Кристофер Негус

Ubuntu® и Debian Linux®

для продвинутых

более 1000 незаменимых команд

2-е издание

WILEY

С&ПИТЕР ®

Ubuntu® Linux® TOOLBOX

1000+ Commands for Ubuntu and Debian® Power Users, Second Edition

Christopher Negus

WILEY

Кристофер Негус

Ubuntu® и Debian Linux®

для продвинутых

более 1000 незаменимых команд

2-е издание

СПИТЕР®

Москва - Санкт-Петербург • Нижний Новгород • Воронеж
Ростов-на-Дону Екатеринбург Самара • Новосибирск
Киев • Харьков Минск
2014

ББК 32.973.2-018.2
УДК 004.451
Н41

Негус К.

Н41 Ubuntu и Debian Linux для продвинутых: более 1000 незаменимых команд.
2-е изд. — СПб.: Питер, 2014. — 384 с.: ил.

ISBN 978-5-496-01075-7

С помощью данного руководства вы научитесь использовать Ubuntu Linux так, как это делают настоящие профессионалы, то есть из командной строки. Откройте для себя более 1000 команд для управления программным обеспечением и системными утилитами, отслеживающими работу ПК и его безопасность, а также программами для работы в Сети и разграничения доступа и др. Книга даст незаменимые знания и навыки использования и администрирования настольных ПК и серверов, работающих под управлением Ubuntu, Debian, Linux Mint и других дистрибутивов Linux.

Из книги вы узнаете, как использовать командную оболочку; искать в Сети программное обеспечение; работать с файлами; проигрывать музыку, просматривать иллюстрации; администрировать файловую систему; делать резервное копирование данных; управлять запущенными процессами и потребляемыми ресурсами; получать доступ к сетевым ресурсам; удаленно администрировать систему и др.

12+ (В соответствии с Федеральным законом от 29 декабря 2010 г. № 436-ФЗ.)

ББК 32.973.2-018.2
УДК 004.451

ISBN 978-1118183526 англ.
ISBN 978-5-496-01075-7

Краткое содержание

Оглавление

Как всегда, я посвящаю свою работу над этой книгой моей жене Шери.

Кристофер Негус

Об авторе

Кристофер Негус (Christopher Negus) обладает более чем 25-летним опытом в обучении и написании книг о Linux и UNIX. Он является автором множества книг, посвященных Linux, включая серию бестселлеров «Red Hat Linux. Библия пользователя» *(Red Hat Linux Bible),* «Linux. Библия пользователя» *(Linux Bible),* «Linux-игрушки» *(Linux Toys)* и «Linux для продвинутых» *(Linux Toolbox).*

Будучи сотрудником Red Hat Inc., Кристофер получил сертификаты, включая такие, как «Сертифицированный инженер Red Hat» (Red Hat Certified Engineer, RHCE), «Сертифицированный инструктор Red Hat» (Red Hat Certified Instructor, RHCI), «Сертифицированный экзаменатор Red Flat» (Red Hat Certified Examiner, RHCX). В настоящее время он занимается написанием технических материалов для портала Red Hat Customer Portal, в которых рассматриваются темы, связанные с Red Hat Enterprise Linux, виртуализацией и облачными вычислениями.

К наградам, полученным Кристофером, относится «Лучшая книга года на тему Linux» за его труд *«Red Lint Linux 8. Библия пользователя»* *(Red Hat Linux 8 Bible),* согласно голосованию читателей журнала *Linux World.* Во время конкурса Readers' Choice Awards, устроенного журналом *LinuxJournal* в 2009 году, в результате голосования его книга *«IJnux. Библия пользователя» (Linux Bible)* вошла в первую пятерку в номинации «Лучшие книги о Linux всех времен» (Favorite Linux Books of All Time).

О технических редакторах

Ричард Блум (Richard Blum) работает в индустрии информационных технологий более 25 лет в качестве как системного, так и сетевого администратора. Он опубликовал множество книг на тему Linux и другого программного обеспечения с открытым исходным кодом и является онлайн-инструктором по веб-программированию и Linux-курсам, которые преподаются в колледжах и университетах США. В свободное время, когда Ричард не исполняет роль компьютерного «ботаника», он наслаждается игрой на пианино и гитаре, любит проводить время с женой Барбарой и двумя дочерьми — Кэти Джейн и Джессикой.

Дэвид Даффи (David Duffey) проживает в Остине, штат Техас, со своей женой и двумя детьми и работает в компании Canonical, которая является коммерческим спонсором Ubuntu. Дэвид немного фанатичен и обладает множеством Linux-сертификатов от Canonical, Red Hat, Novell/SuSe и LPI. Дэвид получил степень бакалавра в математике и информатике в Канзасском университете (Kansas State University), а также степень EMBA исполнительного магистра делового администрирования в Школе бизнеса Маккомбса (McCombs School of Business) Техасского университета (University of Texas).

Благодарности

Я хотел бы поблагодарить Canonical Ltd. и сообщество Ubuntu за постоянную и отличную работу над операционной системой Ubuntu на основе Linux.

Трудясь над этим изданием, я задействовал глубокие профессиональные знания Дэвида Даффи — руководителя серверной партнерской программы в Canonical. Его помощь оказалась неоценимой, когда речь зашла об определении объема этого издания. Я также хочу поблагодарить Ричарда Блума (кто сам является отличным Linux-автором) за тщательное техническое редактирование этого издания.

Выражаю особую благодарность Франсуа Казну (Fran3ois Caen) за его труд в качестве соавтора первого издания данной книги. Кроме того, при работе над первым изданием Томас Блэйдер (Thomas Blader) вышел далеко за пределы своих обязанностей как технического редактора, проявив крайнюю проницательность и обеспечив тщательную проверку на протяжении всей книги. Эрик Фостер-Джонсон (Eric Foster-Johnson) подключился почти в конце работы над первым изданием и обеспечил расширенное описание функциональных возможностей Ubuntu во всей книге.

Спасибо моим друзьям и коллегам из Red Hat, благодаря кому ходить на работу каждый день доставляет радость. В частности, выражаю признательность Генри Хаттону (Henry Hutton) за постоянное ободрение с его стороны и замечательные идеи по продвижению моих книг о Linux. Что касается домашнего фронта, благодарю свою чудесную жену Шери (Sheree) и прекрасного сына Сета (Seth) за то, что я с радостью прихожу и домой тоже.

Я хотел бы выразить благодарность Мэри Джеймс (Mary James) из Wiley за помощь в определении моего видения этой книги, а также Морин Спирс (Maureen Spears) за ее тактичное побуждение меня к тому, чтобы я строго придерживался графика, и ее разработку и редактирование книги твердой рукой. Спасибо Дэниелу Скрибнеру (Daniel Scribner) за управление на этапах производства. Детальное редактирование рукописи, выполненное Нэнси Рапопорт (Nancy Rapoport), придало книге лоск, добиться которого самостоятельно я даже не мог надеяться.

Кристофер Негус

Введение

Огромное, полное энтузиазма сообщество Ubuntu пополнилось тысячами новых пользователей Ubuntu Linux. Если вы являетесь одним из них, то, вероятно, скоро почувствуете, что вам хочется углубиться под «поверхность» приложений и графических инструментов Ubuntu Linux. У вас возникнет желание стать продвинутым пользователем.

Превращение в такового означает приобретение умения работать из командной строки. Немногие интерфейсы обеспечат параметры и гибкость, доступные вам при использовании команд, которые направлены на решение аналогичных задач.

В этой книге приведено более 1000 специфических строк команд, чтобы помочь вам лучше разобраться в Linux. Независимо от того, являетесь вы системным администратором или обычным пользователем настольного компьютера, в этом издании вы найдете команды для создания файловых систем, устранения неполадок в сетях, повышения безопасности, а также почерпнете из него почти все, что вам нужно знать о Linux-системе.

Во время вашего путешествия с использованием командной строки Linux внимание в этой книге будет сосредоточено на Ubuntu — дистрибутиве Linux, разрабатываемом сообществом и спонсируемом компанией Canonical Ltd., а также системе Debian GNU/Linux, которая лежит в основе Ubuntu. Приобретение навыков, необходимых для управления этими системами, поможет вам в работе с вашими собственными Linux-системами, а также позволит узнать то, что вам потребуется сделать как Linux-профессионалу.

Ubuntu штурмует Linux

Со времени своего первого выпуска в 2004 году Ubuntu (www.ubuntu.com) стал самым популярным и, возможно, самым любимым пользователями дистрибутивом Linux. Начиная с названия, которое переводится как *«человечность по отношению к другим»,* и заканчивая сосредоточенностью на поддержке многих языков и особых требований, Ubuntu отражает собственные идеалы, выражающиеся в распространении свободного программного обеспечения за пределами стандартных целевых рынков Linux, основу которых составляют фанаты и корпоративные серверы.

Участники проекта Ubuntu делают все возможное, чтобы помочь новым пользователям в эксплуатации их операционной системы Ubuntu на основе Linux. Ubuntu Live CD позволяют новым пользователям ознакомиться с системой Ubuntu, прежде чем устанавливать ее. Если пользователю понравится Ubuntu, он сможет запустить ее установку на жесткий диск одним щелчком кнопкой мыши, а благо-

даря тому, что система Ubuntu основана на Debian GNU/Linux, для пользователей Ubuntu доступно огромное количество бесплатных приложений из репозиториев программного обеспечения Debian.

Внимание разработчиков Ubuntu сосредоточено на обеспечении простой в использовании настольной системы, однако это не означает, что Ubuntu не имеет коммерческой ценности Linux. На самом деле компания Canonical предлагает платную поддержку корпоративного уровня посредством своей инициативы Landscape **(www.ubuntu.com/management).** Canonical также предоставляет целый спектр платных и бесплатных вариантов поддержки для физических лиц и компаний малого бизнеса **(www.ubuntu.com/support).** Другими словами, перед теми, кто учится работать в Ubuntu, открываются возможности стать профессионалами.

Целевая аудитория этой книги

Это издание предназначено для всех, кто желает освоить мощную функциональность Linux-системы как системный администратор или обычный пользователь. Вы можете быть Linux-энтузиастом, Linux-профессионалом или, возможно, профессионалом в области вычислительной техники, кто все больше и больше осознает, что Windows-системы в его центре обработки данных вытесняются системами Linux.

Главное заключается в том, что вы хотите найти быстрые и эффективные способы сделать так, чтобы Ubuntu и другие системы на основе Debian работали с максимальной производительностью. Этими системами может быть несколько настольных систем в офисе, файл- или принт-сервер в школе или веб-сервер дома, который нужен вам просто для развлечения.

В лучшем случае у вас уже имеется некоторый опыт работы в Linux. Однако если вы являетесь профессионалом в области вычислительной техники с навыками управления операционными системами других типов, например Windows, то сможете легко приспособить свои знания, чтобы научиться использовать специфические команды, которые я описываю в книге.

Что рассматривается в этой книге

Это не книга о Linux для начинающих. Прежде чем приступать к ее чтению, вам лучше всего обзавестись базовыми практическими знаниями о том, что такое Linux, как работает интерпретатор команд, а также что представляют собой процессы, файловые системы и сетевые интерфейсы. Затем данная книга дополнит эти знания информацией, необходимой вам, чтобы выполнять следующие действия.

- **Получать программное обеспечение** — Ubuntu предлагает GUI-инструмент Ubuntu Software Center (Центр приложений Ubuntu) для получения программного обеспечения. Используя такие инструменты, как atp-get, вы освоите лучшие способы поиска, загрузки, установки, обновления и вообще управления программным обеспечением из командной строки.

- **Использовать интерпретатор команд** — найдите четкое описание методик и советы по использованию интерпретатора команд.
- **Взаимодействовать с мультимедиа** — проигрывайте и осуществляйте потоковую передачу мультимедийного содержимого с вашего компьютера. Вы также сможете модифицировать аудио- и графические файлы, а затем конвертировать их содержимое в другие форматы. Что касается видео, то вы получите возможность проигрывать файлы различных форматов, включая коммерческие фильмы.
- **Работать с файлами** — используйте, манипулируйте, конвертируйте и обеспечивайте защиту файлов всевозможных типов в Linux.
- **Администрировать файловые системы** — осуществляйте доступ, форматируйте, разбивайте на разделы и выполняйте мониторинг своих аппаратных файловых запоминающих устройств (жестких дисков, CD/DVD-приводов, дискет, флеш-дисков USB и т. д.). Затем создавайте, форматируйте и проверяйте файловые системы, имеющиеся на этих аппаратных устройствах. Вы даже сможете создавать зашифрованные файловые системы для защиты своих данных.
- **Осуществлять резервное копирование и восстановление данных** — используйте простые команды для сбора, архивации и упаковки своих файлов в практичные резервные архивы. Затем сохраняйте эти архивы локально или на удаленных компьютерах.
- **Работать с процессами** — выводите сведения о запущенных процессах, применяя разные подходы, например, исходя из уровня использования центрального процессора и процессорных ресурсов или согласно идентификатору процесса. Затем вносите изменения в рабочие процессы, чтобы они выполнялись в фоновом или приоритетном режиме. Отправляйте сигналы процессам, чтобы те производили повторное чтение конфигурационных файлов, останавливайте и возобновляйте обработку либо полностью останавливайте (отменяйте) ее.
- **Управлять системой** — вводите команды для проверки системных ресурсов, например, чтобы получить информацию об уровне использования оперативной памяти, уровне выполнения, загрузчиках и модулях ядра.
- **Осуществлять мониторинг сетей** — устанавливайте и разрывайте проводные и беспроводные сетевые подключения. Проверяйте маршрутизацию, DNS-систему и сведения о хостах. Следите за сетевым трафиком.
- **Получать доступ к сетевым ресурсам** — подключайтесь к удаленным файловым системам Linux и Windows с использованием FTP, NFS или Samba-инструментов. Используйте команды с применением интерпретатора команд для работы в Интернете.
- **Осуществлять удаленное администрирование** — выполняйте доступ и администрирование других компьютеров с использованием удаленного входа в систему (ssh, telnet и т. д.) и экрана. Узнайте об интерфейсах удаленного администрирования, таких как Webmin, SWAT и CUPS.
- **Повышать безопасность** — конфигурируйте брандмауэры и системное протоколирование, чтобы обезопасить свои Linux-системы.

- **Конфигурировать хост виртуализации** — сконфигурируйте свою систему Ubuntu как хост-систему KVM, а затем установите и управляйте виртуальными машинами на этом хосте.
- **Получать справочную информацию** — используйте приложения, приведенные в конце книги, для получения дополнительных сведений об интерпретаторе команд (например, касаемо метасимволов и переменных интерпретатора команд) и состоянии системы (посредством /proc).

Хочется надеяться, что если я все сделан правильно, то для поиска строк команд или GUI-инструментов будет проще использовать эту книгу, нежели Google.

Освоив многое из описанного в этой книге, вы добьетесь следующих преимуществ.

- **Сотни команд** — благодаря тому что в небольшом издании, которое вы сможете легко взять с собой, сжат большой объем информации, у вас будет доступ к сотням полезных команд в более чем 1000 командных строк.
- **Критически важная Linux-информация** — в книге приведены ссылки на самые важные сведения в Интернете, позволяющие преуспеть в освоении Linux в целом и Ubuntu в частности.
- **Переносимые знания** — большинство команд и параметров, используемых вами в Ubuntu, будет точно так же работать и в других Linux-системах, основанных на Debian. С другой стороны, разные дистрибутивы Linux предлагают разные графические инструменты администрирования, и даже в определенном дистрибутиве графические инструменты изменяются чаще, чем команды.
- **Быстрое решение проблем** — к моменту, когда другие лишь включат компьютер и запустят графический инструмент администрирования, вы уже успеете ввести полдюжины команд и решить проблему.
- **Непреходящая ценность** — многие команды, описанные в этой книге, использовались в ранних UNIX-системах. Таким образом, вы получите в распоряжение инструменты, которые отражают опыт тысяч специалистов по вычислительной технике более чем за 40 лет.

Полная документация, касающаяся команд, используемых в Linux, состоит из тысяч MAN-страниц, информационных текстовых данных и справочных сообщений, у вас наверняка время от времени будет возникать желание обратиться к источникам за пределами этой книги. К счастью, LIbuntu и другие Linux-системы включают полезную информацию, которая уже в них встроена. Из гл. 1 вы узнаете, как получить доступ к этим сведениям, которые, вероятно, изначально есть в вашей системе Ubuntu.

Как структурирована эта книга

Это издание не является просто справочником (в котором элементы располагаются в алфавитном порядке) или руководством (с пошаговыми инструкциями по выполнению задач). Вместо этого книга разделена на темы и нацелена на то, чтобы

включать столько полезных команд и параметров, сколько мне удалось уместить в ней.

В начале гл. 1 вы получите общее представление о том, что такое система Ubuntu и как она связана с другими Linux-системами вроде различных дистрибутивов на основе Debian. Далее описываются обширные ресурсы, доступные, чтобы подкрепить ваши знания, полученные посредством этой книги (например, MAN-страницы, информационный материал и т. п.). В гл. 2 приведен краткий обзор процесса установки, за которым следует описание полезных команд, например `apt-get`, для получения и управления программным обеспечением для Ubuntu.

Команды, которые обычный пользователь может найти полезными в Linux, представлены в гл. 3, 4, 5 и 6. В гл. 3 приводится описание инструментов для использования интерпретатора команд, в гл. 4 рассматриваются команды для работы с файлами, а в гл. 5 рассказывается о манипуляциях с текстом. Из гл. 6 вы узнаете, как работать с музыкальными, графическими и видеофайлами.

Начиная с гл. 7, вы приступите к изучению тем, связанных с системным администрированием. В этой главе рассматриваются создание и проверка файловых систем, а в гл. 8 описываются команды для выполнения резервного копирования данных. Из гл. 9 вы узнаете, как манипулировать запущенными процессами, а в гл. 10 найдете описание административных инструментов для управления базовыми компонентами и аспектами, такими как, например, аппаратные модули, уровень использования центрального процессора и уровень использования оперативной памяти.

Глава 11 посвящена управлению сетевыми ресурсами и содержит описание того, как конфигурировать и работать с проводными и беспроводными сетевыми интерфейсами. В гл. 12 рассматриваются текстовые команды для поиска и просмотра информации в Интернете, передачи файлов, совместного использования файлов, общения в чатах и работы с электронной почтой. Описание инструментов для удаленного системного администрирования включено в гл. 13.

Из гл. 14 вы узнаете, как повышать безопасность с использованием таких средств, как брандмауэры и протоколирование. В гл. 15 описывается, каким образом вы можете сконфигурировать Ubuntu как хост виртуализации KVM, а затем использовать GUI-инструменты, а также инструменты командной строки для установки виртуальных машин и управления ими со своего хоста KVM Ubuntu.

Далее идут три приложения, содержащие справочную информацию о редактировании текста, параметрах командного процессора (метасимволах и переменных) и системных настройках (из файловой системы /proc).

Что вам понадобится при изучении этой книги

Я надеюсь, что вы оцените красоту моей прозы, однако эта книга не предназначена для того, чтобы вы, взяв ее и бокал вина, свернулись в клубок у камина. Я ожидаю, что вы сядете перед монитором компьютера и попытаетесь подключиться к сети, устранить неполадки в системе или добавить пользователя. Вино — по вашему желанию.

Другими словами, эта книга призвана быть вашим компаньоном в процессе освоения операционной системы Ubuntu или Debian. Если у вас еще не установлена Ubuntu или Debian, загляните в гл. 2, чтобы узнать, как получить и установить любую из этих систем.

Все команды, приведенные в книге, были протестированы в Ubuntu на архитектурах x86 и x86_64. Однако поскольку многие из этих команд существуют уже долгое время (некоторые появились более 30 лет назад, на раннем этапе существования UNIX), большинство этих команд будет работать в Debian-системах именно так, как описано в этой книге, независимо от архитектуры центрального процессора.

Многие из приведенных далее команд также будут работать в других Linux- и UNIX-системах. Внимание в этой книге сосредоточено на Ubuntu, поэтому описания будут более всего отличаться от имеющих место в случае с другими Linux-системами в плане управления пакетами, установки, процесса загрузки и GUI-инструментов администрирования.

Чтобы книга оставалась актуальной продолжительное время, я сосредоточился на последнем выпуске Ubuntu с пометкой Long Term Support (LTS) (долгосрочная поддержка): Ubuntu 12.04. Заявлено, что поддержка этого выпуска будет обеспечиваться до 2019 года.

Условные обозначения

Чтобы помочь вам изучать книгу максимально эффективно и успевать следить за происходящим, я использовал на всем ее протяжении некоторые условные обозначения. В частности, я придумал стили выделения команд, которые позволили мне уместить максимально возможное количество командных строк.

В примерах команд данные, выводимые компьютером (приглашения и сообщения интерпретатора команд), представлены в виде текста, оформленного одним обычным шрифтом, а данные, вводимые в компьютер (то, что вы печатаете), представлены в виде текста, выделенного жирным шрифтом. Кроме того, краткие описания (при наличии таковых) оформлены курсивом. Вот пример:

```
$ Is *jpg                    Показать все JPEG-файлы в текущем каталоге
hat.jpg
dog.jpg
...
```

В целях экономии места выводимые данные иногда сокращаются (или вообще не приводятся). Многоточия (...) используются как индикатор того, что дополнительные выводимые данные были обрезаны. Если команда очень длинная, то в конце каждой строки будет стоять обратный слеш, сообщающих!, что вводимые данные продолжаются на следующей строке. Пример:

```
oggenc NewSong.wav -o NewSong.ogg \
   -a Bernstein -G Classical          \
   -d 06/15/1972 -t "Simple Song" \
   -1 "Bernsteins Mass"               \
   -c info="From Kennedy Center"
```

Как показано в этом примере, вы можете точно так же напечатать обратные слеши, чтобы все соответствующие данные были включены в одну команду, либо просто разместить все данные в одной строке, не используя обратные слеши.

Обычный пользователь может вводить большинство команд в Ubuntu, однако для выполнения некоторых из них ему потребуется обладать привилегиями суперпользователя. Установка Ubuntu осуществляется без пароля суперпользователя, поэтому предполагается, что вы будете использовать команду sudo во время сеанса пользователя Ubuntu для выполнения административных команд. Вот пример:

```
chris@hostl:/tmp$ sudo useradd -m joe
```

Для ясности и с целью экономии места я, как правило, показываю приглашение для обычного пользователя в виде простого знака доллара ($):

```
$                    Обозначает приглашение для обычного пользователя
```

Время от времени вы будете встречать приглашение в виде символа решетки (#), обозначающего, что для выполнения определенной команды вам, вероятно, потребуются привилегии суперпользователя. Таким образом, если вы увидите приглашение в виде символа #, то сможете либо ввести команду sudo в начале командной строки, либо получить привилегии суперпользователя посредством одного из способов, описанных в гл. 3.

Примечания и предупреждения выглядят следующим образом.

ПРИМЕЧАНИЕ----------

Предупреждения, примечания и советы оформлены так.

Что касается стилей текста, то:

- новые термины и важные слова я *выделяю курсивом*, когда они встречаются впервые;
- я привожу клавиатурные комбинации как Ctrl+A;
- имена файлов, URL-адреса и код в тексте я оформляю следующим образом: `persistence.properties`.

Последний вариант выделения — текст, который является описанием, для чего предназначена приведенная далее команда. Например, я могу написать что-то вроде «используйте следующую команду для **отображения содержимого файла».** Выделение описаний таким образом призвано обеспечить быстрые визуальные подсказки, чтобы вы смогли без труда найти на странице команду, точно зная, что она там есть.

От издательства

Ваши замечания, предложения и вопросы отправляйте по адресу электронной почты **vinitski@minsk.piter.com** (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На сайте издательства **http://www.piter.com** вы найдете подробную информацию о наших книгах.

1. Знакомство с Ubuntu

В этой главе:

- введение в Ubuntu Linux;
- поиск Ubuntu-ресурсов;
- изучение быстрых и мощных команд;
- ссылки на полезные утилиты;
- работа в стиле Linux-гуру.

