканировать сразу несколько узлов. Для этого их адреса нужно указать так:

```
nmap 192.168.1.1-100
```

В приведенном примере будут просканированы узлы от 192.168.1.1 до 192.168.1.100. Можно также указать имена узлов через пробел, например, так:

```
nmap host1 host2
```

Чтобы просканировать узел на предмет открытых портов, укажите просто его адрес, никаких опций указывать не нужно, например:

```
nmap 192.168.1.2
```

Вывод будет примерно таким:

```
Interesting ports on den (192.168.1.2):
Not shown: 1712 closed ports
PORT STATE SERVICE
22/tcp open ssh
80/tcp open http

Nmap done: 1 IP address (1 host up) scanned in 0.240 seconds
```

Как уже отмечалось ранее, nmap позволяет также узнать список запущенных служб, для этого нужно использовать опцию `-sV`:

```
nmap -sV 192.168.1.2
```

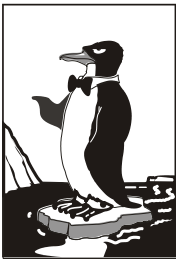
Вот вывод nmap:

```
Starting Nmap 5.21 ( nmap.org ) at 2010-08-23 10:45 EST
Nmap scan report for den (192.168.1.2)
Host is up (0.099s latency).
Not shown: 962 closed ports, 32 filtered ports
PORT STATE SERVICE VERSION
22/tcp open ssh OpenSSH 4.7p1 Debian 8ubuntu1.2 (protocol 2.0)
80/tcp open http Apache httpd 2.2.8 ((Ubuntu) PHP/5.2.4-2ubuntu5.10 with
Suhosin-Patch)
Service Info: OS: Linux

Service detection performed. Please report any incorrect results at
nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.51 seconds
```

Еще один интересный тип сканирования — кто онлайн? Сканер позволяет просканировать сеть и определить доступные узлы. Для этого служит опция `-sP`:

```
nmap 192.168.1.1-254
```

На этом все. Теперь, когда вы знаете, что есть nmap, для вас не составит особого труда ознакомиться со страницей руководства.



Глава 37

Защита сетевых сервисов

37.1. Защита Web-сервера

Apache — довольно безопасный сервис, поэтому его защита сводится к установке определенных прав доступа к его конфигурационным файлам. Для начала установим права 700 к каталогам /etc/httpd/conf и /var/log/httpd:

```
# chmod 700 /etc/httpd/conf
# chmod 700 /var/log/httpd
```

После этого никто, кроме вас, не сможет ни просмотреть, ни изменить конфигурационные файлы и файлы протоколов, которые обычно доступны всем желающим для чтения.

Также нужно защитить конфигурационный файл httpd2.conf от изменения:

```
# chattr +i /etc/httpd/conf httpd2.conf
```

После этого даже вы не сможете изменить данный файл. Если же вам понадобится отредактировать его, тогда нужно снять атрибут `i`:

```
# chattr -i /etc/httpd/conf httpd2.conf
```

37.2. Защита FTP

ProFTPD тоже является весьма защищенным сервисом, а его взлом — только следствием неправильной настройки.

Возьмем директиву `DefaultRoot`, задающую корневой каталог для сервера. Рекомендуется установить значение этой директивы в ~. Как мы знаем, тильда (~) означает домашний каталог пользователя. Следовательно, каждый раз при регистрации пользователя на FTP-сервере корневым каталогом FTP-сервера станет домашний каталог пользователя. В результате, пользователь не сможет прочитать (а при неправильных правах доступа — изменить) важные системные файлы.

Также рекомендуется включить директиву `RequireValidShell`:

```
RequireValidShell on
```

Если данная директива включена, тогда злоумышленник не сможет установить в качестве оболочки какую-нибудь вредоносную программу. FTP-сервер будет проверять, указана ли программа-оболочка в /etc/shells. Если программа не указана в этом файле, то FTP-сервер не будет ее запускать.

37.3. Защита DNS

Серверы DNS обмениваются между собой информацией о зоне. О том, как ограничить передачу зоны, мы говорили в *главе 27*. Но там мы ограничивали передачу зоны по IP-адресу. А что, если злоумышленник каким-то образом подменил целевой DNS-сервер с указанным адресом (например, вывел его из строя и запустил собственный с таким же IP)? Тогда информация о зоне будет передана на сервер злоумышленника.

Чтобы такого не случилось, нужно использовать механизм транзакций TSIG (Transaction SIGnatures). Этот механизм предусматривает перед передачей зоны проверку секретного ключа. Если ключ совпадает, информация о зоне будет передана/принята. Если же ключ не совпадает, информация о зоне не будет передана или не будет принята (если злоумышленник вывел из строя первичный DNS-сервер и пытается передать измененную информацию о зоне на вторичный DNS-сервер).

Первым делом, нужно сгенерировать ключ, который затем указать в файле конфигурации каждого сервера. Для этого используется команда:

```
# dnssec-keygen -a hmac-md5 -b 128 -n HOST host1-host2
```

Вам изменять эту команду не стоит. Команда выведет на экран следующую строку:

```
Khost1-host2.код
```

Также в результате выполнения этой команды будут созданы два файла: Khost1-host2.код.key и host2.код.private. Откройте второй файл. В нем будут следующие строки:

```
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: ключ
```

Строку `ключ` нужно скопировать в буфер обмена (или записать на бумаге, если у вас нет графического интерфейса). После этого добавьте в файлы named.conf обоих DNS-серверов (первичного и вторичного) следующие строки:

```
key host1-host2 {
  algorithm hmac-md5;
  secret "ключ";
};
```

> ***Примечание***
>
> Директиву `key` нужно добавить в начало файла конфигурации.

А теперь будьте внимательны и следуйте моим инструкциям. Откройте файл первичного DNS-сервера и добавьте следующие строки после директивы `key`:

```
server 10.0.0.2  {
key { host1-host2; };
};
```

Затем в директиву options добавьте `allow-transfer`, если вы этого еще не сделали:

```
allow-transfer { 10.0.0.2; };
```

Директива `server` указывает, что для обмена зоной с сервером 10.0.0.2 (это IP-адрес вторичного сервера) нужно использовать ключ `host1-host2`.

Теперь откройте файл конфигурации вторичного DNS-сервера. Добавьте следующие строки:

```
server 10.0.0.1  {
key { host1-host2; };
};
```

В директиву `options` нужно добавить вот такую директиву:

```
allow-transfer { none; };
```

После всего этого нужно перезапустить DNS-сервер:

```
# service named restart
```

37.4. Защита Samba

По умолчанию доступ к серверу Samba предоставляется всем желающим. Это не всегда необходимо. В целях большей безопасности нужно предоставлять доступ только определенным пользователям. Создать пользователей можно с помощью команды `adduser` (в Fedora нужно использовать конфигуратор system-config-users):

```
adduser -s /bin/false win_user
passwd win_user
smbpasswd win_user
```

Первая команда добавляет пользователя и устанавливает в качестве его командной оболочки программу /bin/false, которая запускается, возвращает код **0** и завершает работу. Даже если кто-то узнает пароль пользователя, /bin/false не позволит пользователю зарегистрироваться в системе обычным способом.

Вторая команда устанавливает пароль пользователя. Поскольку пользователь не сможет войти в систему, можно установить любой пароль и даже не запоминать его.

А вот третья команда изменяет пароль пользователя, который он должен будет указать при регистрации на сервере Samba. Этот пароль нужно сообщить пользователю.

Затем следует открыть smb.conf и в секции `global` изменить параметр `security`:

```
security = user
```

37.5. DHCP — привязка к MAC-адресу

Раньше все компьютеры сети настраивались вручную, сейчас настройкой узла занимается DHCP-сервер. Следовательно, если узел сети не получит настроек от DHCP-сервера, то и доступ к сети он не получит.

Со стороны DHCP-сервера можно блокировать доступ нежелательных компьютеров, точнее, разрешить передачу настроек только тем компьютерам, которым нужно. Делается это путем привязки IP-адресов к MAC-адресам (аппаратным адресам сетевых адаптеров). При такой настройке мы "убиваем сразу двух зайцев":

- одному и тому же компьютеру (точнее MAC-адресу) будет всегда назначаться один и тот же IP-адрес, что очень удобно, если система статистики подсчитывает трафик по IP-адресу без аутентификации пользователя;

- компьютеры, MAC-адреса сетевых адаптеров которых вы не "прописали" в конфигурационном файле DHCP-сервера, не получат доступ к сети, потому что не получат сетевые настройки.

Следует понимать, что как дополнительный барьер защита средствами DHCP-сервера вполне сойдет, но в целом она весьма посредственна. Дело в том, что даже если узел не получит сетевые настройки, то их можно указать вручную. А узнать их для злоумышленника не составит особого труда.

Кроме средств DHCP-сервера, нужно еще контролировать пространство IP-адресов вашей сети. Например, вы выделяете своим клиентам адреса из диапазона 192.168.1.50 — 192.168.1.100 (с помощью DHCP), но злоумышленник может указать IP-адрес 192.168.1.101. Если дальше никакие средства (ни прокси-сервер, ни брандмауэр) не осуществляют контроль IP-адресов, толку от контроля MAC-адресов не будет.

К тому же MAC-адрес довольно легко подделать. В Linux MAC-адрес для интерфейса eth0 (первая сетевая плата — у многих она не только первая, но и единственная) можно изменить командами типа:

```
# ifconfig eth0 down
# ifconfig eth0 hw ether XX:XX:XX:XX:XX:XX
# ifconfig eth0 up
```

Здесь: `XX:XX:XX:XX:XX:XX` — MAC-адрес.

В Ubuntu (и других дистрибутивах, где используется Network Manager) MAC-адрес можно изменить с помощью графического интерфейса, конфигуратор YaST в openSUSE, если я не ошибаюсь (а проверять лень, потому что сам для смены адреса использую приведенные ранее команды), тоже позволяет изменить MAC-адрес. В Windows MAC-адрес можно изменить в свойствах сетевой платы, на вкладке **Дополнительно**, свойство **Сетевой адрес** (рис. 37.1).

После смены MAC-адреса, нужно проверить, установился ли он:

- В Linux командой: `ifconfig -a | grep Hwaddr`;
- В Windows: `ipconfig /all` или командой `getmac` (рис. 37.2);
- В FreeBSD: i`fconfig|grep ether`.

Спрашивается, а как злоумышленник узнает, какой MAC-адрес допустим? Для этого ему достаточно подключиться (физически) к вашей сети, что он уже и сделал, раз пытается узнать MAC-адрес (и вправду, зачем мне MAC-адрес одного из компьютеров Пентагона, если я не собираюсь подключаться к его сети?). После этого он может запустить одну из программ для сбора MAC-адресов, например, TCPNetView (вы ее без проблем найдете в Интернете — программа распространяется бесплатно). Кстати, эта программа полезна и для системного администратора — ведь вам же не хочется вводить команду определения MAC-адреса на каждом компьютере? Вы запускаете TCPNetView и получаете MAC-адреса всех подключенных к сети в данный момент компьютеров. Удобно? Я тоже так думаю.

Учитывая все сказанное ранее, можно сделать вывод: защита средствами DHCP-сервера подходит больше для внутреннего контроля, нежели для защиты от взлома сети. Но, повторяю, как дополнительный барьер она вполне действительна.

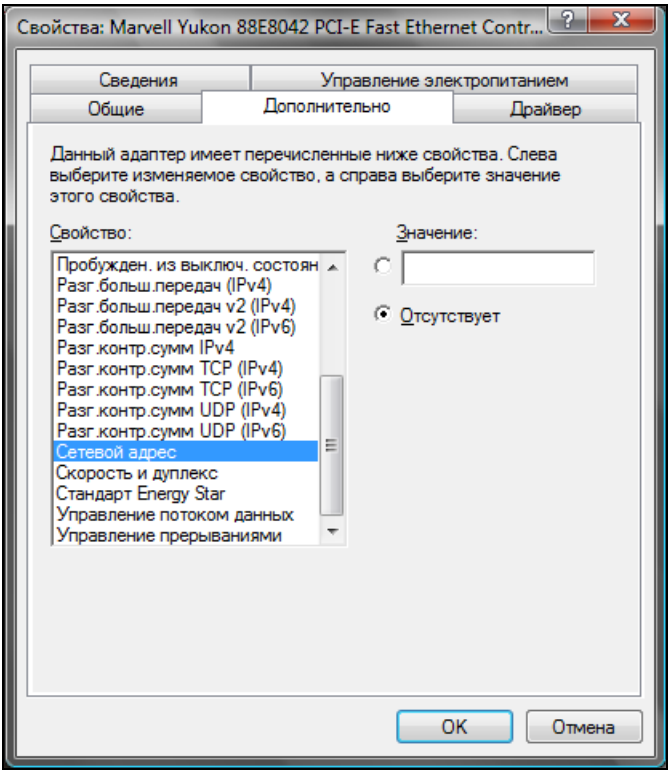


Рис. 37.1. Изменение MAC-адреса сетевого адаптера в Windows

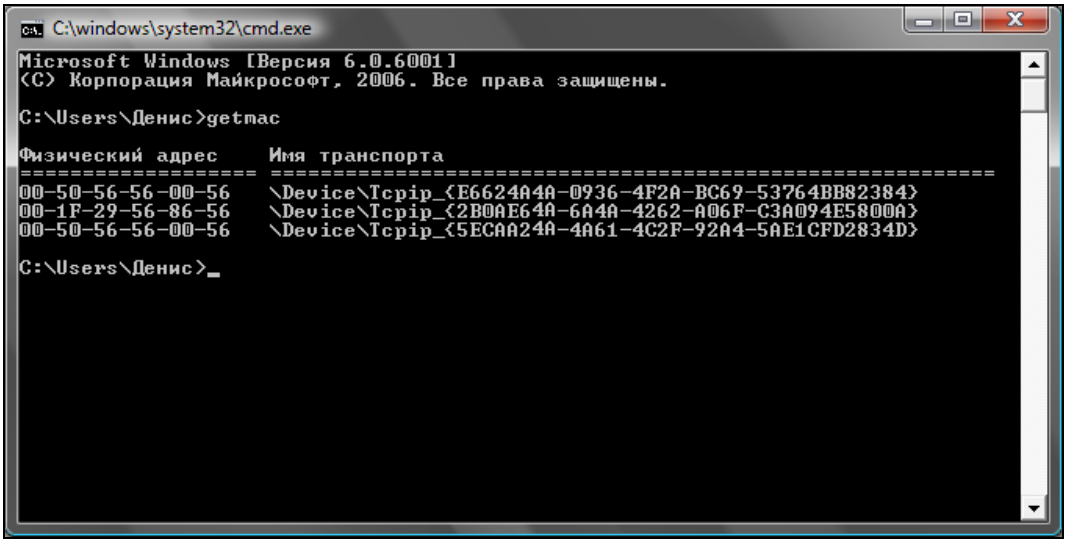


Рис. 37.2. Команда getmac в Windows

Если вы таки надумали реализовать привязку IP-адресов к MAC-адресам, тогда для каждого компьютера сети добавьте в конфигурационный файл следующую конструкцию:

```
host compN {
hardware ethernet xx:xx:xx:xx:xx:xx;
fixed-address IP-адрес;
}
```

Все эти инструкции (для каждого компьютера) нужно добавить в секцию `subnet`, например:

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    …
    host comp3 {
           hardware ethernet 00:40:AA:24:70:2E;
           fixed-address 192.168.1.3;
         }
}
```

37.6. Защита от спама: Greylisting и Qmail

Спам — одна из потенциальных угроз для вашего сервера. Одно дело, что спам — это просто неприятно. Но совсем другое, когда поток спама "забивает" весь жесткий диск вашего сервера, и тогда почтовый сервер вообще не будет функционировать. Это типичный пример атаки на отказ (DoS, Denial of Service).

Для борьбы со спамом используются две общепринятые стратегии: черные списки и серые списки. Что такое черный список, знают все — в черный список заносят адреса машин, с которых производится рассылка спама. Впоследствии наш почтовый сервер отказывается принимать почту с этих адресов.

Однако сейчас редко используются статические адреса — только для серверов, а клиентам в основном назначают динамические адреса, и спамер подключается к Интернету как обычный клиент. Пусть используемый спамером динамический адрес будет внесен в черный список. Но на следующий день этот адрес будет назначен нормальному законопослушному пользователю, и когда он попытается отправить почту, то получит сообщение, что его адрес внесен в черный список.

Серые списки работают иначе. Если адрес компьютера есть в сером списке, то почтовый сервер в первый раз отправляет этому клиенту ответ, что сервер занят и что нужно повторить попытку позже. Спамер не будет повторять отправку письма — ему это не нужно, поскольку он уже начал отправлять письмо по другому адресу. А вот нормальный пользователь, получив такой ответ от сервера, через некоторое время повторит отправку письма. После этого адрес клиента будет внесен в белый список, и сервер всегда будет принимать от него письма.

Здесь мы рассмотрим, как интегрировать механизм серых списков (Greylisting) с Qmail — поскольку в этой книге рассматриваем именно этот почтовый сервер. А вот по следующему адресу вы найдете инструкции по интеграции greylisting с другими MTA: **http://spamlinks.net/filter-server-greylist.htm**.

Итак, нам понадобятся:

- уже установленный MTA Qmail или netqmail (**http://netqmail.org/**);
- патч QMAILQUEUE (**http://www.qmail.org/qmailqueue-patch**), позволяющий вставить другую программу в структуру очереди Qmail;
- программа qmail-qfilter (http://untroubled.org/qmail-qfilter/), выполняющая фильтрацию писем.

Теперь надо создать скрипт /var/qmail/control/qmail-qfilter/greylisting.qfilter:

```
exec /var/qmail/bin/qmail-qfilter \
/var/qmail/control/qmail-qfilter/greylisting.qfilter -- \
/var/qmail/control/qmail-qfilter/yourfilter1 -- \
/var/qmail/control/qmail-qfilter/yourfilter2
```

И установить права доступа к нему:

```
chown root:qmail /var/qmail/control/qmail-qfilter/greylisting.qfilter
chmod 0755 /var/qmail/control/qmail-qfilter/greylisting.qfilter
```

Скопируйте cron-скрипт Greylisting в каталог crontab и установите права доступа:

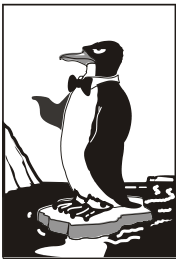
```
chown root:root greylisting.cron
chmod 0755 greylisting.cron
```

Данный скрипт управляет базой данных Greylisting. Для самой базы данных нужно создать каталог и установить разрешения:

```
mkdir -p /var/greylisting
chown qmaild:nofiles /var/greylisting
chmod 0700/var/greylisting
```

Затем установите переменную окружения QMAILQUEUE — в ней нужно указать путь к сценарию /var/qmail/control/qmail-qfilter/greylisting.qfilter. Более подробные инструкции вы найдете по адресу:

http://spamlinks.net/filter-server-greylist.htm#implement-qmail.



Глава 38

Оптимизация сервера

38.1. Общая оптимизация Linux

В этом разделе мы поговорим об общей оптимизации Linux — как сервера, так и рабочей станции, а в следующем — рассмотрим оптимизацию отдельных сетевых сервисов.

38.1.1. Оптимизация подкачки

Операционная система Linux не очень требовательна к оперативной памяти — для нормальной работы даже шлюза небольшой сети вполне хватит 64 Мбайт ОЗУ. Не верите? Посмотрите на рис. 38.1 — из 128 Мбайт использовано всего 33 Мбайт, а своп вообще не используется.

```
[root@localhost ~]# free
             total       used       free     shared    buffers     cached
Mem:        126284      33244      93040          0       4584      16152
-/+ buffers/cache:      12508     113776
Swap:       240964          0     240964
[root@localhost ~]# _
```

Рис. 38.1. Команда `free` — сведения об использовании оперативной памяти

Но это только в том случае, если не запущена система X.Org. После ее запуска Linux превращается в настоящего "обжору", съедающего десятки мегабайтов памяти.

Сама система X.Org тоже не особенно требовательна к памяти, чего не скажешь о графических интерфейсах GNOME и KDE. При использовании GNOME или KDE для комфортной работы необходимо минимум 256 Мбайт оперативной памяти.

Ваша система может работать, мягко говоря, не очень быстро только потому, что ей не хватает оперативной памяти.

Сейчас попытаемся определить, хватает ли вам ОЗУ. Запустите необходимые вам сервисы (например, Apache, ProFPTD, sendmail и т. д.) и попробуйте смоделировать максимальную нагрузку на сервер. Сделать это будет не просто, но все же постарайтесь. Можно кого-нибудь попросить, например, всех пользователей сети, чтобы они одновременно обратились к вашему серверу.

Затем введите команду `free` и посмотрите, сколько мегабайтов оперативной памяти у вас свободно. Также обратите внимание на "остаток" области подкачки (swap). Если и там, и там осталось всего несколько мегабайтов памяти, значит, вам пора покупать еще один модуль ОЗУ. Временно, пока вы его не купили, можно создать на жестком диске дополнительный файл подкачки, что несколько повысит производительность системы. Хочу обратить ваше внимание на то, что это временная мера, ведь производительность жесткого диска существенно ниже производительности оперативной памяти, следовательно, даже если вы добавите файл в 1 Гбайт к области подкачки, это все равно не сравнится с одним настоящим модулем ОЗУ на 256 Мбайт.

В *главе 9* мы научились создавать файл подкачки. Но одного добавления свопфайла мало. Нужно еще оптимизировать работу системы свопинга с помощью коэффициента подкачки. Значение этого коэффициента хранится в файле /proc/sys/vm/swappiness. Минимальное значение коэффициента 0, максимальное — 100. Значение по умолчанию 70.

Теперь о том, как правильно выбрать оптимальное значение. Если вы в основном работаете с небольшими программками и часто переключаетесь между ними, можно установить значение меньше 50, например, 40 или даже 30. В этом случае переключение между приложениями будет мгновенным, но замедлится их работа. Но поскольку эти приложения небольшого размера, то вы этого не заметите.

Если же вы в основном работаете на протяжении дня с громоздкими приложениями, например, Open Office, или занимаетесь обработкой изображений в GIMP, вам лучше установить значение коэффициента, превышающее 70, например, 80 или даже 85. В этом случае переключение между приложениями будет медленное, зато ваше основное приложение будет работать быстро.

Изменить значение коэффициента можно с помощью команды:

```
# echo "значение" > /proc/sys/vm/swappiness
```

Например:

```
# echo "50" > /proc/sys/vm/swappiness
```

38.1.2. Изменение планировщика ввода/вывода

Производительность многозадачной системы в целом сильно зависит от правильного планирования процессов системы. Сейчас мы попытаемся с помощью параметра ядра `elevator` установить нужный нам алгоритм работы ядра, что

позволит существенно повысить производительность системы. Допустимы следующие значения этого параметра:

- `none` — значение по умолчанию;
- `as` — упреждающее планирование;
- `cfg` — "честная очередь";
- `deadline` — планирование крайних сроков.

Для домашнего компьютера больше подойдут значения `as` и `cfq`. В первом случае ядро будет пытаться "угадать" ход программы, а именно — какую операцию ввода/вывода программа "захочет" выполнить в следующий раз. Если ядро будет правильно "угадывать", то производительность системы должна существенно увеличиться. Ясно, что работа данного алгоритма очень зависит от логики программы.

Во втором случае (значение `cfq`) ядро будет равномерно планировать операции ввода/вывода. Данный алгоритм будет работать лучше первого в случае с запутанной логикой программы, когда невозможно предугадать ее следующую операцию.

Последнее значение (`deadline`) больше подходит для сервера, чем для рабочей станции, поэтому существенного прироста от него не ждите.

При загрузке передать параметр ядру можно так:

```
linux elevator=значение
```

Чтобы не вводить параметр каждый раз при загрузке, добавьте его в файл конфигурации загрузчика. Если у вас GRUB, то одна из секций конфигурационного файла /etc/grub/grub.conf будет выглядеть так (листинг 38.1):

Листинг 38.1. Фрагмент файла /etc/grub/grub.conf

```
...
title Linux
root (hd1,0)
kernel /boot/vmlinuz-2.6.9 ro root=/dev/hda1 elevator=as
...
```

Если у вас GRUB2, то откройте файл /etc/default/grub и измените параметр `GRUB_CMDLINE_DEFAULT`, например, так:

```
GRUB_CMDLINE_LINUX_DEFAULT="elevator=as"
```

После изменения файла /etc/default/grub не забудьте запустить команду `update-grub` для обновления вашего /boot/grub/grub.cfg.

38.2. Оптимизация сетевых сервисов

После общей оптимизации сервера нужно занять оптимизацией отдельных сетевых сервисов. Особую нагрузку на сервер производит сервис Samba, поэтому с него и начнем.

38.2.1. Секреты оптимизации Samba

Если открыть файл конфигурации smb.conf, вы найдете в нем параметр `wide links`. Никогда не устанавливайте его в `no`! Так вы существенно снизите производительность Samba. Наоборот, если вы установите его в `yes` (если до этого параметр `wide links` был отключен), то вы можете существенно повысить производительность. Дело в том, что параметр `wide links` определяет, как Samba будет следовать по символическим ссылкам. Сначала Samba следует по символической ссылке, а затем выполняет так называемый `directory path lookup` (системный вызов, определяющий, где завершилась ссылка). Если `wide links = no`, то Samba не будет следовать по символическим ссылкам вне экспортируемой области. Данная операция подразумевает на 6 системных вызовов больше, нежели в случае, если `wide links = yes`. Учитывая, что подобных операций делается очень много, то выключение `wide links` снижает производительность Samba приблизительно на 30%.

Протокол TCP/IP — штука тонкая. Производительность сетевых приложений во многом зависит от того, правильно ли настроен TCP/IP. Samba — настоящее сетевое приложение, которое к тому же работает по протоколу TCP/IP. При использовании TCP/IP, если размер запросов и ответов не фиксирован (как в случае с Samba), рекомендуется применять протокол TCP с опцией `TCP_NODELAY`. Для этого в файл smb.conf нужно добавить строку:

```
socket options = TCP_NODELAY
```

Тесты показывают, что с указанными опциями Samba при больших нагрузках работает в три раза быстрее, чем без указания этих опций. Если Samba используется в локальной сети (в большинстве случаев так оно и есть), рекомендуется еще указать и такую опцию `IPTOS_LOWDELAY`:

```
socket options = IPTOS_LOWDELAY TCP_NODELAY
```

Если есть желание "выжать" из Samba еще больше, тогда установите следующие параметры буферизации: `SO_RCVBUF=8192 SO_SNDBUF=8192`. Например:

```
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
```

38.2.2. Оптимизация ProFTPD

Оптимизировать ProFTPD можно по трем направлениям: ускорить авторизацию, равномерно распределить нагрузку на сервер и помочь серверу избежать перегрузки "узкого" канала.

Начнем с авторизации. Ускорить авторизацию помогут директивы `IdentLookup` и `UseReverseDNS`. Первая управляет использованием протокола ident, но поскольку этот протокол давно не используется, данную директиву можно отключить. Вторая определяет доменное имя клиента по его IP-адресу, но это занимает некоторое время, поэтому для ускорения доступа к FTP-серверу ее тоже нужно отключить. Добавьте в файл конфигурации proftpd.conf следующие строки:

```
IdentLookups off
UseReverseDNS off
```

К авторизации так же относится директива `MaxLoginAttempts`, задающая максимальное число попыток регистрации пользователя на сервере. Ограничьте его следующим значением:

```
MaxLoginAttempts 3
```

Теперь приступим к распределению нагрузки на сервер. Первым делом нужно задать максимальное число клиентов:

```
MaxClients число
```

Понятно, что чем быстрее наш канал подключения к Интернету, тем больше клиентов сервер сможет принять.

С помощью директивы `MaxClientsPerHost` можно установить максимальное число клиентов с одного узла:

```
MaxClientsPerHost число
```

Не нужно устанавливать для этой директивы значение 1. Представьте, что есть сеть, доступ к Интернету которой осуществляется через один сервер — шлюз, имеющий один реальный IP. Получается, что у всех пользователей этой сети один IP. Если установить параметр `MaxClientsPerHost` в 1, то из всей сети на наш FTP сможет зайти только один пользователь. Понятно, что все пользователи сети тоже не будут одновременно заходить на наш FTP, поэтому для директивы `MaxClientsPerHost` нужно установить небольшое значение, например, 2 или 3.

Предположим, что доступ к нашему FTP разрешен только зарегистрированным (а не анонимным) пользователям. Но некоторые пользователи могут "одолжить" свой логин и пароль другим, незарегистрированным на сервере пользователям, чтобы они тоже смогли использовать ресурсы нашего сервера. Это нехорошо, поэтому с помощью директивы `MaxClientsPerUser` мы можем контролировать максимальное число FTP-клиентов от одного пользователя. Вот тут самое время установить значение 1:

```
MaxClientsPerUser 1
```

Но пользователи хотят нас обхитрить. Они заходят под одним и тем же логином, но с разных узлов (например, с разных сетей). Нужно запретить им делать это:

```
MaxHostsPerUser 1
```

Директива `MaxHostsPerUser`, как понятно из ее названия, ограничивает количество узлов на одного пользователя.

Еще нужно установить директиву `MaxInstances`, задающую максимальное число параллельно запущенных экземпляров сервера proftpd (для каждого нового клиента запускается своя копия profptd для обработки его запросов). Ее значение зависит от возможностей вашего сервера. Предположим, что для директивы `MaxClients` мы задали значение 10, то есть одновременно могут работать 10 пользователей. Поскольку мы установили для `MaxClientsPerUser` и `MaxHostsPerUser` значение 1, то для `MaxInstances` можно установить значение 10. Но если мы разрешим использовать каждому пользователю более одного FTP-клиента или разрешим регистрироваться одновременно с разных узлов под одним и тем же логином, тогда нужно увеличить `MaxInstances`. Например, если для `MaxHostsPerUser` мы установили значение 2, то `MaxInstances` будет равен 20 (2×10). В общем, вам, учитывая три значе-

ния (MaxClients, MaxClientsPerUser и MaxHostsPerUser), нужно высчитать максимальное значение MaxInstances, чтобы в моменты пиковой нагрузки все клиенты получили доступ к серверу:

```
MaxInstances 10
```

С помощью директивы MaxLoginAttempts можно задать, сколько раз пользователь может ввести пароль. После последней попытки сервер разорвет соединение. Рекомендуемое значение — 3.

MaxRetrieveFileSize — максимальный размер получаемого файла. Можно не устанавливать, потому как файлы, загружаемые на сервер вами, будете контролировать вы сами, а файлы, которые загружают пользователи, — с помощью следующей директивы (MaxStoreFileSize). Если никто не "зальет" на сервер файл размером, скажем, в 1 Гбайт, то никто не сможет и скачать этот файл.

MaxStoreFileSize — максимальный размер файла, загружаемого на сервер пользователями. Тут все зависит от "ширины" канала и места на диске, даже больше от второго, нежели от первого. Решайте сами…

Нам осталось ограничить скорость передачи данных, чтобы сервис FTP не узурпировал под себя весь трафик. Особенно это важно, если канал "узкий", и на сервере запущены другие сетевые сервисы, например, Apache.

Ограничить пропускную способность можно или с помощью устаревших директив Rate* или с помощью новой TransferRate. Последнюю использовать удобно, если сервер подключен к Интернету по синхронному каналу. Если же сервер подключен по асинхронному каналу, то есть скорости приема и передачи — разные, удобнее использовать директивы Rate*, потому что они могут ограничить как скорость чтения, так и скорость записи:

- RateReadBPS байт-в-секунду — задает скорость чтения данных в байтах в секунду;
- RateWriteBPS байт-в-секунду — максимальная скорость записи данных в байтах в секунду;
- TransferRate байт-в-секунду — одновременно ограничивает как скорость чтения, так и записи.

38.2.3. Оптимизация Apache

Конфигурационный файл сервера Apache httpd.conf находится в каталоге /etc/apache или в /etc/httpd/conf (в зависимости от дистрибутива и версии Apache). В этом файле, как и в proftpd.conf, есть директива MaxClients, позволяющая ограничить число одновременно работающих клиентов.

Чтобы правильно установить это значение, нужно знать, сколько пользователей может одновременно зайти на сервер. При небольшой посещаемости вполне хватит значения 30−50, при большой загрузке количество одновременно работающих клиентов может исчисляться сотнями. Следите за посещаемостью вашего сервера и корректируйте это значение, иначе какая-то часть пользователей может остаться "за бортом", а им это очень не понравится (или же, наоборот, ресурсы сервера будут использоваться нерационально).

Директива `StartServers` задает количество экземпляров сервера, которые будут созданы при запуске исходной копии сервера. Для этой директивы можно установить значение, равное 10% от `MaxClients`. Устанавливать большое значение не нужно, поскольку вы будете нерационально использовать ресурсы компьютера.

Рассмотрим обычную ситуацию. Для `MaxClients` вы установили значение 200, а для `StartServers` — 20. Запросы первых 20 клиентов будут обрабатываться очень быстро, поскольку сервисы уже запущены. Запрос 21 клиента будет обслужен чуть медленнее, поскольку нужно запустить еще одну копию Apache, но не нужно устанавливать в нашем случае (`MaxClients = 200`) для `StartServers` значение больше 20 — ведь не всегда даже 20 человек одновременно заходит на сервер.

Если же на сервере постоянно находится как минимум 20 человек, тогда нужно увеличить и `MaxClients`, и `StartServers`. Хотя бывают исключения, например, для сервера внутренней корпоративной сети. Вы точно знаете, сколько клиентов в вашей сети, следовательно, можно точно знать, какое значение установить для `MaxClients` и `StartServers`. Но, все равно, для `MaxClients` нужно установить чуть большее значение, чем для `StartServers`, — на всякий случай:

```
MaxClients 150
StartServers 100
```

Чтобы еще эффективнее оптимизировать работу Web-сервера, нужно знать, как он работает. Клиент посылает запрос, Web-сервер его обрабатывает и посылает клиенту ответ. После этого соединение можно закрывать и завершать копию Apache, обслуживающую это соединение. Как видите, постоянных соединений, как в случае с FTP-сервером, здесь нет.

Но зачем завершать копию Web-сервера, если сейчас же на сайт зайдет другой пользователь, и опять нужно будет запускать еще одну копию сервера, а это увеличивает загрузку процессора. Поэтому с помощью директивы `MaxSpareServers` можно установить максимальное число серверов, которые будут находиться в памяти уже после закрытия соединения с пользователем — они просто останутся ждать своего пользователя.

Теоретически, чтобы сбалансировать нагрузку, значение для `MaxSpareServers` можно установить такое же, как и для `StartServers`, то есть 10% от `MaxClients`.

Вы не задумывались, что если Web-сервер будет работать в режиме постоянного соединения, как в случае с FTP, то это повысит его производительность? Если вы подумали об этом, то мыслите в правильном направлении. Представим, что у нас на сайте есть форум. Человек редко заходит на форум, чтобы посмотреть одну страничку. Обычно он может находиться на форуме часами. Так зачем же закрывать соединение? Чтобы потом опять тратить время и ресурсы сервера на его открытие? Разрешить постоянные соединения можно с помощью директивы `KeepAlive`. Она задает максимальное число таких соединений:

```
KeepAlive 5
```

А директива `KeepAliveTimeout` задает тайм-аут для постоянного соединения в секундах:

```
KeepAliveTimeout 15
```

Используя все упомянутые в этом разделе директивы, вы сможете добиться существенного повышения производительности вашего Web-сервера.

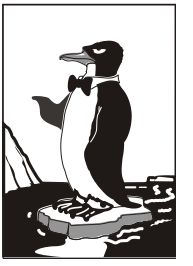
38.2.4. Оптимизация SSH

По адресу **http://www.psc.edu/networking/projects/hpn-ssh/** находится страничка проекта High Performance SSH/SCP — HPN-SSH. Цель данного проекта — ускорение SSH.

На страничке проекта вы найдете патчи, ускоряющие копирование файлов по SCP (Secure copy) в 10 раз. Патчи применяются к исходным кодам OpenSSH, поэтому вам придется их скачать, применить патчи и перекомпилировать OpenSSH. "Исходники" SSH можно скачать на сайте **http://www.openssh.com**.

Сразу скажу, что максимальное ускорение SSH достигается за счет отключения шифрования, то есть в ущерб безопасности. Зачем тогда использовать SSH? Но если производительность для вас важнее, чем безопасность, то можете применить патчи HPN-SSH.

Если вы уже установили RPM/DEB-пакет с OpenSSH, то удалите его. Только после этого откомпилируйте измененные исходные тексты OpenSSH. Помните, чтобы был толк, нужно использовать измененные версии OpenSSH как на сервере, так и на клиенте.



Глава 39

Chroot-окружение

39.1. Песочница

Представьте себе детей, играющих в песочнице. Песочница у нас не обычная, а с высокими бортами, поэтому дети не могут самостоятельно из нее выбраться. Понятно, что в песочнице дети в большей безопасности, нежели чем за ее пределами.

Теперь представим, что дети — это системные и сетевые службы. Система безопасности, называемая chroot-окружением, похожа на песочницу, но она предназначена не для защиты пользователей, а для защиты системы от действий пользователей.

Разберемся, как работает chroot-окружение. Предположим, что у нас есть сетевая служба, например, FTP-сервер. Если пользователю удастся каким-то образом взломать эту службу, то он получит доступ к корневой файловой системе сервера, что нежелательно. Чтобы этого не произошло, сервис изолируют от корневой файловой системы сервера, организуя "песочницу", в которую вызываемую сетевую службу и помещают. Таким образом, при создании chroot-окружения создается набор файлов, содержащий все необходимое для запуска того или иного сетевого сервиса. Под набором файлов подразумевается отдельный каталог, в который копируются все необходимые файлы: конфигурационный файл, исполняемые файлы самого сервиса, библиотеки, вспомогательные программы. Затем производится системный вызов `chroot`, делающий подмену файловой системы, и наш сетевой сервис запускается уже внутри chroot-окружения. Если даже пользователь взломает сервис, то он получит доступ не к файловой системе сервера, а к файловой системе chroot-окружения. Что бы он там ни сделал, его действия не причинят системе никакого вреда.

Но дети взрослеют и со временем могут выйти за пределы песочницы. Точно так же растут возможности злоумышленников — выход за пределы chroot-окружения возможен. Но, несмотря на это, chroot-окружение остается очень мощным барьером для злоумышленников.

39.2. Пример создания chroot-окружения

Давайте рассмотрим создание chroot-окружения для Web-сервера Apache. Первым делом, нужно создать каталог, в котором мы будем формировать chroot-окружение. Пусть это будет каталог chroot:

```
# mkdir /chroot
```

Далее нужно создать все необходимые для работы Apache каталоги (позже мы скопируем в них необходимые файлы):

```
# mkdir -p /chroot/etc
# mkdir -p /chroot/dev
# mkdir -p /chroot//usr/lib
# mkdir -p /chroot/usr/libexec
# mkdir -p /chroot//usr/local/apache/bin
# mkdir -p /chroot//usr/local/apache/logs
# mkdir -p /chroot/usr/local/apache/conf
# mkdir -p /chroot/var/www/html
# mkdir -p /chroot/var/run
```

Установим права доступа:

```
# chown -R root:sys /chroot
```

После этого мы должны создать устройства dev/log и dev/null. Первое необходимо для нормальной работы демона syslogd в chroot-окружении, а второе будет использоваться в качестве домашнего каталога Web-сервера:

```
# mknod /chroot/dev/null c 2 2
# chown root:sys /chroot/dev/null
# chmod 666 /chroot/dev/null

# mknod /chroot/dev/log c 21 5
# chown root:sys /chroot/dev/log
# chmod 666 /chroot/dev/log
```

Теперь скопируем все необходимые для работы Web-сервера файлы:

- конфигурационные файлы (находятся в каталоге /etc/httpd2);
- каталог документов (/var/www/html);
- все остальные файлы.

Чтобы понять, какие файлы нужны для работы Apache, выполните команды:

```
# ldd /usr/sbin/apache2
# strings /usr/sbin/apache2
# strace /usr/sbin/apache2
```

Внимательно следите за выводом этих команд. Если в выводе встретится название файла, данный файл нужно скопировать в каталог /chroot (точнее, в соответствующий подкаталог каталога /chroot). Например, если серверу нужен файл /var/www/html/index.php, то данный файл нужно скопировать в каталог /chroot/var/www/html.

> ***Примечание***
>
> Программа `strace` выводит список всех системных вызовов, которые порождает Apache во время своей работы. Вам нужно обращать внимание только на системные вызовы `open()`.

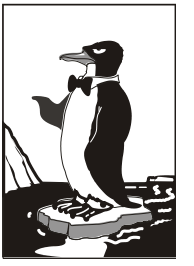
Нам осталось лишь создать базу данных паролей в chroot-окружении и запустить Apache. Для создания базы паролей введите следующие команды:

```
# touch /chroot/etc/passwd
# echo "nobody:x:65534:65534:none:/:/sbin/nologin" >> /chroot/etc/passwd
# echo "www:x:80:80:www:/:/sbin/nologin" >> /chroot/etc/passwd
# touch /chroot/etc/group
# echo "nobody:x:65534:" >> /chroot/etc/group
# echo "www:x:80:" >> /chroot/etc/group
```

Теперь запустим Apache в созданном нами chroot-окружении:

```
# /usr/sbin/chroot /chroot /usr/sbin/apache2
```

Первый аргумент команды chroot — это каталог, в котором мы создали chroot-окружение, а второй — исполняемый файл Web-сервера.



Глава 40

Управление доступом

40.1. Что такое Tomoyo?

Все мы знакомы с системами ограничения доступа SELinux, LIDS и GrSecurity. SELinux и прочие устаревшие системы мы здесь рассматривать не будем — в Интернете, да и в других моих книгах имеется по ним достаточно информации. В настоящее время SELinux — уже не интересно. В этой небольшой главе вместо SELinux вы познакомитесь с модулем безопасности Tomoyo.

Tomoyo исследует поведение каждого процесса, просматривает используемые процессом ресурсы и на основании полученной информации разрешает или запрещает выполнение процесса. Кроме того, Tomoyo можно применять в качестве утилиты системного анализа, то есть этот модуль пригоден для отладки приложений, написания технической документации и изучения принципов работы системы. Инструмент для настоящих хакеров (не забывайте, что хакер — это не тот, кто взламывает и разрушает, а тот, кто создает)!

Естественно, Tomoyo можно использовать для защиты вашей системы, например, для защиты от операций внедрения команд операционной системы, ограничения действий ssh-сервисов и т. д.

Сразу нужно отметить, что данная глава не для начинающих пользователей. Как минимум, вы должны знать, как откомпилировать ядро в вашем дистрибутиве (процедура компиляции ядра в разных дистрибутивах слегка отличается).

40.2. Установка Tomoyo. Готовые LiveCD

Прежде чем установить Tomoyo, можно скачать уже готовые LiveCD, собранные с поддержкой Tomoyo. В практическом плане толку от этих LiveCD мало, но в теоретическом — это как раз то, что вам нужно. Вы можете увидеть Tomoyo в действии, не устанавливая на свой сервер (да, именно на сервер, поскольку на домашний компьютер смысла устанавливать Tomoyo нет). Скачать LiveCD можно по адресам:

- **http://tomoyo.sourceforge.jp/en/1.6.x/1st-step/centos5-live/**;
- **http://tomoyo.sourceforge.jp/en/1.6.x/1st-step/mandriva2009.0/**;
- **http://tomoyo.sourceforge.jp/en/1.6.x/1st-step/f8/**;
- **http://tomoyo.sourceforge.jp/en/1.6.x/1st-step/ubuntu8.04-live/**.

Первая ссылка — дистрибутив CentOS 5 (построен на базе Fedora), собранный с поддержкой Tomoyo. Остальные ссылки — это, соответственно, дистрибутивы Mandriva 2009, Fedora 8 и Ubuntu 8.04. Да, дистрибутивы не очень новые, но для ознакомления вполне сойдут.

Если Tomoyo вам понравился, самое время его скачать и установить. Современная версия Tomoyo требует ядра Linux 2.6.30 или более нового. Самая новая версия ядра на момент написания этих строк — 2.6.32. Инструкция по установке Tomoyo на дистрибутив Linux с более старым ядром находится по адресу: **http://tomoyo.sourceforge.jp/1.6/index.html.en**.

Поддержка Tomoyo уже включена в состав ядра, ее нужно только активировать. А поэтому вам придется перекомпилировать ядро. Посетите сайт **www.kernel.org** и скачайте последнюю версию ядра, например, **http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.31.5.tar.bz2**.

> ***Примечание***
>
> В моем дистрибутиве использовалась версия ядра 2.6.31, поэтому, чтобы не было каких-либо осложнений, я скачал версию 2.6.31.5, а не версию 2.6.32.

Для компиляции ядра нужно установить пакеты gcc, make и ncurces (остальные пакеты будут установлены автоматически).

Распакуйте архив с ядром в /usr/src и введите команду:

```
$ make -s menuconfig
```

Включите параметры ядра **Enable different security models** и **TOMOYO Linux Support**. Затем сохраните конфигурацию ядра и введите команды:

```
$ make -s
$ su
$ make -s modules_install install
```

После установки ядра с поддержкой Tomoyo нужно скачать и откомпилировать утилиты, необходимые для работы с этим модулем:

```
# wget http://osdn.dl.sourceforge.jp/tomoyo/41908/tomoyo-tools-2.2.0-
20090727.tar.gz
# tar -zxf tomoyo-tools-2.2.0-20090727.tar.gz
# make -C tomoyo-tools/ install
```

40.3. Инициализация системы

После установки нового ядра и утилит Tomoyo систему нужно перезагрузить. Следующий шаг — инициализация политик Tomoyo, для чего следует ввести команду:

```
# /usr/lib/tomoyo/tomoyo_init_policy
```

В зависимости от производительности вашего компьютера инициализация может занять несколько минут. Как только инициализация будет завершена, можно запускать редактор политик:

```
# /usr/sbin/tomoyo-editpolicy /etc/tomoyo/
```

Рис. 40.1. Редактор политик

Поскольку вы еще не создавали никаких политик, у вас будет только один домен: **kernel** (рис. 40.1). Когда вы все настроите, доменов будет существенно больше. Можете проверить это, запустив редактор политик, предварительно загрузившись с LiveCD.

Процесс может принадлежать только одному домену, но может во время своего выполнения переходить к другому домену. Процесс не может одновременно принадлежать двум и больше доменам. Ядро принадлежит домену <kernel>, система инициализации init — домену <kernel> /sbin/init, поскольку она была запущена ядром, а процесс, запущенный init, будет находиться в домене "<kernel> /sbin/init процесс". Другими словами, домен здесь — это история выполнения процесса. По домену можно понять, какой процесс является родительским, а какой — дочерним.

Посмотрите на строку домена (см. рис. 40.1):

0: 0 <kernel>

Второе число (в данном случае **0**) — это номер профиля (может быть от 0 до 255). Чтобы просмотреть доступные профили, нажмите клавишу <w> для входа в меню редактора (рис. 40.2), а затем клавишу <p> — для просмотра доступных профилей (рис. 40.3). Строки, содержащие слово **COMMENT**, представляют собой просто комментарии.

- Параметр **MAC_FOR_FILE** регулирует принудительный контроль доступа (MAC, Mandatory Access Control) к файлам. Это самый главный параметр — именно им и отличаются режимы контроля доступа. Существуют четыре профиля (режима доступа), задающие уровень принудительного контроля доступа к файлам:
  - **0** (**disabled**) — контроль доступа к файлам отключен;
  - **1** (**learning**) — обучающий режим, все выполненные операции заносятся в политику как разрешенные;
  - **2** (**permissive**) — разрешающий режим, в этом режиме даже если операция запрещена, она выполняется, но не заносится в политику (полезен для отладки);
  - **3** (**enforcing**) — режим ограничения доступа, если операция запрещена, то она не выполняется, а сообщение о нарушении доступа заносится в журнал.
- Параметр **MAX_ACCEPT_ENTRY** используется для ограничения максимального количества записей в списке доступа. Записи добавляются автоматически в обучающем режиме. По умолчанию используется значение 2040.
- Параметр **TOMOYO_VERBOSE** протоколирует случаи нарушения доступа с помощью syslog.