Независимо от того, используете вы Ubuntu Linux на работе каждый день или прибегаете к этой системе лишь изредка, книга, в которой описываются эффективные и подробные способы сопровождения, мониторинга, обеспечения безопасности и расширения Ubuntu, может оказаться бесценным ресурсом.

Книга «Ubuntu и Debian Linux для продвинутых: более 1000 незаменимых команд. Второе издание» как раз им и является.

Эта книга ориентирована прежде всего на продвинутых пользователей и системных администраторов. Я покажу, как быстро найти и установить программное обеспечение для Ubuntu, а также как обновлять, сопровождать и осуществлять мониторинг «здоровья» и безопасности вашей системы. Короче говоря, я продемонстрирую вам самые эффективные способы использования Ubuntu путем применения некоторых мощных инструментов, имеющихся в вашем распоряжении.

Моя цель состоит в том, чтобы уместить как можно больше полезной информации в небольшом издании, которое вы сможете легко взять с собой. Для этого я описываю следующее.

- **Команды** — множество примеров строк команд демонстрирует верные и полезные способы оперирования командной строкой, которая зачастую кажется устрашающей.
- **GUI-инструменты** — краткие советы по использованию инструментов с графическим интерфейсом для администрирования и конфигурирования вашей системы Ubuntu.
- **Репозитории программного обеспечения** — методы загрузки и установки программного обеспечения, специально созданного для вашей Ubuntu.
- **Онлайн-ресурсы** — где можно найти полезную справочную информацию касаемо Ubuntu, например списки рассылки, на которые можно подписаться, IRC-каналы и другие онлайн-ресурсы.

- **Локальная документация** — инструменты для работы с MAN-страницами, стандартными справочниками Linux и UNIX, а также специфической документацией по устанавливаемому вами программному обеспечению.

Эта книга предназначена для людей, уже знакомых с Linux, поэтому в ней не будет многочисленных скриншотов значков и меню. Вместо этого вы найдете в ней самый короткий путь к тому, чтобы научиться максимально использовать возможности своей системы Ubuntu. Это в первую очередь означает раскрытие тайн командной строки, что позволит вам делать вещи, о которых при использовании Рабочего стола вы могли только мечтать.

То, что вы узнаете из этой книги, поможет вам стать более опытным пользователем систем Ubuntu и Debian, а также Linux в целом. Если это кажется вам заманчивым, пожалуйста, читайте дальше.

Ubuntu, Debian и Linux

Ubuntu — это операционная система на основе Debian GNU/Linux (www.debian.org). Debian существует с начала 1990-х годов и в силу зрелости считается лидирующим дистрибутивом в плане стабильности и безопасности. Система Debian также известна своим строгим следованием принципам свободного программного обеспечения (www.debian.org/intro/free). На ее основе и была создана система Ubuntu.

Система Debian дала начало не только Ubuntu, но и многим другим дистрибутивам Linux (www.debian.org/misc/children-distros). Одни из них происходят непосредственно от Debian, другие являются производными Ubuntu:

- **Xubuntu** — базирующаяся на Xfce настольная система на основе Ubuntu;
- **Kubuntu** — основанная на KDE настольная система на основе Ubuntu;
- **Edubuntu** — производная от Ubuntu, ориентированная на учебные заведения;
- **Linux Mint** — простая в использовании настольная система с корнями Ubuntu и Debian;
- **Knoppix** — система Live CD с рабочим столом KDE, основанная на Debian;
- **Kanotix** — Live CD на основе Debian;
- **Damn Small Linux** — Live CD очень маленького размера (50 Мбайт) на основе Knoppix;
- **Mepis** — настольная система Live CD на основе Ubuntu и Debian.

Xubuntu, Kubuntu и Edubuntu являются, по сути, одним и тем же дистрибутивом Ubuntu на основе Debian. Единственное их различие заключается в рабочем столе по умолчанию, который имеет место в случае с каждым из них, и коллекции приложений, поставляемых с ними. Например, в Kubuntu имеется рабочий стол KDE и менеджер пакетов Adept, которые не устанавливаются в Ubuntu по умолчанию. Дистрибутив Edubuntu ориентирован на работу с учебными приложениями, многие из которых не устанавливаются по умолчанию в других дистрибутивах Ubuntu.

Debian и Ubuntu являются системами с открытым исходным кодом, большинство частей которых создано по универсальной общедоступной лицензии GNU General Public License (www.gnu.org/copyleft/gpl-html), поэтому любой желающий может взять исходный код, распространяемый по лицензии GPL, или произвольный фрагмент GPL-системы и модифицировать его, разобрать на части, использовать как основу, расширить, встроить, переделать, а также свободно распространять соответствующие изменения или модификации. Как правило, единственное требование заключается в том, чтобы вы соблюдали условия лицензии GPL, согласно которой любые изменения, вносимые вами в существующее программное обеспечение, распространяемое по лицензии GPL, должны быть доступны другим для использования таким же образом (см. www.debian.org/social_contract, чтобы узнать об остальных лицензиях, которые признает Debian).

В конечном счете вы получаете не только превосходную систему с бесплатной онлайн-базой всемирной поддержки, но и постоянно развивающийся продукт, движущей силой которого являются люди, увлеченные своим делом. Многие другие дистрибутивы Linux обладают такими же преимуществами; однако система Ubuntu, несомненно, вырвалась вперед по популярности среди Linux-пользователей настольных компьютеров и тех, кто являются новичками.

Понятие выпусков Ubuntu

Приблизительно каждые шесть месяцев появляется новый выпуск Ubuntu. Вы можете выбрать выпуск для использования, посетив страницу Ubuntu Releases (http://releases.ubuntu.com).

Короткие циклы выпусков позволяют Ubuntu предлагать самое новое программное обеспечение с открытым исходным кодом. Однако недостатком этих коротких циклов является то, что многие субъекты бизнеса предпочитают стабильность новейшим прибамбасам. Вот почему для выполнения бизнес-приложений обычно используется дистрибутив Red Hat Enterprise Linux, в случае с которым циклы выпусков, предоставляющих дополнительную функциональность, намного более длительны.

Чтобы решить эту проблему, компания Canonical начала предлагать выпуски Ubuntu с пометкой Long Term Support (LTS) (долгосрочная поддержка). Работая над LTS-выпусками, Canonical прилагает дополнительные усилия с целью обеспечения их стабильности и предлагает более длительные циклы поддержки. Описание циклов поддержки, предоставляемой в случае с LTS-выпусками, можно найти на странице LTS в Ubuntu Wiki (https://wiki.ubuntu.com/LTS).

Чтобы эта книга оставалась актуальной настолько долго, насколько это возможно, внимание в ней сосредоточено на выпуске Ubuntu 12.04 LTS (Precise Pangolin). Для использования той же версии программного обеспечения, которая применяется в этой книге, зайдите на страницу загрузки Precise Pangolin (http://releases.ubuntu.com/precise/) и выберите ISO-образ серверного установочного носителя, подходящий для вашего компьютера (x86- или 64-битную версию).

Если у вас нет точно такой же версии Ubuntu, не беспокойтесь. На большинство приведенных в книге команд вы можете положиться, не опасаясь, что со временем они значительно изменятся. Они сформируют фундамент для использования командной строки, который поможет вам быстрее освоить новую функциональность по мере появления выпусков, которые ее включают.

ПРИМЕЧАНИЕ

У вас когда-нибудь возникал вопрос, откуда взялись такие странные соглашения об именовании Ubuntu (Edgy Eft) и Debian (Woody)? Ответ на него можно найти по адресу https://wiki.ubuntu.com/DevelopmentCodeNames или www.debian.org/doc/manuals/project-history/ch-releases.en.html.

Ubuntu в сравнении с другими дистрибутивами Linux

Если вы войдете в командную строку в системе Ubuntu, Red Hat Enterprise Linux или Fedora, то увидите, что они мало чем отличаются. У этих систем есть общие каталоги и утилиты и функциональность, по большому счету, одинаковая. Так что же отличает Ubuntu от других дистрибутивов Linux? Взгляните на следующее.

- **Рабочий стол Ubuntu Unity** — выбрав свой курс, Ubuntu предлагает собственный рабочий стол Unity вместо GNOME, KDE и других общих интерфейсов рабочего стола Linux. Хотя рабочий стол Unity основан на GNOME, он стремится упростить интерфейс пользователя, чтобы сделать его удобнее на экранах меньшего размера, например нетбуках.

  Основной план Unity — изменить базовую систему управления отображением данных с системы X Window System (используемой в большинстве настольных Linux- и UNIX-систем) на проект Wayland, основанный на OpenGL. Используя Way land, Ubuntu стремится улучшить качество обеспечиваемого пользовательского взаимодействия посредством более сглаженной графики и эффектов. При этом популярные приложения на основе X Window System выполнялись бы в режиме совместимости.

- **Мобильные и развлекательные устройства** — в то время как Ubuntu вторгается в сферу корпоративных вычислений, эта система совершила более естественный переход с настольных устройств на специализированные. Canonical Group (www.canonical.com), которая управляет проектом Ubuntu, анонсировав Ubuntu Phone (www.ubuntu.com/devices/phone). Есть также проект Ubuntu TV (www.ubuntu.com/devices/tv).

- **Упрощенная установка** — сложность загрузки и установки Ubuntu была сведена к нескольким щелчкам кнопкой мыши, а многие решения во время установки принимаются системой автоматически, исходя из того, что может потребоваться или захотеться среднестатистическому пользователю. Упрощенный процесс установки позволил людям эффективно воспользоваться стабильностью пакетов Debian без необходимости принимать сложные решения о разбивке диска на разделы и выборе пакетов.

- **Управление программным обеспечением** — еще одно различие между дистрибутивами Linux заключается в инструментах управления программным обеспечением. Цель соответствующих утилит и систем управления пакетами у Debian и других дистрибутивов Linux одна и та же; вместе с тем действия с ними и их реализация значительно отличаются. Ubuntu и большинство других систем на основе Debian задействуют семейство утилит APT (Advanced Package Tool) для управления программным обеспечением. Вы можете использовать APT для установки, удаления, выполнения запросов и обновления пакетов Debian (DEB). В Red Hat используется система RPM для решения тех же задач для пакетов RPM.
- **Облачные вычисления** — компания Canonical предпринимает активные действия в сфере облачных вычислений. Вместо того чтобы задействовать для работы Ubuntu локальный компьютер, вы можете создать экземпляр Ubuntu в облаке Amazon с использованием Cloudlnit (https://help.ubuntu.com/connmunity/Cloudlnit). Что касается формирования вашей собственной облачной инфраструктуры для других дистрибутивов Linux, то Canonical поддерживает проект OpenStack (www.openstack.org).
- **Администрирование с использованием sudo** — одной из уникальных характеристик системы Ubuntu является намеренная практика блокирования учетной записи суперпользователя и применение взамен sudo (www.gratisoft.us/sudo/intro.html), что позволяет вам выполнять команды с правами доступа суперпользователя для решения задач системного администрирования (подробности о команде sudo вы найдете в гл. 3).

Вход в Linux-систему с правами суперпользователя обеспечивает неограниченный доступ почти ко всем ее компонентам. Удаление всей файловой системы было бы простейшей задачей при наличии прав доступа суперпользователя, поэтому Ubuntu старается ограничить использование этой учетной записи, позволяя задействовать ее, только когда это целесообразно. Большинство дистрибутивов Linux требует, чтобы пользователь входил в систему с правами доступа суперпользователя или выполнял команду su для получения таких прав, чтобы иметь возможность выполнять задачи администрирования; вместе с тем пользователь делает это в системе Ubuntu посредством sudo, используя его или ее пароль для входа в систему, а не отдельный пароль суперпользователя. Каждое выполнение команды sudo регистрируется, что облегчает отслеживание того, кто именно внес изменения (а не только позволяет узнать, что кто-то, введя пароль суперпользователя, сделал это).

Ubuntu обладает особенностями, которые имеют преимущества и недостатки, однако они не ограничивают систему. Ubuntu включает инструменты, позволяющие настраивать, модифицировать, экспериментировать и разбираться в тонкостях сколько вашей душе угодно. Иными словами, идея заключается в том, чтобы у вас была простая в сопровождении, безопасная система с понятным и компактным набором приложений, которая не должна быть ни ограничивающей, ни перегруженной. Все это делает Ubuntu очень гибкой, поэтому вы сможете сразу окунуться в нее и очень быстро ее освоить.

Поиск ресурсов Ubuntu

Сообщество Ubuntu обладает доступным в форме онлайн-ресурсов огромным объемом знаний, которые вы можете почерпнуть. Далее приведен список ссылок на самые популярные и полезные сайты.

- http://ubuntuforums.org — на этом веб-форуме, который поддерживает поиск и представляет собой модерируемую социальную сеть, присутствует разнообразное и талантливое сообщество Ubuntu-пользователей и персонал службы поддержки. Здесь люди делятся друг с другом своими успехами и неудачами, а также предлагают помощь и рекомендации. Если у вас возникнут трудности с чем-либо в Ubuntu, то высока вероятность, что кто-то уже сталкивался с такой же проблемой и нашел ее решение.
- www.ubuntu.com/support — этот сайт предлагает платную поддержку от Canonical Ltd. — компании, которая стоит за Ubuntu. Если вы не хотите тратить время на поиски по форумам или ожидание ответов, знайте, что Canonical Ltd. предлагает консультации по телефону, электронной почте, а также веб-поддержку примерно за $20 в месяц. Доступно также обучение работе в Ubuntu, нацеленное на компании и корпоративных пользователей.
- https://help.ubuntu.com — на этом сайте находится официальная актуальная онлайн-документация по каждому выпуску Ubuntu. По мере выхода свежих выпусков Ubuntu вы можете заходить сюда, чтобы узнать, что нового в них появилось.
- http://screencasts.ubuntu.com — просматривайте записанные сеансы рабочих столов, чтобы узнать, как выполнять в Ubuntu различные операции — от настройки принтера до конфигурирования совместного использования файлов посредством Samba и установки обновлений для поддержания вашей системы Ubuntu в наилучшей форме. Пользователей Ubuntu призывают присоединиться к команде Ubuntu Screencasts Launchpad (https://launchpad.net/~ubuntu-screencasts), чтобы внести свой вклад.
- https://lists.ubuntu.com/mailman/listinfo/ubuntu-users — подписывайтесь на рассылку для пользователей Ubuntu и взаимодействуйте с другими по электронной почте для решения задач, возникающих в любой области, начиная с реализации баз данных MySQL и заканчивая настройкой проблематичных сетевых устройств. Архив материалов прошлых обсуждений можно просмотреть по адресу https://lists.ubuntu.com/archives/ubuntu-users.
- https://wiki.ubuntu.com/IRCResourcePage— если вы заинтересованы в поддержке онлайн-чата IRC, можете посетить ресурсную страницу Ubuntu IRC, чтобы найти руководства, клиентов и чат-серверы, являющиеся доступным источником поддержки, которым можно бесплатно воспользоваться в любое время. Рекомендуется посетить страницу Ubuntu Code of Conduct (www.ubuntu.com/project/about-ubuntu/conduct), если вам еще не доводилось принимать участие в чатах IRC.

Если вы планируете купить аппаратное обеспечение для использования в сочетании со своей Ubuntu или другой Linux-системой, то приведенные далее сайты

могут быть полезными при принятии решения, где вам следует потратить свои деньги.

- www.linux-usb.org — этот сайт нацелен на поддержку практических знаний о USB-устройствах, дружественных к Linux. Здесь имеется система поиска, где вы сможете ввести название или модель устройства от того или иного производителя и сразу получить ответ с информацией, можно ли и насколько удобно использовать это устройство в сочетании с Linux.
- www.linuxfoundation.org/collaborate/workgroups/openprinting — система печати CUPS (http://cups.org) является стандартной системой такого типа, которая применяется в большинстве систем на основе Linux в наши дни. Если модель вашего принтера отсутствует в списке, когда вы пытаетесь добавить новый принтер в свою систему Ubuntu, то, возможно, вам следует поискать на этом сайте обновленный PDD-файл, чтобы добавить его в свою систему CUPS.
- www.sane-project.org — Scanner Access Now Easy (SANE) представляет собой сайт, посвященный теме сканирования документов в Linux. Если вы подбираете сканер или многофункциональное печатающее устройство, загляните на этот сайт, Чтобы узнать, насколько оборудование от того или иного поставщика совместимо с Linux.
- http://tldp.org — The Linux Documentation Project является самым большим собранием руководств, статей с практическими советами и инструкциями и ответов на часто задаваемые вопросы, из которых вы сможете узнать все что угодно, начиная с того, как сварить кофе с помощью Linux, и заканчивая настройкой QoS и управления трафиком.

Несомненно, это не полный список, однако перечисленные выше сайты являются хорошими ресурсами и на них следует заглянуть в первую очередь. Вы также можете попробовать поискать сведения о связанной с Linux поддержке на сайтах поставщиков программного обеспечения, прежде чем делать покупку. Если аппаратное обеспечение совместимо с Linux, то, возможно, на сайтах его поставщиков доступны драйверы или инструкции. Не забывайте о разнообразной информации касательно Linux, которую вы можете найти с помощью своего любимого поисковика.

Наконец, поищите местную группу пользователей Linux (Linux User Group, LUG) в своем регионе. Группа пользователей Linux — это локальное сообщество людей, страстно увлеченных системой Linux и ее реализациями. Вы найдете пользователей с широким спектром практических знаний, от системных администраторов до непрофессиональных пользователей Linux, лиц, занимающихся сопровождением дистрибутивов, а также исполнительных директоров компаний. Члены групп пользователей Linux, как правило, регулярно собираются для групповых дискуссий и проведения презентаций, во время которых демонстрируют найденные ими пути реализации системы Linux и связанных с ней технологий.

Некоторые группы пользователей Linux финансируют местные события вроде инсталлфестов (http://en.wikipedia.org/wiki/Install_fest) или других мероприятий, пропагандирующих Linux. Высока вероятность, что если вы зададите вопрос на собрании членов группы пользователей Linux, то кто-нибудь (и, скорее всего, таких

людей окажется несколько) будет знать ответ. Поисковик поможет найти группу пользователей Linux в вашем регионе, если вы решите вступить в нее. У большинства групп пользователей Linux имеются сайты или списки рассылки, которые можно без труда отыскать в Интернете.

Программное обеспечение для Ubuntu

Большую часть программного обеспечения для Ubuntu можно найти на сайте, где располагаются пакеты (http://packages.ubuntu.com). Стандартные инструменты — Synaptic, APT и Update Manager — являются самыми распространенными средствами установки программного обеспечения в вашу систему Ubuntu (в гл. 2 приведены подробности того, как найти и установить программное обеспечение).

С течением времени на Ubuntu, которая является зрелой Linux-системой, было портировано множество программных пакетов с открытым исходным кодом, предназначенных для работы в ней. Есть даже программные пакеты, предлагаемые для Ubuntu, которые включают несвободные компоненты (только двоичные или программное обеспечение, отягощенное программными патентами или другими ограничениями). Основной момент заключается в том, что нужное программное обеспечение следует искать сначала в официальных репозиториях Ubuntu, а уж затем обращаться к сторонним репозиториям ПО. С другой стороны, когда-нибудь вы, возможно, захотите поэкспериментировать и поискать программное обеспечение, отсутствующее в пакетах Ubuntu. Большинство пакетов будет располагать ключами MD5sum или GPG, которые вы сможете использовать для проверки подлинности загруженного программного обеспечения (www.debian-administration.org/articles/375). Вы также можете столкнуться с проблемами совместимости в случае с нестандартным программным обеспечением, что усложнит процесс обновления. Главное при экспериментировании с нестандартным программным обеспечением — чтобы его тестирование не привело к изменению вашей системы. Список ниже включает сайты, на которые вы можете заглянуть, чтобы узнать, какое другое ПО вам доступно.

ВНИМАНИЕ----------

Следует с осторожностью подходить к смешиванию программного обеспечения в вашей системе Ubuntu с приложениями из мест, которые не являются Ubuntu-источниками. Тщательно проверяйте аутентичность всего, что будете загружать.

- http://freecode.com/ — может похвастаться самой большой веб-коллекцией UNIX- и кросс-платформенного программного обеспечения, тем, заставок, а также программ для Palm-OS. Здесь также имеется раздел дискуссий для каждой программы, чтобы облегчить обсуждение и обратную связь. Эти ребята работают уже очень долгое время, хотя ранее их сайт назывался freshmeat.net.
- http://sourceforge.net — когда разработчики программного обеспечения с открытым исходным кодом собираются вместе, чтобы запустить новый проект, многие из них решают разместить его на ресурсе SourceForge. Он предлагает

веб-пространство, а также инструменты для управления проектами, ресурсами, коммуникациями и кодом. Если вы ищете программное обеспечение, то обязательно попробуйте найти его на SourceForge.

Сосредотачиваемся на Linux-командах

В наши дни многие важные задачи в Linux можно выполнять с помощью как графических интерфейсов, так и команд. Однако командная строка всегда была и попрежнему остается предпочитаемой продвинутыми пользователями Linux.

Графические интерфейсы пользователя (Graphical User Interface, GUI) должны быть интуитивно понятными. Имея некоторый опыт работы на компьютере, вы, вероятно, поймете, например, как добавить пользователя, изменить время и дату или настроить принтер посредством GUI-интерфейса. В подобных случаях я буду уточнять, какой графический инструмент можно использовать для выполнения соответствующей работы. Однако в приведенных далее ситуациях вам, вероятно, придется полагаться на командную строку.

- **Почти всегда что-нибудь идет не так** — задайте вопрос на онлайн-форуме, чтобы решить проблему Linux, с которой вы столкнулись, и оказываемая вам помощь почти всегда будет приходить в форме команд для ввода. Кроме того, инструменты командной строки обычно обеспечивают намного более информативную обратную связь, если проблема заключается в конфигурировании устройства или доступе к файлам и каталогам.
- **Удаленное системное администрирование** — если вы будете администрировать удаленный сервер, то, возможно, графические инструменты окажутся недоступными. Хотя удаленный GUI-доступ (с использованием X-приложений или VNC) и веб-инструменты администрирования могут быть в вашем распоряжении, они обычно позволяют справиться с задачами медленнее, чем командная строка.
- **Параметры, не поддерживаемые GUI-интерфейсом,** — GUI-инструменты администрирования имеют тенденцию обеспечивать самые простые способы решения задач. Более сложные операции зачастую требуют параметров, доступных только в командной строке.
- **Задачи на основе сценариев** — GUI-интерфейсы удобны, если нужно, например, добавить одного пользователя. Однако что, если вы захотите добавить сотню пользователей или собрать комплексные наборы данных о производительности в своей системе? Используя команды, заключенные в то, что называется сценариями интерпретатора команд, вы можете создавать комплексные и рекурсивные задачи, выполнение которых можно будет повторить позднее, просто запустив еще раз соответствующий сценарий.
- **GUI-интерфейс не функционирует как надо или не установлен** — если не доступен никакой графический интерфейс либо установленный GUI-интерфейс не функционирует должным образом, то вам, возможно, придется работать из командной строки. Сбои в работе GUI-интерфейсов могут случаться по многим причинам, например, когда вы используете сторонний, только двоичный

драйвер от NVIDIA, а обновление ядра приводит к тому, что драйвер перестает быть совместимым с ним.

Главное заключается в том, что для раскрытия всех возможностей Linux-системы вам нужно уметь пользоваться командами интерпретатора команд. В случае с Linux доступны тысячи команд для мониторинга и управления всеми областями вашей Linux-системы.

Однако независимо от того, являетесь ли вы Linux-гуру или новичком, одна проблема будет выглядеть угрожающе. Как вспомнить критически важные команды и параметры, когда в интерпретаторе команд вы видите лишь вот это:

```
$
```

Эта книга не является очередным справочником команд или пересказом MAN-страниц. Вместо этого в ней команды в Ubuntu Linux представлены так, чтобы вам было удобно их использовать. Другими словами, вместо того чтобы приводить команды в алфавитном порядке, я сгруппировал и расположил в отдельных главах команды для работы с файловыми системами, подключения к сетям и управления процессами, благодаря чему вы сможете обращаться к командам в соответствии с тем, что вам нужно сделать, а не искать их по имени.

Аналогичным образом я не стану перечислять все параметры, доступные для каждой команды. Вместо этого я приведу демонстрационные примеры самых важных и полезных параметров для использования в сочетании с каждой командой. Затем я покажу, как быстро найти дополнительные параметры, если они вам потребуются, с помощью MAN-страниц, инструмента info и справочных сообщений.

Поиск команд

Некоторых команд, описанных в этой книге, может не быть по умолчанию в вашем дистрибутиве Ubuntu, однако они наверняка доступны посредством APT или других источников. Если интерпретатор команд bash не может найти команду, которую вы ввели в Ubuntu, то это может быть обусловлено одной из следующих причин:

- команда вообще не существует;
- вы сделали ошибку при наборе имени команды (допустили опечатку);
- команды нет ни в одном из каталогов, в которые интерпретатор команд был проинструктирован заглянуть (переменная PATH);
- команда или пакет Ubuntu, который ее содержит, не установлены.

Интерпретатор команд будет реагировать по-разному в зависимости от причины, в силу которой команду не удалось найти. Если команда не установлена, отсутствует в каталогах, указанных в вашей переменной PATH, или недоступна в любом из известных программных пакетов, то вы увидите сообщение command not found (команда не найдена):

```
$ sillycommand
sillycommand: command not found
```

Если окажется, что команда содержится в известном пакете, который просто не установлен, то интерпретатор команд сообщит вам, что нужно ввести, чтобы установить этот пакет. Например, вот что будет, если вы попытаетесь запустить редактор kate, который не установлен:

```
$ kate
The program 'kate' is currently not installed. Install it by typing:
sudo apt-get install kate
```

Если введенная вами команда не найдена, однако ее имя похоже на название одной доступной команды или более, то интерпретатор команд попытается угадать, что вы имели в виду, и предложит пакеты для установки, чтобы вы смогли получить соответствующую команду в свое распоряжение. Вот что вы увидите, если введете cate вместо kate:

```
$ cate
No command 'cate' found, did you mean:
 Command 'kate' from package 'kate' (universe)
 Command 'cat' from package 'coreutils' (main)
 Command 'date' from package 'coreutils' (main)
 Command 'late' from package 'late' (universe)
 Command 'cfte' from package 'fte' (universe)
 Command 'cake' from package 'cakephp-scripts' (universe)
 Command 'yate' from package 'yate' (universe)
 Command 'catg' from package 'nauty' (multiverse)
cate: command not found
```

Если вы обнаружите, что нужная команда присутствует в выводе apt-cache search, либо заподозрите, что она не установлена, вы сможете установить ее из Интернета, введя следующее:

```
sudo apt-get install имя пакета
```

где имя пакета — это название того пакета, который вы желаете установить.

В приведенном далее списке представлен набор команд, которые вы можете выполнить в любом дистрибутиве Linux, чтобы проверить, присутствует ли в системе введенная вами команда.

ПРИМЕЧАНИЕ

Вы можете увидеть многоточие (...), которое используется в выводе, генерируемом кодом, для указания, где несущественная информация была опущена для краткости.

О Показать текущую переменную PATH:

```
S echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:
```

О Найти первый случай употребления команды mount в PATH:

```
$ which mount
/bin/mount
```

О Искать в файловой системе /usr файл или каталог с именем umount:

```
$ find /usr -name umount
/usr/lib/klibc/bi n/umount
```

О Показать, где находится первый двоичный файл и M AN-страница для команды mount:

```
$ whereis mount
mount: /bin/mount /usr/share/man/man8/mount,8.gz
```

О Использовать команду 1 ocate для поиска mount в соответствующем списке (конфигурируемых) каталогов:

```
$ locate mount
...
/usr/bi n/fdmountd
```

О Искать в описаниях MAN-страниц экземпляры ключевого слова — в этом примере им является umount:

```
$ apropos umount
...
umount (8)              - unmount file systems
```

О Показать раздел 8, содержащий M AN-страницу для umount (для выхода нажмите клавишу Q):

```
$ man 8 umount
Reformatting umount(8). please wait...
```

В следующем списке приведены схожие команды, специфичные для систем Ubuntu и Debian.

О Искать в кэшированном списке пакетов, которые могут содержать команду или описание umount:

```
$ apt-cache search umount
gnome-mount - wrapper for (un)mounting and ejecting storage devices
--
```

О Искать в списке установленных пакетов файл с именем umount и показать, в каком пакете он находится:

```
$ dpkg-query -S umount
...
initscripts: /etc/init.d/umountnfs.sh
--
```

О Показать все файлы, содержащиеся в пакете initscripts:

```
$ dpkg -L initscripts
...
/bin/mountpoint
--
```

О Обновить список кэшированных пакетов:

```
$ sudo apt-get update
Password:
Get:l http://security.ubuntu.com feisty-security Release.gpg [191B]
...
--
```

Справочная информация в Ubuntu

Вся оригинальная документация Linux и UNIX была собрана на страницах руководства, обычно называемых *MAN-страницами.* Чуть позднее была предпринята более изощренная попытка документирования, в результате которой появился GNU-инструмент info. Для каждой команды почти всегда доступны справочные сообщения.

Эта справочная информация ориентирована на компоненты. Почти для каждой команды, имеющейся в системе, существуют отдельные MAN-страницы. Они также содержат документацию по устройствам, форматам файлов, системе, информацию о разработчиках, а также сведения о многих других компонентах Linux-системы. Документация, в которой более подробно описываются целые пакеты, обычно находится в подкаталоге каталога /usr/share/doc.

В случае с Ubuntu значительная часть этой документации упакована, поэтому, чтобы прочесть, сначала ее потребуется распаковать. Для этого можно воспользоваться программой gzip, однако проинструктируйте ее, чтобы она только вывела на экран содержимое файлов, не распаковывая их на жесткий диск. Вот команда для распаковки журнала изменений, касающегося команды mount, и вывода на экран его содержимого:

```
$ gzip -dc /usr/share/doc/mount/changelog.Debian.gz j lessutil-1inux \
    (2.20.1-5.1ubuntu2) quantal; urgency=low
```

Инструмент info, MAN-страницы и каталоги /usr/share/doc доступны в большинстве Linux-систем.

Справочные сообщения

Почти все команды в Linux-системе по вашей просьбе выведут на экран краткую информацию по их применению. Часто, чтобы отобразить эти сведения, в сочетании с соответствующей командой используется аргумент -h или --help и ничего более. Приведенная далее команда является примером, как можно указать команде 1 s вывести на экран информацию о ее применении:

```
$ Is --help
Usage: Is [OPTION],.. [FILE],. .
Lust information about the FILEs (the current directory by default).
...
```

Флаг - -he! p позволяет вывести на экран много информации, но вы можете воспользоваться командой постраничного вывода, чтобы ограничить количество данных, выводимых на экран за один раз:

```
$ Is    - help | less
...
```

ПРИМЕЧАНИЕ

Если вам доводилось работать в UNIX-системах, то, вероятно, команда more оказалась первой командой постраничного вывода, которую вы использовали. Команда more доступна в Linux, однако предпочтение в Linux-системах отдается команде less, которая имеет странное название (в переводе означающее «меньше») и по иронии судьбы более функциональна, чем команда more. Она позволяет пролистывать выводимые данные назад и вперед, использовать клавиши со стрелками для прокрутки, а также понимает нажатия клавиш, которые используются в редакторе vi, для навигации и поиска по тексту.

В рассмотренных выше примерах было показано, как вывести на экран справочное сообщение о команде 1 s. Вы также можете отформатировать справочный вывод с помощью команды card, которая отправит данные прямо на принтер по умолчанию, либо сохранить эту информацию в Postscript-файле, чтобы ее можно было просмотреть позднее посредством чего-нибудь вроде утилиты evince, либо конвертировать ее в PDF-файл с использованием утилиты ps2pdf.

MAN-страницы

Вы можете выполнить команду apropos для поиска любого ключевого слова или группы символов в базе данных MAN-страниц. В выводе будут показаны разделы, включающие MAN-страницы, которые содержат слово, заданное вами при поиске с использованием apropos.

```
$ apropos crontab
/etc/anacrontab (5) [anacrontab] - configuration file for anacron
anacrontab (5)          - configuration file for anacron
crontab (1)             - maintain crontab files for individual users (V3)
crontab (5)             - tables for driving cron
```

Здесь в выводе apropos указаны разделы и MAN-страницы, где было найдено слово crontab. В разделах, содержащих MAN-страницы, они сгруппированы по темам. В разделе 1 MAN-страницы посвящены теме «Выполняемые программы или команды интерпретатора команд». В разделе 5 MAN-страницы относятся к теме «Форматы файлов и соглашения». Разделы с MAN-страницами одинаковы во всех Linux-системах, однако могут немного отличаться в других системах типа UNIX. Вы должны иметь возможность просмотреть MAN-страницу для команды man, чтобы узнать, какие разделы представлены в системе, в которой вы работаете:

```
$ man man
Reformatting man(l). please wait...
...
```

В списке ниже приведены номера разделов руководства и типы страниц, которые они содержат:

- 1 — выполняемые программы или команды интерпретатора команд;
- 2 — системные вызовы (функции, предоставляемые ядром);
- 3 — библиотечные вызовы (функции в программных библиотеках);
- 4 — специальные файлы (обычно располагающиеся в /dev);
- 5 — форматы файлов и соглашения, например /etc/passwd;
- 6 — игры;
- 7 — разное (включая макропакеты и соглашения), например шап(7), groff(7);
- 8 — команды системного администрирования (обычно только для суперпользователя);
- 9 — программы ядра (нестандартные).

Исходя из этой информации, вы видите, что введенное для поиска слово crontab встречается в разделе 1 («Выполняемые программы или команды интерпретатора команд»), а также в разделе 5 («Форматы файлов и соглашения»). Вы можете просмотреть MAN-страницы из этих разделов, передавая команде man в качестве аргумента номер каждого из них.

```
$ man 5 crontab
Reformatting crontab(5), please wait...
CR0NTAB(5)                                    CR0NTAB(5)
NAME
    crontab - tables for driving cron
DESCRIPTION
    A crontab file contains instructions to the cron(8) daemon of
    the general form: "run this command at this time on this date'".
...
```

Если не указать номер раздела, то команда man возвратит MAN-страницу из первого раздела, который найдет. В приведенном далее примере man возвращает раздел 1, содержащий MAN-страницы crontab:

```
$ man crontab
Reformatting crontab(l), please wait...
CRONTAB!1)                                    CRONTAB!1)
NAME
    crontab - maintain crontab files for individual users (V3)
...
```

Помимо номеров разделов, команда man принимает несколько аргументов для выполнения разных задач. Вот некоторые примеры.

- Последовательно показывает все разделы с MAN-страницами для crontab:

```
$ man -a crontab
```

- Показывает MAN-страницу из раздела 5 для crontab:

```
$ man 5 crontab
```

О Использует параметр more для постраничного просмотра частей справочника MAN, содержащих crontab:

```
$ man -P more crontab
```

О Эквивалент команды what 1 s:

```
$ man -f crontab
```

О Эквивалент команды apropos:

```
$ man -k crontab
```

Команда whatis — еще одна утилита поиска MAN-страниц. Она отличается от apropos тем, что выводит на экран только описания MAN-страниц, в которых встречается заданное вами ключевое слово. Выполнение команды apropos в отношении команды route возвращает три разные MAN-страницы, на которых было обнаружено упоминание слова route:

```
$ apropos route
NETLINK_ROUTE (7)     - Linux IPv4 routing socket
route (8)             - show / manipulate the IP routing table
traceroute6 (8)       - traces path to a network host
```

Если выполнить whati s в отношении команды route, то будет возвращена только MAN-страница из раздела 8 для команды route:

```
$ whatis route
route (8)             - show / manipulate the IP routing table
```

info-доку менты

В некоторых случаях разработчики позаботились о более полном описании команд, форматов файлов, устройств и других компонентов Linux, разместив соответствующую информацию в базе данных info, которая представляет собой что-то вроде связанного набора страниц интерактивного руководства. Вы можете войти в эту базу данных, просто введя команду info либо открыв соответствующий компонент (для выхода из утилиты info используйте клавишу Q).

```
$ info Is
```

Эта команда выводит информацию о команде Is. Вы можете осуществлять навигацию по утилите info с помощью клавиш T, *I,* <- ~~и~~а также Page Up и Page Down. Из нижеследующего списка вы узнаете больше о навигации в случае с info:

О ? — отобразить основные команды для использования в окнах info;

О Shift+L — вернуться к предыдущему узлу, который вы просматривали;

О N, P, U — перейти соответственно к следующему, предыдущему или располагающемуся вверху узлу;

О Enter — перейти к гиперссылке, которая находится под указателем;

О Shift+R — проследовать по перекрестной ссылке;

О Q или Shift+Q — завершить работу и выйти из info.

Программные пакеты, особенно подробное описание которых доступно в базе данных info, включают gimp, festival, libc, automake, zsh, sed, tar и bash. Файлы, используемые базой данной info, находятся в каталоге /usr/share/info.

Резюме

В этой короткой главе я рассмотрел некоторые различия и сходства Ubuntu Linux с прочими дистрибутивами Linux и другими UNIX-подобными системами. Вы узнали об онлайн-ресурсах, посвященных Ubuntu в частности и Linux в целом.

Вы выяснили, где можно найти программное обеспечение, специфичное для Ubuntu, а также другие Linux-приложения. Вы установили несколько пакетов с использованием Debian-инструмента Advanced Package Tool (APT) и освоили способы поиска команд и MAN-страниц в системе.

Вы, безусловно, можете прочитать книгу от корки до корки, однако она призвана быть справочником по сотням параметров в Ubuntu и Debian Linux, наиболее полезных для продвинутых пользователей и системных администраторов. Информация в этой книге упорядочена по темам, а не по алфавиту, поэтому вам не придется знать команды заранее, чтобы найти необходимую для выполнения соответствующей работы.

Большинство параметров, описанных в этой книге, с тем же успехом будут работать во всех системах на основе Linux, а многие из них смогут задействоваться и в унаследованных UNIX-системах.

2 Установка Ubuntu и добавление программного обеспечения

В этой главе:

- установка Ubuntu;
- работа с репозиториями программного обеспечения;
- получение программного обеспечения с помощью APT;
- управление программным обеспечением посредством инструментов управления пакетами Debian;
- извлечение файлов из пакетов других форматов.

К проверенным временем инструментам, используемым для начальной установки Ubuntu, а также последующего добавления и управления программным обеспечением, относятся утилиты APT (Advanced Package Tool) и dpkg (Debian package). Они являются представителями категории стандартных утилит для управления пакетами, выступающих в качестве внутренних интерфейсов в случае с более знакомыми настольными GUI-инструментами для управления программным обеспечением в Ubuntu и других системах на основе Debian. Эти утилиты для управления пакетами взаимодействуют с файлами .deb из онлайн-репозиториев или локальными файлами . deb, загруженными вами и находящимися на вашем жестком диске.

В этой главе освещаются важные аспекты, которые вам необходимо знать при начальной установке Ubuntu. В ней также представлена информация об онлайн-репозиториях программного обеспечения для Ubuntu. Подробные примеры использования APT, dpkg, tasksel и соответствующих утилит командной строки, включая aptitude, вы также увидите в данной главе.

Получение и установка Ubuntu

Дистрибутив Ubuntu и его близкие родственники Kubuntu, Xubuntu и Edubuntu разрабатывались так, чтобы их было легко использовать, а также с расчетом, что

если кто-либо решит перейти с одной системы на другую, многое будет ему уже хорошо знакомо. Эти дистрибутивы сосредоточены на том, чтобы все оставалось простым и понятным, тем самым помогая ускорить вашу адаптацию к новой системе.

Установщик Ubuntu (Ubiquity) — это первый шаг к осознанию простоты системы Ubuntu, сводящий процесс установки примерно к десяти щелчкам кнопкой мыши. Зайдя в Интернет, вы сможете бесплатно загрузить ISO-образы со страницы загрузки Ubuntu (www.ubuntu.com/download). Здесь вам будет предложено выбрать ISO-образ установочного носителя для настольного компьютера, сервера или облачной системы Ubuntu. После того как вы сделаете выбор, вам предоставят возможность загрузить последнюю версию Ubuntu либо самую свежую версию Ubuntu, для которой обеспечивается долгосрочная поддержка (LTS). Например:

- **Ubuntu 12.10** — появился в октябре 2012 года; поддержка серверной версии этого выпуска рассчитана на 18 месяцев (таким образом, ее поддержка закончится в апреле 2014 года);
- **Ubuntu 12.04 (LTS)** — появился в апреле 2012 года; поскольку это выпуск с долгосрочной поддержкой, он подкрепляется пятью годами гарантированной поддержки (то есть будет поддерживаться до апреля 2017 года).

Существуют 64- и 32-битные версии Ubuntu, доступные при выборе Ubuntu Desktop, Ubuntu Server или Ubuntu Cloud на странице загрузки Ubuntu. 64-битная версия не будет работать, если у вас стандартный компьютер на архитектуре x86 (32-битный); 32-битная версия должна работать на 64-битной машине, но не так эффективно, поэтому выбирайте версию, которая подходит для вашего компьютера.

По мере приобретения вами опыта взаимодействия с Ubuntu вы можете изучить другие пути получения в ваше распоряжение установленной системы Ubuntu. Вот некоторые методы, к которым можно прибегнуть на странице загрузки.

- **Juju** — используя Juju, системный администратор может взять развернутую систему Ubuntu и сохранить ее в том, что называется *наборами charm*. Эти наборы затем можно использовать для воспроизведения системы Ubuntu и ее развертывания в разных облачных средах.
- **Cloud Guest** — чтобы просто опробовать Ubuntu в облаке, вы можете выбрать Cloud Guest. В результате вы получите возможность попользоваться Ubuntu с некоторыми предварительно сконфигурированными приложениями в облаке в течение одного часа.

Если вы новичок в Ubuntu, рекомендую загрузить соответствующий ISO-образ установочного носителя и инсталлировать его на дополнительный компьютер, который, возможно, у вас имеется. Для этого нужно лишь выбрать версию, которая подходит для вашего компьютера, и загрузить ее.

По завершении загрузки вам, возможно, захочется посетить страницу HowToMD5SUM и получить тй5зит-файл для загруженной вами версии Ubuntu (https://help.ubuntu.com/community/HowToMD5SUM). Он может помочь проверить целостность ISO-образа. Такая цифровая подпись будет доступна в случае с большей

частью программного обеспечения с открытым исходным кодом; рекомендую проверить ее перед установкой или записью ISO-образа на CD/'DVD.

ПРИМЕЧАНИЕ----------

Если вы хотите быть более уверенными в безопасности загрузок, чем при использовании только контрольных сумм MD5, обратите внимание на SecureApt. Дополнительную информацию, как APT задействует цифровую аутентификацию и шифрование, можно найти в разделе SecureApt на справочном сайте Ubuntu (https://help.ubuntu.com/community/SecureApt).

Подготовка к установке

Если вы собираетесь удалить все с жесткого диска компьютера и установить Ubuntu, вам не придется заниматься предварительной подготовкой к установке. Если же вы хотите сохранить какие-либо данные с жесткого диска, то сделайте их резервную копию, прежде чем двигаться дальше. Чтобы установить Ubuntu, сохранив при этом данные на жестком диске, вам, возможно, потребуется изменить размеры существующих разделов жесткого диска и заново организовать их на нем. Информацию об изменении размеров разделов диска и командах для разбивки диска на разделы можно найти в гл. 7.

Выбор параметров установки

Для тестирования приведенных в этой книге команд я использовал серверный установочный носитель с 64-битной версией Ubuntu 12.04 (LTS). Вы можете выбрать другие версии установочных носителей. Команды должны сравнительно мало отличаться от приведенных в этой книге независимо от того, установите вы серверную или настольную систему либо даже воспользуетесь другим выпуском.

ПРИМЕЧАНИЕ----------

Если вы выберете серверный установочный носитель, то имейте в виду, что он обеспечит минимальную систему с возможностью добавления нескольких служб. Настольное программное обеспечение не будет установлено, однако вы сможете добавить его позднее в ходе изучения этой главы с помощью различных инструментов командной строки. Если хотите, вместо этого можно просто начать с установки настольного варианта.

После загрузки с установочного CD вас попросят выбрать язык для использования. Затем появится меню со следующими параметрами.

- Install Ubuntu Server (Установить Ubuntu Server) — немедленно начать процесс установки (выберите этот вариант).
- Check disc for defects (Проверить диск на наличие ошибок) — проверить CD на предмет проблем чтения с него, чтобы выявить их сейчас, а не посередине процесса установки.
- Test memory (Проверить память) — если вы подозреваете наличие неполадок с оперативной памятью вашего компьютера, то Ubuntu позволит запустить Memtest86 (www.memtest.org/) для стресс-теста оперативной памяти с целью выявления ошибок.

- Boot from first hard drive (Загрузиться с первого жесткого диска) — если вы случайно загрузились с CD-ROM, находящегося в приводе, то просто выберите этот пункт меню, чтобы загрузиться с первого жесткого диска.
- Rescue a broken system (Восстановить поврежденную систему) — установочный CD также может быть использован как диск для аварийного восстановления системы. Используйте его, если ваша Ubuntu вдруг перестанет загружаться, а вы захотите добраться до содержимого жесткого диска, чтобы устранить неполадку (соответствующий вариант при новой установке выбирать не нужно).

Вы можете подробнее узнать об установке Ubuntu из руководства по инсталляции в соответствии с типом и версией используемого вами установочного носителя. Например, касательно установки серверной версии Ubuntu 12.04 LTS загляните по следующему адресу: https://help.ubuntu.com/12.04/serverguide/installation.html.

Ответы на вопросы во время установки

Большинство окон, которые вы увидите во время установки Ubuntu, будут интуитивно понятными. Далее приведен список с кратким обзором этих окон наряду с советами в случаях, когда вам может понадобиться помощь.

- Install welcome (Добро пожаловать в программу установки) — выберите нужный язык.
- Where are you? (Где вы находитесь?) — укажите страну, в которой вы находитесь, чтобы определить ваш часовой пояс (для всемирных серверов иногда задаются настройки в соответствии со средним временем по Гринвичу).
- Keyboard layout (Раскладка клавиатуры) — вы можете понажимать клавиши, чтобы установщик определил раскладку клавиатуры, либо выбрать ее из списка доступных.
- Enter the hostname (Ввести имя хоста) — укажите одно имя хоста, которое будет представлять вашу систему. Например, вы просто напечатали бы abc, если бы полное имя вашего хоста было abc.example.cot.
- Set up users and passwords (Настроить учетные записи пользователей и пароли) — введите полное имя для пользователя, более короткое имя пользователя и пароль. Этот пользователь будет обладать привилегиями суперпользователя (в Ubuntu учетная запись суперпользователя не конфигурируется по умолчанию). Вас также спросят, желаете ли вы зашифровать домашний каталог (это позволит лучше защитить данные от того, кто может похитить ваш жесткий диск, однако вам придется запомнить пароль, который будет использоваться при шифровании этого каталога, поскольку в противном случае ваш домашний каталог окажется недоступным и для вас тоже).
- Configure your clock (Настроить ваши часы) — установщик попытается наладить связь с сетевым сервером времени, чтобы настроить время, и подскажет часовой пояс (исходя из вашего текущего местоположения). Вы можете выбрать часовой пояс, который он предложит, либо указать другой. Как уже отмечалось

ранее, некоторые администраторы серверов предпочитают задавать часовые пояса, ориентируясь на среднее время по Гринвичу, если требуется синхронизировать серверы, используемые по всему миру.

- Prepare disk space (Подготовить дисковое пространство) — выберите Guided (Авто) для разбивки диска на разделы, если хотите поручить Ubuntu решение, как разметить диск. Выберите Manual (Вручную), если хотите сами определить, как диск будет разбиваться на разделы. Рекомендую выбрать автоматическую разбивку диска на разделы с использованием LVM. Кроме того, если вы не используете все доступное пространство, то сможете прибегнуть к LVM-командам позднее (при изучении гл. 7), чтобы увеличить или уменьшить размеры своих LVM-разделов.
- Start the installation (Начать установку) — когда разбивка диска на разделы завершится, начнется установка.