```
Press one of below keys to switch window.

s      <<< System Policy Editor >>>
e      <<< Exception Policy Editor >>>
d      <<< Domain Transition Editor >>>
a      <<< Domain Policy Editor >>>
p      <<< Profile Editor >>>
m      <<< Manager Policy Editor >>>
q      Quit this editor.
-
```

Рис. 40.2. Меню редактора политик

```
<<< Profile Editor >>>      16 entries    '?' for help

_   0:   0-COMMENT=-----Disabled Mode-----
    1:   0-MAC_FOR_FILE=disabled
    2:   0-MAX_ACCEPT_ENTRY=2048
    3:   0-TOMOYO_VERBOSE=disabled
    4:   1-COMMENT=-----Learning Mode-----
    5:   1-MAC_FOR_FILE=learning
    6:   1-MAX_ACCEPT_ENTRY=2048
    7:   1-TOMOYO_VERBOSE=disabled
    8:   2-COMMENT=-----Permissive Mode-----
    9:   2-MAC_FOR_FILE=permissive
   10:   2-MAX_ACCEPT_ENTRY=2048
   11:   2-TOMOYO_VERBOSE=enabled
   12:   3-COMMENT=-----Enforcing Mode-----
   13:   3-MAC_FOR_FILE=enforcing
   14:   3-MAX_ACCEPT_ENTRY=2048
   15:   3-TOMOYO_VERBOSE=enabled
```

Рис. 40.3. Доступные профили

```
<<< Exception Policy Editor >>>      939 entries    '?' for help

_   0: alias              /bin/bash /bin/sh
    1: alias              /bin/ed /bin/red
    2: alias              /bin/gawk /bin/awk
    3: alias              /bin/gawk /usr/bin/awk
    4: alias              /bin/grep /bin/egrep
    5: alias              /bin/grep /bin/fgrep
    6: alias              /bin/hostname /bin/dnsdomainname
    7: alias              /bin/hostname /bin/domainname
    8: alias              /bin/hostname /bin/nisdomainname
    9: alias              /bin/hostname /bin/ypdomainname
   10: alias              /bin/mail /bin/mailx
   11: alias              /bin/mail /usr/bin/Mail
   12: alias              /bin/tar /bin/gtar
   13: alias              /bin/tcsh /bin/csh
   14: alias              /bin/traceroute /bin/tcptraceroute
   15: alias              /bin/traceroute /bin/traceroute6
   16: alias              /bin/traceroute /bin/tracert
   17: alias              /bin/vi /bin/ex
   18: alias              /bin/vi /bin/rvi
   19: alias              /bin/vi /bin/rview
   20: alias              /bin/vi /bin/view
   21: alias              /etc/sysconfig/network-scripts/ifdown-ippp /etc/sysconf
```

Рис. 40.4. Исключения

В политику Tomoyo по умолчанию добавлены исключения, которые необходимы для нормальной работы системы. Для просмотра исключений нажмите клавишу <e> (рис. 40.4). Для выхода из редактора политик нажмите клавишу <q>.

Попробуем настроить Tomoyo в автоматическом обучающем режиме И создадим, например, политику для DNS-сервера. Запустите его:

```
# service named start
```

Потом запустите редактор политик. Перейдите к процессу named, используя стрелки <↑> и <↓>. Для изменения профиля named нажмите клавишу <s>, а затем введите `1`, что соответствует номеру профиля MAC_FOR_FILE. Строка, относящаяся к named, теперь будет выглядеть так:

```
число: 1          *       /usr/sbin/named
```

Значение 1 соответствует обучающему режиму (Learning Mode). В обучающем режиме нужно определить, какие файлы использует DNS-сервер при запуске, в процессе работы и при завершении работы. Поэтому DNS-сервер нужно перезапустить:

```
# service named restart
```

Снова запустите редактор политик и перейдите к процессу named. Нажмите клавишу <Enter>, чтобы просмотреть, какие разрешения предоставила система DNS-серверу во время перезапуска. После этого выйдите из редактора и сохраните созданную политику:

```
# /usr/sbin/tomoyo-savepolicy
```

Для загрузки политики используется команда:

```
# /usr/sbin/tomoyo-loadpolicy af
```

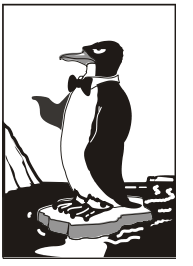
При сохранении политики в каталоге /etc/tomoyo создаются два файла: exception_policy.conf и domain_policy.conf. Первый — это политика исключений, а второй — политика домена. Параметр `a` при загрузке политики указывает, что загрузить нужно оба файла, а параметр `f` присоединяет загружаемую политику к той, что сейчас находится в ядре. Если параметр `f` не указывать, то политика, имеющаяся в ядре, будет перезаписана загружаемой политикой.

Теперь перейдем в разрешающий режим. Запустите редактор политики и установите для процесса named профиль `2`. Действия DNS-сервера запрещаться не будут, но мы получим сообщение о нарушении доступа, что позволит выяснить, какие еще файлы нужны DNS-серверу. Когда все будет настроено, нужно выбрать профиль `3`.

После создания политики для DNS-сервера, можно приступать к созданию политик и для других процессов. Помните, что некоторые процессы могут запускать другие процессы. Например, Web-сервер может запускать sendmail для отправки писем и perl для запуска Perl-сценариев. Поэтому когда вы исследуете процесс, смотрите, какие процессы он запускает. Если вы для родительского процесса установили какой-то профиль, то этот же профиль нужно установить и для всех дочерних процессов.

Создание политик Tomoyo — дело кропотливое, хотя и не очень сложное. Дополнительную информацию можно получить по следующим адресам:

- **http://tomoyo.sourceforge.jp/2.2/tuning.html.en**;
- **http://tomoyo.sourceforge.jp/2.2/enforcing.html.en**.



Глава 41

Виртуальные частные сети

41.1. Для чего нужна виртуальная частная сеть?

Предположим, что пользователям нашей организации нужно обращаться к ресурсам корпоративной сети, когда они находятся за ее пределами, например, в другом городе. Первое, что приходит в голову — это настроить сервер удаленного доступа (Remote Access Server, RAS или dial-in сервер). Пользователь с помощью модема "дозванивается" к серверу удаленного доступа, сервер идентифицирует пользователя, после чего последний подключается к сети предприятия и работает в ней, как обычно (разве что скорость передачи данных будет значительно ниже).

Но использование RAS — затея довольно дорогая и неудобная. Во-первых, нужно организовать модемный пул, а это недешево и накладно: нужна или многоканальная линия, или же несколько телефонных линий (чтобы обеспечить одновременную работу нескольких пользователей). Во-вторых, придется оплачивать междугородние и даже международные звонки пользователей (для удобства самих пользователей можно организовать callback-режим). В-третьих, далеко не всегда у пользователя есть возможность подключиться к телефонной сети. В-четвертых, RAS не может обеспечить связь нескольких филиалов компании.

Выходом из сложившейся ситуации является использование *виртуальной частной сети* (Virtual Private Network, VPN). В случае с VPN данные передаются по каналам Интернета. Это существенно упрощает и удешевляет нашу задачу. Доступ к Интернету есть везде, пользователи сами смогут выбирать провайдера и способ (соответственно, и скорость) подключения к Интернету. Понятно, чтобы оградить себя от перехвата информации, данные при передаче через VPN шифруются. Вот основные преимущества VPN:

- не нужно никакого дополнительного оборудования (модемного пула) и каких-либо дополнительных ресурсов (например, многоканальной телефонной линии). Все, что нужно, — это подключение к Интернету, а поскольку нет такого частного предприятия, которое не было бы подключено к Интернету, будем считать, что все необходимое для организации VPN уже есть;
- безопасность передачи данных по сравнению с обычной передачей данных по Интернету;

- возможность как соединения филиалов компании, так и подключения отдельных пользователей к корпоративной сети. При этом мобильные пользователи могут подключаться к Интернету с помощью GPRS, что делает подключение к VPN максимально гибким — пользователю не придется искать свободную телефонную розетку.

41.2. Необходимое программное обеспечение

Для организации соединений типа сеть-сеть, то есть для связи двух сетей одной компании в одну VPN, используется протокол IpSec. В Linux его реализация называется OpenS/WAN (**http://www.openswan.org**). OpenS/WAN — это потомок самой популярной Linux-реализации IpSec — FreeS/WAN (**http://www.freeswan.org**). Проект OpenS/WAN — более современный и поддерживает ядра 2.4 и 2.6, в то время как FreeS/WAN поддерживает только старые ядра (2.2 и 2.4).

Примечание

В этой книге мы рассмотрим установку OpenS/WAN 2.4.*x* в систему с ядром 2.6.*x*.

Для подключения удаленных пользователей к корпоративной сети используется протокол PPTP (Point to Point Tunneling Protocol). Настройку этого протокола мы также рассмотрим в этой главе.

41.3. Канал для передачи данных VPN

41.3.1. Соединение сеть-сеть

Предположим, что нам нужно объединить два офиса компании. Один пусть находится в Москве, а другой, скажем, во Владивостоке. Для связи офисов будут использоваться VPN-маршрутизаторы. В роли такого маршрутизатора может выступать любой Linux-компьютер с установленным программным обеспечением (OpenS/WAN).

Важно правильно выбрать канал для подключения VPN-маршрутизатора к Интернету. Канал не должен быть "узким", иначе данные между филиалами компании будут передаваться очень медленно. Обычные выделенные линии, понятно, отпадают. Также отпадают различные беспроводные соединения, вроде RadioEthernet. Конечно, беспроводное соединение — это удобно, но его качество очень сильно зависит от "зашумленности" эфира и от погоды. Если на улице плохая погода, скорость заметно падает, не говоря уже о том, что беспроводная точка доступа может сгореть во время грозы. Беспроводные соединения можно охарактеризовать как не очень надежные — ведь чуть ли не после каждой грозы вам придется менять точку доступа. Выключить точку доступа на время непогоды может себе позволить отдельный пользователь, но не предприятие. Поэтому беспроводные соединения нас не устраивают.

Если нужна надежность и независимость, можно выбрать синхронные (двунаправленные) спутниковые соединения, но в этом случае оборудование, да и содержание такого канала (лицензия на передатчик, оплата), будут совсем не дешевыми.

Если организация может себе позволить такое удовольствие, то, уверен, не пожалеет о своем решении.

Наиболее оптимальным для многих организаций будет использование DSL-соединений. Вполне приличная скорость соединения — до нескольких мегабит в секунду, да и такие соединения надежнее беспроводных (имеется в виду RadioEthernet, а не спутниковое соединение).

Если вы остановили свой выбор на DSL-соединении, то нужно выбирать SDSL-соединение — оно синхронное, то есть скорость приема и передачи данных будет одинаковой. Кроме SDSL-соединения, есть еще и ADSL-соединение — оно асинхронно, и скорость приема в несколько раз ниже скорости передачи. Такое соединение больше подходит для домашнего использования, чем для связи офисов компании.

41.3.2. Соединение клиент-сеть

В этом случае пользователь может сам выбрать то соединение, которое предпочтительно для него. В большинстве случаев мобильные пользователи в командировках будут использовать GPRS-соединения. Да, недостатков у GPRS достаточно (самые главные — низкая скорость передачи данных и дороговизна), но зато подключиться к родной сети можно практически откуда угодно — мобильный телефон всегда под рукой, лишь бы быть в зоне покрытия мобильного оператора.

41.4. Настройка соединения сеть-сеть

41.4.1. Установка OpenS/WAN

По адресу **http://www.openswan.org/download/binaries/** вы найдете уже откомпилированные пакеты OpenS/WAN для дистрибутивов Fedora Core, Mandriva, Mandrake, OpenWRT, Red Hat, RHEL, SUSE.

Перед установкой пакетов, возможно, понадобится перекомпиляция ядра. Вам нужно включить опции **PF_KEY**, **AH**, **ESP** и все опции в группе **CryptoAPI**.

Примечание

В этой книге для уменьшения ее размера глава о перекомпиляции ядра удалена, но для всех читателей я сделал небольшой бонус. По следующей ссылке вы найдете главу из подготавливаемого сейчас 3-го издания моей книги "Linux. От новичка к профессионалу", в которой рассмотрены загрузочные сообщения, сама перекомпиляция ядра и информация о ядре реального времени: **http://dkws.org.ua/mybooks/kernel.pdf**.

41.4.2. Немного терминологии

Итак, мы связываем два офиса компании, один из которых находится в Москве, а другой — во Владивостоке (рис. 41.1). Москва находится на западе (слева), Владивосток — на востоке (справа), поэтому московскую сеть мы назовем left, а сеть Владивостока — right. Это не принципиально, но так принято.

Рис. 41.1. Пример VPN-сети

Понятно, что VPN-маршрутизатор будет выходить в Интернет через какой-то обычный маршрутизатор. В терминологии VPN маршрутизатор, через который подключается к Интернету левый VPN-маршрутизатор, называется leftnexthop (соответственно правый — rightnexthop).

41.4.3. Генерирование ключей

Перед настройкой OpenS/WAN нужно сгенерировать ключи на обоих VPN-маршрутизаторах. Для этого на каждом VPN-маршрутизаторе введите команду:

```
# ipsec newhostkey
```

Просмотреть ключ можно с помощью одной из команд:

```
# ipsec showhostkey --left
# ipsec showhostkey --right
```

41.4.4. Конфигурационный файл

OpenS/WAN использует один основной файл конфигурации: /etc/ipsec/ipsec.conf. Этот файл состоит из трех разделов: общие настройки (`config setup`), настройки по умолчанию (`conn %default`) и настройки соединения (`conn <название соединения>`). Понятно, что последних разделов может быть несколько, поскольку каждый такой раздел задает параметры конкретного соединения.

Рассмотрим пример раздела, содержащий параметры по умолчанию (листинг 41.1).

Листинг 41.1. Параметры по умолчанию

```
config setup
        # указывает интерфейсы, которые будут использоваться для
        # VPN-соединений
        interfaces=%defaultroute

        # управляют протоколированием KLIPS (Kernel IP Security) и
        # демоном Pluto
        klipsdebug=none
```

```
plutodebug=none

# Эти параметры лучше не изменять
plutoload=%search
plutostart=%search
```

ПРИМЕЧАНИЕ

Табуляция перед именем директивы (именно директивы, а не раздела файла конфигурации) обязательна! Иначе при обработке конфигурационного файла будет выведено сообщение об ошибке: **… has wrong number of fields …**.

Обратите внимание на параметр `interfaces`. В большинстве случаев подойдет значение `%defaultroute`, но можно указать имя интерфейса явно, например:

```
interfaces="ipsec0=ppp1"
```

Теперь рассмотрим раздел с настройками по умолчанию. Данный раздел вообще не обязателен, но если он есть, то обычно в нем указываются две директивы: `authby` и `keyingtries`. Первая задает метод аутентификации, а вторая — количество попыток установки соединения (по умолчанию `0`, то есть соединение будет устанавливаться бесконечно, пока не будет установлено). Пример данного раздела приведен в листинге 41.2.

Листинг 41.2. Пример раздела настроек по умолчанию

```
conn %default

        authby=rsasig
        keyingtries=3
```

Основной раздел конфигурационного файла описывает VPN-соединения. Для написания этого раздела нам нужно обратиться к рис. 41.1. В конфигурационном файле обоих VPN-маршрутизаторов следует указать сведения, приведенные в табл. 41.1.

Таблица 41.1. *Параметры VPN-соединения типа "сеть-сеть"*

Директива	Назначение
`left`	IP-адрес левого VPN-маршрутизатора (вместо него можно указать значение `%defaultroute`). В нашем случае это 192.168.1.1
`leftsubnet`	IP-адрес левой сети. В нашем случае это 192.168.1.0/24
`leftnexthop`	IP-адрес левого маршрутизатора (можно указать значение `%defaultroute`)
`leftrsasigkey`	Ключ левого маршрутизатора (можно узнать с помощью команды `ipsec showhostkey --left`)
`leftid`	Идентификатор левой сети. Например, @moscow.firma.ru. Можно в разделе `config setup` указать опцию `uniqueids=yes`. Это избавит вас от указания идентификаторов сети

Таблица 41.1 *(окончание)*

Директива	Назначение
`right`	IP-адрес правого VPN-маршрутизатора (вместо него можно указать значение `%defaultroute`). В нашем случае это 192.168.2.1
`rightsubnet`	IP-адрес правой сети (192.168.2.0/24)
`rightnexthop`	IP-адрес правого маршрутизатора (можно указать значение `%defaultroute`)
`rightrsasigkey`	Ключ правого маршрутизатора (можно узнать с помощью команды `ipsec showhostkey --right`)
`rightid`	Идентификатор правой сети. Например, `@vladivostok.firma.ru`
`leftfirewall`	Если левая сторона защищена брандмауэром, то нужно установить значение yes для этой директивы
`auto`	Управляет автоматической установкой соединений. Если указать `auto=start`, соединение будет автоматически установлено. Чтобы директива auto работала, нужно в `config setup` указать `plutostart=%search`

Пример раздела параметров соединения приведен в листинге 41.3.

Листинг 41.3. Пример раздела параметров VPN-соединения

```
conn my_vpn
       left=192.168.1.1
       leftsubnet=192.168.1.0/24
       leftnexthop=10.0.0.1
       leftrsasigkey= 0sAQtyjh9345...
       leftid=@moscow.firma.ru

       right=192.168.2.1
       rightsubnet=192.168.2.0/24
       rightnexthop=10.1.0.1
       rightrsasigkey=0sAQ65jh92...
       rightid=@vladivostok.firma.ru

       auto=start
```

Полная версия файла ipsec.conf представлена в листинге 41.4.

Листинг 41.4. Пример файла ipsec.conf

```
config setup
       # указывает интерфейсы, которые будут использоваться для
       # VPN-соединений
       interfaces=%defaultroute
```

```
        # управляют протоколированием KLIPS (Kernel IP Security) и
        # демоном Pluto
        klipsdebug=none
        plutodebug=none

        # Эти параметры лучше не изменять
        plutoload=%search
        plutostart=%search

conn %default

        authby=rsasig
        keyingtries=3

conn my_vpn
        left=192.168.1.1
        leftsubnet=192.168.1.0/24
        leftnexthop=10.0.0.1
        leftrsasigkey= 0sAQtyjh9345...
        leftid=@moscow.firma.ru

        right=192.168.2.1
        rightsubnet=192.168.2.0/24
        rightnexthop=10.1.0.1
        rightrsasigkey=0sAQ65jh92...
        rightid=@vladivostok.firma.ru

        auto=start
```

41.4.5. Установка VPN-соединения

Для запуска демона OpenS/WAN нужно выполнить команду:

```
ipsec start
```

При этом будут запущены все соединения, для которых вы указали `auto=start`. Команду `ipsec start` нужно выполнить на обеих сторонах.

Проверить, запущено ли соединение, можно с помощью команды:

```
ipsec look
```

41.4.6. Настройка брандмауэра iptables

Для работы IpSec нужно должным образом настроить брандмауэр `iptables`, а именно — разрешить порт 500, который используется для обмена сертификатами и ключами:

```
iptables -A INPUT -i eth0 -p udp -s $IP --sport 500 --dport 500 -j ACCEPT
iptables -A OUTPUT -o eth0 -p udp -d $IP --sport 500 --dport 500 -j ACCEPT
```

```
iptables -A INPUT -i eth0 -p 50 -s $IP -j ACCEPT
iptables -A OUTPUT -o eth0 -p 50 -d $IP -j ACCEPT

iptables -A INPUT -i eth0 -p 51 -s $IP -j ACCEPT
iptables -A OUTPUT -o eth0 -p 51 -d $IP -j ACCEPT

iptables -A FORWARD -p all -s 192.168.2.0/24 -d 192.168.1.0/24 -j ACCEPT
iptables -A FORWARD -p all -s 192.168.1.0/24 -d 192.168.2.0/24 -j ACCEPT
```

В нашем случае:

- `$IP` — это IP-адрес шлюза на противоположной стороне, то есть 192.168.2.1 для левой стороны и 192.168.1.1 для правой стороны;
- `eth0` — это внешний интерфейс сети;
- `ipsec0` — это VPN-интерфейс.

41.5. Настройка соединения клиент-сеть

В этом разделе мы рассмотрим настройку соединения клиент-сеть, когда нужно обеспечить подключение отдельного пользователя к локальной сети. Для настройки соединения такого типа используется протокол PPTP.

Нам понадобятся следующие пакеты:

- pptpd или pptp-server — PPTP-сервер;
- pptp-linux, pptp-client, pptp-adsl — PPTP-клиент.

Названия пакетов (зависят от дистрибутива), правда, могут немного отличаться. Найти данные пакеты можно с помощью сайтов **http://rpmfind.net** (или **rpm.pbone.net**), если у вас Red Hat-совместимый дистрибутив, и **http://packages.ubuntu.com**, если у вас Ubuntu или Debian.

Еще нам нужен пакет ppp, но в большинстве случаев он устанавливается по умолчанию, поэтому вам нужно только проверить его наличие в вашей системе.

Нужно отметить, что VPN-сервер в современных дистрибутивах настраивается на порядок проще, чем в дистрибутивах, основанных на ядре 2.4. Ведь в старых дистрибутивах вам нужно было добавить поддержку MPPE (патч) для ppp и ядра, а в новых дистрибутивах, основанных на ядре 2.6, всего этого делать не нужно. Не придется даже перекомпилировать ядро, поскольку в большинстве случаев расширение MPEE включено по умолчанию. Почему в большинстве случаев? Откуда же я знаю, какой у вас дистрибутив? Может, у вас какой-то экзотический дистрибутив, разработчики которого посчитали, что MPPE вам не нужен, и отключили его.

> ***Пояснение***
>
> Microsoft Point-to-Point Encryption (MPPE) — протокол шифрования данных, используемый поверх соединений PPP.

41.5.1. Редактирование конфигурационных файлов

После установки пакета ppptd (или pptp-server) можно отредактировать его конфигурационный файл /etc/pptpd.conf (листинг 41.5).

Листинг 41.5. Конфигурационный файл /etc/pptpd.conf

```
speed 115200
option /etc/ppp/options.vpn
debug
#
remoteip 192.168.1.12-22
```

Чтобы основной конфигурационный файл был компактным, дополнительные опции вынесем в файл /etc/ppp/options.vpn. Думаю, назначение этих опций понятно и без моих комментариев. IP-адреса VPN-клиентов вам нужно изменить (параметр `remoteip`). В этом примере предполагается, что максимум может быть 10 VPN-клиентов, которым будут назначены IP-адреса из диапазона 192.168.1.12 — 192.168.1.22.

Теперь отредактируем файл /etc/ppp/options.vpn (листинг 41.6) — понятно, его еще нужно создать.

Листинг 41.6. Конфигурационный файл /etc/ppp/options.vpn

```
ipparam PoPToP

lock

mtu 1000
mru 1000

ms-dns 192.168.1.1
name server.com
# нужен для того, чтобы после подключения к серверу удаленный
# пользователь мог обратиться к узлам виртуальной сети
# Чтобы эта опция работала правильно, нужно включить
# форвардинг пакетов (Ipv4 Forwarding)
proxyarp
# нужен, если вы планируете использовать аутентификацию
auth
# если аутентификация не нужна, тогда укажите
#noauth

# Протоколы аутентификации, если noauth, то они не нужны
refuse-pap
refuse-chap
refuse-chapms
require-mschap-v2

ipcp-accept-local
```

```
ipcp-accept-remote
lcp-echo-failure 30
lcp-echo-interval 5
# Если deflate = 0, то сжатие не используется
#deflate 0
```

Этот файл конфигурации немного сложнее, чем предыдущий. Если особо разбираться, что есть что, не хочется, тогда просто измените IP-адрес DNS-сервера (опция `ms-dns`) и имя узла (опция `name`). В не самых свежих версиях ppp вместо опций `refuse-pap`, `refuse-chap`, `refuse-chapms` и `require-mschap-v2` нужно использовать опции (соответственно): `-pap`, `-chap`, `-chapms` и `+chapms-v2`.

Данные опции управляют аутентификацией VPN-пользователя (мы используем протокол аутентификации MS CHAP v2 как самый безопасный).

Практически все настроено. Осталось только отредактировать файл /etc/ppp/options. Добавьте в него всего одну опцию:

```
lock
```

Имена VPN-пользователей можно определить в файле /etc/ppp/chap-secrets. Формат этого файла такой:

```
имя сервер.домен пароль IP
```

> ***Примечание***
>
> Если аутентификация вам не требуется, то не надо редактировать файл /etc/ppp/chap-secrets и создавать VPN-пользователей. При настройке клиента не нужно указывать имя пользователя и пароль, следует также отключить шифрование.

Вот небольшой пример:

```
vpn1 server.com "" *
```

Здесь `vpn1` — имя пользователя, `server.com` — имя нашего VPN-сервера. Пароль мы указали пустой, это означает, что пароль будет браться из /etc/shadow. IP-адрес мы тоже не указывали — VPN-пользователь сможет аутентифицироваться с любого IP. Пользователь `vpn1` должен существовать в системе (добавить пользователя можно командой `adduser`).

Вот сейчас все готово. Для запуска PPTP-сервера используется команда:

```
service pptpd start (или /etc/init.d/pptpd start)
```

Не забудьте разрешить на брандмауэре прохождение пакетов на порты 47 и 1723 (ведь наверняка в вашей сети есть брандмауэр):