В процессе установки вы, возможно, получите приглашение сделать следующее:

- сконфигурировать прокси-сервер, если понадобится добраться до репозиториев программного обеспечения для Ubuntu, чтобы получить приложения, необходимые для завершения установки;
- решить, должно ли выполняться автоматическое обновление программного обеспечения;
- выбрать, какое программное обеспечение следует установить (я просто добавил OpenSSH, чтобы можно было удаленно войти в систему на сервере, остальное добавлю позже);
- установить загрузчик GRUB (инсталлируйте его в главную загрузочную запись, если у вас не установлено никаких других операционных систем).

Во время работы установщика будут доступны множественные терминальные сеансы посредством комбинаций клавиш Ctrl+Alt+F2 и Ctrl+Alt+F3, если вам потребуется приглашение интерпретатора команд. Кроме того, сочетание Ctrl+Alt+F4 позволит отображать по ходу установки любые сообщения или сведения об ошибках, произошедших во время этого процесса. Для возврата в окно установки используйте Ctrl+Alt+Fl.

Работа с программными пакетами Debian

Если вы предпочитаете использовать GUI-инструмент для установки программного обеспечения, то знайте, что с рабочего стола или во время сеанса ssh с использованием параметра -X ssh для туннелирования XI1 доступно приложение Synaptic Package Manager (см. гл. 13). Утилита aptitude обеспечивает красивый (основанный на тексте) внешний интерфейс с использованием curses в случае с APT, если выполнить ее без аргументов. Кроме того, внешний интерфейс dselect для утилиты dpkg доступен в большинстве Debian-систем, однако научиться использовать его может оказаться трудной задачей.

В Ubuntu применяется формат пакетов Debian (обычно это архивы .ar) — стандартный инструмент, используемый при создании программных пакетов для систем на основе Debian. Благодаря тому что программные компоненты собираются в отдельные пакеты Debian (с расширением . deb), программное обеспечение может не только представлять собой самостоятельный архив с приложениями, но также содержать множество сведений о содержимом пакета. Эти метаданные могут включать описания программного обеспечения, информацию о зависимостях, архитектуре компьютера, поставщике, размере, лицензировании и другие сведения.

Когда базовая система Ubuntu будет установлена, вы сможете добавлять, удалять и иным образом управлять файлами .deb в соответствии с тем, как вы будете использовать эту систему. Файлы .deb используются для установки основной массы программного обеспечения в Ubuntu, Kubuntu, Xubuntu, Edubuntu и большинстве других систем на основе Debian. Инструмент aptitude отлично подходит для удовлетворения большинства повседневных нужд, связанных с программным обеспечением: однако существует множество других инструментов для управления пакетами, и, возможно, у вас время от времени будет возникать необходимость прибегнуть к некоторым из них.

- APT — используйте APT для загрузки и установки пакетов из онлайн-репозиториев. APT-команды (apt-get, apt-cache и т. д.) можно применять для установки пакетов, расположенных локально на вашем жестком диске. Однако обычно APT используется для работы с программным обеспечением, которое находится в Интернете.
- dpkg — применяйте его для работы с файлами .deb, расположенными на CD-ROM или другом дисковом носителе. Команда dpkg может принимать параметры для конфигурирования, установки и получения информации о программном обеспечении.
- aptitude — используйте его в командной строке для работы с онлайн-репозиториями. Инструмент aptitude рекомендуется выбирать в первую очередь, поскольку он автоматически позаботится о выполнении некоторых действий, которые вам придется выполнять вручную при работе с dpkg или APT.

Эта глава включает разделы, посвященные каждой из этих утилит, в которых описываются обстоятельства, наиболее соответствующие применению каждого инструмента.

ПРИМЕЧАНИЕ----------

Для получения дополнительной информации об этих инструментах управления пакетами обратитесь к MAN-страницам, посвященным APT и dpkg.

Ubuntu (и другие дистрибутивы *buntu) устанавливается с одного CD-ROM или DVD. По завершении инсталляции вы сможете ввести команду apt-cache stats для получения отчета об общем количестве доступных пакетов:

```
$ apt-cache stats
Total package names : 50267 (1,005 k)
```

```
Total package structures: 85142 (4,768 k)
Normal packages: 58468
...
```

Как видите, в минимальной базовой конфигурации Ubuntu доступно более 50 000 программных пакетов. Сообщество Debian предусмотрительно включает только программное обеспечение, которое подходит для повторного распространения.

ПРИМЕЧАНИЕ

В учебном руководстве по Debian, которое можно найти по адресу www.debian.org/doc/manuals/debian-tutorial/ch-introduction.html, отмечается: «Хотя разработчики Debian верят в свободное программное обеспечение, бывают ситуации, в которых людям хочется или требуется установить проприетарное программное обеспечение на свои компьютеры. Разработчики Debian по возможности поддержат это; несмотря на то что проприетарное программное обеспечение не включено в основной дистрибутив, иногда оно доступно на FTP-сайте в каталоге несвободных программ, при этом растет количество пакетов, используемых исключительно для установки проприетарного программного обеспечения, которое нам не разрешено распространять самостоятельно».

Canonical Group придерживается аналогичных стандартов в случае с Ubuntu (www.ubuntu.com/community/ubuntustory/licensing), предлагая программное обеспечение, разделенное на четыре категории — main, restricted, universe и multiverse.

- **Main** — содержит программное обеспечение, которое является свободно распространяемым и поддерживается командой Ubuntu. Значительная часть этого ПО устанавливается при инсталляции Ubuntu.
- **Restricted** — включает программное обеспечение, которое является общим для многих Linux-систем и поддерживается командой Ubuntu, однако может не распространяться по абсолютно свободной лицензии.
- **Universe** — содержит моментальный снимок почти всего программного обеспечения с открытым исходным кодом, имеющегося в мире Linux и доступного по лицензиям, которые могут не быть такими же свободными, как другие. Для ПО в этом компоненте не гарантируются полная безопасность и техническая поддержка.
- **Multiverse** — включает программное обеспечение, которое не соответствует концепции *свободного* программного обеспечения, являющейся основным компонентом лицензионной политики Ubuntu. Никакой поддержки программного обеспечения в этом компоненте не предусмотрено, и вам решать, обосновано ли лицензирование этих приложений.

Работа с программными пакетами

В следующих разделах описываются основы управления пакетами с пояснением, что происходит «за кулисами» и как устанавливать пакеты. Изучение данного материала — необходимый первый шаг, который нужно сделать перед изучением других инструментов, например aptitude.

Команда dpkg является очень мощным инструментом, если речь идет об установке, одиночных пакетов . deb, однако она не позволяет тщательно проверять и устанавливать зависимости, необходимые различным приложениям, а также работать с репозиториями программного обеспечения наподобие компонентов Ubuntu, упоминавшихся ранее. APT же позволяет разрешать и устанавливать зависимости, а также обращаться к сконфигурированным репозиториям, однако не используется для установки файлов . deb, расположенных на жестком или другом локальном диске.

В некоторых других дистрибутивах Linux задействуются системы управления пакетами, схожие с APT. Дистрибутивы, основанные на (производные от) Red Hat (в том числе Red Hat Enterprise Linux, CentOS, Fedora и Mandriva), включают инструменты, например yum, rpm, urpmi и smart, позволяющие управлять программным обеспечением. Несмотря на то что эти инструменты отличаются от используемых в Ubuntu, идеи в случае с ними схожи; конфигурационный файл присутствует, чтобы сообщить инструменту управления пакетами, где в Интернете можно найти самые свежие программные пакеты. Инструмент управления пакетами затем будет работать совместно с установщиком, чтобы снабдить систему нужным программным обеспечением.

Подобная система из, так сказать, сборщика пакетов из Интернета и внутреннего инструмента управления пакетами является очень мощной комбинацией, призванной разрешать вопросы, касающиеся зависимостей, осуществлять цифровую аутентификацию целостности программного обеспечения, легко поддерживать систему в актуальном состоянии, а также давать возможность лицам, занимающимся сопровождением дистрибутивов, без труда и в больших масштабах распространять внесенные ими изменения.

Реагирование на сообщения об ошибках, связанных с локалями. Работая в командной строке в Ubuntu, когда будете пытаться установить пакеты, вы можете увидеть сообщение об ошибке, связанной с локалью, похожее на одно из следующих:

```
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
locale: Cannot set LC_CTYPE to default locale: No such file/directory
```

Похоже, что проблема связана с языковыми настройками или с чем-то, что касается интернационализированных кодировок в целом. Один из способов, позволяющих решить эту задачу, заключается в экспорте переменной среды LC_ALL и задании для нее значения согласно настройке LANG:

```
$ export LC_ALL="$LANG"
```

На справочных сайтах предлагаются другие возможные способы устранения этой проблемы, однако изменения, внесенные в результате применения показанного способа, будет проще всего отменить, если они приведут к еще большему количеству проблем, чем было изначально. Он также должен сработать независимо от языка, на котором вы говорите. Следует отметить, что вам придется вводить эту команду каждый раз, когда вы будете открывать локальный интерпретатор команд или интерпретатор команд ssh. Вы можете автоматизировать выполнение этой задачи, поместив соответствующую команду в файл ~/ .bashrc.

Активизация большего количества репозиториев для APT

В предыдущих выпусках Ubuntu репозитории multiverse и universe были деактивизированы по умолчанию. Однако теперь они по умолчанию активизированы, в силу чего обновления и поиск программного обеспечения вы сможете осуществлять, имея в своем распоряжении намного больше вариантов. У вас, возможно, вызовет беспокойство, что поддержка, лицензирование и патчи могут быть недоступны в случае с репозиториями multiverse и universe. Это может оказаться проблемой, если вы подумываете об установке, во время которой будете придерживаться определенных политик и процедур.

Чтобы деактивизировать репозиторий universe или multiverse, откройте в текстовом редакторе файл `/etc/apt / sources .list` и снабдите комментариями строки, в которых активизирован компонент multiverse или universe. Вы, возможно, захотите отметить места с комментариями с целью подчеркнуть, что именно вы их оставили, как это было сделано посредством #сп в приведенных далее примерах:

```
#cn deb http://us.archive.ubuntu.com/ubuntu/ precise universe
#cn deb-src http://us.archive.ubuntu.com/ubuntu/ precise universe
#cn deb http://us.archive.uDuntu.com/ubuntu/ precise-updates universe
#cn deb-src http://us.archive.ubuntu.com/ubuntu/ precise-updates universe
#cn deb http://us.archive.ubuntu.com/ubuntu/ precise multiverse
#cn deb-src http://us.archive.ubuntu.com/ubuntu/ precise multiverse
#cn deb http://us.archive.ubuntu.com/ubuntu/ precise-updates multiverse
#cn deb-src http://us.archive.ubuntu.com/ubuntu/ precise-updates multiverse
```

Еще одна причина отредактировать файл `/etc/apt/sources .list` заключается в добавлении дополнительных репозиториев, которые могут предлагаться частными лицами или компаниями. Чтобы иметь возможность редактировать этот файл, вы должны обладать правами доступа суперпользователя:

```
$ sudo vi /etc/apt/sources.list
```

Вставьте строку, начинающуюся с deb (в случае с предварительно подготовленными пакетами), или deb-src (для пакетов с исходным кодом), а затем URL-адрес репозитория вместе с дистрибутивом (как, например, preci se в приведенном ранее примере), а также описания компонентов (inverse в примерах выше). Обычно вы будете описывать как `contrib` компоненты, создаваемые сторонними разработчиками в качестве вклада (то есть не из проекта Ubuntu), а также как free или `non-free`. По большей части вы должны получать всю эту информацию с сайта, который предоставляет соответствующий репозиторий.

Если вы решите добавить другие сторонние репозитории, обязательно проверьте аутентичность источника, предлагающего программное обеспечение, прежде чем модифицировать свою Linux-систему. В наши дни это не является большой проблемой в случае Linux, однако очень просто добавить дефектное или вредоносное программное обеспечение в свою систему, если проявить невнимательность и не соблюсти меры предосторожности.

Используйте программное обеспечение только из хорошо знакомых источников и всегда держите под рукой средства для проверки загружаемого ПО перед его установкой.

Добавление коллекций программного обеспечения с помощью tasksel

Если вы начнете с установки минимальной системы, что было описано в разделе «Получение и установка Ubuntu» данной главы, то, возможно, найдете удобным добавление групп пакетов для получения всей требуемой функциональности. Это избавит вас от необходимости выискивать все пакеты, которые нужны для обеспечения определенной функциональности.

ПРИМЕЧАНИЕ----------

Если вы выберете коллекцию программного обеспечения с использованием tasksel, загрузится и установится множество пакетов. Например, когда я устанавливал рабочий стол Ubuntu с применением tasksel, было загружено и установлено свыше 900 пакетов. Процесс занял более 30 минут, притом, что у меня был доступ в Интернет с приличной скоростью, поэтому убедитесь, что вам действительно это нужно, прежде чем выбирать коллекцию пакетов.

Для запуска tasksel просто введите следующее:

```
$ sudo tasksel
```

Появится меню доступных коллекций программного обеспечения. Если окажется, что в вашей системе не установлен интерфейс рабочего стола, то вы сможете выбрать тот или иной вариант из нескольких разных интерфейсов рабочего стола. Используйте соответствующую клавишу со стрелкой, чтобы перейти к нужной коллекции, затем нажмите Пробел, чтобы выбрать ее (при этом появится звездочка), после этого нажмите Tab, чтобы выделить кнопку OK, а затем Enter для начала установки.

Вот пример выбора базового рабочего стола Ubuntu:

```
[ ] Ubuntu LXDE Desktop
[*] Ubuntu desktop
[ ] Ubuntu desktop USB
```

Когда пакеты будут загружены и установлены, вы, возможно, получите приглашение выполнить дополнительное конфигурирование. В некоторых случаях вам может потребоваться перезагрузить компьютер, чтобы изменения вступили в силу.

Управление программным обеспечением с помощью APT

Функциональные возможности dpkg и APT могут задействоваться, дополняя друг друга, однако в большинстве случаев APT будет достаточно для управления любым программным обеспечением, которое вам потребуется установить, загрузить,

обновить, проверить, а также для осуществления поиска любой системы на основе Debian. Упоминая APT, я имею в виду набор команд для управления программным обеспечением, к которым относятся apt-get, apt-cache, apt-key и др.

Для получения краткой справочной информации о возможностях APT посредством командной строки используйте в ней параметр -h в сочетании с любой интересующей вас APT-командой. Некоторые команды APT требуют наличия привилегий суперпользователя, поэтому им должна предшествовать команда sudo, чтобы они функционировали должным образом. Далее показано, как APT-команды используются на практике.

ПРИМЕЧАНИЕ

Утилита aptitude предпочтительнее APT; тем не менее, чтобы рассказать вам об основных положениях, я сначала рассмотрю APT.

- Эта команда обращается к /etc/apt/sources .list и обновляет базу данных доступных пакетов. Обязательно вводите эту команду при каждом изменении sources.list;

```
$ sudo apt-get update
```

- Данная команда выполняет нечувствительный к регистру поиск в базе данных пакетов в соответствии с заданным *ключевым, словом.* Возвращаются имена пакетов и описания, в которых было найдено искомое ключевое слово:

```
$ apt-cache search ключевое слово
```

- Чтобы загрузить и установить *пакет* согласно заданному имени, присутствующему в базе данных пакетов, используйте приведенную далее команду. Начиная с версии APT 0.6, эта команда автоматически проверяет аутентичность пакетов, используя известные ей GPG-ключи (http://wiki.debian.org/SecureApt):

```
$ sudo apt-get install пакет
```

- Чтобы только загрузить *пакет*, разместив его в /var/cache/apt/archi ves, используйте следующую команду:

```
$ sudo apt-get -d install пакет
```

- Для вывода на экран информации о программном обеспечении, содержащемся в указанном *пакете*, используйте эту команду:

```
$ apt-cache show пакет
```

- Чтобы произвести проверку на предмет наличия обновлений для всех установленных пакетов с последующим выводом приглашения загрузить и установить их, примените следующую команду:

```
$ sudo apt-get upgrade
```

- Данная команда обновляет всю систему в соответствии с новым выпуском, даже если это подразумевает удаление пакетов (такой метод обновления системы не считается предпочтительным):

```
$ sudo apt-get dist-upgrade
```

- Вводите эту команду всякий раз, когда вам необходимо удалить частично загруженные пакеты либо пакеты, которые больше не установлены:

```
$ sudo apt-get autoclean
```

- Для удаления всех кэшированных пакетов из /var/cache/apt/archives с целью высвобождения дискового пространства используйте следующую команду:

```
$ sudo apt-get clean
```

- Эта команда удаляет указанный пакет и все его конфигурационные файлы. Чтобы сохранить конфигурационные файлы, потребуется убрать ключевое слово --purge:

```
$ sudo apt-get --purge remove пакет
```

- Данная команда запускает проверку работоспособности на предмет поврежденных пакетов. Она попытается устранить проблемы, о которых будут свидетельствовать сообщения unmet dependency (взаимозависимости не удовлетворены):

```
$ sudo apt-get -f install
```

- Для вывода на экран информации об установленных APT-утилитах используйте следующую команду:

```
$ apt-config -V
```

- Эта команда отображает GPG-ключи, известные APT:

```
$ sudo apt-key list
```

- Для вывода на экран статистики по всем установленным пакетам используйте данную команду:

```
$ apt-cache stats
```

- Задействуйте эту команду для вывода на экран зависимостей пакета (будь он установлен или нет):

```
$ apt-cache depends
```

- Для вывода списка всех установленных в системе пакетов вам потребуется следующая команда:

```
$ apt-cache pkgnames
```

Поиск пакетов с помощью APT

Для поиска пакетов в любом доступном репозитории вы можете **выполнить запрос на предмет наличия нового программного обеспечения** с использованием команды

apt-cache. Например, чтобы найти игру в танки под названием bzflag, можно ввести следующее:

```
$ apt-cache search bzflag
bzflag - a 3D first person tank battle game
bzflag-client - BZFlag client
bzflag-data - BZFlag data file
bzflag-server - bzfs - BZFlag game server
```

Вы также можете **попросить APT показать информацию** о пакете bzflag:

```
$ apt-cache show bzflag
Package: bzflag
Priority: optional
Section: universe/games
Installed-Size; 212
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Original-Maintainer: Tim Riker <Tim@Rikers.org>
...
```

Сколько дополнительного программного обеспечения потребуется для обновления bzflag? **Осуществите проверку на предмет зависимостей,** введя следующее:

```
$ apt-cache depends bzflag
  Depends: bzflag-client
  Depends: bzflag-server
...
```

Установка пакетов с помощью APT

Используя sudo в сочетании с APT или другим инструментом управления пакетами, **вы можете устанавливать приложения, доступные в любом репозитории программного обеспечения,** который вы активизировали для Ubuntu. Далее приведена процедура установки bzflag с помощью APT:

```
$ sudo apt-get install bzflag
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
   bzflag-client bzflag-data bzflag-serverThe following NEW packages will be installed:
  bzflag bzflag-client bzflag-data bzflag-server
0 upgraded, 4 newly installed. 0 to remove and 61 not upgraded.
Need to get 12.0 MB of archives.
After this operation. 19.4 MB of additional disk space will be used.
Do you want to continue [Y/n]? Y
...
```

Как только инсталляция завершится и у вас при этом будет установлен рабочий стол, вы сможете запустить программу bzflag, введя bzfl ад в командной строке либо выбрав ее в меню **Games** (Игры) на рабочем столе Ubuntu.

Обновление пакетов с помощью APT

С течением времени пакеты изменяются и появляются их свежие версии, в которых присутствуют новые замечательные функциональные возможности и устранены проблемы. Вы можете задействовать APT для обновления своей системы в соответствии с новыми версиями, придерживаясь двухэтапного процесса.

Сначала **осуществите проверку на предмет наличия обновлений** для пакетов, известных вашей системе Ubuntu, использовав параметр update в сочетании с apt-get:

```
$ sudo apt-get update
```

Эта команда выполнит поиск новых версий пакетов, доступных для загрузки, в репозиториях и обновит список пакетов и версий, кэшированных вашей системой Ubuntu.

Затем обновите пакеты в своей системе Ubuntu, использовав параметр upgrade с apt-get:

```
$ sudo apt-get upgrade
```

Следует обновлять список пакетов до обновления самих пакетов, поэтому хорошее решение — совместное выполнение данных команд во всех случаях. Разделите эти команды точкой с запятой, чтобы дать указание интерпретатору команд выполнить их обе, одну за другой:

```
$ sudo apt-get update; sudo apt-get upgrade
Get:1 http://us.archi ve.ubuntu.com/ubuntu/ preci se-updates/mai n
base-files amd64 6.5ubuntu6.4 [69.9 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu/ precise-updates/main dpkg
amd64 1.16.1.2ubuntu7.1 [1,830 kB]
...
The following packages will be upgraded:
  accountsservice apparmor apt apt-transport-https apt-utils
aptitude base-files
  bind9-host busybox-initramfs busybox-static coreutils dbus dnsutils
dpkg gnupg gpgv
...
59 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
Need to get 43.9 MB of archives.
After this operation, 569 kB of additional disk space will be used.
Do you want to continue [Y/n]? Y
```

Обновление одиночного программного пакета

Обновление одиночного системного пакета в Ubuntu — достаточно простая процедура, для осуществления которой нужно ввести apt-get install <название пакета>. Вам потребуется выполнить эту команду с использованием sudo, чтобы получить

права доступа суперпользователя. Старая версия будет автоматически обновлена в соответствии с новейшей из доступных.

ПРИМЕЧАНИЕ----------

Это может показаться нелогичным, но использование параметра upgrade в сочетании с apt-get приводит к обновлению всех пакетов. Параметр install обеспечивает инсталляцию нового пакета либо установку обновления для одного определенного пакета или более.

Сначала **проверьте версию установленного на текущий момент приложения.** В рассматриваемом примере я исхожу из того, что более ранняя версия приложения minicom, предназначенного для коммуникаций посредством линий последовательной передачи, уже установлена в вашей системе. Как и большинство команд, программа minicom поддерживает возможность вывода на экран номера своей текущей версии:

```
$ minicom --version
minicom version 2.2 (compiled Mar 7.2007)
...
```

Теперь задействуйте APT для установки самой последней версии пакета minicom из репозиториев. APT сообщит вам, что будет произведено обновление пакета:

```
$ sudo apt-get install minicom
...
The following packages will be upgraded:
  minicom
```

Теперь снова узнайте у программы minicom ее версию, и вы увидите, что она действительно была обновлена:

```
$ minicom --version
minicom version 2.5 (compiled May 2 2011)
...
```

Удаление пакетов с помощью APT

Вы можете **удалить пакет** из своей системы Ubuntu, снабдив apt-get параметром remove. Вы получите приглашение подтвердить удаление программного обеспечения, прежде чем оно будет действительно осуществлено:

```
$ sudo apt-get remove bzflag
Reading package lists... Done
Building dependency tree
Reading state information... Done
...
The following packages will be REMOVED:
  bzflag
0  upgraded, 0 newly installed, Г to remove and 2 not upgraded.
After this operation, 217 kB disk space will be freed.
Do you want to continue [Y/n]? n
```

Очистка кэша от пакетов с помощью APT

После того как вы произведете начальную установку выпуска Ubuntu, эта система будет хранить загруженные пакеты кэшированными в /var/cache/apt/ для ускорения их загрузки, если они когда-нибудь потребуются вам снова. Со временем они могут начать занимать много пространства на диске. Вы можете очистить имеющийся кэш пакетов, однако придется снова загружать тот или иной удаленный пакет позднее, если того потребуют зависимости. Для очистки кэша необходимо снабдить apt-get параметром cl еап. Я покажу, как все это работает, сначала выполнив Linux-команду find в отношении каталога /var/cache/apt/, чтобыузнать, какие пакеты являются кэшированными на текущий момент:

```
$ find /var/cache/apt/ -name \*.deb
/var/cache/apt/archi ves/nauti1us_lX3a3.4.2-0ubuntu6_amd64.deb
/var/cache/apt/archives/gwibber-service-identica_3.4-0ubuntu2.1_all.deb
/var/cache/apt/archives/gi rl.2-peas-l.0_1.2.0-lubuntul_amd64.deb
```

Теперь удалим все пакеты, кэшированные в каталоге /var/cache/apt/, а затем убедимся, что они были удалены, снова выполнив команду find:

```
$ sudo apt-get clean
$ find /var/cache/apt/ -name \*.deb
```

Соответствующий каталог теперь пуст.