```
iptables --append INPUT --protocol 47 --jump ACCEPT
iptables --append INPUT --protocol tcp --match tcp --destination-port 1723
--jump ACCEPT
```

Подробно настройка брандмауэра для взаимодействия с PPTP описана на страничке **http://asplinux.net/node/1918.**

А на следующей страничке подробно описаны ошибки PPTP-клиента, что пригодится при анализе журналов системы в случае возникновения проблем: **http://pptpclient.sourceforge.net/howto-diagnosis.phtml#running_pptp**.

41.5.2. Настройка Linux-клиента

В этом разделе мы рассмотрим настройку клиента, работающего под управлением Linux. Понятно, что на VPN-сервере не обязательно устанавливать PPTP-клиент.

Для установки VPN-клиента надо установить пакет pptp-linux (рис. 41.2) или pptp-client. После установки запустите сценарий pptp-command. Сценарий отобразит меню из четырех пунктов (рис. 41.3) — выберите пункт **Setup**.

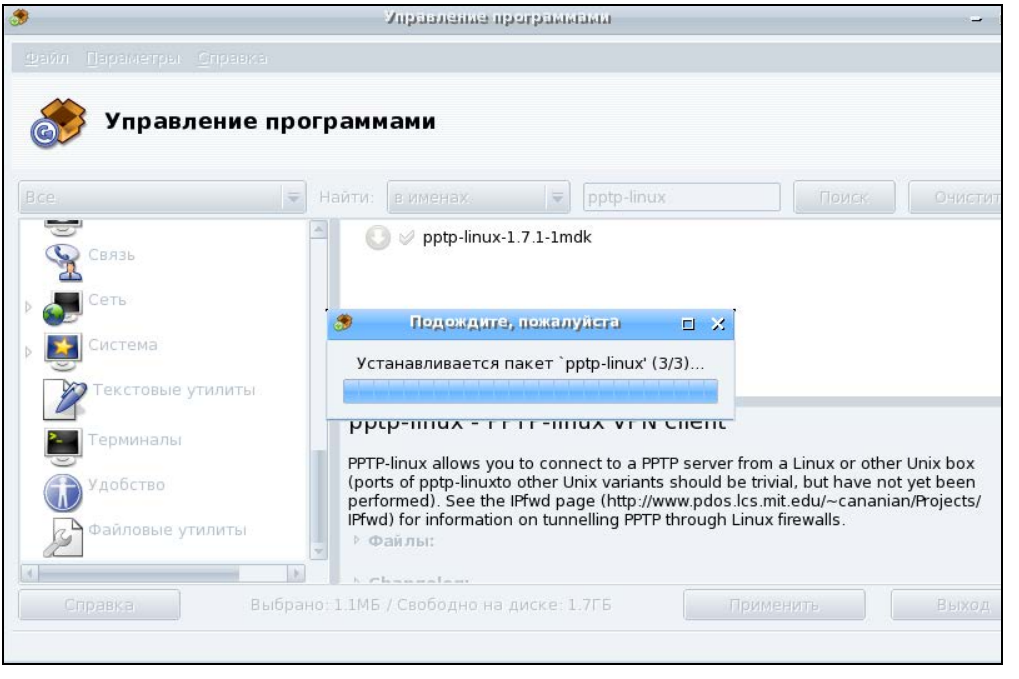


Рис. 41.2. Установка пакета pptp-linux в Linux Mandriva

Количество пунктов меню увеличится (см. рис. 41.3), выберите пункт **Manage CHAP secrets**, а затем команду **Add a New CHAP secret**. Сценарий попросит ввести вас имя локальной машины, имя удаленной машины (вводить необязательно), имя пользователя и пароль.

Затем вы вернетесь в меню настройки. Выберите теперь команду **Add a new PPTP tunnel**, а затем команду **Other** и введите имя и IP-адрес VPN-сервера, а также параметры маршрутизации.

Вернувшись в меню настройки, выберите пункт **Configure resolv.conf** и укажите IP-адреса DNS-серверов.

Настройка почти закончена. В уже хорошо знакомом вам меню выберите команду **Select a default tunnel**, позволяющую выбрать туннель по умолчанию. Следует выбрать туннель, который вы только что создали.

Для подключения к VPN запустите опять сценарий pptp-command и выберите команду **Start**. Понятно, что перед этим вы должны подключиться к Интернету.

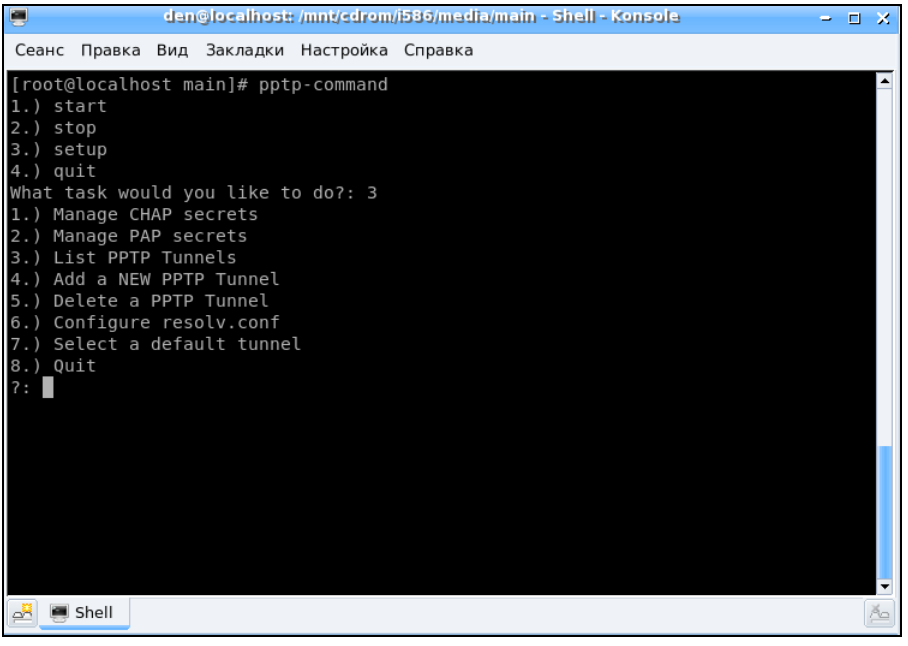


Рис. 41.3. Сценарий pptp-command

В Ubuntu в окне **Сетевые соединения** (NetworkManager) имеется вкладка VPN, но кнопка **Добавить** не активна. А все потому, что не установлены пакеты, реализующие поддержку VPN. Чтобы настроить VPN-соединение через NetworkManager, вам нужно скачать с packages.ubuntu.com пакеты network-manager-pptp и network-manager-pptp-gnome. Конечно, если вы используете не протокол PPTP, а какой-нибудь другой, тогда придется скачать и установить соответствующие пакеты, например, network-manager-vpnc. Если быть предельно точным, то надо установить следующие пакеты (и все пакеты, от которых зависят эти пакеты):

- pptp-linux;
- network-manager-pptp;
- network-manager-pptp-gnome;
- network-manager-vpnc;
- network-manager-openvpn;
- network-manager-openvpn-gnome;
- network-manager-strongswan.

Первые три пакета необходимы для поддержки PPTP, четвертый — для протокола VPNC (Cisco), следующие два — для протокола OpenVPN, последний — для strongSwan.

А как же скачать пакеты, если соединение с Интернетом осуществляется по VPN? Существуют три варианта:

- перезагрузиться в Windows, если она установлена, подключиться к Интернету, скачать пакеты;

- найти другой компьютер или использовать альтернативное соединение (например, 3G-модем или мобильный телефон);
- использовать дистрибутив Denix (**http://denix.dkws.org.ua**) — в него по умолчанию включены все пакеты, необходимые для установки VPN-соединений.

> ***Примечание***
>
> Ради справедливости нужно отметить, что в Ubuntu 10.04 появилась поддержка VPN "из коробки", но поддерживается только PPTP, а вот если нужно установить соединение по другому протоколу — см. только что описанные действия.
>
> Если у вас все-таки Ubuntu 9.*x* и переходить на 10.*x* вы пока не планируете, обязательно прочитайте мою статью по настройке VPN-соединения в Ubuntu 9: **http://www.dkws.org.ua/index.php?page=show&file=a/ubuntu/vpn-ubuntu9**.

Скачанные пакеты можно установить командой `dpkg` (подробно об этом вы прочитаете в соответствующей главе книги).

Подробно процесс настройки VPN-соединения в Mandriva описан по адресу: **http://wiki.mandriva.com/ru/Настройка_VPN_в_Mandriva**.

При настройке VPN-соединения в Mandriva могут возникнуть некоторые проблемы. Решение проблем описано по адресу: **http://www.dkws.org.ua/phpbb2/viewtopic.php?p=18432**.

41.5.3. Настройка Windows-клиента

В Windows 2000/XP

Настройка VPN-подключения в Windows 2000/XP намного проще. Во-первых, вам не нужно устанавливать VPN-клиент — он уже входит в состав Windows и установлен по умолчанию. Во-вторых, интерфейс мастера новых подключений в Windows дружелюбнее и привычнее интерфейса pptp-command (хотя кому как).

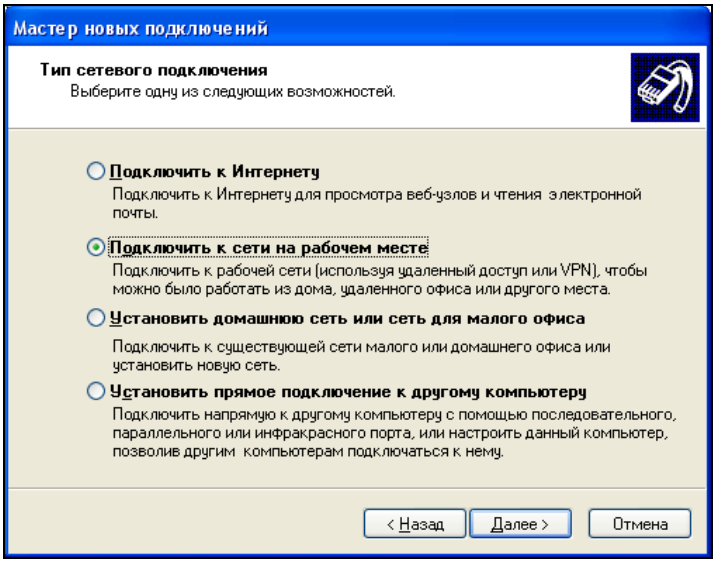


Рис. 41.4. Мастер новых подключений

Чтобы создать VPN-подключение, выполните команду меню **Пуск | Настройка | Сетевые подключения | Создание нового подключения**. В окне мастера новых подключений выберите **Подключить к сети на рабочем месте** (рис. 41.4). Затем выберите **Подключение к виртуальной частной сети** (рис. 41.5).

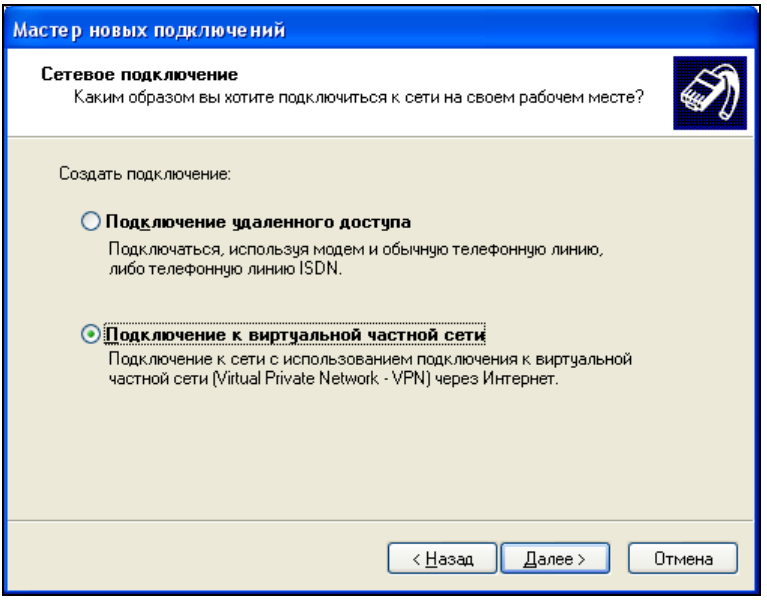


Рис. 41.5. Подключение к VPN

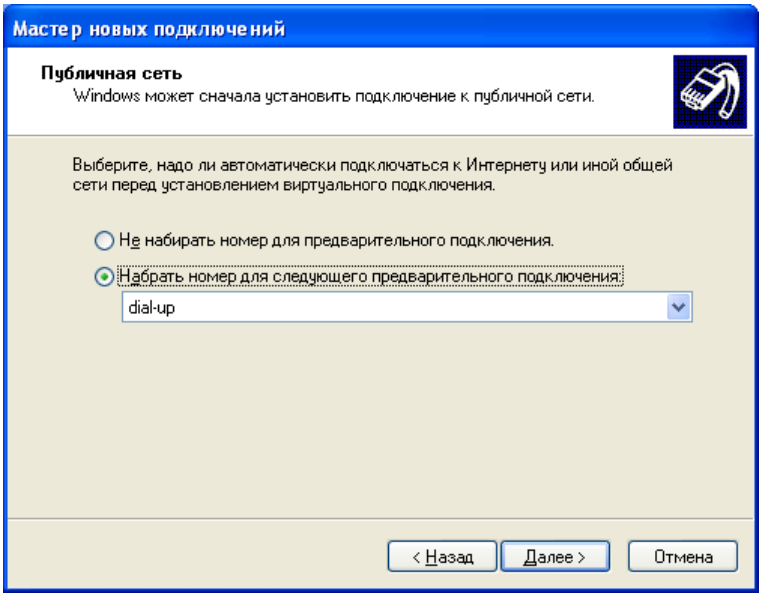


Рис. 41.6. Выбор подключения к Интернету

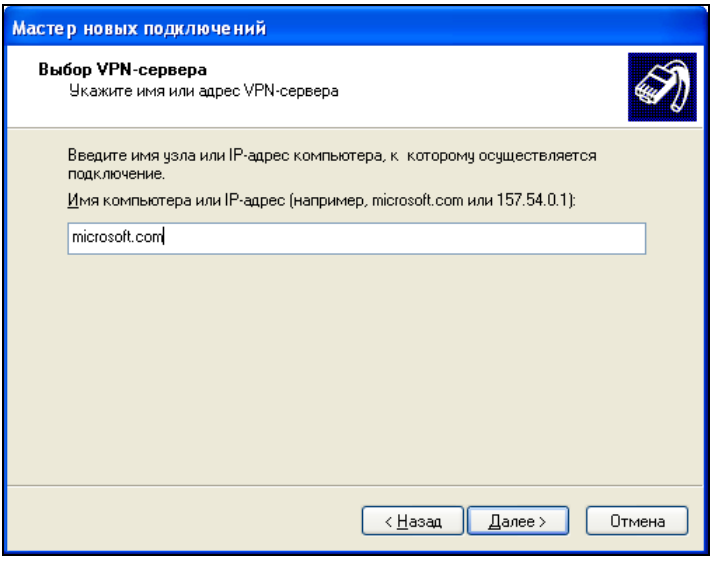


Рис. 41.7. Ввод имени VPN-сервера

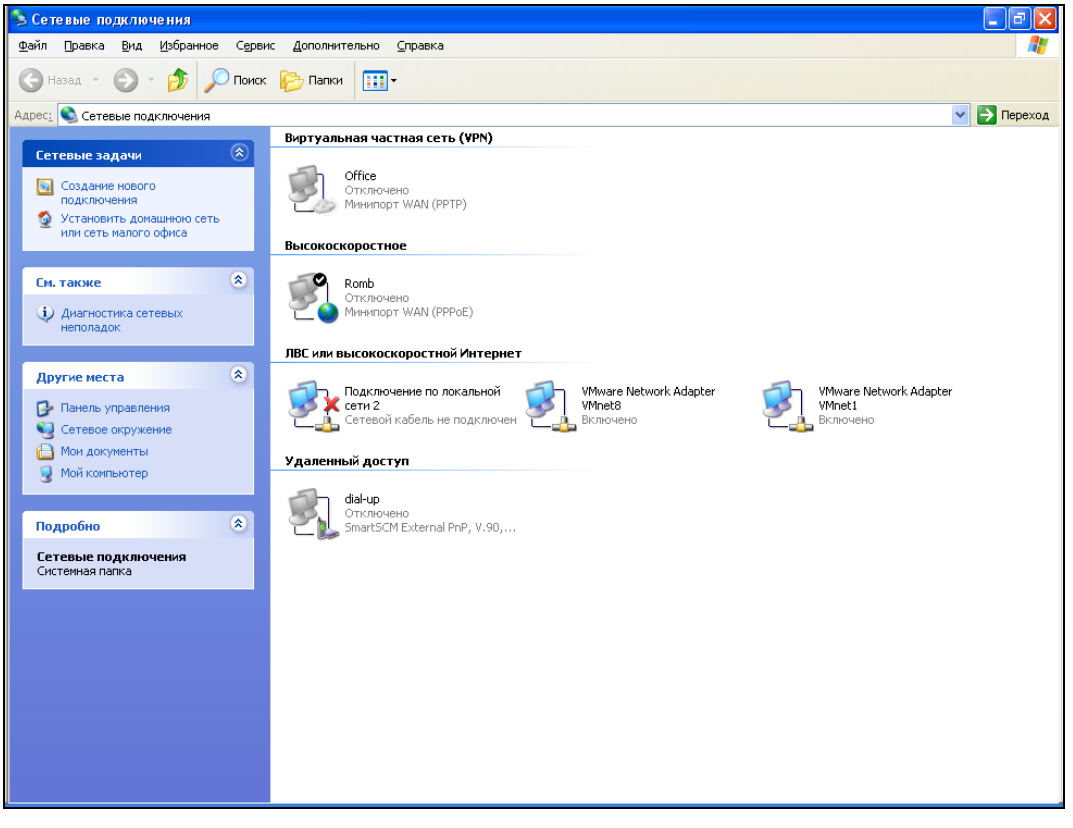


Рис. 41.8. Сетевые подключения

Перед подключением к VPN нужно установить соединение с Интернетом, поэтому мастер новых подключений предложит вам выбрать соединение с Интернетом, которое будет установлено перед подключением к VPN (рис. 41.6).

Затем вам останется лишь ввести параметры подключения: имя VPN-сервера (рис. 41.7), имя пользователя и пароль. Вместо имени VPN-сервера можно ввести его IP-адрес.

После создания VPN-подключения его можно запустить из системной папки **Сетевые подключения** (рис. 41.8).

В Windows Vista/Windows 7

В Windows Vista/7 VPN-подключение создается аналогично. Выберите команду **Пуск | Подключение**. В открывшемся окне (рис. 41.9) перейдите по ссылке **Установка подключения или сети**. В открывшемся окне выберите **Подключение к рабочему месту** (рис. 41.10).

Далее процесс настройки ничем не отличается от процесса настройки в Windows XP: нужно выбрать интернет-соединение (рис. 41.11), ввести имя VPN-сервера и т. д.

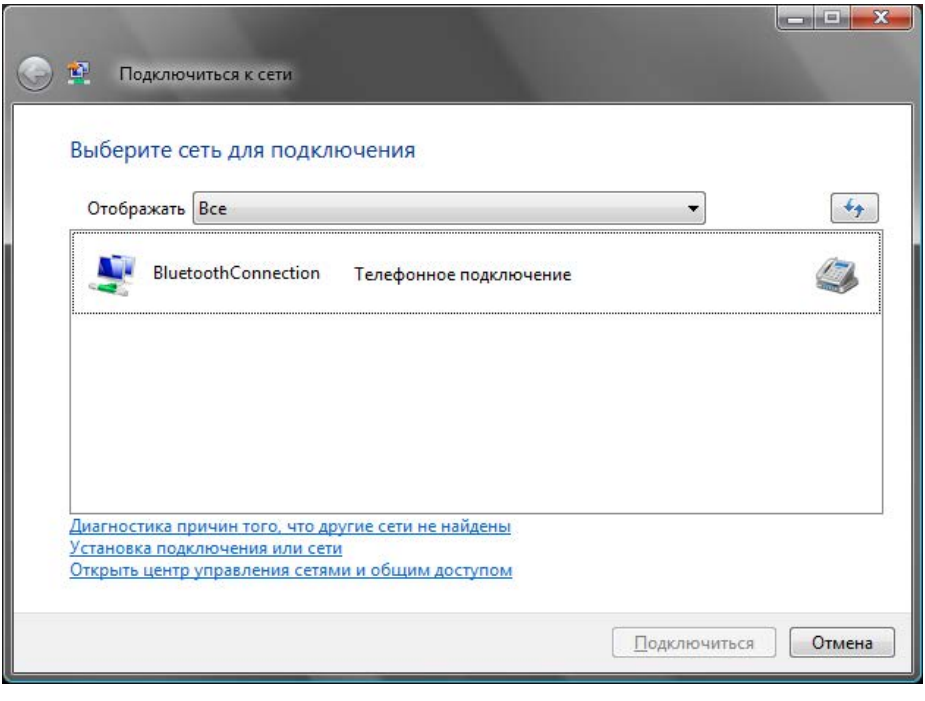


Рис. 41.9. Подключиться к сети

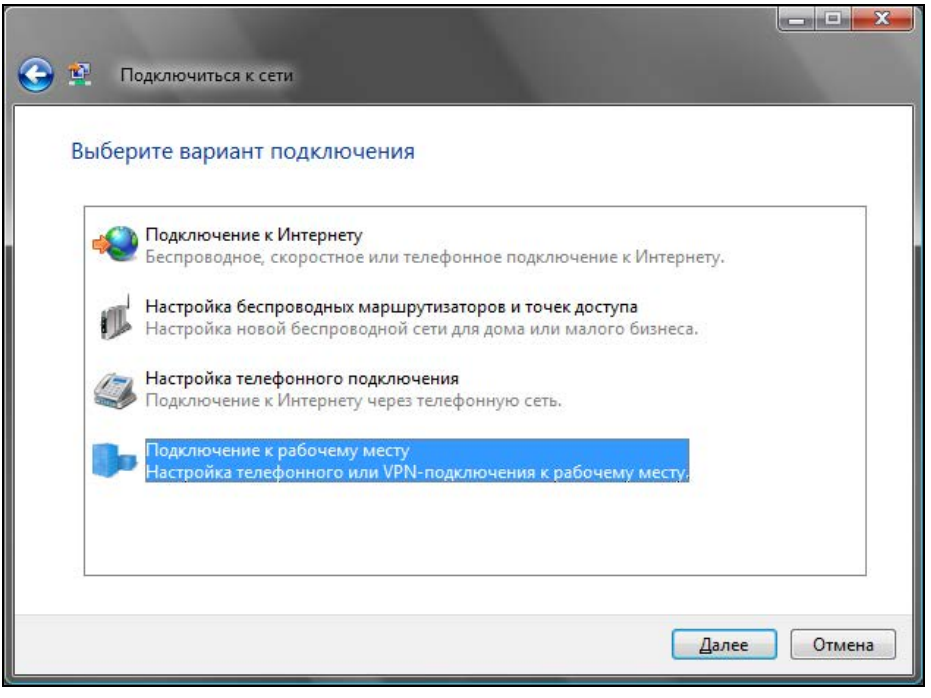


Рис. 41.10. Выбор варианта подключения

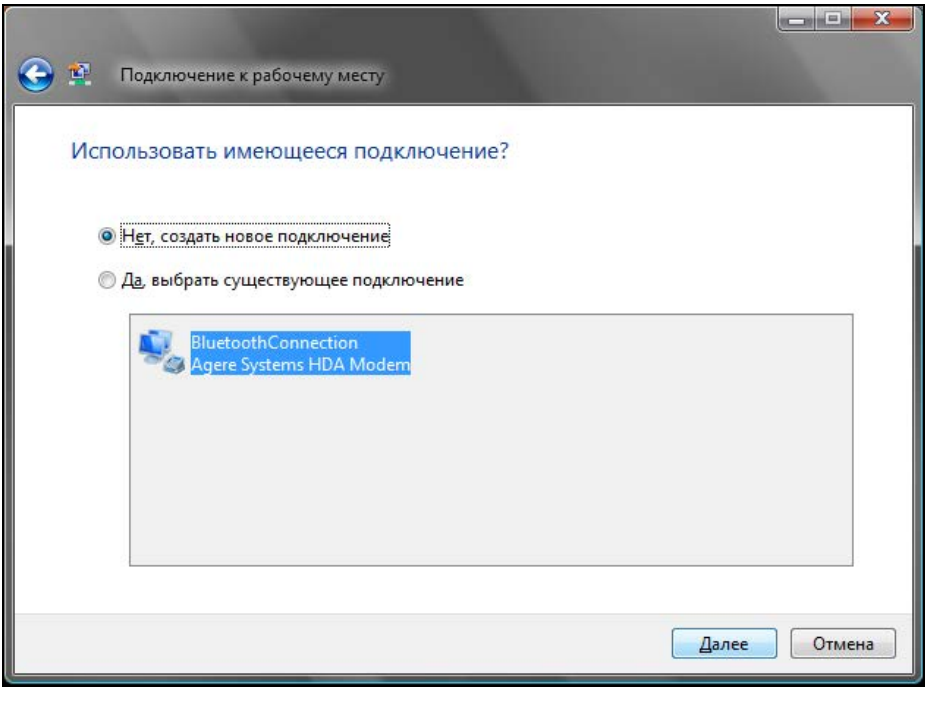
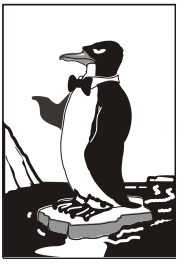


Рис. 41.11. Выбор интернет-соединения



Глава 42

Прокси-сервер Squid и антивирус ClamAV

42.1. Зачем нужен прокси-сервер в локальной сети?

С помощью прокси-сервера Squid можно очень эффективно управлять ресурсами своей сети, например, кэшировать трафик (http), "обрезать" баннеры, указать, какие файлы можно скачивать пользователям, а какие — нет, также можно определить максимальный объем передаваемого объекта и даже ограничить пропускную способность пользователей определенного класса.

Основная функция прокси-сервера — это кэширование трафика. Если в сети используется прокси-сервер, можно сократить кэш браузеров клиентов практически до нуля — он уже не будет нужен, поскольку кэширование станет выполнять прокси-сервер. Тем более, что он выполняет кэширование всех клиентов сети, и уже запрошенные кем-то ранее страницы окажутся доступны другим пользователям. Это означает, что если кто-то заходил на сайт **firma.ru**, то у всех остальных пользователей сети этот сайт будет открываться практически мгновенно, потому что его уже кэшировали.

Даже если у вас всего один компьютер, все равно есть смысл использовать Squid, хотя бы для того, чтобы "обрезать" баннеры — так можно сэкономить на трафике, да и страницы начнут открываться быстрее, поскольку многочисленные баннеры грузиться не будут.

Squid не сложен в настройке, во всяком случае он не сложнее Samba и подобных сетевых сервисов. Прежде всего установите пакет squid. После установки пакета у вас в системе появится новый сервис — squid. Основной конфигурационный файл — /etc/squid/squid.conf.

42.1.1 Базовая настройка Squid

Приступим к редактированию основного конфигурационного файла /etc/squid/squid.conf (листинг 42.1).

Листинг 42.1. Файл /etc/squid/squid.conf

```
# порт для прослушивания запросов клиентов
# задается в формате http_port <порт> или http_port <узел>:<порт>
```

```
# последний случай подходит, если SQUID запущен на машине с несколькими
# сетевыми интерфейсами
http_port 192.168.0.1:3128

# адрес прокси провайдера, нужно согласовать с провайдером
# cach_peer proxy.your_isp.com

# объем оперативки в байтах, который будет использоваться прокси-сервером
# (85 Мбайт) не устанавливайте более трети физического объема оперативки,
# если данная машина должна использовать еще для чего-либо
# можно задать в мегабайтах, но тогда между числом и MB обязательно
# должен быть пробел: cache_mem 85 MB
cache_mem 87040

# где будет размещен кэш.
# первое число — это размер кэша в Мбайт, не устанавливайте кэш на весь
# раздел, если нужно, чтобы он занимал весь раздел, отнимите от размера
# раздела 20% и укажите это значение. Например, если раздел 1024 Мбайт, то
# для кэша — только 820 Мбайт; второе — количество каталогов первого уровня
# третье — к-во каталогов второго уровня
cache_dir /usr/local/squid 1024 16 256

# максимальный размер кэшируемого объекта
# если размер объекта превышает указанный здесь, то объект не будет
# сохранен на диске
# maximum_object_size 4096 KB

# хосты, с которых разрешен доступ к прокси
acl allowed_hosts src 192.168.1.0/255.255.255.0
acl localhost src127.0.0.1/255.255.255.255

# разрешенные порты:
acl allow_ports port 80                    # http
acl allow_ports port 21                    # ftp
# SSL-порты
acl SSL_ports port 443 563