ПРИМЕЧАНИЕ----------

Если вы используете параметр -h в сочетании с apt-get, то, возможно, заметите, что в Ubuntu-версии APT имеется Super Cow Powers. Подробнее о Super Cow Powers вы сможете узнать, введя apt-get с параметром moo.

Загрузка пакетов с помощью APT

Если вы хотите загрузить пакет из репозитория программного обеспечения, не устанавливая его, можете воспользоваться apt-get в сочетании с параметром download. Когда загрузка пакета будет завершена, вы сможете исследовать его с помощью команды dpkg перед установкой (как описано в следующем разделе). Чтобы загрузить пакет minicom в текущий каталог, введите следующее:

```
$ sudo apt-get download minicom
```

Управление программным обеспечением с помощью dpkg

Утилита dpkg (и родственные ей команды) работает на более низком уровне, чем APT-утилиты. APT использует dpkg «за кулисами» для управления программным обеспечением в вашей системе Ubuntu. APT и dpkg работают во многом так же, как yum и rpm в дистрибутивах Linux на основе Red Hat. Обычно функциональности APT будет хватать, чтобы вы могли справиться практически с любой задачей,

однако в некоторых ситуациях потребуется dpkg, например, чтобы узнать, какой пакет ассоциирован с определенным файлом в вашей системе. Далее приведены некоторые общие команды и операции dpkg.

ПРИМЕЧАНИЕ

Утилита dpkg задействует флаг -D для обозначения отладочной информации, которая будет выводиться на экран во время выполнения различных операций. Если вы захотите, чтобы на экран выводилось больше информации, чем по умолчанию, попробуйте использовать -D1 в сочетании с некоторыми командами dpkg. В соответствующей секции MAN-страницы dpkg приведены уровни вывода в случае использования флага -D.

- Эта команда показывает файлы, установленные с использованием файла пакета .deb (вместо whatever.deb укажите имя пакета, который хотите запросить). Нужно указать полный или относительный путь к файлу .deb:

```
$ dpkg -c whatever.deb
```

- Данная команда отображает информацию о файле пакета .deb:

```
$ dpkg -I whatever.deb
```

- Эта команда отображает информацию о пакете:

```
$ dpkg -p whatever.deb
```

- Следующая команда показывает пакеты, в которых был обнаружен файл с определенным названием. Файловое имя может включать путь к файлу или только имя файла:

```
$ dkpg —S имя файла
```

- Эта команда показывает все установленные пакеты. Она также принимает параметры для вывода на экран более специфической информации:

```
$ dpkg -1
```

- Данная команда показывает все файлы, которые были установлены из пакета (этот пакет должен уже быть установлен):

```
$ dpkg -L пакет
```

- Параметр -s позволяет вывести на экран статут определенного пакета:

```
$ dpkg —s пакет
```

- Эта команда устанавливает определенный файл пакета . deb:

```
$ sudo dpkg -i whatever.deb
```

- Следующая команда удаляет определенный пакет из системы, но оставляет конфигурационные файлы (таким образом, если этот пакет будет переустановлен позднее, он сможет возобновить использование ваших конфигурационных файлов):

```
$ sudo dpkg -r пакет
```

- Эта команда удаляет определенный пакет и его конфигурационные файлы:

```
$ sudo dpkg -P пакет
```

- Данная команда извлекает файлы, содержащиеся в файле пакета . deb, в целевой каталог. Следует отметить, что она сбросит права доступа к целевому каталогу либо создаст соответствующий каталог, если его еще нет:

```
$ sudo dpkg -x whatever.deb /tmp/whatever
```

- Выполняя команду dpkg, любой пользователь может запрашивать базу данных пакетов. Чтобы задействовать эту команду для установки или удаления программного обеспечения из вашей системы, вы должны обладать привилегиями суперпользователя.

Установка пакета с помощью dpkg

Команда dpkg сосредоточена на работе только с пакетами, в то время как apt-get заботится о таких досадных деталях, как выяснение, в каком репозитории находится нужный пакет, и его загрузка оттуда. Для простоты в приведенном далее примере apt-get используется для загрузки пакета, после чего задействуется dpkg для его установки, чтобы вы ощутили, что такое базовая команда dpkg:

```
$ sudo apt-get download minicom
Get:1 Downloading minicom 2.5-2 [297 kB]
Fetched 297 kB in Os (302 kB/s)
```

Эта команда **загружает пакет в текущий каталог.**

Теперь **инсталлируйте файл .deb** с помощью команды dpkg -i (для **установки):**

```
$ sudo dpkg -i /var/cache/apt/archives/minicom_2.5-2_amd64.deb
Selecting previously deselected package minicom.
(Reading database ... 185336 files and directories currently installed.)
Unpacking minicom (from .../mimcom_2.5-2_amd64.deb) ...
Setting up minicom (2.5-2)      -
Processing triggers for MAN-db ...
```

Удаление пакета с помощью dpkg

Чтобы **удалить установленный пакет с помощью dpkg,** используйте параметр -г следующим образом:

```
$ sudo dpkg -r minicom
(Reading database ... 185407 files and directories currently installed.)
Removing minicom
Processing triggers for MAN-db ...
```

Если вы захотите удалить сразу и пакет, и его конфигурационные файлы либо удалить конфигурационные файлы после удаления пакета, используйте следующую команду:

```
$ sudo dpkg -P minicom
(Reading database ... 185335 files and directories currently installed.)
Removing minicom ...
Purging configuration files for minicom ...
```

Извлечение файлов из файла .deb с помощью dpkg

Пакеты Debian и Ubuntu поставляются упакованными в одиночные файлы .deb. Каждый файл . deb содержит один файл или более, образующих пакет как таковой, к которым относятся, например, содержащие предварительно подготовленные команды; являющиеся файлами поддержки; включающие документацию и, возможно, исходный код.

Файл .deb, по сути, представляет собой архив файлов, которые вы хотите установить на компьютер, плюс некоторые заголовочные и управляющие данные, которые идентифицируют программное обеспечение (описания, контрольные суммы, информация о сборке и т. д.).

Вы можете **извлечь множество информации, содержащейся в пакете, с помощью команды dpkg.** В приведенном ниже примере в каталог /tmp извлекаются файлы из файла .deb, располагающегося в каталоге/var/cache/apt/archi ves, которые относятся к пакету rsync:

```
$ mkdir /tmp/rsync_contents
$ sudo dpkg -x /var/cache/apt/archives/rsyncJ3.0.9-lubuntul_amd64.deb \
      /tmp/rsync_contents
$ Is /tmp/rsync_contents/
etc lib usr
```

Вы можете заменить файл rsync_3.0.9-lubuntul_amd64. deb, показанный в предыдущей команде, на любой загруженный вами файл . deb.

Запрос информации о пакетах .deb

В приведенном далее примере показано, как **выполнить запрос касательно установленных пакетов с целью найти пакет rsync и отобразить информацию о его версии:**

```
$ dpkg -p rsync
...
Version: 3.0.9-lubuntul
```

О Используйте параметр -I для получения информации о файле .deb, расположенном в любом каталоге:

```
$ dpkg -I rsyncJ3.0.9-lubuntul_amd64.deb
new debian package, version 2.0.
...
```

О Для **постраничного просмотра списка всех пакетов, установленных в вашей системе,** введите следующую команду:

```
$ dpkg -1 | less
...
и accountsservice 0.6.15-2ubuntu9.4 query and manipulate user acct info
...
```

О Чтобы просто **увидеть листинг определенного пакета,** используйте параметр -1 с именем этого пакета:

```
$ dpkg -1 rsync
...
ii rsync 3.0.9-lubuntul fast, versatile, remote (and local) file-copyi
```

О Проверьте файл в вашей системе, чтобы узнать, к какому пакету этот файл относится, если таковой существует:

```
$ dpkg -S /usr/bin/rsync
rsync: /usr/bin/rsync
```

Теперь, когда вы знаете, как выбрать пакет (-ы), который (-е) хотите запросить, давайте выудим из него еще немного информации. В приведенном далее примере **показаны стандартные подробности об установленном пакете:**

```
$ dpkg -s rsync
Package: rsync
Status: install ok installed
Priority: optional
Section: net
Installed-Size: 638
...
```

Эта команда **показывает содержимое файла .deb** в локальном каталоге:

```
$ dpkg -c /var/cache/apt/archives/rsync_3.0.9-lubuntul_amd64.deb
drwxr-xr-x   root/root          0      2011-11-08 14:05
drwxr-xr-x   root/root          0 2011-11-08 14:05  ./etc/
drwxr-xr-x   root/root          0 2011-11-08 14:05  ,/etc/default/
-rw-r--r--   root/root       1768 2011-11-08 14:05  ,/etc/default/rsync
drwxr-xr-x   root/root          0 2011-11-08 14:05  ./etc/i ni t.d/
-rwxr-xr-x root/root         4395 2011-11-08 14:05  ,/etc/init.d/rsync
...
```

В этом примере демонстрируется **извлечение управляющих сценариев, расположенных в файле .deb, в целевой каталог на диске.** Будьте внимательны при извлечении, поскольку эта команда сбросит права доступа к целевому каталогу в значение 0755 (55 означает, что пользователи, за исключением вас, будут обладать ограниченными правами доступа к /tmp, а большинство приложений предполагает, что у них имеются широкие права доступа к /tmp). Для этого примера я создам каталог /tmp/my_$RANDOM (my_ и произвольное число), чтобы работать в нем:

```
$ cd /var/cache/apt/archives
$ sudo dpkg -e rsync_3.0.9-lubuntul_amd64.deb /tmp/my_$RANDOM
$ Is -lart /tmp/my_25445/
total 32
total 32
-rwxr-xr-x   1 root  root   494 Nov  8   2011 prerm
-rwxr-xr-x   1 root  root   113 Nov  8   2011 postrm
-rwxr-xr-x   1 root  root   712 Nov  8   2011 postinst
-rw-r--r--   1 root  root    37 Nov  8   2011 conffiles
-rw-r--r--   1 root  root  1012 Nov  8   2011 control
-rw-r--r--   1 root  root  1743 Nov  8   2011 md5sums
```

Чтобы **извлечь в каталог все файлы, которые не являются управляющими** и содержатся в файле .deb, используйте параметр -x, как показано далее (опять-таки имейте в виду, что права доступа к целевому каталогу будут сброшены и станут 0755):

```
$ sudo dpkg -x minicom_2.5-2_amd64.deb /tmp/dx $RAND0M
$ Is -lart /tmp/dx_4921/
total 16
drwxr-xr-x      4root   root 4096 May   2   2011 usr
drwxr-xr-x      3root   root 4096 May   2   2011 etc
drwxr-xr-x      4root   root 4096 May   2   2011 .
drwxrwxrwt   15root   root 4096   Jan 27  22:03 ..
```

Чтобы **увидеть установленные файлы, используемые пакетом в системе,** задействуйте параметр -L:

```
$ dpkg -L minicom
/ .
/usr
/usr/share
/usr/share/man
/usr/share/man/manl
/usr/share/man/manl/minicom.l.gz
/usr/share/man/manl/xminicom.l.gz
/usr/share/man/manl/asci i-xfr.1.gz
. . .
```

Если пакет не был удален полностью, то вы, возможно, увидите какие-то оставшиеся конфигурационные файлы:

```
$ dpkg -L minicom
/etc
/etc/mi ni com
/etc/mini com/mi ni com.users
```

В приведенных выше примерах были описаны базовые варианты использования утилиты dpkg, однако рассмотренный перечень ни в коей мере не является исчерпывающим. К другим доступным случаям относятся предполагающие реконфигурирование пакетов (dpkg-reconfigure); сообщение dpkg о том, какие пакеты следует игнорировать (dpkg hold); задание состояний выбора. Дополнительную информацию вы найдете на MAN-странице dpkg.

Управление программным обеспечением с помощью aptitude

Инструменты dpkg и APT существуют уже долгое время и работают хорошо, однако у обоих есть много нюансов, в которых следует разбираться, чтобы правильно использовать эти инструменты. Программа aptitude пытается все упростить, автоматизируя некоторые важные операции с пакетами (например, выполнение apt-get update перед обновлением или установкой), одновременно обеспечивая достаточ-

ную гибкость, чтобы быть полезным. В силу этих причин я рекомендую использовать aptitude в командной строке во всех возможных случаях.

Программа aptitude нацелена на то, чтобы быть как curses-приложением, так и инструментом командной строки. В этом разделе я сосредоточусь на использовании ее как инструмента командной строки. В приведенных далее примерах подробно разбираются способы управления программными пакетами с помощью команды aptitude. Следует отметить, что большинство параметров аналогично используемым с командой apt-get.

ПРИМЕЧАНИЕ----------

Дополнительную информацию о навигации по curses-интерфейсу aptitude и прочих деталях можно получить, заглянув в руководство AptitudeSurvivalGuide (https://help.ubuntu.com/community/AptitudeSurvivalGuide) или Aptitude User's Manual (http://people.debian.org/~dburrows/aptitude-doc/en/), либо введя man aptitude в командную строку.

- Выполнение команды apti tude без параметров запускает curses-интерфейс. Нажмите Ctrl+T для доступа к меню и клавишу Q для выхода:

```
$ sudo aptitude
```

- Эта команда выводит на экран справочные сведения по использованию aptitude:

```
$ aptitude help
```

- Данная команда показывает пакеты, соответствующие определенному *ключевому слову*.

```
$ aptitude search ключевое слово
```

- Эта команда обновляет индексы доступных пакетов из APT-источников:

```
$ sudo aptitude update
```

- Данная команда обновляет все используемые пакеты в соответствии с их последними версиями:

```
$ sudo aptitude upgrade
```

- Чтобы вывести на экран информацию об определенном пакете, установленном или нет, используйте следующую команду:

```
$ aptitude show пакет
```

- Эта команда загружает определенный пакет, но не устанавливает его:

```
$ sudo aptitude download пакет
```

- Данная команда удаляет все загруженные файлы .deb из каталога /var/cache/apt/archi ves:

```
$ sudo aptitude clean
```

- Следующая команда удаляет все устаревшие файлы . deb из каталога /var/cache/apt/archi ves. Это поддерживает текущий кэш, не позволяя ему заполнить все свободное пространство на диске:

```
$ sudo aptitude autoclean
```

- Данная команда устанавливает определенный пакет в системе. Следует отметить, что есть набор параметров для выбора конкретных версий и использования подстановочных символов:

```
$ sudo aptitude install пакет
```

- Следующая команда удаляет определенный пакет из системы:

```
$ sudo aptitude remove пакет
```

- Эта команда обновляет все пакеты в соответствии с их последними версиями, удаляя или устанавливая пакеты по мере необходимости. Параметр upgrade предпочтительнее, чем dist-upgrade:

```
$ sudo aptitude dist-upgrade
```

Обновление пакетов с помощью aptitude

По умолчанию aptitude всегда будет выполнять apt-get update перед обновлением или установкой. Однако вы можете ввести команду для **выполнения только обновления:**

```
$ sudo aptitude update
Ign http://security.ubuntu.com precise-security InRelease
Hit http://security.ubuntu.com precise-security Release.gpg
Ign http://us.archive.ubuntu.com precise InRelease
...
```

Если вы захотите **обновить все пакеты в системе,** то можете ввести с командой aptitude параметр upgrade. Это приведет к установке всех новых пакетов, ожидающих в репозиториях (в приведенном ниже примере новых пакетов в наличии не оказалось):

```
$ sudo aptitude upgrade
Resolving dependencies...
The following NEW packages will be installed:
  linux-image-3.2.0-36-generic{a}
The following packages will be upgraded:
  linux-image-server linux-server
2 packages upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 38.5 MB of archives. After unpacking 149 MB will be used.
Do you want to continue? [Y/n/?] Y
```

Запрос информации о пакетах с помощью aptitude

Вы можете **выполнять поиск посредством aptitude с применением ключевых слов или полных имен пакетов** так же, как и при использовании других инструментов управления пакетами. В приведенном здесь примере осуществляется поиск с использованием ключевого слова mi ni c, в результате чего возвращается информация о пакете minicom и плагине в виде оконного менеджера xfce:

```
$ aptitude search mini c
i   minicom          - friendly menu driven serial communication program
$ aptitude show minicom
Package: minicom
State: not installed
Version: 2.5-2
Priority: optional
Section: universe/comm
...
```

Установка пакетов с помощью aptitude

Ранее вы загружали пакет с использованием apt-get. Здесь вы воспользуетесь aptitude для **загрузки пакета, не устанавливая его:**

```
$ sudo aptitude download minicom
...
Get:1 http://us.archive.ubuntu.com precise/main minicom 2.5-2 [297kB]
Fetched 276 kB in Os (413 kB/s)
```

Если вы хотите **установить пакет minicom,** можете вызвать apti tude следующим образом:

```
$ sudo aptitude install minicom
...
Need to get 111 kB/408 kB of archives. After unpacking
1,503 kB will be used.
Do you want to continue? [Y/n/?] n
```

Если у вас имеется **набор пакетов, которые вы хотите установить,** можете снабдить aptitude подстановочным символом для осуществления выборки. В данном примере я устанавливаю все пакеты, содержащие слово minic (как и при осуществлении поиска с помощью aptitude в приведенном выше примере). Здесь также осуществляется выборка всех зависимостей для каждого пакета с использованием того, что в случае с aptitude называется *обнаружителем совпадений.* Используйте обнаружитель совпадений -п в качестве префикса вашего ключевого слова для установки всех пакетов, содержащих слово minic:

```
$ sudo aptitude install "-nminic"
...
The following NEW packages will be automatically installed:
gcc-4.6-base:1386{a} 11bc6:i 386{a} 1ibcpan-mini-perl
    1ibencode-local e-perl{a}
  libfile-homedir-perl{a} 1ibfile-1isting-perl(a} libfile-which-perl{a}
0  packages upgraded, 37 newly installed, 0 to remove and 0 not upgraded.
Need to get 6.038 kB/6,335 kB of archives. After unpacking
    18.1  MB will oe used.
...
Accept this solution? [Y/n/q/?] n
```

Удаление пакетов с помощью aptitude

Удалять пакеты посредством aptitude так же легко, как устанавливать их. Потребуется лишь передать параметр remove:

```
$ sudo aptitude remove mini com
...
The following packages are unused and will be REMOVED:
  1 rzsz
The following packages will be REMOVED:
  minicom
0 packages upgraded. 0 newly installed, 2 to remove and 0 not upgraded.
Need to get OB of archives. After unpacking 1401kB will be freed.
Do you want to continue? [Y/n/?] Y
```

Очистка кэша от пакетов с помощью aptitude

По мере того как вы будете устанавливать программное обеспечение с помощью инструмента aptitude, он будет загружать файлы . deb и помещать их в каталог /var/cache/apt/archives. Со временем у вас возникнет необходимость **очистить кэш от этих файлов,** воспользовавшись параметром с! еап или, по крайней мере, параметром autocl еап, чтобы высвободить пространство на диске. Если вы заглянете в кэш, то увидите, что там уже есть кое-какие файлы:

```
$ Is /var/cache/apt/archives/
lock
lrzsz_0.12.21-5_amd64.deb
minicom_2.5-2_amd64.deb rsync_3.0.9-lubuntul_amd64.deb
partial
rsync_3.0.9-lubuntul_amd64.deb
```

Для удаления этих файлов посредством команды aptitude вам потребуется лишь прибегнуть к параметру clean или autoclean:

```
$ sudo aptitude clean
```

Повторный ввод команды 1 s покажет, что пакеты действительно были удалены, так что если у вас низкая скорость доступа в Интернет, а загрузка последних обновлений заняла целую неделю, то вам, возможно, следует подумать дважды, прежде чем делать описанное выше, либо воспользоваться параметром autoclean, который обеспечивает удаление только устаревших пакетов.

Полезные комбинации параметров aptitude

Параметр -v **позволяет получать более подробную информацию при выполнении операций посредством aptitude.** Используя его многократно, вы можете иметь

нечто большее, чем обычные сведения, выводимые на экран при выполнении действий. Если вы вызовете aptitude с использованием -v, то увидите контрольную сумму MD5 соответствующего пакета. Это своего рода цифровой отпечаток пальца, который можно использовать для проверки, является ли пакет подлинным и не был ли он поврежден. **Задействовав -w, вы сможете получить еще больше информации:**

```
$ aptitude show -vv minicom
Package: minicom
State: installed
...
Fi1ename: pool/uni verse/m/mi ni com/mi ni com_2.5-2_amd64.deb
MD5sum: c2f403'lblfc6e688a783871cdca9890e
...
```

Вы можете использовать параметр -s в сочетании с командой aptitude, чтобы сообщить ей, что вы желаете **симулировать то, что произошло бы, без выполнения операции на самом деле.** Это можно сделать независимо от того, о каком действии с использованием aptitude идет речь:

```
$ sudo aptitude -s install minicom
The following NEW packages will be installed:
...
Do you want to continue? [Y/n/?] y
Would download/install/remove packages.
```

Если добавить параметр -v к -s, то на экране появится еще более подробный вывод:

```
$ sudo aptitude -vs install minicom
Reading package lists... Done
...
Do you want to continue? [Y/n/?] y
Inst Irzsz (0.12.21-5 Ubuntu:12.04/precise [amd64])
Inst minicom (2.5-2 Ubuntu:12.04/precise [amd64])
Conf Irzsz (0.12.21-5 Ubuntu:12.04/precise [amd64])
Conf minicom (2.5-2 Ubuntu:12.04/precise [amd64])
```

Если вы **не хотите получать приглашение** ответить на вопрос «Хотите продолжить?», то можете заранее предусмотреть ответ на него, добавив параметр -y своей команде:

```
$ sudo aptitude -vs -y install "-nine"
Reading package lists... Done
Building dependency tree
Reading state information... Done
Reading extended state information
Initializing package states... Done
...
```

Будьте очень осторожны, применяя параметр -y, поскольку в случае с aptitude отсутствует возможность отмены внесенных изменений.

Наконец, aptitude -h возвратит справочную информацию, которую вы сможете использовать, если вам потребуется освежить свои знания. Интересно выяснить, что в Ubuntu-версии aptitude кое-чего не хватает:

```
$ aptitude -h
aptitude 0.6.6
...
This aptitude does not have Super Cow Powers.
```

Нет Super Cow Powers? Я любопытен, поэтому передаю aptitude параметр moo:

```
$ aptitude moo
There are no Easter Eggs in this program.
```

«Пасхальные яйца» (Easter eggs) — это элементы, скрытые в программе в качестве сюрприза. Хм, вероятно, вы можете воспользоваться параметром -v для вывода на экран более подробной информации:

```
$ aptitude -v moo
There really are no Easter Eggs in this program.
```

Возможно, вы захотите увидеть еще более подробный вывод. Настоятельно потребуйте у aptitude еще немного информации:

```
$ aptitude -vv moo
Didn't I already tell you that there are no Easter Eggs in this program?
```

Вы начали замечать некую закономерность? Возможно, обеспечение вывода на экран более подробной информации куда-нибудь вас приведет.

Проверка установленных пакетов с помощью debsums

Возможны ситуации, в которых вы засомневаетесь в поведении двоичного файла или пакета, установленного в системе. Он может функционировать некорректно либо вообще не запускаться. Проблемы в виде поврежденных пакетов могут возникать из-за нестабильных сетевых подключений или сбоев электроэнергии. Кроме того, взломщики могут попытаться подменить мощные команды своими версиями, чтобы нанести больше вреда. Соответственно, целесообразно проверять файлы в системе на соответствие информации, находящейся в определенном пакете.

Программа debsums — это утилита для Ubuntu и других систем на основе Debian, которая **сверяет контрольные суммы MD5 всех установленных пакетов** с md5sum-файлами, расположенными в каталоге /var/l ib/dpkg/info.

Для установки этой программы вам потребуется ввести следующую команду:

```
$ sudo aptitude install debsums
```

Датее приведены некоторые из самых полезных параметров, используемых при выполнении debsums. Всю подробную информацию вы сможете найти на MAN-странице, посвященной debsums.

$ debsums -a	*Проверить все файлы, включая конфигурационные, пропускаемые по умолчанию.*
$ debsums -e	*Проверить только конфигурационные файлы пакетов.*
$ debsums -c	*Вывести на stdout только измененные файлы.*
$ debsums -1	*Показать файлы, в случае с которыми отсутствует тс155ит-информация.*
$ debsums -s	*Показать информацию только об ошибках; обо всем остальном никаких сведений' на экран не выводить.*
$ debsums пакет	*Показать пакеты, которые необходимо проанализировать посредством debsums.*

ПРИМЕЧАНИЕ

Для большинства операций вам не потребуется запускать эту утилиту с правами суперпользователя (с помощью sudo). У обычных пользователей может не быть доступа с правом чтения в случае с некоторыми файлами, поэтому потребуется использовать sudo, если вы получите сообщение наподобие следующего: debsums: can't open at file/etc/at.deny (Permission denied) (debsums: не могу открыть файл /etc/at.deny (Отсутствуют необходимые права доступа)).

Если вы введете команду debsums без параметров, она проверит каждый известный ей файл в системе. Ее вывод можно будет перенаправить в файл, если он потребуется вам позднее. Имя файла, которое отобразит команда debsums, будет сопровождаться надписью OK в правой части вывода, если с контрольной суммой MD5 в случае с этим файлом все окажется в порядке. На экран могут также выводиться сообщения наподобие md5sums missing (Контрольные суммы MD5 отсутствуют) касательно определенного файла или слово REPLACED (Заменен), если контрольная сумма MD5 не совпадает.

Следует остерегаться ошибочных результатов. Если вы решите использовать этот инструмент в качестве основного средства оценки позднее, понадобится сконфигурировать все так, как вы хотите, и заново сгенерировать контрольные суммы MD5 для того, что отсутствует или является некорректным. Таким образом, вы будете знать, что обладаете последней информацией.

Приведенная датее команда **сверит все файлы в системе с имеющимися md5sum-файлами.** Как видите, некоторые файлы отсутствуют или заменены. Вам потребовалось бы убедиться, что у системы еще нет проблем с этими файлами, прежде чем вы стати бы заново генерировать контрольные суммы MD5 для всего:

```
$ debsums
/usr/bi n/acpi                                          OK
/usr/share/man/manl/acpi.l.gz                           OK
/usr/share/doc/acpi/README                              OK
/usr/share/doc/acpi/AUTHORS                             OK
...
/usr/share/app-install/i cons/pybli ographi c.png        OK
debsums:    nomdSsums for bsdutiis
debsums:    nomdSsums forbzip2
debsums:    nomdSsums forcdrecord
...
/usr/share/locale-langpack/en_AU/LC_MESSAGES/adduser.mo      REPLACED
/usr/share/locale-langpack/en_AU/LC_MESSAGES/alsa-utils.mo OK
...
```

Если вы захотите сохранить эту информацию в файле, то можете **перенаправить оба потока — stdout и stderr — в файл.** Амперсанд, размещенный в конце строки, заставит команду выполняться в фоновом режиме, благодаря чему вы сможете продолжить работу в интерпретаторе команд:

```
$ debsums &> /tmp/foo &
```

Чтобы **проверить конфигурационные файлы, распространяемые с каждым пакетом,** на предмет наличия изменений, введите debsums с параметром -a:

```
$ debsums -a
/usr/bin/acpi                                   OK
/usr/share/man/manl/acpi.l.gz                   OK
...
```

Если хотите проверить **только конфигурационные файлы, игнорируя все остальные,** используйте параметр -e. Это хороший способ узнать, не отредактировали ли вы случайно конфигурационный файл, вносить изменения в который не собирались. Как видите, некоторые конфигурационные X-файлы были изменены:

```
$ debsums -e
...
/etc/Xll/Xresources/xl1-common                  OK
/etc/Xll/Xsessi on                              FAILED
/etc/Xll/rgb.txt                                OK
/etc/init.d/xll-common                          OK
/etc/Xll/Xsession.d/50xll-common_determine-startup  OK
/etc/Xll/Xsession.d/30xll-common_xresources     OK
/etc/Xll/Xsession.d/20xll-common_process-args   OK
/etc/Xll/Xsession.options                       FAILED
...
```

Команда debsums выдает множество информации, а вы, возможно, захотите **увидеть только измененные файлы.** Для этого следует ввести debsums с параметром -c:

```
$ debsums -c
debsums: no md5sums for binutils
debsums: no md5sums for libaudio
...
```

В случае ввода предыдущей команды на экране отобразятся сообщения касательно файлов, не имеющих исходной тббзит-информации, с которой можно было бы свериться. Вы можете **выполнить проверку на предмет наличия файлов, не имеющих тббвит-информации,** введя debsums с параметром -1:

```
$ debsums -1
binutilslibaudio2...
```

Если вы захотите отобразить с помощью debsums только сведения об ошибках, используйте параметр -s, посредством которого **сообщите debsums, что, за исключением информации об ошибках, никаких других сведений на экран выводить не нужно.**

```
$ debsums -s
debsums: no mdSsums for binutils
debsums: no md5sums for 1ibaudio2
...
```

Чтобы **проверить определенный пакет,** передайте `debsums` имя этого пакета в качестве аргумента:

```
$ debsums coreutils
/bin/cat                                    OK
/bin/chgrp                                  ок
/bin/chmod                                  OK
...
```

Приведенная далее команда проверяет только файлы, которые указаны в тббзит-файле пакета, расположенном в каталоге `/var/Tib/dpkg/info`; таким образом, если в пакете отсутствует тс15яит-файл, вы увидите сообщение об ошибке:

```
$ debsums libaudio2
debsums: no md5sums for 11baudio2
```

Чтобы сгенерировать отсутствующие шббзит-данные для любого пакета, используйте комбинацию из dpkg, утилиты md5sum и небольшого количества кода сценария интерпретатора команд. Сначала введите dpkg -L, чтобы увидеть перечень всех файлов в пакете, о которых знает dpkg. Список, который возвратит dpkg, будет включать и другие строки данных, помимо имен файлов, так что вам потребуется передать по каналу этот вывод дгер и отфильтровать все, что не начинается со знака слеша.

Во второй строке понадобится провести проверку с использованием интерпретатора команд на предмет того, является ли строка вывода dpkg каталогом или файлом (имена каталогов тоже начинаются со знака слеша). Если окажется, что это файл, будет выполнена команда `md5sum` в отношении этой строки вывода, которая на данном этапе должна быть именем файла. Наконец, в третьей строке вам потребуется сохранить весь вывод в текстовом файле с использованием того же соглашения об именовании, которое имеет место в случае с шббзиш-файлами в каталоге /var/1ib/dpkg/info.

В приведенном далее примере показано, как создать соответствующий сценарий для пакета rsync. Однако вы можете выбрать для использования любой пакет, в случае с которым отсутствуют контрольные суммы MD5.

```
$ for file in 'dpkg -L rsync | grep ^/'; do
test -f "$file" && md5sum "$fi1e":
done > /tmp/rsync.md5sums
```

Эта команда позволит вам получить md5sum-6a3y данных, которую вы сможете записать на CI) и использовать для проверки системы. Если контрольные суммы MD5 будут находиться на компакт-диске, то их нельзя случайно удалить, и с ними ничего не случится, если возникнут проблемы с файловой системой на жестком диске. Если вы захотите проверить свои контрольные суммы MD5 позже, можете

воспользоваться командой md5sum с параметром -c, передав ей также имя md5sum-файла данных:

```
$ md5sum -c /tmp/rsync.md5sums
...
/usr/bin/rsync: OK
/usr/share/doc/rsync/examples/rsyncd.conf: OK
/usr/share/doc/rsync/README.gz: OK
/usr/share/doc/rsync/TODO.gz: OK
...
```

Чтобы задействовать файл `rsync.md5sum` в сочетании с `debsums`, вам потребуется внести одно изменение, которое создаст проблемы для `md5sum`, однако необходимо при использовании `debsums`: удалить слеш, стоящий перед именем файла. Вы можете сделать это в текстовом редакторе либо применив еще немного кода сценария интерпретатора команд:

```
$ cat /tmp/rsync.md5sums
302916114c29191cd9c8cb51d67ee60a /usr/bin/rsync
...
```

Чтобы убрать слеш, стоящий перед `/usr/bin/rsync`, можете воспользоваться текстовым редактором или потоковым редактором (sed):

```
$ sed -e 's# /# #g' /tmp/rsync.md5sums > /tmp/rsync.debsums
$ cat /tmp/rsync.debsums
302916114c29191cd9c8cb51d67ee60a usr/bin/rsync
```

Удалив стоящий впереди слеш, вы сможете скопировать `rsync. debsums` в каталог /var/1 `ib/`, и у `debsums` появится возможность использовать его:

```
$ sudo mv /tmp/rsync.debsums /var/1ib/dpkg/info/rsync.md5sums
$ debsums rsync
/usr/bin/rsync                                  OK
/usr/share/doc/rsync/examples/rsyncd.conf       OK
/usr/share/doc/rsync/README.gz                  OK
...
```

Создание пакетов .deb

Путем перестройки файла .deb, применяемого для создания пакета Debian, вы можете изменить его так, чтобы он больше соответствовал вашему способу использования программного обеспечения (например, включив тёбвит-файл). Для начала вам потребуется извлечь содержимое файла . deb, который вы хотите изменить, в рабочий каталог. Затем вы сможете модифицировать дерево файлов и управляющие файлы в соответствии со своими нуждами.

Например, вы могли бы загрузить и извлечь содержимое пакета rsync и управляющие файлы в текущий каталог, введя следующие команды (каталог $ RANDOM в вашем случае будет, конечно, другим):

```
$ aptitude download rsync
```

Далее извлеките содержимое пакета и управляющие файлы из загруженного файла. Каталог SRAND0M можно найти, напечатав /tmp/rsync_ и нажав клавишу Tab:

```
$ sudo dpkg -x rsync_3.0.9-lubuntul_amd64 /tmp/rsync_$RANDOM
S sudo dpkg -e rsync_3.0.9-lubuntul_amd64.deb /tmp/rsync_17197/
```

Теперь перейдите в свой каталог пакета, в который вы извлекли содержимое файла . deb, и оглядитесь. Вы должны увидеть структуру каталога, очень похожую на эту:

```
$ cd /tmp/rsync 17197
$ Is -lart
total 36
-rw-Г--Г--  1 root  root    37 Nov   8   2011  conffiles
-rw-r--r--  1 root  root  1012 Nov   8   2011  control
drwxr-xr-x  4 root  root 4096  Nov   8   2011  etc
drwxr-xr-x  3 root  root  4096 Nov   8   2011  lib
-rw-r--r--  1 root  root  1743 Nov   8   2011  md5sums
-rwxr-xr-x  1 root  root   712 Nov   8   2011  postinst
-rwxr-xr-x  1 root  root   113 Nov   8   2011  postrm
-rwxr-xr-x  1 root  root   494 Nov   8   2011  prerm
drwxr-xr-x  4 root  root 4096  Nov   8   2011  usr
```

Далее вам необходимо сконфигурировать каталог пакета таким образом, чтобы обеспечить соответствие форматам, которые потребуются dpkg для создания файла .deb. Это подразумевает создание подкаталога с именем rsyncJ3.0.9-lcnl. 1/DEBIAN и перемещение в него установочных файлов. Управляющий файл как таковой является отформатированным особым образом файлом, который содержит поля заголовка и содержимого и разбирается инструментами управления пакетами с целью вывода на экран информации о пакете:

```
$ sudo mkdir -p rsync_3.0.9-lcnl.1/DEBIAN
$ sudo mv control conffiles prerm postrm postinst rsyncJ3.0.9-lcnl.l/DEBIAN
```

Вам также потребуется переместить каталоги etc/ и usr/ в каталог rsync_3.0.9-lcnl.1:

```
$ sudo mv usr etc rsync_3.0.9-lcnl.l
```

Если вы сделали все правильно, у вас останется лишь каталог rsync_3.0.9 - lcnl. 1, каталог 1ı b и тй5зит-файл в текущем каталоге.

Теперь переместите созданный вами ранее пк15зиш-файл в свой подкаталог DEBIAN и переименуйте его в md5sums. Благодаря этому debsums получит в свое распоряжение контрольные суммы MD5 для выполнения проверки:

```
$ sudo mv /var/1ib/dpkg/info/rsync.md5sums rsync_3.0.9-lcnl.l/DEBIAN/md5sums
```

Датее отредактируйте управляющий файл, чтобы изменить некоторую информацию. Вы, конечно, не захотите устанавливать свою модифицированную версию rsync, содержащую те же сведения о пакете, что и оригинал. Откройте управляющий файл в vi или другом редакторе и измените строку Version так, чтобы она полностью соответствовала строке, приведенной ниже. Вы заметите, что после слова Version стоит двоеточие; это поле заголовка. Информационное поле следует

сразу за ним. Убедитесь, что после двоеточия имеется пробел, и не добавляйте никаких дополнительных символов возврата каретки или пробелов в код в файле. Подход к форматированию в данном случае очень придирчивый:

```
$ sudo vi rsync_3.0.9-lcnl.l/DEBIAN/control
...
Version: rsync_3.0.9-lcnl.l
...
```

Ниже вы сможете внести дополнения в поле Description. Оно будет отображаться в описаниях при каждом просмотре кем-нибудь подробностей о пакете. Обратите внимание на пробел перед словами fast remote. Он является частью особого форматирования, позволяющей dpkg отличить текст описания от многострочного заголовка. Обязательно поставьте пробел в первом столбце, если описание будет переноситься на следующую строку:

```
...
Description: Modified by CN 2013-02-01 to include md5sums.
 fast remote file copy program (like rep)
...
```

Теперь создайте свой новый пакет, используя dpkg -b и имя каталога управляющего файла, который вы сгенерировали. Вы увидите на экране предупреждение, что Original-Maintainer является полем, определяемым пользователем. Это сообщение можно спокойно проигнорировать.

```
$ sudo dpkg -b rsync_2.6.9 -lcnl.1
warning, 'rsync_2.6.9-lcnl.1/DEBIAN/control' contains user-defined field
'Original-Maintainer'
dpkg-deb: building package 'rsync' in 'rsync_3.0.9-lcnl.1.deb'.
dpkg-deb: ignoring 1 warnings about the control file(s)
```

Итак, у вас появился новый файл .deb и вы можете попросить dpkg отобразить информацию о нем. Просто введите dpkg с параметром -I, и вы увидите сведения о новом пакете:

```
$ dpkg -I rsync_3.0.9-lcnl.l.deb
 new debian package, version 2.0.
 size 1004 bytes: control archive» 712 bytes.
      970 bytes, 21 lines control
 Package: rsync
 Version: 3.0.9-len1.1
...
```

На данном этапе вы могли бы установить пакет rsync. Рассмотренное выше упражнение является главным образом демонстрацией создания пользовательского пакета, для чего вам не потребуется вникать в тонкости системы. Тем не менее в приведенном далее коде показано, что этот пакет установится и будет функционировать как обычный пакет Debian. Вы также захотите, чтобы debsums тоже можно было использовать. Обратите внимание, что dpkg сообщит вам о замене новой версии на более старую:

```
$ sudo dpkg -i rsync_3.0.9-lcnl.1.deb
dpkg - warning: downgrading rsync from 3.0.9-lubuntul_amd64
      to 3.0.9-lcnl.l.
(Reading database ... 88107 files and directories currently installed.)
Preparing to replace rsync 3.0.9-lubuntul_amd64
     (using rsync_3.0.9-lcnl.l.deb) ...
Unpacking replacement rsync ...
Setting up rsync (3.0.9-lcnl.l) ...
```

Теперь утилита debsums располагает пДбзит-файлами для использования при выполнении проверки, и везде, где установлен ваш новый пакет rsync, вот это будет одинаковым:

```
$ debsums rsync
/usr/bin/rsync                                    OK
/usr/share/doc/rsync/examples/rsyncd.conf         OK
/usr/share/doc/rsync/README.gz                    OK
...
```

Вы также можете попросить dpkg показать информацию о вашем пакете rsync, воспользовавшись параметром -l, чтобы убедиться, что установлена его новая версия:

```
$ dpkg -1 rsync
...
ii rsync 3.0.9-lcnl.l Modified by CN 2013-02-01 to include md5sums.
```

ПРИМЕЧАНИЕ

Дополнительные сведения о создании файлов .deb можно найти в практическом руководстве Debian Binary Package Building HOWTO (http://tldp.org/HOWTO/Debian-Binary-Package-Building-HOWTO). MAN-страница dpkg-deb тоже является хорошим источником информации на тему создания пакетов .deb.

Резюме

Программное обеспечение для Ubuntu и других дистрибутивов на основе Debian заключено в пакеты формата .deb. Для начальной инсталляции Ubuntu применяется установщик Ubiquity. Используя меню загрузки, вы можете загрузить полноценную среду Ubuntu и выполнять установку оттуда либо запустить Ubuntu с CD-ROM.

Для инсталляции дополнительного программного обеспечения можно использовать утилиты aptitude и APT, позволяющие получать пакеты из онлайн-репозиториев. Для установки пакетов, расположенных локаяьно на вашем жестком диске, а также создания пользовательских пакетов Debian вы можете задействовать утилиту dpkg. APT, aptitude и dpkg позволяют выполнять запросы касательно программного обеспечения. Для проверки установленных пакетов можно прибегнуть к утилитам debsums и md5sum.

3 Использование интерпретатора команд

В этой главе:

- получение доступа к интерпретатору команд;
- использование истории команд и завершения ввода команд;
- задание псевдонимов;
- получение доступа суперпользователя;
- написание простых сценариев интерпретатора команд.

Использование интерпретатора командного языка (обычно называемого просто *интерпретатором команд)* восходит к ранним годам существования первых UNIX-систем. Помимо очевидного их применения для выполнения команд, интерпретаторы команд обладают множеством встроенных параметров, таких, например, как переменные среды, псевдонимы, а также большим количеством функций для программирования.

Хотя интерпретатор команд, чаще всего используемый в Linux-системах, называется *Bourne Again Shell (bash),* также доступны и другие (такие как sh, csh, ksh и tcsh). Во многих случаях эти интерпретаторы команд (скажем, sh) в действительности являются символьными ссылками на программы других интерпретаторов команд, например bash. В Ubuntu Linux sh является символьной ссылкой на `/bin/dash`. Интерпретатор команд sh важен, поскольку в большинстве сценариев интерпретатора команд он определяется как инструмент, используемый для запуска сценариев. Что касается интерактивного использования, то интерпретатором команд, применяемым по умолчанию, является bash.

В этой главе представлена информация, которая поможет вам в использовании интерпретаторов команд Linux в целом и bash в частности.

Окна терминала и доступ к интерпретатору команд

Самый распространенный способ получения доступа к интерпретатору команд из графического интерфейса Linux — использование окна терминала. Из графического интерфейса часто можно получить доступ к виртуальным терминалам, чтобы затем добраться до интерпретатора команд. При отсутствии графического интер-

фейса, используя вход в систему в текстовом режиме, вы, зайдя в систему, обычно будете попадать сразу в интерпретатор команд.

Окна терминала

Чтобы открыть окно терминала при использовании Unity (рабочий стол Ubuntu по умолчанию), выберите значок **Dashboard** (Панель управления), а затем начните печатать текст в поле поиска. Найдите появившийся на панели управления значок **Terminal** (Терминал) и выберите его. После этого откроется окно **gnome-terminal,** в котором вы увидите приглашение интерпретатора команд bash. На рис. 3.1 приведен пример окна **gnome-terminal.**

Команды, показанные на рис. 3.1, говорят о том, что *текущим интерпретатором* команд является bash (/bi n/bash), *текущим пользователем* — пользователь настольного компьютера, открывший окно **(chris),** а *текущим каталогом* — домашний каталог пользователя (/home/chris). Имя пользователя **(chris)** и имя хоста (ubuntutb) отображаются в строке заголовка и приглашении.

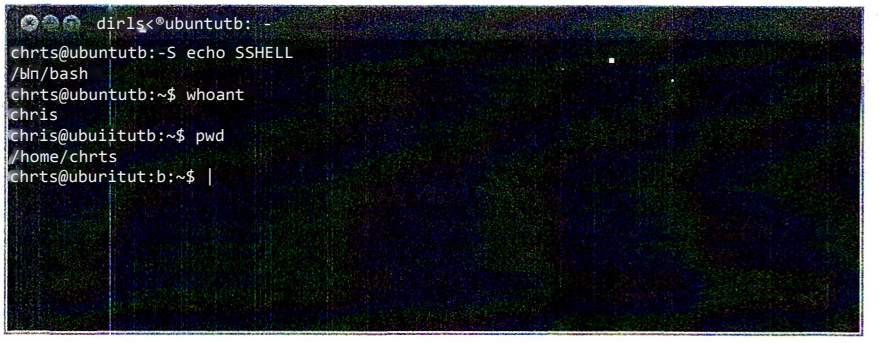

Рис. 3.1. Ввод команд интерпретатора команд в окне gnome-terminal

Окно **gnome-terminal** не только позволяет вам получить доступ к интерпретатору команд, но и предоставляет соответствующие элементы для управления вашими интерпретаторами команд. Вот несколько примеров:

- выберите **File** (Файл) ► **Open Tab** (Открыть вкладку), чтобы **открыть другой интерпретатор команд на новой вкладке;**
- выполните команду **File** (Файл) ► **Open Terminal** (Открыть терминал), чтобы **открыть новое окно терминала;**
- выберите **Terminal** (Терминал) ► **Set Title** (Задать заголовок), чтобы **задать новый заголовок в строке заголовка.**

Вы также можете использовать сочетания клавиш для работы с окном терминала. Откройте **интерпретатор команд на новой вкладке,** нажав **Shift+Ctri+T; откройте новое окно терминала** с помощью **Shift+Ctrl+N; закройте вкладку,** нажав **Shift+Ctrl+W; закройте окно терминала** посредством **Shift+ctrl+Q.** Выделите текст и скопируйте его, нажав **Shift+Ctrl+C,** а затем **вставьте его в то же самое или другое окно,** нажав **Shift+Ctrl+V** либо среднюю кнопку мыши.

ПРИМЕЧАНИЕ

В большинстве приложений наподобие текстового процессора LibreOffice функцию копирования вызывает нажатие клавиш Ctrl+C, а не Shift+Ctrl+C, а функцию вставки — Ctrl+V, а не Shift+Ctrl+V. Комбинация клавиш Ctrl+C имеет специальное назначение в окне интерпретатора команд (обеспечивает отправку сигнала программе, который обычно приводит к завершению ее работы), поэтому в окне gnome-terminal нужные функции графического рабочего стола задействуются с использованием клавиши Shift в качестве модификатора.

Другие сочетания клавиш для управления окнами терминала предусматривают использование **F11** для переключения окна в *полноэкранный режим* и обратно. Нажмите **Ctrl+Shift++,** чтобы **увеличить** (сделать текст крупнее), или **Ctrl+-** (это **Ctrl** и знак минуса), чтобы **уменьшить** (сделать текст мельче). **Переключаться между вкладками** можно с помощью **Ctrl+Page Up** и **Ctrl+Page Down** (соответственно предыдущая и следующая вкладка) либо с использованием **Alt+1, Alt+2, Alt+3** и т. д. для перехода на первую, вторую, третью (и т. д.) вкладку. Нажмите **Ctrl+D,** чтобы выйти из интерпретатора команд, что приведет к закрытию текущей вкладки или всего окна терминала (если соответствующая вкладка окажется последней).