# запрещаем все порты, кроме указанных в allow_ports
http_access deny !allow_ports
```

```
# запрещаем метод CONNECT для всех портов, кроме указанных в
# acl SSL_ports:
http_access deny CONNECT !SSL_ports

# запретим доступ всем, кроме тех, кому можно
http_access allow localhost
http_access allow allowed_hosts
http_access allow SSL_ports
http_access deny all

# пропишем пользователей, которым разрешено пользоваться squid
# (ppt, admin):
ident_lookup on
acl allowed_users ppt admin
http_access allow allowed_users
http_access deny all
```

Базовый конфигурационный файл с успехом выполняет только функцию кэширования, а в следующем разделе мы поговорим о более тонкой настройке Squid.

42.1.2. Практические примеры настройки Squid

Управление доступом

Управление доступом осуществляется с помощью ACL (Access Control List) — списков управления доступом. Разберемся, как работать с ACL. Создадим список AllowedPorts:

```
acl AllowedPorts port 80 8080 3128
```

Имя списка — `AllowedPorts`, тип списка — `port`. Далее мы можем использовать этот список в `http_access` для разрешения/запрещения указанных портов:

```
http_access allow AllowedPorts   # разрешение портов
http_access deny AllowedPorts    # запрещение портов
```

Кроме типа `port` часто используются следующие типы списков:

- `proto` — протокол (HTTP или FTP);
- `method` — метод передачи данных (GET или POST);
- `src` — IP-адреса (или диапазоны адресов) клиентов;
- `dst` — IP-адреса/URL сайтов, к которым обращаются клиенты.

Вы также можете создать список узлов, которым разрешен доступ к прокси:

```
acl allowed_hosts src "/etc/squid/allowed-hosts.txt"
Сам файл /etc/squid/allowed-hosts.txt будет выглядеть так:
# den
```

```
192.168.0.2/255.255.255.255
# admin
192.168.0.3/255.255.255.255
```

Отдельный файл использовать удобнее, чтобы не "засорять" основной конфигурационный файл. Обратите внимание: права доступа к файлу allowed-hosts.txt должны быть такие же, как к squid.conf.

Создание черного списка URL

Теперь попробуем создать черный список URL:

```
acl blacklist url_regex adult
http_access deny blaklist
http_access allow all
```

Данный черный список не пропускает URL, содержащие слово `adult`. По аналогии можно было бы создать отдельный файл и записать в него все "плохие" URL (но это довольно накладно, проще использовать регулярные выражения).

Отказ от баннеров

С помощью ACL можно отказаться и от баннеров — принцип тот же. Для этого добавьте в файл конфигурации следующие ACL:

```
acl banners urlpath_regex "/etc/squid/banners.txt"
http_access deny banners
```

В файл banners.txt нужно внести URL баннерных сетей, например,

```
^http://www.clickhere.ru
^http://banner.kiev.ua
...
```

Создание этого файла пусть будет вашим домашним заданием — все равно все баннерные сети в книге не приведешь.

42.1.3. Управление прокси-сервером

Для запуска, перезапуска и остановки прокси-сервера нужно использовать следующие команды:

```
# service squid start
# service squid restart
# service squid stop
```

42.1.4. Настройка клиентов

Все браузеры на компьютерах вашей сети нужно настроить на использование порта 3128 (именно этот порт мы установили в конфигурационном файле). На рис. 42.1 изображена настройка браузера Opera.

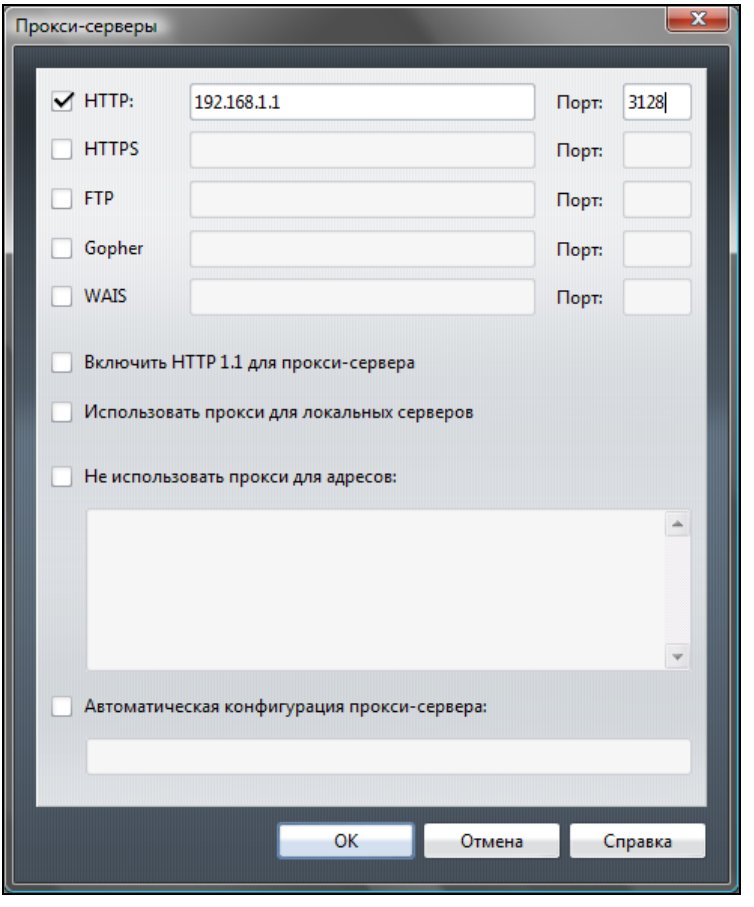


Рис. 42.1. Настройка клиента

42.1.5. Прозрачный прокси-сервер

С прокси-сервером часто связаны две проблемы. Первая заключается в том, что для работы через прокси-сервер приходится настраивать всех клиентов. Если сеть большая, скажем, 100 компьютеров, можете себе представить, сколько это займет времени — ведь нужно подойти к каждому компьютеру. Даже если на настройку одного компьютера уйдет 5 минут, то на всю работу потребуется 500 минут — целый рабочий день. И настройкой браузера может дело и не обойтись. Ведь у пользователей могут быть и другие интернет-программы, работающие с WWW/FTP, которые также нужно будет настроить. Проблема настройки — не самая страшная. Понятно, что если в сети организации 100 или более компьютеров, то и администратор будет не один. А вдвоем-втроем можно настроить все 100 компьютеров за 2−3 часа.

Вторая проблема — более серьезная. Представим, что в сети у нас есть продвинутые пользователи (а они-таки есть), которые знают, для чего используется про-

кси-сервер. Они могут просто изменить настройки и вместо работы через прокси использовать прямое соединение с Интернетом, то есть станут работать в обход Squid. Вы так старались, создавая список черных URL (преимущество это сайты для взрослых и всевозможные чаты/форумы), а они с помощью пары щелчков мышью сведут все ваши старания к нулю.

Обе проблемы можно решить, если настроить "прозрачный" прокси-сервер — пользователи даже не будут подозревать, что он есть. Во-первых, это решит проблемы с настройкой — вам не нужно настраивать браузеры пользователей, потому что все HTTP-запросы будут автоматически поступать на прокси-сервер. Во-вторых, прозрачный прокси обеспечит принудительное кэширование информации, и, соответственно, принудительный контроль за страницами, которые посещают пользователи.

Для настройки прозрачного прокси вам нужно изменить как конфигурационный файл самого прокси-сервера, так и правила брандмауэра iptables *(см. главу 15)*. Вот правила iptables:

```
iptables -t nat --new-chain TransProxy
# только порт 80 (HTTP) и 443 (SSL, https) — остальные обрабатывать не бу-
дем
iptables -t nat -A PREROUTING -p tcp --dport 80 -j TransProxy
iptables -t nat -A PREROUTING -p tcp --dport 443 -j TransProxy
iptables -t nat -A TransProxy -d 127.0.0.1/8 -j ACCEPT
# укажите IP-адрес своей сети
iptables -t nat -A TransProxy -d 192.168.1.0/24 -j ACCEPT
# все запросы перенаправляются на прокси-сервер 192.168.1.1, порт 3128
iptables -t nat -A TransProxy -p TCP -j DNAT --to 192.168.1.1:3128
```

Теперь займемся настройкой Squid. В конфигурационный файл squid.conf добавьте следующие директивы:

```
# серверу назначается реальный IP-адрес, его и нужно указать
tcp_outgoing_address ваш_реальный_IP
httpd_accel_host virtual
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Более подробно о настройке iptables рассказано в *главе 15*. Хочу лишь напомнить, что iptables обычно устанавливается на шлюзе — компьютере, который предоставляет доступ к Интернету других компьютеров сети. На этом же компьютере должен быть установлен и Squid.

42.1.6. Расширение squidGuard

Чуть ранее были приведены примеры создания черных списков, ограничивающие доступ к сайтам с запрещенным контентом. Но пока вы сформируете базу черных списков, пройдет время. Для автоматизации этого процесса вы можете использовать squidGuard, имеющий уже готовые черные списки, сформированные большим сообществом пользователей и тщательно проверенные разработчиками squidGuard.

Расширение squidGuard не только сэкономит трафик, но и эффективно защитит вашу сеть от запрещенного контента.

Черный список squidGuard обновляется постоянно. Скачать squidGuard можно на сайте **http://www.squidguard.org/**. Там же вы найдете альтернативные черные списки. В базе squidGuard внесены самые известные сайты с запрещенным контентом, а именно: насилие, порнография, наркотики, азартные игры и т. д.

Для установки squidGuard достаточно установить одноименный пакет. После установки черный список узлов будет помещен в каталог /usr/share/squidGuard-1-3-0/db (версия squidGuard у вас может быть иная). В некоторых дистрибутивах или если вы устанавливаете squidGuard с исходных кодов, база будет помещена в каталог /usr/local/squidGuard/db.

Чтобы база данных была самой актуальной, скачайте последнюю версию базы по адресу: **http://www.squidguard.org/blacklists.html**. Это файл blacklist.tar.gz. Его нужно распаковать в каталог /usr/share/squidGuard-1-3-0/db или в /usr/local/squidGuard/db:

```
cp blacklist.tar.gz /usr/local/squidGuard/db
gzip -d blacklist.tar.gz
tar xfv blacklist.tar
```

После этого следует немного отредактировать файл конфигурации squidGuard. Скопируйте файл /etc/squid/squidGuard.conf.sample в файл /etc/squid/squidGuard.conf и откройте его в текстовом редакторе. Весь файл редактировать не нужно, полный листинг этого файла тоже приводить не стану — он слишком длинный.

Первым делом нужно указать путь к базе и к журналам:

```
dbhome /usr/local/squidGuard/db
logdir /var/log/squidGuard
```

Теперь опишем разрешенное время работы:

```
# s = Вс, m = Пн, t =Вт, w = Ср, h = Чт, f = Пт, a = Сб

time workhours {
        weekly m 08:00-12:00 13:00-19:00
        weekly t 08:00-11:00 12:00-19:00
        weekly w 08:00-12:00 12:00-18:00
        weekly h 08:00-13:00 13:00-18:00
        weekly f 08:00-12:00 13:30-18:00
        weekly a 11:20-14:00
        weekly s 11:32-14:00
}
```

Опишем две зоны. К первой будут относиться наши пользователи, а ко второй — администраторы сети. Пользователи, не относящиеся к первым двум группам, вообще не будут иметь доступа к Интернету.

```
src users {
   ip 192.168.1.5-192.168.1.200
   }

   src admins {
```

```
ip 192.168.1.1-192.168.1.4
}
```

Опишем списки доступа, определяющие, кому и к каким узлам разрешен доступ. Администраторам разрешаем доступ ко всем узлам, кроме рекламных баннеров,
а вот пользователям запрещаем доступ по максимуму.

```
acl {
  admins {

 pass !advertising all

# запрещенные запросы перенаправляем на следующий адрес
redirect http://server.ru/error.html
}

users {
pass !adult !audio-video !forums !hacking !redirector !warez !ads !aggres-
sive !drugs
                      !gambling !publicite !violence !banneddestination !ad-
vertising all

redirect http://server.ru/error.html
}

# остальным пользователям доступ к Интернету запрещен (pass none)
default {
pass none
redirect http://server.ru/error.html
}
}
```

Почти все. Осталось только "прописать" расширение squidGuard в конфигурационном файле Squid. Откройте файл /etc/squid/squid.conf и добавьте в него следующие строки:

```
redirector_bypass on
redirect_program /usr/local/squidGuard/bin/squidGuard
redirect_children 1
```

Сохраните файл и перезапустите Squid:

```
# service squid restart
```

Теперь выполните следующую команду:

```
tac /var/log/squidGuard/squidGuard.log | less
```

Вы должны увидеть сообщение о том, что squidGuard запущен (**started**) и готов к обработке запросов (**ready for requests**). Если вы увидели заветные строки, значит, вы все сделали правильно.

42.2. Антивирусная защита

42.2.1. Зачем нужен антивирус в Linux

Linux считается одной из самых безопасных операционных систем. Она устойчива, ее сетевые сервисы надежны и … для Linux существует очень мало вирусов. Почему? Давайте подумаем. Представим на некоторое время, что мы — вирусописатели. Для какой бы операционной системы вы бы написали вирус? Для той, в которой работает на данный момент большинство компьютеров и которая более доступна в плане внедрения вируса? Или для той, которая не так популярна, как первая, и в несколько раз неприступнее? Думаю, вы бы выбрали первый вариант. Вот такой вариант как раз и есть Windows. Начнем с того, что для DOS было написано очень много вирусов, и все они по наследству перешли в Windows. Но система Windows несла в себе не только новые функции, но и новые ошибки, каждая из которых порождала новую волну вирусов. Не успевали в Microsoft закрыть одну "дыру", как появлялась следующая. Чего только стоит дырявый Internet Explorer, через который буквально за 10−15 минут в Интернете может проникнуть в систему целая армия троянов, сетевых червей и прочей нечисти. Windows с ее передовыми и непроверенными технологиями — отличная цель для вирусописателей. Ведь вирусописатели, в какой-то степени, творческие люди. И им интересно, чтобы их "творение" развивалось. А в Linux развитие вируса пресекает сама операционная система. Предположим, что Linux-пользователь скачал какой-то вирус для Linux. И даже запустил его. Максимум, что может сделать вирус — это повредить файлы в домашнем каталоге пользователя. Ведь для всего остального у него не хватит полномочий. А если вирус запустит пользователь root? Да, вирус в этом случае сможет нанести ущерб системе. Но, скажем так, это единичный случай. Все грамотные Linux-пользователи никогда не запускают ничего подозрительного под пользователем root и вообще ежедневную работу выполняют под обычным пользователем, а под пользователем root выполняют только системно-важные операции, а просмотр WWW к ним, как мы знаем, не относится. Да и Linux-браузеры не содержат такого огромного количества "дыр", как IE.

Если вирусов под Linux нет, спрашивается — зачем же тогда нужен антивирус? Антивирус нужен как раз для обеспечения безопасности Windows-машин. Большинство антивирусов для Linux предназначены для установки на шлюзах — машинах, которые предоставляют доступ к Интернету. Установив антивирус на шлюзе, вы сможете контролировать трафик, проходящий через шлюз. Таким образом, вы защитите Windows-машины от проникновения вируса. Охрану ставят на входе, не так ли? Конечно, антивирус на шлюзе — это не панацея. Не нужно рассчитывать, что он на все 100% обезопасит вашу сеть. Желательно, чтобы на каждой Windows-машине был установлен отдельный антивирус, работающий в режиме монитора.

В этой главе мы рассмотрим бесплатный антивирус ClamAV (**http://www.clamav.net**). Почему именно ClamAV, а не какой-нибудь коммерческий антивирус вроде DrWeb или Kaspersky AntiVirus? Коммерческие антивирусы сопровождаются хорошей документацией, в которой вы разберетесь и без моих комментариев, да и не хочется отбирать хлеб у службы поддержки коммерческих антивирусов.

42.2.2. Установка ClamAV

Для работы ClamAV нужно установить четыре пакета (если пакетов нет в составе вашего дистрибутива, то их можно скачать с сайта **www.clamav.net**):

- clamav — сканер;
- clamav-db (или clamav-base) — антивирусная база данных;
- clamd (или clamav-daemon) — демон Clam (в новых версиях Clam демон clamd входит в состав пакета clamav);
- clamav-freshclam — утилита обновления антивируса (иногда она входит в состав пакета clamav — все зависит от дистрибутива).

Сразу после установки нужно установить соединение с Интернетом (если оно еще не установлено) и выполнить обновление антивирусной базы данных:

```
# /etc/init.d/clamd start
# freshclam
```

Первая команда запускает демон Clam, чтобы у `freshclam` (выполняет обновление базы данных) была возможность сообщить демону об удачном обновлении баз данных.

> ***Примечание***
>
> Команды `clamd` и `freshclam` нужно запускать от имени пользователя root. Напомню, что для этого не нужно входить в систему как root — достаточно использовать команды `su` или `sudo`.

42.2.3. Проверка файловой системы

Сомневаюсь, что в вашей файловой системе будут вирусы (не забываем, что мы используем одну из самых безопасных операционных систем), но все же лучше запустить сканер:

```
# clamscan -r /
```

Эта команда проверит всю файловую систему. Если нужно проверить только отдельный каталог, то вместо `/` укажите имя каталога.

42.2.4. Прозрачная проверка почты

Пришло время настроить "прозрачный" почтовый антивирус. Почтовый антивирус чрезвычайно актуален, ведь большинство так называемых сетевых червей распространяются именно с помощью электронной почты.

Конечно, антивирус ClamAV можно использовать и в режиме обычного сканера, но наиболее интересен он в режиме почтового антивируса. Чуть раньше было сказано, что данный антивирус является прозрачным. Почему прозрачным? Обычный почтовый антивирус "прикручивается" к MTA-агентам путем внесения изменений в их конфигурационные файлы. Агент MTA "знает", что прежде чем передать письмо, его нужно проверить, вызвав прописанный в конфигурационном файле антивирус. Прозрачный антивирус действует независимо от MTA-агента. Более того, MTA-агент даже не подозревает о его существовании. Это очень удобно, хотя

бы потому, что нам не нужно изменять конфигурацию MTA-агента. Вы когда-нибудь "прикручивали" антивирус, например, к sendmail? Если нет, то обязательно попробуйте, когда у вас будет свободное время. После этого вы оцените технологию "прозрачности" ClamAV.

Но простота внедрения — это не единственное преимущество ClamAV. Представьте, что у вас есть почтовый сервер, на котором вы развернули почтовый антивирус. Все бы хорошо — почта ведь проверяется. Но! Ведь у ваших сотрудников есть ящики не только на локальном почтовом сервере. Наверняка найдется несколько человек (если не подавляющее большинство), у которых есть почтовые ящики на бесплатных почтовых серверах, например, на Mail.Ru. В этом случае вирус может попасть в вашу сеть, когда пользователь получает почту с сервера Mail.Ru. Наш антивирус будет бессилен, поскольку он контролирует только наш локальный сервер. Правильно настроенный ClamAV будет проверять абсолютно все почтовые соединения, то есть соединения с 25 и 110 портами любых серверов.

Сам ClamAV является обычным антивирусом, а "прозрачным" его делает сервер P3Scan, скачать который можно по адресу **http://sourceforge.net/projects/p3scan/**.

Антивирус у нас уже установлен и работает, поэтому можно приступить к настройке P3Scan. Работать все будет так: iptables брандмауэра будет перенаправлять пакеты на порт, на котором запущен P3Scan. После этого начинает работать ClamAV, которому P3Scan передает для проверки почту. Неинфицированная почта будет отправлена клиенту.

Теперь, собственно, настройка. Отредактируйте файл /etc/p3scan/p3scan.conf следующим образом:

```
virusregexp = .*: (.*) FOUND
scanner  = /usr/bin/clamdscan --no-summary -i
scannertype = basic
```

Если нужно, измените путь к ClamAV.

Все, что осталось сделать — это создать правило перенаправления POP3-трафика на порт 8110 (на этом порту работает P3Scan):

```
# iptables -t nat -A PREROUTING -p tcp --dport 110 -j REDIRECT --to 8110
```

42.2.5. Проверка Web-трафика

Почта — это не единственный способ распространения сетевых червей и прочей нечисти. Очень много вирусов распространяются по WWW, поэтому нам нужно (на шлюзе) перехватить WWW-трафик, проверить его антивирусом, и если трафик "чистый", передать его пользователю.

Работать прозрачный антивирус Web-трафика будет на базе уже установленного и настроенного прокси-сервера Squid — Squid будет получать запрашиваемый пользователем по WWW файл и с помощью программы Viralator передавать его антивирусу. Кроме программы Viralator, есть и другие программы, которые можно использовать для этой цели, но работать с Viralator проще. Также можно организовать передачу файлов между прокси-сервером и антивирусом с помощью стандартных редиректоров Squid, но они не всегда работают корректно, поэтому мы их использовать не будем.

Скачать программу Viralator можно на сайте **http://viralator.sourceforge.net/**. Кроме Viralator, нам понадобится запущенный на шлюзе Web-сервер Apache — через него и будет запускаться сценарий Viralator.

Теперь можно приступить к настройке. Настройку Squid рассматривать не будем — с ними мы уже знакомы. На уже настроенный Squid нужно установить SquidGuard и отредактировать его конфигурационный файл /etc/squid/squidGuard.conf (листинг 42.2).

Листинг 42.2. Конфигурационный файл etc/squid/squidGuard.conf

```
# Путь к базе SquidGuard и журналам
dbhome /usr/share/squidGuard-1.2.0/db
logdir /var/log/squidGuard

dest files {
expressionlist files-to-check.reg
  }
  acl {

# 10.0.0.1 — это IP Web-сервера, на котором установлен Viralator
default {
pass !files all
redirect
http://10.0.0.1/cgi-bin/viralator.cgi?url=%u
   }
  }
```

Этот конфигурационный файл заставляет SquidGuard передавать файлы, имена которых соответствуют регулярному выражению из файла files-to-check.reg, сценарию viralator.cgi, расположенному на Web-сервере.

Нам нужно создать файл /usr/share/squidGuard-1.2.0/db/files-to-check.reg и поместить в него следующее регулярное выражение:

```
(\.exe$|\.bat$|\.zip$|\.bin$|\.sys$|\.rar$)
```

Как несложно догадаться, эта строка задает типы файлов для проверки — такие типы файлов потенциально могут содержать вирусы. Можете отредактировать эту строку так, как считаете нужным.

Мы пока что связали сценарий Viralator со SquidGuard, но не связали сам SquidGuard со Squid. Для этого откройте файл /etc/squid/squid.conf и добавьте в него следующие строки:

```
redirector_bypass on
redirect_program /usr/local/squidGuard/bin/squidGuard

# максимальное количество копий SquidGuard в памяти
```

```
redirect_children 20
redirector_access deny SSL_ports
redirector_access deny localhost
```

Теперь нужно настроить Apache. Откройте его конфигурационный файл /etc/httpd/conf/httpd.conf и отредактируйте следующие директивы:

```
# указываем IP нашего Web-сервера
Listen 10.0.0.1:80
ServerName 10.0.0.1
```

Не забудьте после этого запустить Apache. Теперь приступим непосредственно к настройке Viralator. Сценарий viralator распакуйте в каталог /var/www/cgi-bin, после чего следует изменить владельца и права доступа сценария:

```
# chown apache:apache /var/www/cgi-bin/viralator.cgi
# chmod +x /var/www/cgi-bin/viralator.cgi
```

Сценарий Viralator требует дополнительный Perl-модуль LWP. Для установки этого модуля нужно ввести команду:

```
# perl -MCPAN -e shell
```

А когда увидите приглашение `cpan>`, то введите команду:

```
install LWP
```

После этого перейдите в каталог /var/www/cgi-bin (именно в него вы должны были распаковать архив с viralator). В этом каталоге будет подкаталог etc, а в нем — подкаталог viralator. Этот каталог viralator нужно скопировать в каталог /etc. После чего удалите каталог etc из каталога /var/www/cgi-bin.

Почти все готово. Осталось только отредактировать конфигурационный файл Viralator: /etc/viralator/viralator.conf (листинг 42.3).

Листинг 42.3. Файл /etc/viralator/viralator.conf

```
servername -> 10.0.0.1       # IP-адрес Web-сервера
antivirus -> CLAMAV  # мы используем ClamAV
virusscanner -> clamscan     # так называется программа-сканер
scannerpath -> /usr/bin      # а это путь к сканеру
viruscmd -> --remove         # опция сканера для удаления вирусов
alert -> FOUND        # сообщение сканера о том, что найден вирус
downloads -> /var/www/html/downloads       # этот каталог нужно создать
downloadsdir -> /downloads
default_language -> english.txt     # язык по умолчанию (русского нет)

# остальное можно не изменять
scannersummary -> true
popupfast -> false
```

```
popupback -> false
popupwidth -> 600
popupheight -> 400
filechmod -> 644
BAR -> bar.png
PROGRESS -> progress.png
```

Создайте каталог downloads и установите права доступа:

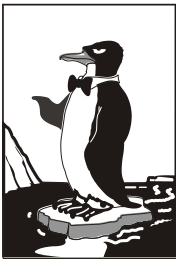
```
# mkdir /var/www/html/downloads
# chown apache:apache /var/www/html/downloads
# chmod 777 /var/www/html/downloads
```

Все настроено! Теперь машины наших клиентов нужно настроить на использование нашего прокси-сервера (10.0.0.1, порт 3128) и приступить к тестированию!

ЧАСТЬ VIII

Теория и практика системного администратора

В восьмой, заключительной, части книги мы поговорим о суровых буднях системного администратора. С одной стороны, мы будем говорить о практике, а с другой стороны, все эти разговоры — чистой воды теория. Потому что все равно у вас будет все иначе, чем написано в книге.