Окно **gnome-terminal** также поддерживает профили (выберите **Edit** (Редактировать) ► **Profiles** (Профили)). Одни настройки профилей — косметические (позволяют выделять текст жирным шрифтом, обеспечивать мерцание курсора, звуковые сигналы терминала, использовать цвета, изображения и прозрачность). Другие настройки — функциональные. Например, по умолчанию терминал позволяет прокручивать 512 строк, сохраняя их в памяти. Некоторые пользователи хотят прокручивать еще дальше и согласны выделить на это больше памяти.

Если вы будете запускать **gnome-terminal** вручную, то сможете добавить параметры. Вот несколько примеров:

$ gnome-terminal -x alsamixer	*Запустить вместе с alsamixer (для выхода нажмите Esc).*
$ gnome-terminal --tab --tab --tab	*Запустить терминал с тремя открытыми вкладками.*
$ gnome-terminal --geometry 80x20	*Запустить терминал размером 80 символов на 20 строк.*
$ gnome-terminal --zoom=2	*Запустить терминал с более крупным шрифтом.*

Помимо окна **gnome-terminal,** есть окна множества других терминалов, например xterm (базовый эмулятор терминала, сопутствующий системе X Window System), aterm (эмулятор терминала, созданный по образцу эмулятора Afterstep X VT VT102) и konsole (эмулятор терминала, поставляемый вместе с рабочим столом KDE). Проект рабочего стола Enlightenment предлагает терминал eterm (который включает такие функции, как журналы сообщений на фоне экрана).

Работа с виртуальными консолями

При загрузке Ubuntu в многопользовательском режиме (уровень выполнения — 2, 3 или 5) создается шесть виртуальных консолей (именуемых начиная с ttyl

и до tty6) с входом в систему в текстовом режиме. Если используется рабочий стол X Window System, то X, вероятно, функционирует в виртуальной консоли 7. Если X не выполняется, то, скорее всего, вы смотрите на виртуальную консоль 1.

Вы можете **переключиться на другую виртуальную консоль** из X, нажимая **Ctrl+Alt+Fl, Ctrl+Alt+F2** и т. д. до **F6.** Из текстовой виртуальной консоли можно переключиться, используя **Alt+Fl, Alt+F2** и т. д. Нажмите **Alt+F7,** чтобы вернуться в GUI-интерфейс X. Каждая консоль позволяет вам входить в систему с использованием разных учетных записей пользователей. Переключение с целью заглянуть в другую консоль не повлияет на запущенные процессы в любой из них. При переключении на виртуальную консоль с 1 по 6 вы будете видеть примерно следующее приглашение войти в систему:

```
Ubuntu 12.04.2 LTS ubuntutb tty2
ubuntutb login:
```

Каждым виртуальным терминалом управляют отдельные процессы getty. Введите приведенную ниже команду, чтобы увидеть, как выглядят процессы getty, прежде чем входить в любой из виртуальных терминалов:

```
$ ps awx |    grep -v  grep  | grep getty
 4366 tty4      Ss+     0:00   /sbin/getty  -8  38400  tty4
 4367 tty5      Ss+     0:00   /sbin/getty  -8  38400  tty5
 4372 tty2      Ss+     0:00   /sbin/getty  -8  38400  tty2
 4373 tty3      Ss+     0:00   /sbin/getty  -8  38400  tty3
 4374 ttyl      Ss+     0:00   /sbin/getty  -8   38400  ttyl
 4375 tty6      Ss+     0:00 /sbin/getty     -8 38400 tty6
```

После того как я войду в первую консоль, getty обработает мой вход в систему, а затем запустит интерпретатор команд bash:

```
$ ps awx | grep -v grep | grep tty
 4366 tty4    Ss+  0:00  /sbin/getty     -8 38400tty4
 4367 tty5    Ss+  0:00  /sbin/getty     -8 38400tty5
 4372 tty2    Ss    0:00  /bin/login --
 4373 tty3    Ss+  0:00  /sbin/getty     -8 38400tty3
 4374 ttyl    Ss+  0:00  /sbin/getty     -8 38400 ttyl
 4375 tty6    Ss+  0:00  /sbin/getty     -8 38400tty6
 7214    tty2 S+   0:00  -bash
 1153 tty7 Ss+ 5:59 /usr/bin/X :0 -auth /var/run/1ightdm/root/:0
      -nolisten tcp vt7 -novtswitch -background none
```

Конфигурационные файлы виртуальных консолей располагаются в каталоге /etc/init. Для каждой виртуальной консоли предусматривается сценарий, например, ttyl. conf для консоли ttyl, tty2. conf для консоли tty2 и т. д.

ПРИМЕЧАНИЕ----------

В других версиях Linux конфигурационные настройки консолей хранятся в одном файле — /etc /inittab. Демон init использует /etc/inittab в качестве своего конфигурационного файла. В Ubuntu Linux, с другой стороны, задействуется версия init из пакета upstart, которая использует каталог /etc/init для размещения своих конфигурационных файлов.

Интерпретатор команд

Когда вы откроете интерпретатор команд (посредством либо входа в систему в текстовом режиме, либо окна терминала), его среда сконфигурируется исходя из того, какой пользователь запустил этот интерпретатор команд. Настройки интерпретатора команд bash для интерпретаторов команд всех пользователей располагаются в нескольких файлах. Вы можете сделать так, чтобы ваши собственные версии этих файлов переопределяли системные настройки. Эти настройки содержатся в файлах двух типов — файлах запуска и файлах инициализации.

bash задействует файлы запуска в случае с любым интерпретатором команд, который является тем, что назначается при входе в систему. Эти файлы определяют настройки, которые будут применяться при каждом вашем входе в систему, bash задействует файлы инициализации в случае с интерпретаторами команд, которые работают в интерактивном режиме, то есть без запуска сценария интерпретатора команд.

bash применяет общесистемный файл запуска /etc/profile, а также некоторые файлы с точкой из домашнего каталога пользователя для индивидуальных настроек (при наличии таковых): ,bash_profile, .bashjogin и .profile. Он также использует расположенные в каталоге /etc/profi 1 e. d сценарии, имена которых оканчиваются на . sh.

bash задействует общесистемный файл инициализации /etc/bash. bashrc, а также файл .bashrc из вашего домашнего каталога (для индивидуальных настроек). Эти файлы задействуются при каждом новом открытии интерпретатора команд bash.

ПРИМЕЧАНИЕ----------

В других версиях Linux общесистемные файлы хранятся в /etc/bashrc.

При выходе из интерпретатора команд, назначаемого при входе в систему (например, из виртуальной консоли), будут выполнены все команды, указанные в вашем файле ~/. bashjogout. Модификация настроек в этих файлах приведет к перманентному изменению настроек интерпретатора команд пользователя, однако не повлияет на интерпретаторы команд, которые уже запущены (прочие интерпретаторы команд используют другие конфигурационные файлы).

Есть множество способов, посредством которых вы можете просматривать и изменять среду своего интерпретатора команд. Один из главных заключается в изменении того, каким пользователем вы являетесь, — в частности, чтобы вы стали суперпользователем (об этом будет рассказано в разделе «Получение полномочий суперпользователя» далее в этой главе).

История bash

Bourne Again Shell (bash) — это интерпретатор команд, по умолчанию используемый в большинстве современных Linux-систем, а также во многих других системах, например Mac OS X. В bash, как и в других интерпретаторах команд, имеется такая встроенная функция, как история, позволяющая вам просматривать, изменять и повторно использовать команды, которые выполнялись ранее.

Она может оказаться очень полезной, поскольку многие команды Linux длинные и сложные.

При запуске bash он считывает файл ~/. bash_hi story и загружает его в память. Для этого файла задается значение SHISTFILE.

ПРИМЕЧАНИЕ

Дополнительные сведения о том, как работать с переменными среды интерпретатора команд вроде $HISTFILE, вы найдете в разделе «Использование переменных среды» далее в этой главе.

Во время сеанса bash команды добавляются в историю, которая содержится в памяти. При выходе из bash история, находящаяся в памяти, записывается обратно в файл . bash_hi story. **Количество команд, размещаемых в истории во время сеанса bash,** задается посредством SHISTFILE, а **количество команд, фактически сохраняемых в файле истории,** — посредством SHISTFILESIZE:

```
$ echo SHISTFILE SHISTSIZE SHISTFILESIZE
/home/chris/.bash_history 1000 2000
```

Чтобы **просмотреть всю историю,** введите history. Для **отображения определенного количества предыдущих команд в истории** укажите после history нужное число. Приведенный далее код отобразит на экране пять предыдущих команд из вашей истории:

```
$ history 5
975   mkdir extras
976   mv *doc extras/
977   is -CF
978   vi house.txt
979   history 5
```

Для **перемещения среди команд** в вашей истории используйте клавиши t и *I*. Как только команда отобразится на экране, вы сможете воспользоваться клавиатурой для **редактирования текущей команды** так же, как и любой другой, — посредством клавиш ←, →, **Delete, Backspace** и т. д. Вот некоторые другие способы повторного вызова и выполнения команд из вашей истории bash:

```
$ !!                 Выполнить предыдущую команду
$ 1997               Выполнить команду номер 997 из истории
ls -CF
$ 1997 *doc          Присоединить *doc к команде 997 из истории
ls -CF *doc
$ !?CF?              Выполнить предыдущую строку команд, содержащую строку CF
ls -CF *doc
$ ! 1 s              Выполнить предыдущую команду Is
ls -CF *doc
$ П s:s/CF/1         Выполнить предыдущую команду Is, заменив CF на 1
ls -1 *doc
```

Еще один способ **отредактировать историю команд** — применение команды fc. Используя fc, откройте выбранную команду, которая имеется в истории, с помощью редактора nano. Напечатайте в nano необходимые изменения, а затем

нажмите **Ctrl+O** и **Ctrl+X,** чтобы сохранить их. Перейдите в другой редактор, задав соответствующее значение для переменной FCEDIT (к примеру, export FCEDIT=gedi t) либо в строке команды fc. Например:

```
$ fc 978                    Отредактировать команду номер 978, а затем выполнить ее
$ fc                        Отредактировать предыдущую команду, а затем выполнить ее
$ fc -e /usr/bin/vim 989 Использовать vim для редактирования команды 989
```

Используйте **Ctrl+R** для **поиска строки в истории.** Например, если нажать **Ctrl+R,** а затем ввести строку ss, результат будет следующим:

```
# <Ctrl+r>
(reverse-i-search)'ss': sudo /usr/bin/less /var/1og/messages
```

Нажмите несколько раз комбинацию клавиш **Ctrl+R** для **обратного поиска в своем списке истории** других случаев использования строки ss.

ПРИМЕЧАНИЕ----------

По умолчанию для редактирования истории команд bash используются команды в стиле Emacs. Если вы предпочитаете редактор vi, можете прибегнуть к редактированию своей истории в стиле vi, воспользовавшись командой set, чтобы задать vi в качестве редактора. Для этого введите следующее: set -o vi. Добавьте эту строку в свой файл .bashrc, как было описано ранее, чтобы vi задействовался как ваш редактор истории при каждом открытии интерпретатора команд.

Завершение ввода в командной строке

Вы можете нажимать клавишу **Tab** для завершения ввода информации различного рода в командной строке. Вот несколько примеров, в которых демонстрируется ввод частичного имени с последующим нажатием **Tab,** чтобы **bash попытался завершить ввод нужной вам информации** в командной строке:

```
$ ifc<Tab>                  Завершение ввода команды: завершает ввод команды в
                            виде ifconfig
$ cd /home/ch<Tab>          Завершение ввода имени каталога с файлами:
                            завершает ввод имени каталога в виде /home/chris
$ cd ~jo<Tab>               Завершение ввода пути к домашнему каталогу
                            пользователя: завершает ввод в виде /home/john
$ echo $PA<Tab>             Завершение ввода имени переменной среды: завершает
                            ввод в виде $PATH
$ ping @<Tab><Tab>          Завершение ввода пути к hosts: показать
                            хосты из /etc/hosts
(Pdavinci .example.com      (Pri tchi e. exampl e. com (Pthompson. exampl e. com
@1 ocalhost                 (Pzooey
```

Перенаправление stdin и stdout

При вводе команды в интерпретаторе команд она выполняется в интерактивном режиме. Результирующий процесс обладает двумя потоками вывода — stdout для нормального командного вывода и stderr для вывода ошибок.

В примере ниже показано, что если /tmpp не будет найден, то сообщение об ошибке будет выведено на stderr, а вывод из листинга /tmp (который будет найден) пойдет на stdout:

```
$ Is /tmp /tmpp
Is: /tmpp: No such file or directory
/tmp/:
gconfd-chris keyring-b41WuB keyring-ItEWbz mapping-chris orbit-chris
```

По умолчанию весь вывод направляется на экран. Используйте знак «больше» (>) для **направления вывода в файл.** Вернее, вы можете направить стандартный поток вывода (используя >) или стандартный поток ошибок (используя сочетание 2>) в файл. Вот примеры:

```
$ Is /tmp /tnwnp > output.txt
Is: /tmpp: No such file or directory

$ Is /tmp /tmmp 2> errors.txt
/tmp/:
gconfd-chris keyring-b41WuB keyring-ItEWbz mapping-chris orbit-chris
$ Is /tmp /tmmp 2> errors.txt > output.txt

$ Is /tmp /trrnip > everything.txt 2>&1
```

В первом примере stdout перенаправляется в файл output.txt, а srderr по-прежнему направляется на экран. Во втором примере srderr (поток 2) направляется в errors. txt, в то время как stdout направляется на экран. В третьем примере объединены первые два примера. В последнем примере оба потока направляются в файл everything.txt. **Чтобы добавить данные в файл,** а не перезаписывать его, используйте два знака «больше»:

```
$ Is /tmp » output.txt
```

Если вы хотите никогда не видеть поток вывода, то можете просто **направлять его в специальный файл-битоприемник** (/dev/nul 1). В этом случае stderr будет отбрасываться, а sdtout — отображаться:

```
$ Is /tmp 2> /dev/nul1
```

ПРИМЕЧАНИЕ----------

Другой ситуацией, в которой вам может понадобиться перенаправить stderr, является случай, когда вы будете выполнять задания с использованием crontab. Вы могли бы перенаправить stderr в почтовое сообщение, которое будет отослано владельцу crontab. Таким образом, все сообщения об ошибках могут отправляться лицу, выполняющему задания.

Подобно тому как вы можете направить стандартный вывод от команды, у вас также есть возможность **направить стандартный ввод к команде.** Например, приведенная далее команда отправляет по электронной почте файл /etc/hosts пользователю с именем **chris** в локальной системе:

```
$ mail chris < /etc/hosts
```

Используя каналы, вы можете **перенаправить вывод от одного процесса к другому,** а не только в файлы. Вот пример, где вывод команды 1 s передается по каналу команде sort -г для сортировки этого вывода в обратном порядке:

```
$ Is /tmp | sort -г
```

В следующем примере **используется сочетание канала и перенаправления** (stdout команды 1 s сортируется, а stderr сбрасывается в битоприемник):

```
$ Is /tmp/ /tmmp 2> /dev/nul1 | sort -г
```

Каналы можно использовать для передачи множества вещей:

```
$ dpkg-query -1 | grep -i sql | wc -1
$ ps auwx | grep firefox
$ ps auwx I less
$ whereis -m bash | awk '{print $2}'
```

Первая строка команд в приведенном выше примере отображает все установленные пакеты, берет те из них, в именах которых есть sql (независимо от регистра) и подсчитывает количество оставшихся строк (фактически считает количество пакетов, в именах которых присутствует sql). Вторая строка команд отображает процессы Firefox, взятые из длинного списка процессов (при этом предполагается, что браузер Firefox запущен), а также все процессы, в строках команд которых упоминается слово firefox, как, например, в команде, используемой в этом примере. Третья строка команд позволяет вам постранично просмотреть список процессов. Последняя строка отображает слово bash:, за которым следует путь к MAN-странице bash, а затем выводит на экран только путь к соответствующей MAN-странице (второй элемент в строке).

Используя обратные одинарные кавычки, вы **можете выполнить сначала одну часть строки команд, а затем передать ее вывод остальной части строки команд.** Вот примеры:

```
$ dpkg-query -S 'which ps'
$ Is -1 'which bash'
```

Первая строка команд в предыдущем примере находит полный путь команды ps, а таюке пакет, содержащий эту команду ps. Вторая строка команд находит полный путь к команде bash и генерирует длинный список (Is -1), касающийся этой команды.

Более продвинутый и эффективный способ **взять вывод одной команды и передать его в качестве параметров другой команде** заключается в использовании команды xargs. Например:

```
$ Is /bin/b* | xargs /usr/bin/dpkg-query -S
```

Вы можете снабдить xargs параметром -t для генерирования подробного вывода строки команд перед выполнением этой команды. Вы также можете сделать так, чтобы команда xargs передала каждую строку вывода от 1 s в качестве ввода отдельным командам dpkg-query. Скобки {} используются в качестве заполнителя строки:

```
$ Is /bin/b* | xargs -t -I{} dpkg-query -S {}
dpkg-query -S /bin/bash
```

```
bash: /bin/bash
dpkg-query -S /bin/bunzip2
Dzi p2: /b‘in/burizíp2
dpkg-query -S /bin/bzcat
bzip2: /bin/bzcat
dpkg-query -S /bin/bzcmp
bzip2: /bin/bzcmp
dpkg-query -S /bin/bzdiff
bzip2: /bin/bzdiff
dpkg-query -S /bin/bzegrep
bzip2: /bin/bzegrep
dpkg-query -S /bin/bzexe
bzip2: /bin/bzexe
dpkg-query -S /bin/bzfgrep
bzip2: /bin/bzfgrep
dpkg-query -S /bin/bzgrep
bzip2: /bin/bzgrep
dpkg-query -S /bin/bzip2
bzip2: /bIn/bzlp2
dpkg-query -S /bin/bzip2recover
bzip2: /bin/bzip2recover
dpkg-query -S /bi n/bzless
bzi p2: /bI n/bzl ess
dpkg-query -S /bin/bzmore
bzip2: /bin/bzmore
```

Как вы можете видеть из вывода, отдельные команды dpkg-query -S выполняются в случае с каждым параметром, передаваемым 1 s.

Псевдоним

Для **задания и просмотра псевдонимов** используется команда alias. Некоторые псевдонимы уже определены в общесистемных или пользовательских файлах инициализации командного процессора, о которых шла речь ранее. Вот как можно **отобразить все псевдонимы, заданные на текущий момент:**

```
$ alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --co1or=auto’
alias grep='grep --color=auto'
alias l=‘ls -CF’
alias la=’ls -A’
alias 11='ls -alF’
alias 1s=’1s --color=auto
```

Следует отметить, что некоторые псевдонимы используются просто как способ добавить параметры к поведению команды по умолчанию (например, задается 1 s --col or=auto, чтобы вывод команды 1 s всегда отображался в цвете).

Вы можете **определить свои собственные псевдонимы для текущей версии bash,** как показано далее:

```
$ alias lala='1s -la'
```

Добавьте эту строку в свой файл ~/. bashrc, чтобы определение имело место при каждом новом сеансе bash. **Удалите псевдоним из текущего сеанса bash** с использованием команды unalias следующим образом:

```
$ unalias lala        Удалить псевдоним, заданный ранее для команды lala
$ unalias -a          Удалить все псевдонимы, заданные для команд
```

Наблюдение за командами

Если вы захотите понаблюдать за командой, вывод которой меняется, используйте команду watch. Например, чтобы проследить, какая средняя нагрузка создается в вашем случае, вы можете воспользоваться следующим:

```
$ watch 'cat /proc/loadavg'
```

watch будет запускать повторное выполнение команды cat каждые две секунды. Для завершения выполнения команды нужно нажать Ctrl+C. Чтобы изменить частоту обновления на 10 секунд, введите вот это:

```
$ watch -п 10 'Is -Г
```

Чтобы выделить разницу между обновлениями на экране, введите следующее:

```
$ watch -d 'Is -Г
```

Нажмите Ctrl+C, чтобы завершить выполнение команды watch. Следует отметить, что для выделения различий необходимо, чтобы файлы изменялись.

Наблюдение за файлами

Вы можете использовать команду watch для наблюдения за размером файлов. Например, чтобы следить за большим ISO-файлом с именем mydownload.iso по мере его загрузки, используйте следующую команду:

```
$ watch 'Is -1 mydownload.iso'
```

Для наблюдения, как содержимое обычного текстового файла увеличивается в размерах с течением времени, можно воспользоваться командой tail. Например, вы можете проследить за добавлением сообщений в файл /var/1 og/messages и отобразить их на экране с помощью команды, приведенной далее:

```
$ sudo tail -f /var/1og/messages
```

Нажатие Ctrl+C приведет к завершению выполнения команды tail.

Получение полномочий суперпользователя

Открыв интерпретатор команд, вы сможете вводить команды, а также обращаться к файлам и каталогам в соответствии со своим идентификатором пользователя/группы и правами доступа, заданными для этих компонентов. Доступ ко многим системным параметрам есть только у суперпользователя.

Есть три основных способа получить полномочия суперпользователя:

- выполнить одну команду с привилегиями суперпользователя с помощью sudo;
- войти в систему как суперпользователь;
- временно стать суперпользователем с помощью команды su.

Ubuntu Linux предоставляет пользователям возможность выполнять команду su. Фактически пароль для суперпользователя даже не задается по умолчанию, поэтому в большинстве случаев для выполнения административной команды (например, useradd для добавления нового пользователя) вы указывали бы перед ней команду sudo. Например:

```
$ sudo useradd -m joe Позволяет суперпользователю добавить
                      нового пользователя с именем joe
```

В Ubuntu по умолчанию установлены ограничения, в силу которых войти в систему с правами доступа суперпользователя нельзя, поэтому Ubuntu также не позволяет использовать команду su, которая обычно применяется в других Linux-системах, чтобы стать суперпользователем.

Если вам потребуется выполнить последовательность команд как суперпользователю, вы можете ввести приведенную далее команду, чтобы открыть интерпретатор команд от имени суперпользователя:

```
$ sudo bash              Открыть интерпретатор команд от имени суперпользователя
[sudo] password for chris:
'kirfrk'kirk'k
#
```

Если решите, что вам нужно добавить пароль для суперпользователя (что позволит входить в систему как суперпользователю или применять команду su, чтобы временно стать суперпользователем), вы сможете сделать это также с помощью команды sudo:

```
$ sudo passwd root       Задать пароль для суперпользователя
```

Однако большинство пользователей настольных компьютеров с Ubuntu просто применяет sudo и никогда не задает пароль для суперпользователя.

Делегирование полномочий с помощью sudo

Команда sudo позволяет четко делегировать полномочия пользователям, не обладающим правами доступа суперпользователя. Инструмент sudo — отличное средство для предоставления особых привилегий, когда пользователей множество, и протоколирования всего, что ваши пользователи будут делать, обладая этими привилегиями. Если не задано иное, sudo будет выполняться от имени суперпользователя. В Ubuntu Linux для выполнения привилегированных команд предназначена команда sudo, а не su. Конфигурационные настройки sudo можно найти в файле /etc/sudoers.

ВНИМАНИЕ----------

Никогда не редактируйте этот файл в обычном текстовом редакторе. Вместо этого всегда пользуйтесь командой visudo. Она откроет файл sudoers в редакторе, временно заблокирует его, не позволяя другим использовать этот файл, и проведет проверку на ошибки перед тем, как вы завершите работу.

Доступ к файлу /etc/sudoers ограничен, поэтому вам потребуется использовать команду sudo, что отредактировать его. Например:

```
$ sudo visudo
```

Команда `visudo` запускает редактор — по умолчанию им является nano, о котором уже говорилось ранее.

Если заглянете в файл sudoers, входящий в состав вашего дистрибутива, то увидите разные пустые секции, разделенные комментариями, и один активный оператор:

```
root ALL=(ALL:ALL) ALL
```

Это означает, что суперпользователю разрешено выполнять любую команду на любом хосте от имени любого пользователя.