Глава 43

Стратегия администрирования

43.1. О чем эта глава?

Вы можете быть самым профессиональным администратором в мире, у вас может быть самая лучшая команда (из таких же профессионалов, как вы), но без четкой стратегии администрирования вы далеко не зайдете.

Инструкции, приведенные в этой главе, одинаково пригодятся как единственному администратору в компании, работающему на полставки, так и руководителю IT-отдела.

Внутри любой IT-службы можно выделить следующие подразделения:

- **руководство** — сюда входят руководитель отдела ("верховный" администратор) и несколько его заместителей. Как показывает практика, это подразделение не должно быть большим. Одного-трех человек вполне достаточно (но не более 5% от всего IT-отдела);
- **отдел администрирования** — к этому подразделению относятся как раз сами администраторы, выполняющие всю основную работу по поддержанию всей сети в рабочем состоянии. Размер отдела, наверное, самый большой, хотя многое зависит от профиля самой компании;
- **отдел разработки** — создается по мере необходимости. Одни компании занимаются разработкой программного обеспечения, другим нужны программисты для доработки уже существующих программных продуктов. Например, если компания занимается разработкой программного обеспечения, то отдел разработки будет самым большим. А вот в некоторых организациях, наоборот, отдела разработки может и не быть. Экономия…
- **отдел поддержки** — занимается либо поддержкой собственных пользователей, либо поддержкой клиентов (если компания занимается разработкой программного обеспечения). Сотрудники этого отдела также должны заниматься обучением работе с собственным программным обеспечением.

Подобная структура может существовать в компании среднего или крупного размера. Иногда весь IT-отдел состоит из двух-трех человек: руководителя (главный администратор) и двух помощников. Руководитель заменяет все подразделение управления, а помощники выполняют функции администрирования и поддержки пользователей. Отдела разработки в таких компаниях, как правило, нет: или один из администраторов выполняет некоторые функции программиста (например,

"допиливает" 1С), или же компания нанимает сторонних специалистов (для разовой работы так выходит намного дешевле).

Когда ваш отдел разрастается, помните о золотом правиле: один руководитель на десять человек. Как только число человек превысит 10, нужно разделяться. Например, 5 человек будут администрировать сеть, а 5 — заниматься поддержкой пользователей. Вы же — будете управлять всеми ими. Поверьте, это не просто. В идеале один человек может управлять группой не более 10 человек. Если в группе окажется больше 10 человек, нужно разделить ее на несколько групп и в каждой группе назначить главного. Тогда вы будете управлять всего несколькими руководителями групп, а они уже — своими группами.

43.2. И руководство, и пользователи довольны. Миф или реальность?

Часто бывает так, что либо руководство, либо пользователи недовольны. Самый сложный случай, когда недовольны и руководство, и пользователи. Например, руководство недовольно сроками внедрения новой системы управления предприятием, но вы задерживаетесь с ее внедрением, поскольку не успеваете параллельно заниматься поддержкой пользователей. А текущие проблемы (у кого-то зависает компьютер, "упал" Интернет, нужно заменить картридж в принтере) занимают, действительно, много времени. Хуже всего, когда выполнение одних обязанностей мешает вам исполнять другие. В итоге руководство будет недовольно вдвойне: вы не выполнили план и на вас (на ваш отдел) есть жалобы пользователей.

Давайте разберемся, как сделать так, чтобы все были довольны. С руководством, с одной стороны, проще — нужно выполнять их указания и вовремя предоставлять отчеты о проделанной работе. А как же пользователи? А пользователь доволен в следующих случаях:

- когда его компьютер работает без сбоев и зависаний;
- когда "работает Интернет", и не просто работает, а быстро работает;
- когда установлено программное обеспечение, с которым умеет работать сотрудник (желательно, чтобы оно работало без сбоев);
- когда есть приветливая служба поддержки.

Казалось бы, все логично и просто. Но вот компьютер пользователя начал зависать и самопроизвольно перезагружаться. Проблема с оперативной памятью налицо. В компьютере установлено два модуля оперативной памяти. Вы один извлекли. В результате компьютер стал работать нормально, но медленнее. Теперь пользователь жалуется, что компьютер работает медленно. Где взять новый модуль? Правильно, в магазине. Но не покупать же его за свои средства? Приходится обращаться непосредственно к начальству (либо к финансовому директору, бухгалтеру и т. д.), объяснять проблему и получать "добро" на покупку нового модуля.

Но часто бывает так. Директор спрашивает, работает ли компьютер в данный момент? Вы отвечаете, что работает, но медленно. Но, услышав слово "работает", остальное он слышать не хочет. Денег нет, экономия, кризис и т. д. Это с одной стороны. А с другой стороны — пользователь, которому все равно, как вы сделаете,

чтобы компьютер работал быстрее. Он ведь должен жаловаться именно вам. Получается, что администратор как бы между двух огней. С одной стороны — руководство и экономия, с другой — пользователи. К этому нужно привыкать и пытаться находить компромиссы. Этому вас научит книга по психологии, а не самоучитель системного администратора.

С Интернетом тоже часто бывают проблемы. Вспоминаю свой опыт работы "верховным" системным администратором. Мой рабочий день начинался в 9 часов утра, собственно, раньше добраться до места работы не получалось. Зато некоторые пользователи, в том числе и финансовый директор, любили приходить пораньше. А Интернета "нету". И дело не в настройках сервера. А просто по утрам он "падал" — такой был провайдер. К моему приходу на работу меня уже поджидали недовольные пользователи. Звонок в службу поддержки провайдера, полчаса ожидания, соединение установлено. Хорошо, что в число "любителей Интернета" входил финансовый директор. Закончилось все сменой провайдера, и договор с новым провайдером подписывал именно он.

Идем дальше — программное обеспечение. Понимаю, что знания пользователей — это их личные проблемы. Вы не обязаны обучать каждого пользователя компьютерной грамотности. Ведь есть же специальные курсы. Но бывает так, что в целях экономии компании переходят на открытое программное обеспечение. Например, вместо привычного пользователям пакета MS Office устанавливается бесплатный OpenOffice.Org. Он похож на MS Office, но как бы ни была хороша копия — это не оригинал. А еще хуже, когда устанавливается Linux. Нет, Linux — отличная операционная система, но уж больно она отличается от привычной Windows. Некоторого программного обеспечения для Linux нет, поэтому запускать его придется в эмуляторе wine. Не исключено, что под эмулятором программы будут работать не так, как нам бы этого хотелось.

Экономия экономией, но о пользователях тоже нужно заботиться. Какой прок от бесплатной ОС и от экономии, если первые полдня пользователь не может выполнить привычные действия, а другую половину дня администратор объясняет ему, как работать в Linux (да, вместо того, чтобы заниматься более полезными делами)? Поэтому переход на Linux нужно начинать постепенно. В первую очередь Linux нужно устанавливать на компьютеры, где не нужно специальное программное обеспечение. Если на компьютере установлен браузер, почтовый клиент и пакет MS Office (типичный компьютер типичного менеджера) — это первый кандидат для перехода на Linux. Также нужно смотреть на уровень квалификации самого пользователя. Если перед вами женщина предпенсионного возраста, то на ее компьютер Linux лучше установить в последнюю очередь. Linux не так уж и сложен, просто к нему нужно привыкнуть. А постепенный переход означает, что вы будете получать 5 жалоб в день, а не 100.

Разберемся, что не нравится пользователям (это тоже важно учитывать):

- простои (даже запланированные);
- поврежденные или нечаянно удаленные файлы;
- обновления (пользователям не нравятся изменения, особенно если они вносят некоторые неудобства в привычный рабочий процесс);
- долгие объяснения, почему система (сеть) не работает.

Вообще пользователю не нравится, когда система (сеть) не работает. Им хочется, чтобы система работала круглосуточно, а почему она работает (или не работает) — они знать не желают. Да, проблемы — ваши. Всем интересен результат, а обеспечить этот результат должны вы. И вообще, как бы обидно это ни звучало, пользователям вообще все равно — существуете вы или нет. Они вспоминают об администраторе только тогда, когда что-то ломается.

А о простоях нужно предупреждать пользователей заранее, например, за день — чтобы они успели закончить текущие дела. Хотя все равно найдутся недовольные, так что особенно можете не обращать на них внимание. Есть такая категория людей, которая всем и всегда недовольны.

Относительно удаленных файлов — тут решать вам: или вы сообщаете пользователям, что они и только они отвечают за создание резервных копий своих собственных файлов, или же придется заниматься резервных копированием самостоятельно. Когда пользователи вынуждены сами делать резервные копии — гора с плеч (с ваших), но вы — плохой администратор. А если "бэкапы" делаете вы, то вы — хороший администратор. Но забот у вас станет намного больше, чем вы думаете. И придется подыскать хороший программный комплект для резервного копирования по сети (не будете же вы вручную скачивать данные с каждого компьютера?). Я рекомендую пакет Amanda Network Backup (**http://www.amanda.org**) — он бесплатный и может использоваться как в Linux, так и в Windows.

Итак, будем считать, что и руководство, и пользователи — довольны. Но что это мы только о них и говорим? А чего бы хотелось самим администраторам? Примерно вот этого:

- имеются все необходимые для выполнения обязанностей ресурсы (в том числе и финансовые, и неограниченный доступ к Интернету);
- имеется нормальный парк "железа", а не музейные экспонаты, год выпуска которых можно установить только путем экспертизы;
- чтобы не приходилось отвлекаться от выполнения первоочередных обязанностей (нужно донастроить брандмауэр iptables, а тут пришел пользователь, который не знает, на какую кнопку нажать);
- чтобы можно было творчески подойти к решению задачи без лишнего контроля и придирок руководства;
- чтобы количество рабочих часов было нормированным (ненормированный рабочий день нравится далеко не всем — есть свои интересы помимо работы).

Но на практике, к сожалению, бывает наоборот. Старые компьютеры, ограниченный доступ к Интернету (с учетом трафика даже для администратора — и такое проходили), множество недовольных пользователей (компьютеры старые, постоянно зависают и медленно работают). Так что нужно быть готовым к разным ситуациям.

43.3. Роль главного администратора

Данная часть главы будет полезна руководителю IT-отдела или главному администратору. Вот основные обязанности руководителя:

- определение направления деятельности, предоставление необходимых ресурсов (да, "выбивать" денежку от вышестоящего начальства придется именно вам);

- набор и увольнение персонала;
- отчеты перед вышестоящим начальством;
- распределение задач, контроль их выполнения;
- решение конфликтов между сотрудниками;
- контроль развития сети (чтобы она не потеряла масштабируемости);
- ведение локальной документации (схема сети, расположение компьютеров, установленное программное обеспечение и т. д.).

Может, вам показалось, что в этом списке кое-чего не хватает? Действительно, может, чего-то и не хватает — все зависит от специфики организации, в которой работает администратор. Одно могу сказать точно: руководитель IT-отдела не должен напрямую общаться с пользователями. Решением их проблем пусть занимаются другие администраторы. Однако пользователи могут жаловаться главному администратору на действия (или бездействие) других администраторов, например, нахамил, игнорирует просьбы и т. д.

Особого внимания заслуживают набор и увольнение персонала — от этого зависит, с кем вы будете работать. Управление персоналом — нелегкая задача. Ведь нужно оценивать не только технические знания, но и личностные качества. Человек может быть отличным специалистом в своей области, но как человек… Продолжать не буду, иначе на обложке появится надпись "Осторожно! Ненормативная лексика". Если вы все-таки хотите взять такого человека в отдел, то явно не в службу поддержки пользователей. Пусть настраивает серверы или выполняет ту работу, которую лучше него никто не сможет сделать. Однако, если есть возможность, стоит все же вообще отказаться от приема такого сотрудника в отдел.

Технические знания сотрудника оценить довольно легко (при условии, конечно, что вы сам — профессионал своего дела). Несколько коварных вопросов, и вы будете знать уровень профессиональных знаний будущего сотрудника. Если у вас есть время и вам не лень, можно разработать даже систему тестирования знаний. Совсем не обязательно разрабатывать программный комплекс, достаточно создать в текстовом редакторе что-то наподобие анкеты, распечатать ее, и пусть кандидаты в сотрудники отвечают на вопросы этой анкеты. А вы уже оцените уровень знаний.

Надо, тем не менее, понимать, что анкетирование дает только относительное представление о знаниях и способностях сотрудника. Например, вы задаете вопрос, на который человек не знает правильного ответа, поскольку никогда с этим не сталкивался. Но если человек способен к самообучению, то он найдет информацию в Интернете и сможет решить поставленную задачу. Увы, это можно узнать только на практике. А если каждому предоставлять доступ к Интернету во время заполнения анкеты, то все будут специалистами высочайшего класса. Но как раз для этого и существует испытательный срок (обычно 3 месяца). За этот срок сотрудник проявит себя и как специалист, и как человек. Вам важно определить, вписывается ли сотрудник в коллектив, как он общается с пользователями, клиентами и другими администраторами. Если человек не вписался в коллектив, то его лучше уволить, даже если с профессиональной стороны к нему претензий нет. Если его оставить,

то головная боль в будущем вам обеспечена — вы только и будете заниматься разрешением конфликтов внутри коллектива.

Есть два способа сформировать команду профессионалов. Первый заключается в том, что вы нанимаете уже опытных администраторов. Второй — вы обучаете имеющихся сотрудников со средним уровнем квалификации. С одной стороны, опытный администратор сразу включается в работу. С другой, у опытных администраторов есть один недостаток — их приходится от многого отучать. Так происходит со всеми профессионалами своего дела, а не только с администраторами. Например, водитель-дальнобойщик, попав за руль "легковушки", при переключении передач делает "перегазовку", а этого не только не нужно делать на "легковушке", но и нельзя, так как вредно для сцепления. Опытные администраторы также сразу требуют привилегированного доступа ко всему, но им его сразу давать нельзя, поскольку вы еще ничего о них не знаете — можно ли им доверять или нет.

Когда же вы выращиваете администраторов в своем коллективе, то всегда можете обучить их так, как вам нужно. Но на все это нужно время, которого иногда мало.

Кроме управления персоналом вам придется заниматься распределением заданий и контролировать их выполнение. При этом избегайте следующего:

- не мешайте своим сотрудникам — нужно управлять задачами, но не вмешиваться. Если вы постоянно стоите "над головой" у младших администраторов, это будет им только мешать;
- недоразумений при выполнении заданий — иногда задания не выполняются, потому что человек, который должен выполнить то или иное задание, думает, что его должен выполнить кто-то другой. Каждый сотрудник должен знать свое задание и сроки его выполнения;
- излишнего расходования ресурсов — иногда руководители, чтобы быть точно уверенными в выполнении задания, поручают одно и то же задание нескольким сотрудникам или отделам (чтобы каждый сотрудник или каждый отдел лично работал над заданием). Такая тактика в некоторых случаях себя оправдывает, но не забывайте, что действия сотрудников нужно кому-то координировать. Например, вы поручили Иванову и Петрову настроить Apache. Иванов внес одни изменения в конфигурационный файл, а Петров, ничего не подозревающий о действия Иванова, через час внес в этот файл другие изменения. В итоге сервер так и не работает. Поэтому, если предположить, что Иванов с заданием справится, лучше поручить Петрову другое задание — так будет рациональнее. Это как ехать на двух машинах в булочную только из-за перестраховки, что одна из машин может поломаться. Целесообразнее выбрать "самую исправную" машину и отправить ее за хлебом. Шутка...

Руководителю IT-отдела постоянно приходится общаться с вышестоящим руководством. Именно от него можно получить средства, необходимые для выполнения работы (на покупку расходных материалов, комплектующих, программного обеспечения). Отчеты о проделанной работе тоже нужно составлять для начальства.

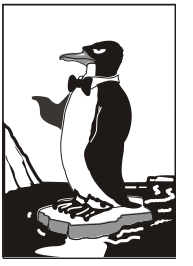
Говоря с начальством, помните, что оно не имеет никакого представления о том, чем занимаются системные администраторы. Поэтому старайтесь говорить (состав-

лять отчеты) максимально понятно. Старайтесь избегать профессионального сленга и использования непонятных начальству аббревиатур, иначе вы не убедите их в необходимости покупки нового оборудования и найма дополнительных сотрудников.

Когда пишете отчет для начальства, придерживайтесь пяти простых правил:

- все должно быть максимально понятным;
- старайтесь не использовать непонятных (для начальства!) терминов, обязательно поясняйте, что означает тот или иной термин;
- расшифровывайте аббревиатуры (также обязателен перевод расшифровки аббревиатуры на русский язык);
- когда требуете средства на новое оборудование, обязательно поясняйте, что даст замена старого оборудования новым;
- когда требуете расширение штата, обязательно объясните, чем будет заниматься каждый новый сотрудник.

Надеюсь, рекомендации, приведенные в этой главе, хоть как-то помогут вам. В следующей главе мы поговорим об уходе за аппаратными средствами.



Глава 44

Уход за "железом"

44.1. Обязанности администратора

В обязанности администратора очень часто входит не только настройка программного обеспечения, но и уход за "железом". Да, чистка всего компьютерного парка. И если внешнюю чистку можно возложить на плечи самих пользователей (ну не будете же вы всем мониторы протирать), то внутреннюю чистку должны производить только вы сами.

Правда, чистка — это не единственное занятие администратора. Вам также придется заниматься ремонтом (в пределах разумного) и модернизацией компьютеров предприятия.

Некоторые предприятия, отличающиеся особо большим парком компьютеров, нанимают сторонние фирмы для чистки и профилактики компьютеров. Если вы работаете на таком предприятии — вам повезло. Вы будете заниматься только настройкой программного обеспечения. Настроили Linux-серверы и забыли, останется только время от времени Windows на рабочих станциях переустанавливать. Однако в большинстве случаев на таких услугах экономят, и все действия по уходу за компьютерами (всеми компьютерами, а не только серверами) должен выполнять системный администратор.

Теоретически, парк до 30 компьютеров в состоянии обслуживать один человек. А вот если компьютеров больше, то нужен как минимум еще один человек — помощник. Поэтому если вы оказались на предприятии, где больше 30 компьютеров и всего один администратор (то есть вы), смело требуйте расширения штата ИТ-отдела. На первый взгляд кажется, что и 50 компьютеров вы "потянете". Но это не так. Поверьте мне на слово. Обслуживать даже 30 компьютеров — это довольно тяжело. Представьте, что ночью была гроза, а утром вы обнаружили, что 5 компьютеров (да, грозозащита — это хорошо, но современные реалии таковы, что она присутствует далеко не всегда) не работают. По закону подлости — это компьютеры директора, главного бухгалтера, ведущего менеджера и других немаловажных лиц на предприятии. Какой компьютер нужно отремонтировать первым? Куда бежать, если сгорел блок питания? В общем, приятного мало. А вот когда у вас есть помощники, проблема решится быстрее.

Есть и другая ситуация, которая намного проще предыдущей. Предположим, что вам нужно проложить сеть в другой корпус предприятия. Эту операцию намного

проще выполнять с помощником, нежели самому. Точнее, без помощника вам вообще не обойтись.

Подобных примеров могу привести очень много, но толку от них не будет. Лучше приступим к формированию собственного обменного фонда.

44.2. "Про запас", или обменный фонд

Как администратору, вам нужно сформировать собственный обменный фонд аппаратных средств и периферийных устройств. Некоторые комплектующие и периферийные устройства имеют "привычку" выходить из строя с завидной регулярностью. Пример: дешевые китайские мыши, клавиатуры. Вы донастраиваете сервер, а у главного бухгалтера поломалась мышь. В итоге вам нужно все бросить и ехать в ближайший компьютерный магазин, вместо того, чтобы спокойно доделать текущую работу. А все это время и нервы. А если у вас будет запасная мышка, то проблема решится за пару минут.

Кроме клавиатур и мышек в обменном фонде должны быть:

- **жесткие диски** — обычно хватит одного-двух запасных SATA-дисков (они наиболее распространены на современных компьютерах). Если в вашем компьютерном парке есть компьютеры с устаревшими IDE-дисками, запасайтесь и ими. Сейчас достать жесткий диск (новый) с IDE-интерфейсом довольно сложно — в наличии они бывают не всегда, а под заказ приходится ждать 1–2 дня. Полагаю, через год найти IDE-диск будет еще сложнее. И тогда в случае выхода из строя IDE-диска придется покупать новую материнскую плату (скорее всего, процессор и оперативную память тоже придется менять) или специальный SATA-контроллер для подключения нового SATA-диска. Ну, в самом деле, не выбрасывать же компьютер из-за одного жесткого диска? Понимаю, что он устарел, но для работы с текстом (офисный компьютер — это не игровая станция) он вполне подходит.

  Если найти новый IDE-диск вы не можете, попробуйте купить б/у. Вот только перед покупкой его желательно проверить на наличие плохих секторов (в Linux для этого используется программа badblocks). Много покупать б/у-дисков не стоит — это коты в мешке — сегодня работают, а завтра — "посыпались". Так что уже сегодня можете "порадовать" руководство, что вам нужно минимум 2 IDE-диска (если они, конечно, используются на ваших компьютерах) и 1–2 SATA-диска. SATA-диски всегда есть в наличии, да и учитывая, что они более "молодые", вероятность выхода их из строя ниже, чем у IDE-дисков. Поэтому одного-двух вполне хватит;
- **блоки питания ATX** — никогда не угадаешь, когда блок питания выйдет из строя. Поэтому нужно запастись парочкой блоков питания. Не покупайте дешевые блоки мощностью 250 Вт. Для современного компьютера 300 Вт — это необходимый минимум, а лучше покупать блоки питания мощностью 350 Вт. Иначе "голодание" вашему компьютеру обеспечено. Запасные блоки питания также пригодятся вам при чистке основных блоков питания. Чистить блок питания рекомендуется раз в год, но процесс чистки занимает минут 20–30, и чтобы

сервер не простаивал, вы можете установить запасной блок питания, а основной блок питания почистить без всякой спешки;

- **картриджи для принтеров** — заниматься самостоятельной заправкой тонера нельзя — это вредно для здоровья. Лучше отнести картридж в сервисный центр. На заправку картриджа потребуется 1–2 дня (учитывая загрузку сервисного центра), а чтобы в это время принтер не простаивал, нужен запасной картридж. Хорошо, если все принтеры на предприятии одного производителя и одной модели. В противном случае придется покупать несколько разных картриджей. Не спешите с покупкой картриджей — нужно выяснить, какие принтеры используются чаще всего, а потом уже планировать закупку картриджей для этих принтеров;
- **носители информации** — у системного администратора просто должен быть запас DVD-болванок и флешек. Вот только раздавать болванки и флешки всем пользователям не рекомендую — иначе они очень быстро израсходуются. Используйте эти носители для своих целей, например, для резервного копирования;
- **привод DVD-RW** — в приводе DVD (как и CD) много механических частей, а это означает, что его вероятность выхода из строя довольно высока. Лучше всего купить USB-привод. Его можно использовать как временную подмену вышедшему из строя приводу как обычного компьютера, так и ноутбука;
- **коммутатор** (switch) — иногда эти "коробочки" ломаются (или сгорают отдельные порты). Чтобы привести сеть в чувство, требуется замена коммутатора. Хорошо, если запасной коммутатор есть под рукой. Нужно покупать коммутатор как минимум на 16 портов, а еще лучше — на 24 порта. Чем больше компьютеров смогут работать после выхода из строя коммутатора, тем лучше.

Вот вроде бы и все. Переходим к чистке компьютеров.

44.3. Чистка компьютеров. Профилактика системы охлаждения

Чистку компьютера следует производить хотя бы раз в полгода, а чистку блока питания — раз в год. Для чистки компьютера нужно обзавестись обычным пылесосом. Можно даже использовать компактный автомобильный пылесос. Стоит такой пылесос недорого, зато он очень удобен.

Думаю, понятно, что чистка должна быть сухой. Никаких влажных салфеток и моющих средств! При чистке компьютера особое внимание уделите радиатору процессора. Возможно, придется снять процессор, чтобы хорошо очистить радиатор. Радиатор, забитый пылью, отнюдь не способствует отводу тепла, а выполняет обратную функцию — еще больше нагревает процессор.

Если вентилятор процессора или блока питания издает посторонние звуки (помимо естественного шума), его нужно смазать. Для смазки подойдет любое машинное масло — хватит одной-двух капель масла. Но это только в том случае, если вентиляторы оснащены специальным отверстием для смазки. Оно находится под наклейкой по центру вентилятора. Обычно на вентиляторах блока питания такое отверстие имеется, а вот что касается вентилятора процессора, то тут ваши шансы найти такое отверстие равны 50%. На одних вентиляторах такое отверстие есть,

на других — нет. Иногда из-за этого приходится менять вентилятор — он шумит, а смазать его нельзя.

Со стационарными компьютерами все очень просто. А вот с ноутбуками все намного сложнее. Чтобы добраться до вентилятора процессора, приходится полностью разбирать некоторые модели ноутбуков. Вот пример разборки ноутбука HP: **http://smanuals.ru/electronics-repair/hp-compaq-6730s-6735s-disassembly-replace-cooling-fan.html**. Без этого руководства я бы никогда не добрался до вентилятора процессора. А если бы и добрался, то наверняка что-нибудь сломал. Отсюда вывод — перед разборкой ноутбука попробуйте найти соответствующее руководство в Интернете. Если же такого руководства нет, а заднюю крышку все равно снять не получается, лучше обратитесь в сервисный центр. В противном случае ремонт выйдет дороже, чем обычная чистка.

Кстати, уборку в серверном помещении я бы не доверял уборщицам. А то у нас часто бывает, что любая уборщица хуже лучшего хакера. Одно неаккуратное движение шваброй, и сервер "умер". Кто будет убирать в вашем серверном помещении — решайте сами. Но убирать там нужно регулярно.

44.4. Охлаждение компьютеров

Нормальная температура для работы компьютеров — примерно 20 °C. Но сами понимаете, что такие условия встречаются далеко не всегда, особенно летом. Сейчас программа SpeedFan показывает температуру ядра процессора 71 °C (ноутбук HP 6735s), температура в помещении около 30 °C, пора включать кондиционер.

А вот теперь начинается самое интересное. Многие из нас привыкли подбирать кондиционер по площади помещения, в котором он будет установлен. Но это в корне не правильно. Нужно рассчитать общую тепловую нагрузку, а затем подбирать кондиционер, соответствующий этому параметру.

Тепловая мощность измеряется в БТЕ (BTU, British thermal unit, британская термическая единица). 1 Вт примерно равен 3.412 БТЕ/час. Пусть в помещении находятся 10 компьютеров, каждый из которых потребляет по 400 Вт. Рассчитаем тепловую мощность:

$$10 \times 400 \times 3.412 = 13648 \text{ БТЕ/час}$$

Кроме компьютеров источниками тепла являются сами пользователи и осветительные приборы. Пусть в помещении включено 5 лампочек по 100 Вт каждая, рассчитаем тепловую мощность:

$$5 \times 100 \times 3.412 = 1706 \text{ БТЕ/час}$$

Один пользователь выделяет тепла на 300 БТЕ/час. Выходит, наши 10 пользователей создадут нагрузку в 3000 БТЕ/час.

Осталось учесть тепловую нагрузку от окон, стен и потолка. Например, если у вас солнечная сторона, то нагрузка от окон будет больше, чем на противоположной стороне здания. Также если вы находитесь на последнем этаже, то нагрузка будет еще и от крыши, которая летом постоянно нагревается. Все это пусть считает специалист по установке кондиционеров. Вы же добавите к полученному показателю свои значения. Осталось сравнить общую тепловую нагрузку с эффективностью

охлаждения кондиционера (параметр EER), подробно об этом можно прочитать тут: **http://en.wikipedia.org/wiki/Energy_efficiency_ratio**.

Некоторые компании экономят на охлаждении, устанавливая кондиционеры только в кабинетах директоров, а "рабочий класс" обречен спасаться от жары с помощью вентиляторов. В этом случае ничем компьютеру особо не поможешь. Старайтесь размещать системные блоки так, чтобы горячий "выхлоп" одного компьютера не попадал на воздухозаборник другого.

Пользователям ноутбуков можно помочь с помощью подставок для охлаждения корпуса ноутбука. Эти подставки оснащены одним или двумя вентиляторами, охлаждающими корпус установленного сверху ноутбука (рис. 44.1). КПД подставки довольно низкий, поскольку охлаждается корпус, но не сам процессор. Однако такая подставка лучше, чем ничего — она ведь также охлаждает и жесткий диск ноутбука. Толк от такой подставки будет, если ноутбук предварительно очищен от пыли. Иначе воздух от вентиляторов просто не будет проникать в вентиляционные отверстия корпуса ноутбука.

Рис. 44.1. Подставка для охлаждения ноутбука

У таких подставок есть еще один недостаток — цена. Кондиционер начального уровня стоит примерно 4500 рублей. А вот одна такая подставка— от 1000 до 2000 рублей. Выходит, четыре подставки равны одному кондиционеру. Если руководство компании не потратилось на кондиционер, то сомневаюсь, что оно выделит деньги на подставки.

44.5. Стойки для оборудования

Для большего порядка в серверной рекомендуется применять серверные стойки или шкафы. Шкаф стоит дороже, но зато его можно закрыть на ключ, что предотвращает несанкционированный физический доступ к серверу, даже если кто-то случайно окажется в серверной комнате. Типичный серверный шкаф изображен на рис. 44.2.

Как видите, кроме самих серверов в шкаф можно поместить и сетевое оборудование, что очень удобно. На рис. 44.3 изображена серверная стойка.

Рис. 44.2. Серверный шкаф

Рис. 44.3. Серверная стойка

44.6. Влажность

Идеальная влажность для компьютерных систем — от 40 до 55%. Если влажность низкая, возникнут проблемы с электростатическими зарядами. Если же влажность слишком высокая, влага будет конденсироваться на платах, что вызовет окисление контактов и замыкание. Бороться с высокой влажностью можно с помощью кондиционеров с функцией осушения воздуха. А вот если влажность низкая, подойдут увлажнители воздуха (самый дешевый стоит около 2000 рублей).

Что касается статического электричества, то, прежде чем приступить к разборке компьютера, лучше всего полностью выключить его (из розетки в том числе) и воспользоваться антистатическим браслетом (рис. 44.4). Такой браслет надевается на запястье и подсоединяется к "заземлению" — к третьему штырьку в розетке, если таковой имеется…

Рис. 44.4. Антистатический браслет

44.7. Инструмент системного администратора

Любому системному администратору пригодятся следующие инструменты:

- набор плоских и крестообразных отверток — всегда нужен, ведь разбирать и собирать компьютеры придется часто;
- набор мелких ювелирных отверток — вы их будете использовать редко, но хорошо, если таковые будут под рукой, чтобы потом не пришлось в спешке их покупать;
- набор шестигранных ключей, набор ключей типа TORX (шестигранная звезда) — без таких ключей некоторые устройства (например, ноутбуки) разобрать не получится;
- пинцет — некоторые мелкие вещи, которые вы уроните в системный блок, намного удобнее доставать с помощью пинцета;
- фонарик — освещение не всегда отличное, поэтому фонарик тоже нужен, желательно купить светодиодный фонарик — он лучше светит;
- ножницы — на всякий случай;
- инструмент для обжима витой пары — думаю, зачем он нужен, говорить не нужно;
- запасные разъемы RJ-45 — стоят копейки, поэтому чем больше, тем лучше (в пределах разумного, конечно);
- инструмент для зачистки проводов — для зачистки проводов можно также использовать и инструмент для обжима витой пары, но лучше купить отдельный специальный инструмент;
- антистатический браслет — ранее мы говорили, зачем он нужен;

- кроссовер — может пригодиться для прямого (в обход коммутатора) соединения двух компьютеров;
- обжатый Ethernet-кабель — пригодится для подключения вашего ноутбука к коммутатору (для тестирования соединения), минимальная длина такого кабеля — 1 метр;
- цифровой мультиметр — объединяет в себе несколько измерительных приборов, как правило, это вольтметр, амперметр и омметр (рис. 44.5).

Рис. 44.5. Цифровой мультиметр

Заключение

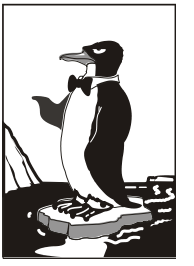
Вместо заключения порекомендую две своих книги. Первая — "Linux. От новичка к профессионалу. 2-е изд." Можно сказать, что это наиболее полная книга по Linux, написанная мною. Ознакомиться с книгой можно по адресу: **http://bhv.ru/books/book.php?id=186944**. Надо также отметить, что в настоящее время подготавливается и 3-е издание этой популярной книги (выйдет из печати ориентировочно в декабре 2010 года), в котором будут учтены все изменения, произошедшие в мире Linux со времени выхода 2-го издания.

Вторая книга посвящена настройке беспроводной сети. В книге, которую вы держите в руках, конечно, есть пример монтажа беспроводной сети, но в книге "Беспроводная сеть дома и в офисе" настройка сети описана подробнее. Ознакомиться с книгой можно на сайте издательства: **http://bhv.ru/books/book.php?id=185666**.

Если у вас есть какие-либо вопросы, комментарии или просто пожелания, связаться со мной можно на форуме сайта **www.dkws.org.ua**. Все читатели могут рассчитывать на *посильную* помощь автора по настройке Linux. Это вовсе не означает, что настраивать сервер за вас буду я, но если что-то не будет получаться, советом обязательно помогу.

Денис Колисниченко

ПРИЛОЖЕНИЯ



Приложение 1

Параметры ядра

Параметры ядра позволяют управлять поведением ядра. Как уже было отмечено в предыдущих главах книги, мы можем передать параметры ядра непосредственно при загрузке, используя меню загрузчика, или же прописать параметры ядра в файлах конфигурации загрузчика. Первый случай подходит для "одноразового" использования того или иного параметра, а второй — если параметр нужен для корректной работы системы, поэтому, чтобы не указывать его каждый раз при загрузке Linux, намного проще внести его в файл конфигурации загрузчика.

В SUSE для передачи параметров ядра нужно выбрать загрузочную метку, а затем ввести нужные вам параметры в поле **Параметры загрузки** (рис. П1.1).

Параметров ядра очень много, поэтому в табл. П.1 собраны самые полезные.

Рис. П1.1. Редактирование параметров ядра

Таблица П.1. *Некоторые параметры ядра Linux*

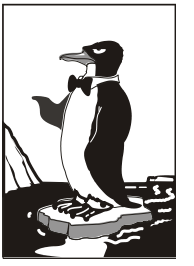
Параметр	Описание
`root=устройство`	Позволяет указать корневую файловую систему Например, `root=/dev/hda5`
`ro`	Монтирует корневую файловую систему в режиме "только чтение". Используется по умолчанию. После проверки файловой системы программой fsck корневая файловая система перемонтируется в режим `rw`
`rw`	Монтирует корневую файловую систему в режиме "чтение/запись". При использовании этого параметра нельзя запускать программы типа fsck. Перед запуском fsck нужно перемонтировать корневую файловую систему в режиме `ro`
`mem=`	Определяет объем памяти, установленной в компьютере. Иногда ядро неправильно определяет объем оперативной памяти. Вы можете помочь ему в этом, указав параметр `mem`. Только указывать его нужно правильно, например: `mem=768M` После числа обязательно должна следовать буква `M`, иначе ядро "подумает", что объем оперативной памяти 768 байтов
`init=`	Позволяет задать программу инициализации. По умолчанию используется программа `/sbin/init`, но вы можете задать другую программу, например, `/bin/bash`, если вам нужно обойти сценарии `init` (например, когда вы забыли пароль root)
`reboot=`	Позволяет задать тип перезагрузки компьютера. Возможные значения: `cold` и `warm`, т. е. "холодная" или "горячая" перезагрузка
`single`	Однопользовательский режим для администрирования системы, например в случае отказа
`nodmraid`	Отключает программные RAID-массивы, организованные на уровне BIOS
`noapic`	Полезен, если вы при загрузке увидите сообщение: **kernel panic - not syncing: IO-APIC + timer doesn't work!** Подробнее об этом параметре вы можете прочитать по адресу: **http://www.dkws.org.ua/phpbb2//viewtopic.php?topic=2973&forum=5**
`nopcmcia`	Отключает PCMCIA-карты (для ноутбуков). Полезен, если вы подозреваете, что у вас проблемы с PCMCIA-картой
`nodma`	Отключается DMA (Direct Memory Access, прямой доступ к памяти) для всех IDE-устройств
`noapm`	Отключает APM (Advanced Power Management) — расширенное управление питанием
`nousb`	Отключает поддержку USB
`noscsi`	Отключает поддержку SCSI

Таблица П.1 *(окончание)*

Параметр	Описание
`pci=noacpi`	Не использовать ACPI для управления PCI-прерываниями
`apci=off`	Полностью отключает ACPI (Advanced Configuration and Power Interface). Полезен на некоторых ноутбуках, когда не удается установить (а потом загрузить) Linux
`edd=off`	Отключает поддержку EDD (Enhanced Disk Device). Полезен, если установка зависает при определении параметров жесткого диска или же параметры жесткого диска, определенные программой установки, не соответствуют реальным (например, не совпадает размер и количество разделов)

ПРИМЕЧАНИЕ

С дополнительными параметрами ядра вы можете ознакомиться по адресу **http://dkws.org.ua/phpbb2/viewtopic.php?t=3031**.



Приложение 2

Суперсервер xinetd

П2.1. Сетевые сервисы и суперсервер

Сетевые сервисы могут запускаться автономно или только по требованию, то есть при получении от клиента запроса. Автономно запускаются те сервисы, от которых клиент ожидает немедленной реакции, например, Web-сервер, FTP-сервер, DNS-сервер. Другие сетевые сервисы, например, finger, tftp, могут позволить себе запуск по требованию. Но как передать запрос сервису, который не запущен? Ведь процесс не запущен, следовательно, некому и принять запрос от клиента.

Для запуска сетевых сервисов по требованию используется суперсервер xinetd. Данный сервер всегда находится в памяти и принимает на себя все запросы (кроме запросов, адресованных к автономным службам). Затем он анализирует запрос и запускает необходимую сетевую службу для его обработки. Такая схема позволяет экономить системные ресурсы, потому что не нужно держать в памяти все редко используемые сетевые сервисы.

Суперсервер xinetd имеется в дистрибутивах Fedora, Mandriva и всех их клонах, но вы не найдете его в Ubuntu — там его роль выполняет система инициализации upstart.

П2.2. Конфигурационный файл суперсервера

Конфигурационный файл xinetd называется /etc/xinetd.conf. В современных дистрибутивах этот файл довольно небольшой (листинг П2.1), потому что осталось мало служб, запускаемых с помощью xinetd, — в основном используется автономный запуск.

Листинг П2.1. Пример конфигурационного файла /etc/xinetd.conf.

```
defaults
{
# максимальное число одновременно запущенных экземпляров сервера
        instances               = 60
# параметры протоколирования
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
```

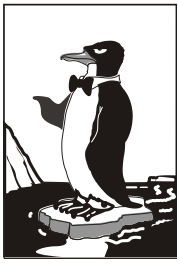
```
        log_on_failure          = HOST

# Параметр cps
# первый аргумент — количество соединений в секунду
# второй аргумент — число секунд, по истечении которых сервис снова будет
доступен
# после превышения первого аргумента cps
    cps                  = 25 30
}

# каталог, содержащий конфигурационные файлы отдельных сетевых сервисов
includedir /etc/xinetd.d
```

В каталоге, заданном директивой `includedir`, содержатся конфигурационные файлы сетевых сервисов. Каждый сервис описывается в собственном файле. Сервис описывается так:

```
service  название
{
параметры
}
```

Вот пример описания сервиса rsync:

```
service rsync
{
# сервис отключен
# чтобы его включить нужно указать disable = no или вообще не указывать
disable
    disable= yes
# тип сокета (stream для TCP, dgram для UDP, raw — для сервисов, требующих
# прямого обращения к протоколу IP)
    socket_type     = stream
# для TCP нужно установить значение no, для UDP — yes
    wait            = no
# пользователь, от имени которого работает сервис
    user            = root
# вызываемый сервер (исполнимый файл сетевой службы)
    server          = /usr/bin/rsync
# аргументы, которые будут переданы серверу (зависит от сервера)
    server_args     = --daemon
# что протоколировать при сбое (USERID — ID пользователя,
# HOST — имя удаленного узла)
    log_on_failure  += USERID
}
```



Приложение 3

Команды Linux

В Linux есть команды, которые нужно знать каждому пользователю Linux. Их мы и рассмотрим в этом приложении. Для большего удобства команды разбиты на группы: общие команды, команды для работы с файлами и каталогами, ссылками, правами доступа, команды для работы с текстом, команды для работы в Интернете и команды системного администратора.

Все представленные здесь команды предназначены для работы в консоли, то есть в текстовом режиме. Понятно, что большинство современных дистрибутивов запускаются в графическом режиме, поэтому некоторые пользователи Linux даже не подозревают о том, что существует консоль. Да, таково новое поколение Linux-пользователей, которым проще использовать графический файловый менеджер, чем вводить команды. Но если вы хотите стать квалифицированным пользователем Linux, то просто обязаны знать, как работать в консоли, иначе уподобитесь Windows-пользователям, которые при каждом сбое переустанавливают операционную систему… Если вы пропустили *главу 8*, в которой рассматривается работа с консолью, настоятельно рекомендую прочитать ее!

П3.1. Общие команды

П3.1.1. Команда *arch* — вывод архитектуры компьютера

Данная команда поможет узнать тип аппаратной платформы, например: i386, i586, i686 и др.

Пример использования:

```
$ arch
i686
```

П3.1.2. Команда *clear* — очистка экрана

Команда `clear` очищает экран при работе в консоли (терминале).

Пример использования:

```
$ clear
```

П3.1.3. Команда *date*

Команда `date` используется для вывода текущей даты. Эта команда может использоваться также для установки даты, если запущена от имени администратора.

Пример использования:

```
$ date
# date 1609171707
```

Первая команда выводит дату, а вторая команда — устанавливает дату (при условии, что команда запущена от имени root) 16 сентября (1609) 2007 года (07) и время 17:17. Как видите, установка даты осуществляется в формате MMddhhmmYY (MM — месяц, dd — число, hh — часы, mm — минуты, YY — год).

Команда `date` может вывести дату в указанном вам формате. Для изучения форматов даты введите команду `man date`.

П3.1.4. Команда *echo*

Команда `echo` выводит текстовую строку, указанную в качестве аргумента, например:

```
$ echo "Hello world!"
Hello world!
```

Обычно данная команда используется в сценариях командного интерпретатора для вывода сообщений на экран.

П3.1.5. Команда *exit* — выход из системы

Для завершения сеанса работы в системе (при условии, что вы работаете в консоли) нужно использовать команду `exit`. Если не завершить сеанс работы, кто угодно сможет работать в системе под вашим именем (понятно, что во время вашего отсутствия за компьютером).

П3.1.6. Команда *man* — вывод справки

Команда `man` используется для получения справки о любой команде системы. Например, команда `man ls` выведет справку об использовании команды `ls`, которая выводит содержимое каталога. О том, как правильно использовать саму справочную систему, вам расскажет команда `man man`.

П3.1.7. Команда *passwd* — изменение пароля

С этой командой мы уже знакомы. Данная команда обеспечивает изменение пароля пользователя, который ее запустил. Суперпользователь root имеет право изменить пароль любого пользователя так:

```
# passwd имя_пользователя
```

П3.1.8. Команда *startx* — запуск графического интерфейса X.Org

Linux может запускаться на разных уровнях запуска. На пятом уровне запуска графический интерфейс X.Org (бывшее название X Window) запускается автомати-

чески, если он был вообще установлен. На третьем же уровне запуск графического интерфейса не производится. Если же вам очень он нужен, то его можно запустить с помощью команды `startx`. Никаких параметров не нужно.

П3.1.9. Команда *uptime* — информация о работе системы

Команда `uptime` (рис. П3.1) выводит статистическую информацию о работе системы: сколько времени прошло с момента последней перезагрузки (собственно, это и есть время "uptime"), сколько пользователей в данный момент подключено к системе и среднюю загрузку системы за последние 1, 5 и 15 минут.

```
[den@localhost ~]$ uptime
 13:02:26 up 6 min,  3 users,  load average: 0.57, 0.81, 0.44
[den@localhost ~]$
```

Рис. П3.1. Команда `uptime`

П3.1.10. Команда *users* — информация о пользователях

Данная команда выводит пользователей, подключенных к системе в данный момент (рис. П3.2).

```
[den@localhost ~]$ users
den den
[den@localhost ~]$
```

Рис. П3.2. Команда `users`

Из рис. П3.2 видно, что пользователь den подключился к системе двумя способами: вошел в консоли и в графическом режиме (или по FTP, ssh, telnet — способы подключения к системе разные).

П3.1.11. Команды *w*, *who* и *whoami* — информация о пользователях

Эти три родственные команды выводят следующую информацию (рис. П3.3):

- команда `w` — список пользователей, подключенных к системе; виртуальный терминал, с которого работает пользователь; время входа в систему для каждого пользователя, статистику использования системы (IDLE — время простоя, JCPU — использование процессора), выполняемые каждым пользователем задачи;
- команда `who` — список пользователей, подключенных к системе; время и дату входа каждого пользователя;
- команда `whoami` — имя пользователя, который ввел команду.

```
[den@localhost ~]$ w
 13:04:08 up 7 min,  3 users,  load average: 0,18, 0,60, 0,41
USER     TTY        LOGIN@   IDLE   JCPU   PCPU WHAT
den      pts/0     12:59    4:54   0.00s  1.14s kded [kdeinit] --new-startup
den      pts/1     13:02    0.00s  0.15s  0.01s w
[den@localhost ~]$ who
den      pts/0        2007-07-23 12:59
den      pts/1        2007-07-23 13:02
[den@localhost ~]$ whoami
den
[den@localhost ~]$
```

Рис. П3.3. Команды `w`, `who` и `whoami`

П3.1.12. Команда *xf86config* — настройка графической подсистемы

Текстовый конфигуратор системы X.Org (она же X Window). Использовать его нужно, только если в вашем дистрибутиве нет более удобных графических или псевдографических конфигураторов.

П3.2. Команды для работы с файлами и каталогами, ссылками, правами доступа

Здесь мы рассмотрим основные команды Linux для работы с файлами, каталогами и ссылками, а также команды, устанавливающие права доступа к файлам и каталогам.

П3.2.1. Работа с файлами

Основные команды для работы с файлами приведены в табл. П3.1.

Таблица П3.1. *Основные команды Linux, предназначенные для работы с файлами*

Команда	Назначение
`touch <файл>`	Создает пустой файл
`cat <файл>`	Просмотр текстового файла
`tac <файл>`	Вывод содержимого текстового файла в обратном порядке, то есть сначала выводится последняя строка, потом предпоследняя и т. д.
`cp <файл1> <файл2>`	Копирует файл `<файл1>` в файл `<файл2>`. Если `<файл2>` существует, программа попросит разрешение на его перезапись
`mv <файл1> <файл2>`	Перемещает файл `<файл1>` в файл `<файл2>`. Эту же команду можно использовать и для переименования файла
`rm <файл>`	Удаляет файл
`locate <файл>`	Производит быстрый поиск файла

Таблица П3.1 *(окончание)*

Команда	Назначение
`which <программа>`	Выводит каталог, в котором находится программа, если она вообще установлена. Поиск производится в каталогах, указанных в переменной окружения `PATH` (это путь поиска программ)
`less <файл>`	Используется для удобного просмотра файла с возможностью скроллинга (постраничной прокрутки)

Рассмотрим небольшую серию команд (протокол выполнения этих команд приведен на рис. П3.4):

```
touch file.txt
echo "some text" > file.txt
cat file.txt
cp file.txt file-copy.txt
cat file-copy.txt
rm file.txt
cat file.txt
mv file-copy.txt file.txt
cat file.txt
```

```
[root@localhost ~]# touch file.txt
[root@localhost ~]# echo "some text" > file.txt
[root@localhost ~]# cat file.txt
some text
[root@localhost ~]# cp file.txt file-copy.txt
[root@localhost ~]# cat file-copy.txt
some text
[root@localhost ~]# rm file.txt
rm: удалить обычный файл `file.txt'? y
[root@localhost ~]# cat file.txt
cat: file.txt: No such file or directory
[root@localhost ~]# mv file-copy.txt file.txt
[root@localhost ~]# cat file.txt
some text
[root@localhost ~]#
```

Рис. П3.4. Операции с файлом

Первая команда (`touch`) создает в текущем каталоге файл file.txt. Вторая команда (`echo`) записывает строку `some text` в этот же файл. Обратите внимание на символ > — это символ перенаправления ввода/вывода, о котором мы говорили в *главе 8.*

Третья команда (`cat`) выводит содержимое файла — в файле записанная нами строка `some text`. Четвертая команда (`cp`) копирует файл file.txt в файл с именем file-copy.txt. После этого мы опять используем команду `cat`, чтобы вывести содержимое файла file-copy.txt, — надо же убедиться, что файл действительно скопировался.

Шестая команда (`rm`) удаляет файл file.txt. При удалении система спрашивает, хотите ли вы удалить файл. Если хотите удалить, то нужно нажать клавишу <Y>, а если нет, то <N>. Точно ли файл удален? Убедимся в этом: введите команду `cat file.txt`. Система нам сообщает, что нет такого файла.

Восьмая команда (`mv`) переименовывает файл file-copy.txt в файл file.txt. Последняя команда выводит исходный файл file.txt.

Думаю, особых проблем с этими командами у вас не возникло, тем более что принцип действия этих команд вам должен быть знаком по командам DOS, которые, как квалифицированный пользователь Windows, вы должны знать наизусть.

Вместо имени файла иногда очень удобно указать *маску имени файла*. Например, у нас есть много временных файлов, имена которых заканчиваются фрагментом tmp. Для их удаления можно воспользоваться командой: `rm *tmp`. Если же требуется удалить все файлы в текущем каталоге, следует просто указать звездочку: `rm *`.

Аналогично, можно использовать символ `?`, который, в отличие от звездочки, заменяющей последовательность символов произвольной длины, заменяет всего один символ. Например, нам нужно удалить все файлы, имена которых состоят из трех букв и начинаются на s:

```
rm s??
```

Будут удалены файлы s14, sqm, sr6 и т. п., но не будут тронуты файлы, имена которых состоят более чем из трех букв и которые не начинаются на s.

Маски имен можно также использовать и при работе с каталогами.

П3.2.2. Работа с каталогами

Основные команды для работы с каталогами приведены в табл. П3.2.

***Таблица П3.2.** Основные команды для работы с каталогами*

Команда	Описание
`mkdir <каталог>`	Создание каталога
`cd <каталог>`	Изменение каталога
`ls <каталог>`	Вывод содержимого каталога
`rmdir <каталог>`	Удаление пустого каталога
`rm -r <каталог>`	Рекурсивное удаление каталога

При указании имени каталога можно использовать следующие символы:

- `.` — означает текущий каталог. Если вы введете команду `cat ./file`, то она выведет файл file, который находится в текущем каталоге;
- `..` — родительский каталог. Например, команда `cd ..` переведет вас на один уровень вверх по дереву файловой системы;
- `~` — домашний каталог пользователя (об этом мы поговорим позже).

Теперь рассмотрим пример работы с каталогами на практике. Выполните следующие команды:

```
mkdir directory
cd directory
```

```
touch file1.txt
touch file2.txt
ls
cd ..
ls directory
rm directory
rmdir directory
rm -r directory
```

Первая команда (`mkdir`) создает каталог directory в текущем каталоге. Вторая команда (`cd`) переводит (изменяет каталог) в только что созданный каталог. Следующие две команды `touch` создают в новом каталоге два файла — file1.txt и file2.txt.

Команда `ls` без указания каталога выводит содержимое текущего каталога. Команда `cd ..` переводит в родительский каталог. Как уже было отмечено, в Linux родительский каталог обозначается так: `..` (две точки), а текущий так: `.` (одна точка). То есть, находясь в каталоге directory, мы можем обращаться к файлам file1.txt и file2.txt без указания каталога или же так: `./file1.txt` и `./file2.txt`.

> ***Внимание!***
>
> Еще раз обратите внимание: в Linux в отличие от Windows для разделения элементов пути используется прямой слэш (`/`), а не обратный (`\`)!

Кроме обозначений `..` и `.` в Linux часто используется обозначение `~` (тильда) — это *домашний каталог*. Предположим, что наш домашний каталог /home/den. В нем мы создали подкаталог dir и поместили в него файл file1.txt. Полный путь к файлу можно записать так:

```
/home/den/dir/file1.txt
```

или же так:

```
~/dir/file1.txt
```

Как видите, тильда (`~`) заменяет часть пути. Удобно? Конечно!

Поскольку мы находимся в родительском для каталога directory каталоге, чтобы вывести содержимое только что созданного каталога в команде `ls`, нам нужно четко указать имя каталога:

```
ls directory
```

Команда `rm` используется для удаления каталога. Но что мы видим — система отказывается удалять каталог! Пробуем удалить его командой `rmdir`, но и тут отказ. Система сообщает нам, что каталог не пустой, то есть содержит файлы. Поэтому для удаления каталога необходимо сначала удалить все имеющиеся в нем файлы и вложенные каталоги. Чтобы выйти из положения, проще указать параметр `-r` команды `rm` для рекурсивного удаления каталога. В этом случае сначала будут удалены все подкаталоги (и все файлы в этих подкаталогах), а затем будет удален и сам каталог (рис. П3.5).

Отметим также команды `cp` (копирования) и `mv` (перемещения/переименования). Работают они сходно — для копирования (перемещения/переименования) сначала указывается каталог-источник, а потом каталог-назначение.

```
[root@localhost ~]# mkdir directory
[root@localhost ~]# cd directory
[root@localhost directory]# touch file.txt
[root@localhost directory]# touch file2.txt
[root@localhost directory]# ls
file2.txt  file.txt
[root@localhost directory]# cd ..
[root@localhost ~]# ls directory
file2.txt  file.txt
[root@localhost ~]# rm directory
rm: невозможно удалить каталог `directory': Is a directory
[root@localhost ~]# rmdir directory
rmdir: `directory': Directory not empty
[root@localhost ~]# rm -r directory
rm: спуститься в каталог `directory'? y
rm: удалить пустой обычный файл `directory/file.txt'? y
rm: удалить пустой обычный файл `directory/file2.txt'? y
rm: удалить Каталог `directory'? y
[root@localhost ~]#
```

Рис. П3.5. Операции с каталогами

П3.2.3. Команда *ln* — создание ссылок, жестких и символических

Для создания ссылок используется команда `ln`:

```
ln file.txt link1
ln -s file.txt link2
```

Первая команда создает жесткую ссылку link1, ссылающуюся на текстовый файл file.txt. Вторая команда создает символическую ссылку link2, которая ссылается на этот же текстовый файл file.txt.

Модифицируя ссылку (все равно какую — link1 или link2), вы автоматически модифицируете исходный файл — file.txt.

Особого внимания заслуживает операция удаления. По идее, если вы удаляете ссылку link2, файл file.txt также должен быть удален, но не тут-то было — вы не можете его удалить до тех пор, пока на него указывает хоть одна жесткая ссылка. При удалении ссылки link2 просто будет удалена символьная ссылка, но жесткая ссылка и сам файл останутся. Если же вы удалите ссылку link1, будет удален и файл file.txt, поскольку на него больше не ссылается ни одна жесткая ссылка.

П3.2.4. Команда *chmod* — права доступа к файлам и каталогам

Права доступа к файлам и каталогам задаются командой `chmod`. Существуют два способа указания прав доступа: *символьный* (когда указываются символы, задающие право доступа, — `r`, `w`, `x`) и *абсолютный*.

Так уж заведено, что в мире UNIX чаще пользуются абсолютным методом. Разберемся, в чем он заключается. Рассмотрим следующий набор прав доступа:

```
rw-r-----
```

Данный набор прав доступа предоставляет владельцу право чтения и модификации файла (`rw-`), запускать данный файл владелец не может. Члены группы владельца могут только просматривать файл (`r--`), а все остальные пользователи не имеют вообще никакого доступа к файлу.

Возьмем отдельный набор прав, например для владельца: `rw-`.

Чтение разрешено — мысленно записываем 1, запись разрешена — запоминаем еще 1, а вот выполнение запрещено, поэтому запоминаем 0. Получается число 110. Если из двоичной системы перевести число 110 в восьмеричную, получится число 6. Для перевода можно воспользоваться табл. П3.3.

Таблица П3.3. *Преобразование чисел из двоичной системы в восьмеричную*

Двоичная система	Восьмеричная система	Двоичная система	Восьмеричная система
000	0	100	4
001	1	101	5
010	2	110	6
011	3	111	7

Аналогично произведем разбор прав для членов группы владельца. Получится двоичное 100, то есть восьмеричное 9. С третьим набором (`---`) все вообще просто — это 000, то есть 0.

Записываем полученные числа в восьмеричной системе в порядке владелец-группа-остальные. Получится число 640 — это и есть права доступа. Для того чтобы установить эти права доступа, выполните команду:

```
chmod 640 <имя_файла>
```

Наиболее популярные права доступа:

- 644 — владельцу можно читать и изменять файл, остальным пользователям — только читать;
- 666 — читать и изменять файл можно всем пользователям;
- 777 — всем можно читать, изменять и выполнять файл.

> ***Примечание***
>
> Напомню, что для каталога право выполнения — это право просмотра оглавления каталога.

Иногда символьный метод оказывается проще. Например, у нас есть файл script, который нужно сделать исполнимым, для этого можно применить команду:

```
chmod +x script
```

Для того чтобы снять право выполнения, указывается параметр `-x`:

```
chmod -x script
```

Подробнее о символьном методе вы сможете прочитать в руководстве по команде `chmod` (выполнив команду `man chmod`).

П3.2.5. Команда *chown* — смена владельца файла

Если вы хотите "подарить" кому-то файл, то есть сделать какого-то пользователя владельцем файла, нужно применить команду `chown`:

```
chown пользователь файл
```

> ***Примечание***
>
> Возможно, что после изменения владельца файла вы сами не сможете получить к нему доступ, ведь владельцем будете уже не вы.

П3.2.6. Команда *chattr* — изменение атрибутов файла, запрет изменения файла

С помощью команды `chattr` можно изменить атрибуты файла. Параметр `+` устанавливает атрибут, а параметр `-` атрибут снимает. Например:

```
# chattr +i /boot/grub/menu.lst
```

Данная команда устанавливает атрибут `i`, запрещающий любое изменение, переименование и удаление файла. Установить этот атрибут, равно как и снять его, имеет право только суперпользователь или процесс с возможностью `CAP_LINUX_IMMUTABLE`. Чтобы изменить файл, нужно очистить атрибут с помощью команды:

```
# chattr -i /boot/grub/menu.lst
```

Если установить атрибут `j`, то все данные, прежде чем они будут записаны непосредственно в файл, будут сохранены в журнале файловой системы ext3/ext4. Использование атрибута `j` имеет смысл, только если файловая система смонтирована с опциями `data=ordered` или `data=writeback` *(см. разд. 9.7)*. Когда файловая система смонтирована с опцией `data=journal`, этот атрибут не имеет значения, поскольку все данные файла и так уже журналируются. Об остальных атрибутах вы сможете прочитать в справочной системе:

```
man chattr
```

П3.2.7. Команда *mkfs* — создание файловой системы

С помощью команды `mkfs` мы можем создать файловую систему на разделе жесткого диска, например: `mkfs.ext2 /dev/hda1`.

Вообще, создать файловую систему нужного типа (если эта файловая система поддерживается ядром вашей системы) можно с помощью команды `mkfs.<имя_файловой_системы>`, например:

```
mkfs.ext3
mkfs.vfat
mkfs.reiserfs
```

Подробнее прочитать об этом можно, введя команду `man mkfs.<имя файловой системы>`.

П3.2.8. Команда *fsck* — проверка и восстановление файловой системы

Для проверки файловой системы используется команда `fsck`. Использовать ее нужно так:

```
fsck <раздел>
```

Например:

```
fsck /dev/sda5
```

Перед использованием этой команды следует размонтировать проверяемую файловую систему. Если требуется проверить корневую файловую систему, то надо загрузиться с LiveCD и запустить `fsck` для проверки нужного раздела.

Если же жесткий диск "посыпался", то есть появились "плохие" блоки, нужно, не дожидаясь полной потери данных, выполнить следующие действия:

- Выполнить команду `fsck -c <раздел>` (данная команда пометит "плохие" блоки).
- Сделать резервную копию всех важных данных.
- Отправиться в магазин за новым жестким диском и перенести данные со старого жесткого диска на новый. Проверить жесткий диск на наличие плохих секторов можно программой `badblocks`.

> ***Примечание***
>
> Программа `fsck` может проверять не только файловые системы ext2/ext3. Для проверки, например, vfat, можно использовать команду `fsck.vfat <раздел>`.

Для восстановления "упавшей" таблицы разделов можно использовать программу `gpart`. Только используйте ее осторожно и внимательно читайте все сообщения, выводимые программой.

П3.2.9. Команда *chroot* — смена корневой файловой системы

Предположим, мы установили Windows после установки Linux, и программа установки Windows перезаписала начальный загрузчик. Теперь Windows загружается, а Linux — нет. Что делать? Нужно загрузиться с LiveCD и выполнить команду:

```
# chroot <раздел, содержащий корневую файловую систему>
```

Например, если Linux был установлен в раздел /dev/sda5, нужно ввести команду:

```
# chroot /dev/sda5
```

Данная команда сменит корневую файловую систему, то есть вы загрузите ядро Linux с LiveCD, а затем сделаете подмену корневой файловой системы. Вам останется только ввести команду записи загрузчика (например, `lilo`) для восстановления начального загрузчика.

П3.2.10. Установка скорости CD/DVD

Программа `hdparm` позволяет ограничить скорость оптического привода (CD-ROM/DVD-ROM). Иногда нужно ограничить скорость, чтобы информация была считана

без ошибок (как правило, если поверхность носителя информации немного повреждена). Рассмотрим команду ограничения скорости:

```
# hdparm -q -E<множитель> <устройство>
```

Множитель — это и есть скорость, например 1× соответствует скорости 150 кбит/с для CD, 1385 кбит/с — для DVD. Чтобы установить вторую (2×, 300 кбит/с) скорость чтения для CD, используется команда:

```
# hdparm -q -E2 /dev/cdrom
```

Для ограничения скорости DVD можно использовать следующую команду:

```
# hdparm -q -E1 /dev/dvd
```

П3.2.11. Монтирование каталога к каталогу

В Linux можно подмонтировать каталог к каталогу, а не только каталог к устройству. Делается это с помощью все той же команды `mount`, запущенной с параметром `--bind`:

```
# mount --bind исходный_каталог каталог_назначения
```

П3.2.12. Команды поиска файлов

Для поиска файлов в Linux используется команда `find`. Это довольно мощная утилита со сложным синтаксисом и далеко не всегда нужная обычному пользователю. Намного проще будет установить файловый менеджер mc и использовать встроенную функцию поиска.

Но команду `find` мы все же рассмотрим, по крайней мере, ее основы. Синтаксис команды следующий:

```
find список_поиска выражение
```

Мощность программы `find` заключается во множестве самых разных параметров поиска, которые из-за этого не так легко запомнить. К тому же `find` может выполнять команды для найденных файлов. Например, вы можете найти временные файлы и сразу удалить их.

Подробно опции команды `find` мы рассматривать не будем — это вы можете сделать самостоятельно с помощью команды `man find`. Зато рассмотрим несколько примеров использования этой команды.

- Найти файлы с именем a.out (точнее, в имени которых содержится строка "a.out"), поиск начать с корневого каталога (/):

  ```
  find / -name a.out
  ```

- Найти файлы по маске *.txt:

  ```
  find / -name '*.txt'
  ```

- Найти файлы нулевого размера, поиск начать с текущего каталога (.):

  ```
  find . -size 0c
  ```

  Впрочем, для поиска пустых файлов намного проще использовать параметр `-empty`:

  ```
  find . -empty
  ```

- Найти файлы, размер которых от 100 до 150 Мбайт, поиск производить в домашнем каталоге и всех его подкаталогах:

```
find ~ -size +100M -size -150M
```

- Найти все временные файлы и удалить их (для каждого найденного файла будет запущена команда `rm`):

```
# find / -name *.tmp -ok rm {} \;
```

Вместо параметра `-ok` здесь можно использовать параметр `-exec`, который также запускает указанную после него команду, но не запрашивает подтверждение выполнения этой команды для каждого файла.

Кроме команды `find` можно использовать команды `which` и `locate`. Первая выводит полный путь к программе или к сценарию, если программа или сценарий находится в списке каталогов, заданном в переменной окружения `PATH`:

```
which sendmail
```

Программа `locate` ищет в базе данных демона located файлы, соответствующие заданному образцу. Недостаток этой команды в том, что located имеется далеко не во всех дистрибутивах, поэтому команды `locate` у вас может и не быть. Зато если located имеется и запущен, поиск файлов будет осуществляться быстрее, чем с помощью `find`.

П3.2.13. Создание файла подкачки

Оперативная память — это очень критичный для Linux ресурс. Даже более критичный, чем частота процессора, поэтому нехватка оперативной памяти очень остро ощущается в Linux. Иногда работать становится просто невыносимо.

При установке Linux создается раздел подкачки, который используется, если системе не хватает оперативной памяти, — на него сгружается не используемая в данный момент информация, а в оперативную память с жесткого диска подгружаются необходимые процессору данные. Ясно, что система с разделом подкачки работает медленнее, чем с модулем оперативной памяти, но все же она работает быстрее и стабильнее, нежели вообще без раздела подкачки.

Если вы пожадничали и при установке Linux создали маленький раздел подкачки, делу можно помочь даже без переразметки жесткого диска. Мы можем создать файл подкачки, который будет использоваться в паре с разделом подкачки.

Сейчас мы создадим файл /swap_file размером 128 Мбайт:

```
# dd if=/dev/zero of=/swap_file bs=1k count=131072
```

Файл /swap_file пока еще нельзя назвать файлом подкачки, поскольку мы его не отформатировали как файл подкачки. Сделаем это:

```
# mkswap /swap_file 131072
```

Теперь осталось активировать только что созданный файл подкачки:

```
# swapon /swap_file
```

Последнюю команду нужно добавить в файл /etc/rc.d/rc.sysinit (или в /etc/rc.local в Debian/Ubuntu) для того, чтобы не вводить ее при каждом запуске системы.

П3.3. Команды для работы с текстом

П3.3.1. Команда *diff* — сравнение файлов

Команда используется для сравнения двух файлов. Формат вызова программы `diff` такой:

```
diff параметры файл1 файл2
```

Результат сравнения выводится так: отличающиеся строки помечаются символами > и <. Строка из первого файла помечается символом <, а строка из второго файла — символом >.

Самые полезные параметры программы `diff` приведены в табл. П3.4.

Таблица П3.4. *Некоторые параметры программы* `diff`

Параметр	Описание
`-b`	Программа будет игнорировать пробельные символы в конце строки
`-B`	Игнорирует пустые строки
`-e`	Используется для создания сценария для редактора ed, который будет использоваться для превращения первого файла во второй
`-w`	Игнорирует пробельные символы
`-y`	Вывод в два столбца
`-r`	Используется для сравнения файлов в подкаталогах. Вместо первого файла указывается первый каталог, вместо второго файла указывается, соответственно, второй каталог

П3.3.2. Команда *grep* — текстовый фильтр

Предположим, что у нас есть файл протокола /var/log/messages и вы хотите вывести все сообщения, связанные с демоном pppd. Понятно, что вручную выделить все нужные сообщения будет довольно трудно. Но с помощью `grep` можно автоматизировать данную задачу:

```
cat /var/log/messages | grep ppp
```

Команда `cat /var/log/messages` передаст содержимое файла /var/log/messages на стандартный ввод программы `grep`, которая, в свою очередь, выделит строки, содержащие строку 'ppp'.

Вообще, просматривать журналы удобнее с помощью команды `tac`, которая выводит строки файла в обратном порядке — ведь сообщения дописываются в конец журнала, следовательно, если выводит строки в обратном порядке, то сначала получим самые новые сообщения, а потом уже все остальные:

```
tac /var/log/messages | grep ppp
```

П3.3.3. Команды *more* и *less* — постраничный вывод

Большой текстовый файл намного удобнее просматривать с помощью программ `less` или `more`. Программа `less` удобнее, чем `more`, если она есть в вашей системе:

```
tac /var/log/messages | grep ppp | less
```

П3.3.4. Команды *head* и *tail* — вывод начала и хвоста файла

Команда `head` выводит первые десять строк файла, а `tail` — последние десять. Вообще количество строк может регулироваться с помощью параметра `-n`.

Пример использования:

```
head -n 10 /var/log/messages
tail -n 15 /var/log/messages
```

П3.3.5. Команда *wc* — подсчет слов в файле

Команда `wc` используется для подсчета слов в текстовом файле для подсчета количества строк (если задан параметр `-l`) и символов (параметр `-c`).

Пример использования:

```
wc /var/log/messages
wc -l /var/log/messages
wc -c /var/log/messages
```

П3.4. Команды для работы с Интернетом

П3.4.1. Команда *ftp* — стандартный FTP-клиент

Для открытия соединения с любым FTP-сервером введите команду:

```
ftp <имя или адрес FTP-сервера>
```

Можно просто ввести команду `ftp`, а в ответ на приглашение

```
ftp>
```

ввести команду:

```
open <имя или адрес FTP-сервера>
```

Лично мне больше нравится первый вариант, поскольку он позволяет сэкономить время. При подключении к серверу вы сможете ввести имя пользователя и пароль:

```
[den@dhsilabs ~]$ ftp
ftp> open ftp.narod.ru
Connected to ftp.narod.ru.
220 ftp.narod.ru (Libra FTP daemon 0.17 20050906)
500 Unrecognized command AUTH
Name (ftp.narod.ru:den): den
```

```
331 Password required
Password:
230 Logged in, proceed
Remote system type is UNIX.
ftp>
```

Подключившись к серверу, вы можете ввести команду `help`, чтобы просмотреть список доступных команд. Для получения справки по той или иной команде введите `help <имя_команды>` (рис. П3.6). Наиболее популярные команды приведены в табл. П3.5.

```
331 Password required
Password:
230 Logged in, proceed
Remote system type is UNIX.
ftp> help
Commands may be abbreviated.  Commands are:

!               cr              mdir            proxy           send
$               delete          mget            sendport        site
account         debug           mkdir           put             size
append          dir             mls             pwd             status
ascii           disconnect      mode            quit            struct
bell            form            modtime         quote           system
binary          get             mput            recv            sunique
bye             glob            newer           reget           tenex
case            hash            nmap            rstatus         trace
ccc             help            nlist           rhelp           type
cd              idle            ntrans          rename          user
cdup            image           open            reset           umask
chmod           lcd             passive         restart         verbose
clear           ls              private         rmdir           ?
close           macdef          prompt          runique
cprotect        mdelete         protect         safe
ftp>
```

Рис. П3.6. Список команд FTP-клиента

Таблица П3.5. *Некоторые команды FTP-клиента*

Команда	**Описание**
`ls`	Вывод содержимого каталога
`get`	Загрузить файл с сервера
`put`	Загрузить файл на сервер
`mget`	Получить несколько файлов с сервера. Допускается использование масок файлов, например, *.rpm
`mput`	Загрузить несколько файлов на сервер
`cd`	Изменить каталог
`mkdir`	Создать каталог
`rmdir`	Удалить пустой каталог
`delete`	Удалить файл

Кроме ftp, в Linux есть и другие текстовые FTP-клиенты, например, NcFTP (**http://www.ncftp.com**), lukemftp (**ftp://ftp.netbsd.org/pub/NetBSD/misc/lukemftp/**), lftp (**http://ftp.yars.free.net/projects/lftp/**) и др. Все эти FTP-клиенты не входят в состав дистрибутива, их нужно устанавливать самостоятельно. Но стоит ли это делать — решать вам. Ведь все они подобны стандартному клиенту ftp и обладают двумя-тремя дополнительными функциями, которые, возможно, вам и не понадобятся. Например, NcFTP умеет докачивать файлы, а lftp — загружать одновременно несколько файлов. В любом случае вы можете изучить документацию по тому или иному FTP-клиенту (ее легко найти в Интернете), а потом решить, стоит ли его использовать или нет.

П3.4.2. Команда *lynx* — текстовый браузер

Если графический режим недоступен (например, на сервере), а по сети побродить хочется, можно использовать текстовый браузер lynx. В некоторых дистрибутивах вместо lynx используются браузеры links и elinks, но суть остается та же — просмотр страниц Интернета в текстовом режиме.

П3.4.3. Команда *mail* — чтение почты и отправка сообщений

Программа `mail` — это простейший клиент для чтения и отправки почты. Позволяет читать только почту, принятую вашей системой. Если же нужно принять почту с других POP3-серверов, тогда нужно использовать другие почтовые клиенты, которые могут работать в консоли, например, `mutt` или `pine`.

Для чтения предназначенных вам сообщений введите команду `mail` без параметров. Если хотите написать кому-то письмо, передайте в качестве параметра электронный адрес этого человека:

```
mail ivanov@firma.ru
```

П3.5. Команды системного администратора

П3.5.1. Команды *free* и *df* — информация о системных ресурсах

Команда `free` выводит информацию об использовании оперативной и виртуальной памяти, а `df` — об использовании дискового пространства. Из рис. П3.7 видно, что в системе установлено всего 384 Мбайт ОЗУ, из них 247 Мбайт занято и 137 Мбайт — свободно. На жестком диске /dev/sda1 всего 2,8 Гбайт дискового пространства, из них свободно — 1,66 Гбайт.

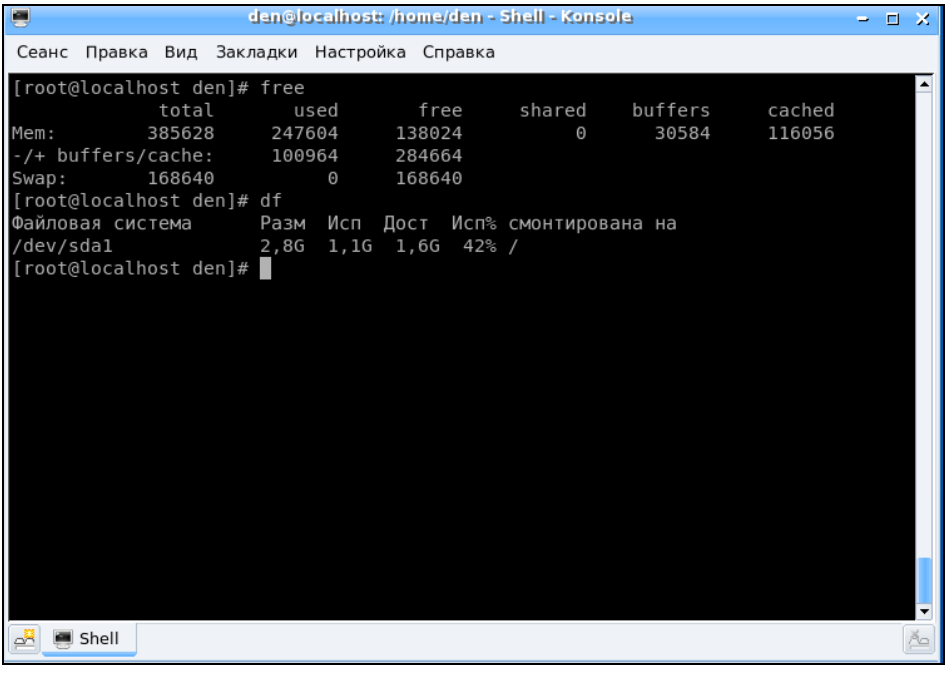


Рис. П3.7. Команды `free` и `df`

П3.5.2. Команда *md5sum* — вычисление контрольного кода MD5

С целью проверки подлинности некоторых файлов, передаваемых через Интернет, используется алгоритм MD5, точнее контрольный код, вычисленный с использованием этого алгоритма. Разработчик программы выкладывает в Интернете пакет со своей программой и на своем сайте публикует контрольный код. Вы скачиваете пакет и вычисляете его контрольный код. Если коды отличаются, то файл при передаче был поврежден (или это другая версия пакета, которая, возможно, была подложена злоумышленником с целью установки вражеского кода в вашу систему).

Использовать программу нужно так:

```
md5sum файл
```

Предметный указатель

(окончание рубрики см. на стр. 520)

М

О

П

Р

С

Т

Ф

(окончание рубрики см. на стр. 522)

Ч, Ш

Э

Я