В Ubuntu Linux добавлено приведенное далее, чтобы все пользователи, входящие в группу `admin` или `sudo` (в соответствии с указанным в файле `/etc/group`), могли получить привилегии суперпользователя:

```
# Члены группы admin могут получить привилегии суперпользователя
35admin ALL=(ALL) ALL
# Разрешить членам группы sudo выполнять любую команду
i&sudo ALL= (ALL:ALL) NOPASSWD: ALL
```

Если пользователь из группы sudo запросит права суперпользователя, он получит доступ к ним без ввода пароля. Вы можете модифицировать соответствующую строку sudoers, чтобы привилегированному пользователю не приходилось вводить пароль (его собственный пароль пользователя) при выполнении sudo, как показано далее:

```
&sudo ALL=(ALL:ALL) ALL
```

После установки Ubuntu Linux созданная вами учетная запись была автоматически добавлена в группу sudo. Чтобы дополнительные пользователи смогли получить привилегии суперпользователя, вы можете просто добавить их в группу sudo в файле /etc/group. Например, чтобы предоставить права sudo имеющемуся пользователю joe, вы можете модифицировать соответствующую строку в /etc/group, чтобы она приобрела следующий вид:

```
Sudo:x:27:chris,joe
```

У вас также есть возможность предоставить пользователю ограниченные привилегии суперпользователя вместо указанных в файле /etc/sudoers. К примеру, вы можете добавить приведенную ниже строку, задав в качестве значения первого поля учетную запись в своей системе:

```
joe ALL= /usr/bin/less /var/1og/messages
```

ПРИМЕЧАНИЕ

Предыдущая настройка позволит соответствующему пользователю выполнять команду less с привилегиями суперпользователя. Это предполагает угрозу безопасности, поскольку команда less может позволить этому пользователю получить больше информации о системе путем изучения других системных файлов.

Теперь пользователь с именем **joe** (или любой другой добавленный вами) сможет сделать следующее:

```
$ sudo /usr/bin/less /var/1og/messages
Password:
```

Когда пользователь **joe** введет свой пароль, он сможет постранично просмотреть файл / var/1 og/messages. В тот момент также задается временная метка. Следующие пять минут (по умолчанию) у этого пользователя будет возможность задействовать приведенную выше команду, при этом он не получит приглашение ввести пароль, чтобы выполнить ее.

Тем не менее обычно вам следует добавлять таких пользователей в sudoers или группу admin и не создавать отдельные записи в файле /etc/sudoers.

Каждое выполнение sudo будет регистрироваться в /var/1 og/secure:

```
Feb 24 21:58:57 local host sudo: joe : TTY=pts/3 ; PW0=/home/joe ;
USER=root; COMMAND=/usr/bin/less /var/log/messages
```

Далее добавьте эту строку в /etc/sudoers:

```
joe          serverl=(chris) /bin/ls /home/chris
```

Теперь пользователь с именем **joe** сможет сделать следующее:

```
$ sudo -u chris /bin/ls /home/chris
```

Данная команда sudo выполняется от имени пользователя **chris,** работая только на хосте serverl. В некоторых организациях осуществляется централизованное управление и развертывание файла /etc/sudoers на всех хостах, поэтому целесообразно указать права доступа к sudo на определенных хостах.

Команда sudo также поддерживает назначение псевдонимов, использование предопределенных групп пользователей, команд и хостов. Чтобы увидеть соответствующие примеры, загляните в файл /etc/sudoers в своей Linux-системе.

Команда su

Если в какой-то момент вы решите добавить пароль для учетной записи суперпользователя, открыв интерпретатор команд от имени обычного пользователя, можете воспользоваться командой su (super user – «суперпользователь»), **чтобы стать суперпользователем.** Вы также можете указать команду su с целью стать другим пользователем, не являющимся суперпользователем. Далее описывается, как работает команда su.

Просто задействовав su, как показано в следующем коде, вы не получите доступа к интерпретатору команд, назначаемому при входе в систему, со средой суперпользователя:

```
S su
Password: *****
# echo $PATH
/us r/1 oca 1/sbin:/usr/local/bin:/usr/sbi n:/usr/bin:/sbin:/bin:/usr/games
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/bin:
/usr/bi n: /bin: /usr/XHR6/bin: /home/joe/bin
```

После выполнения su пользователь по-прежнему будет располагать PATИ пользователя с именем **joe.** Чтобы **активизировать среду суперпользователя,** введите команду su с параметром в виде тире (-):

```
# exit
$ su -
Password: *****
# echo SPATH
/usr/kerberos/sbin:/usr/kerberos/bi n:/usr/1ocal/sbi n:
/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bi n
```

В большинстве случаев следует использовать su, если только у вас нет особой причины не делать этого. Если не обозначен ни один пользователь, то su по умолчанию выполнится от имени суперпользователя. Однако su также можно применить для того, чтобы **стать одним из других пользователей:**

```
$ su - chris
```

Кроме того, команду su можно задействовать для **выполнения одной команды от имени определенного пользователя:**

```
$ su -c whoami
Password: ******
root
# su -c 'less /var/1og/messages'
```

Несмотря на то что во втором примере вы вошли в систему как обычный пользователь, в результате выполнения команды whoami с su -c на экране появится информация, свидетельствующая о том, что вы являетесь суперпользователем. В предыдущем примере для идентификации /var/1 og/messages как параметра для less потребовалось заключить строку команды less в кавычки. Как показано выше, команда whoami может быть полезна для **определения, от имени какого пользователя вы выполняете команду в настоящее время:**

```
$ whoami
chris
```

Переменные среды

Небольшие фрагменты информации, полезные для вашего интерпретатора команд, находятся в том, что называется *переменными.* По соглашению имена всех переменных интерпретатора команд указываются в верхнем регистре (хотя придерживаться этого правила не обязательно). Если вы используете bash, то за вас для некоторых переменных будут заданы значения с использованием разных сценариев запуска bash, описанных ранее.

Указанное для переменной значение определяет, является ли она локальной (доступной только текущему интерпретатору команд) или же *переменной среды* (которая наследуется другими интерпретаторами команд или приложениями, запускаемыми из этих интерпретаторов). Чтобы определенная переменная смогла стать переменной среды, ее нужно экспортировать.

Чтобы **отобразить в алфавитном порядке все переменные текущего интерпретатора команд,** для которых уже заданы значения в случае с вашим интерпретатором команд, введите следующее:

```
$ set | less
BASH=/bi n/bash
C0L0RS=/etc/DIR_C0L0RS.xterm
COLUMNS-118
DISPLAY»:0.0
HOME=/home/chns
HOSTNAME=ubuntutb
...
```

Данный вывод содержит лишь несколько примеров переменных среды, которые вы увидите. Команда set показывает локальные переменные, переменные среды, а также функции. Команда env выводит на экран только переменные среды.

Вы также можете самостоятельно **задавать значения для любых переменных, а затем изменять эти значения.** Например, чтобы указать значение 123 для переменной ABC (а затем отобразить содержимое ABC), введите следующее:

```
$ ABC=123
$ echo $ABC
123
```

Переменная ABC существует только в рамках интерпретатора команд, в котором она была создана. Если вы начнете выполнение команды из этого интерпретатора команд (Is, cat, fi refox и т. д.), то новый процесс не увидит эту переменную. Запустите новый процесс bash и проверьте это:

```
$ bash
$ echo $ABC

$
```

Вы можете **сделать переменные частью среды и обеспечить возможность их наследования** дочерними процессами, экспортировав эти переменные:

```
$ export ABC=123

$ bash
$ echo $ABC
123
```

Кроме того, вы можете **выполнить конкатенацию строки и существующей переменной:**

```
$ export PATH=$PATH:/home/chri s/bin
```

Наберите следующее, чтобы **вывести на экран переменные среды своего bash:**

```
$ env
```

При создании своих собственных переменных среды избегайте имен, которые уже широко используются в системе для переменных среды. Список переменных среды интерпретатора команд приведен в приложении Б.

Создание простых сценариев интерпретатора команд

Сценарии интерпретатора команд подходят для автоматизации повторяющихся задач, выполняемых с его использованием, bash и другие интерпретаторы команд включают базовые конструкции, встречающиеся в разных языках программирования, например циклы, проверки, операторы case и т. д. Основное отличие состоит в том, что в случае с интерпретаторами команд имеют место переменные только одного типа — строковые.

Редактирование и запуск сценария

Сценарии интерпретатора команд представляют собой простые текстовые файлы, содержащие команды, функции, псевдонимы или любые другие компоненты, выполнение которых можно запустить из интерпретатора команд. Преимущество сценария интерпретатора команд заключается в том, что вы можете использовать его для формирования целого набора команд, благодаря чему их можно легко применять снова (без необходимости заново печатать) или даже выполнять автоматически (например, в случае задания cron).

Вы можете создавать сценарии интерпретатора команд, используя свой любимый текстовый редактор (например, vi). Чтобы запустить файл сценария интерпретатора команд, этот файл должен быть выполняемым. К примеру, если вы создадите сценарий интерпретатора команд с файловым именем myscript.sh, то сможете **сделать его выполняемым** так:

```
$ chmod u+x myscript.sh
```

Кроме того, первая строка ваших сценариев bash всегда должна иметь следующий вид:

```
#!/bin/bash
```

Знак # в данном случае означает начало комментария. Синтаксис #! выступает в качестве комментария для интерпретаторов команд, которые не понимают этот особый синтаксис. Часть /bi n/bash сообщает функционирующему интерпретатору команд, будь то bash или другой, какую программу следует использовать для запуска сценария (поскольку исторически сложилось, что не все системы включали в себя bash, вы часто будете видеть /bin/bash в роли команды для запуска сценария).

Как и в случае с любой командой, помимо необходимости быть выполняемым, создаваемый вами сценарий интерпретатора команд также должен быть либо указан в PATH, либо идентифицироваться посредством своего полного или относительного пути при запуске. Другими словами, если вы попытаетесь запустить свой сценарий, то можете получить следующий результат:

```
$ myscript.sh
bash: myscript.sh: command not found
```

В данном примере каталог, содержащий myscri pt. sh, не указан в вашей переменной PATH. Чтобы решить эту проблему, вы можете отредактировать PATH, скопировать соответствующий сценарий в каталог, указанный в вашей переменной PATH, либо ввести полный или относительный путь к своему сценарию. Вот четыре только что описанных варианта с использованием PATH, приведенные в соответствующем порядке:

```
S mkdir -/bin ; cp myscript.sh -/bin/ ; PATH=$PATH:~/bin
$ cp myscript.sh /usr/Tocal/bin
$ ./myscript.sh
$ /tmp/myscript.sh
```

Избегайте использования точки в переменной среды PATH как индикатора, что команды могут выполняться из текущего каталога. Этот подход может привести к появлению команд, обладающих теми же именами, что и важные, хорошо известные команды (например, T s или cat), которые могут быть переименованы, если команды с аналогичными названиями окажутся в текущем каталоге. Это представляет серьезную угрозу безопасности.

Добавление содержимого в сценарий

Сценарии интерпретатора команд могут представлять собой простые последовательности команд, однако вы можете использовать их тем же путем, что и код на любом языке программирования. Например, сценарий может выдавать разные результаты, если вы будете предоставлять ему разные вводимые данные. В этом разделе описывается, как использовать составные команды, например операторы i f/then, case и циклы for/whi 1 e в ваших сценариях интерпретатора команд.

Код в приведенном далее примере присваивает строку abc переменной MYSTRING. Затем он проверяет вывод с целью узнать, равно ли ее значение abc, и действует далее согласно результату проверки. Соответствующая проверка заключена в квадратные скобки ([]):

```
MYSTRING=abc
if [ SMYSTRING = abc ] : then
echo "The variable is abc"
fi
```

Чтобы **провести проверку с использованием логического отрицания,** задействуйте ! = вместо =, как показано далее:

```
if [ SMYSTRING != abc ] : then
echo "SMYSTRING is not abc":
fi
```

Ниже приведены примеры **проведения проверки чисел** :

```
MYNUMBER=1
if [   $MYNUMBER-eq 1 ] ;then   echo "MYNUMBER equals 1"; fi
if [   $MYNUMBER -lt 2 ] ;then   echo "MYNUMBER <2":   fi
if [   $MYNUMBER-le 1 ] ;then   echo "MYNUMBER <=1"; fi
if [   $MYNUMBER-gt 0 ] ;then   echo "MYNUMBER >0";   fi
if [   $MYNUMBER-ge 1 ] ;then   echo "MYNUMBER >=1" ; fi
```

Рассмотрим некоторые **проверки, затрагивающие имена файлов.** В приведенном далее примере показано, что вы можете проверить, существует ли файл (-e), является он обычным файлом (-f) или каталогом (-d). Эти проверки проводятся с помощью операторов i f/then. Если не обнаружится ни одного соответствия, для генерирования результата будет использоваться оператор el se.

```
filename="$HOME"
if [ -e $filename ] ;       then echo "$filename exists"; fi
if [ -f "$filename" ]        ; then
    echo "$filename is     a regular file"
elif [ -d "$filename"        ] ; then
    echo "$filename is     a directory"
else
    echo "I have no idea what $filename is       "
fi
```

В табл. 3.1 приведены примеры тестов, которые вы можете проводить в отношении файлов, строк и переменных.

Таблица 3.1. Операторы для тестирования

Оператор	Проводимая проверка
-a файл	Наличие файла (то же, что -e)
-b файл	Является ли файл файлом специального блочного устройства
-c файл	Является ли файл файлом специального символьного устройства (например, последовательного)
-d файл	Является ли файл каталогом
-e файл	Существует ли файл (то же, что -a)
-f файл	Существует ли файл и является ли он обычным (например, не является каталогом, сокетом, каналом, ссылкой или файлом устройства)
-g файл	Задан ли для файла бит set-group-id
-h файл	Не является ли файл символьной ссылкой (то же, что -L)
-k файл	Задан ли для файла бит закрепления
-L файл	Не является ли файл символьной ссылкой (то же, что -h)
-n строка	Превышает ли длина строки 0 байт
-o файл	Являетесь ли вы владельцем файла

Оператор	Проводимая проверка
-p файл	Не является ли файл именованным каналом
-r файл	Доступен ли вам файл для чтения
-s файл	Существует ли файл и превышает ли его размер 0 байт
—S файл	Существует ли файл и является ли он сокетом
-t файл	Связан ли дескриптор файла с терминалом
-и файл	Проверяет, какой файл обладает множеством битов set-user-id (установка идентификатора пользователя)
—w файл	Доступен ли вам файл для записи
-x файл	Доступен ли вам файл для выполнения
-z строка	Составляет ли длина строки 0 байт
выражение1 -а выражение2	Являются ли оба (первое и второе) выражения истинными
выражение1 -о выражение2	Является ли любое из двух выражений истинным
файл1 -nt файл2	Является ли первый файл более новым, чем второй (с использованием временной метки модификации)
файл! -ot файл.2	Является ли первый файл более старым, чем второй (с использованием временной метки модификации)
файл1 -ef файл2	Связаны ли два файла посредством ссылки (жесткой или символьной)
переменная1 = переменная2	Равно ли значение первой переменной значению второй
переменная! -eq переменная2	Равно ли значение первой переменной значению второй
переменная! -де переменная2	Больше ли или равно значение первой переменной значению второй
переменная! -gt переменная2	Больше ли значение первой переменной значения второй
переменная! -1е переменная2	Меньше ли или равно значение первой переменной значению второй
переменная! -It переменная2	Меньше ли значение первой переменной значения второй
переменная! != переменная2	Неравенство значений первой и второй переменных
переменная! -пе переменная2	Неравенство значений первой и второй переменных

Еще одной часто применяемой конструкцией является команда case. Используя оператор case, вы можете проводить проверку на предмет различных вариантов и предпринимать последующие действия в зависимости от результата. Как и оператор switch в языках программирования, операторы case могут использоваться вместо нескольких вложенных операторов if.

```
case "$VAR" in
     stringl)
          { actionl };;
     stri ng2)
          { action2 };;
     *)
          { default action } ;;
esac
```

Вы можете найти примеры использования case в системных сценариях запуска (initscripts), находящихся в каталоге /etc/init.d/. Каждый сценарий init выполняет действия, исходя из параметра, который был ему передан (start, stop и т. д.), а выборка осуществляется с помощью большой конструкции case.

ПРИМЕЧАНИЕ----------

Сценарии init, которые ранее располагались в каталоге etc/init.d и соответствующих каталогах /etc/rc?.d, теперь хранятся в каталоге /etc/init в Ubuntu.

Интерпретатор команд bash также предлагает **стандартные конструкции для выполнения циклов,** которые показаны в примерах ниже. В первом примере все значения переменной NUMBER (от 0 до 9) располагаются в строке for:

```
for NUMBER in 0 1 2 3 4 5 б 7 8 9
do
    echo The number is $NUMBER
done
```

В приведенных далее примерах вывод, генерируемый командой Is (список файлов), обеспечивает переменные, в отношении которых оператор for выполняет действия:

```
for FILE in '/bin/ls': do echo $FILE; done
```

Вместо того чтобы предавать оператору for весь список значений, вы можете увеличивать значение и **продолжать выполнение цикла while, пока условие не будет удовлетворено.** В следующем примере значение VAR начинается с 0 и увеличивается во время выполнения цикла whi 1 e, пока не станет равным 3:

```
"VAR=0"
while [ $VAR -It 3 ]: do
    echo $VAR
    VAR=$[$VAR+1]
done
```

Другой способ добиться результата, который обеспечивает только что продемонстрированный оператор while, заключается в использовании оператора until:

```
VAR=0
until [ $VAR -eq 3 ]; do echo $VAR: VAR=$[$VAR+1]; done
```

Если вы только начинаете осваивать написание программ интерпретатора команд, то обратитесь к руководству Bash Guide For Beginners **(http://tldp.org/LDP/Bash-Beginners-Guide/html/index.html).** Используйте это руководство наряду со справоч-

ным материалом наподобие MAN-страницы bash, чтобы изучить примеры хороших методик написания сценариев интерпретатора команд.

Резюме

Несмотря на усовершенствования графических интерфейсов пользователя, интерпретатор команд по-прежнему остается самым популярным инструментом, применяемым продвинутыми пользователями для работы с Linux-системами. Bourne Again Shell (bash) — это наиболее распространенный интерпретатор команд, используемый в сочетании с Linux. Он включает множество полезных параметров для повторного вызова команд (история), завершения ввода команд, задания псевдонимов и перенаправления вывода и ввода в случае с командами. Вы можете сами создавать мощные команды, используя простые методики написания сценариев интерпретатора команд.

4 Работа с файлами

В этой главе:

- настройка прав доступа;
- обход файловой системы;
- создание/копирование файлов;
- использование жестких/символьных ссылок;
- изменение атрибутов файлов;
- поиск файлов;
- отображение списка файлов и проверка файлов.

В файловой системе Linux все можно рассматривать как файлы. К ним относятся файлы данных, каталоги, устройства, именованные каналы, ссылки и файлы других типов. С каждым файлом связан набор данных, который определяет, кто и как может получить доступ к нему. В этой главе рассматривается множество команд для исследования и работы с файлами.

Типы файлов

Каталоги и обычные файлы, несомненно, являются файлами тех типов, которые вы будете наиболее часто использовать. Однако существуют и другие виды файлов, и с ними вы будете сталкиваться при использовании Linux. Командная строка предоставляет множество способов создания, поиска и вывода на экран файлов разных типов.

Файлы, которые обеспечивают доступ к аппаратным компонентам вашего компьютера, называются *файлами устройств.* Есть символьные и блочные устройства. Существуют также *жесткие ссылки* и *гибкие ссылки,* которые вы можете использовать, чтобы сделать один и тот же файл доступным из разных мест. Реже используются *именованные каналы* и *сокеты,* обеспечивающие точки доступа для процессов, которые позволяют им взаимодействовать.

Обычные файлы

К обычным файлам относятся файлы данных (документы, музыка, изображения, архивы и т. д.) и команды (двоичные файлы и сценарии). Вы можете определить

тип файла, воспользовавшись командой fi 1e. В приведенном далее примере показано, что вы можете перейти в каталог, содержащий документацию по интерпретатору команд bash (доступный, если у вас установлен пакет doc-base), и **использовать команду file для просмотра типов файлов** в этом каталоге:

```
$ cd /usr/share/doc/
$ file doc-base/install-docs.html
doc-base/install-docs.html: XML document text
$ file doc-base/copyright
doc-base/copyright: ASCII English text
$ file doc-base/doc-base.html
doc-base/doc-base.html/: di rectory
$ file doc/doc-base/changelog.gz
doc-base/changelog.gz: gzip compressed data, from Unix, max compression
$ file shared-mime-info/shared-mime-info-spec.pdf
shared-mime-info/shared-mime-info-spec.pdf: PDF document, version 1,4
```

Команда file отображает файлы документов различных форматов, расположенные в каталогах с документаций по Ubuntu. Она способна заглянуть внутрь файлов и выяснить, что тот или иной файл содержит текст, который был сжат; является файлом в формате PDF или PostScript, который можно отправить на принтер; содержит обычный текст или HTML-разметку (веб-страница). На экран выводится даже подкаталог, имя которого оказывается неожиданным, поскольку является странным для каталога (doc-base.html).

Вы можете создавать обычные файлы с помощью любого приложения, которое позволяет сохранять данные. Если вы просто хотите **создать пустые файлы для начала,** есть множество способов сделать это. Вот два примера:

```
$ touch /tmp/newfi 1 e.txt        Создать пустой файл
$ > /tmp/newfi1e2.txt             Создать пустой файл
```

Отображение подробной информации о файле — это еще один способ определить его тип. Например:

```
$ Is -1 /tmp/newfile2.txt Показать информацию о файле для определения его типа
-rw-r--r-- 1 chris chris 0 Sep 5 14:19 newfile2
```

Знак тире на месте первого символа в десятисимвольной информации о правах доступа (-rw-r--r--) выступает в роли индикатора, что соответствующий элемент является обычным файлом (описание прав доступа вы найдете в разделе «Настройка прав доступа к файлам и папкам» данной главы). Команды тоже являются обычными файлами, но сохраняются как выполняемые файлы. Вот несколько примеров:

```
$ Is -1 /usr/bin/apt-key
-rwxr-xr-x 1 root root 8067 Dec 12 08:48 /usr/bin/apt-key
$ file /usr/bin/apt-key
/usr/bin/apt-key: POSIX shell script, UTF-8 Unicode text executable
$ file /bin/ls
/bin/ls: ELF 64-bit LS8 executable, x86-64, version 1 (SYSV),
...
```

Как вы можете видеть, команда apt-key является выполняемой благодаря настройкам x для владельца, группы и пр. Введя команду file в отношении apt-key, вы увидите, что это сценарий интерпретатора команд. Команды, которые не являются сценариями, представляют собой двоичные выполняемые файлы, как, например, команда Is, приведенная ранее.

Каталоги

Каталог — это контейнер для файлов и подкаталогов. Каталоги располагаются иерархически, начиная с корневого (/) и заканчивая множеством подкаталогов, каждый из которых отделяется слешем. Каталоги называют *папками* при доступе к ним посредством графических файловых менеджеров.

Чтобы создать новые каталоги для размещения данных, вы можете воспользоваться командой mkdi r. Вот примеры использования mkdi r для **создания каталогов различными способами:**

$ mkdir /tmp/new	*Создать каталог new в /tmp*
$ mkdir -p /tmp/a/b/c/new	*Создавать родительские каталоги для new по мере необходимости*
$ mkdir -т 700 **/tmp/new2**	*Создать new2 с правами доступа drwx------*

В первом примере команда mkdi r просто добавляет каталог new в существующий каталог /tmp. Во втором примере она создает каталоги по мере необходимости (подкаталоги a, b и c), чтобы создать итоговый каталог new. В последнем примере команде mkdi r передается параметр -т для настройки прав доступа к каталогу.

Вы можете **идентифицировать файл как каталог** благодаря тому, что первым символом в десятисимвольной строке прав доступа является d:

```
$ file /tmp/new
/tmp/new: directory
$ Is -Id /tmp/new
drwxrwxr-x 2 chris chris 4096 Feb 10 15:40 /tmp/new
```

Касательно каталогов также следует отметить, что если вы хотите, чтобы другие могли использовать определенный каталог в качестве своего текущего, то для него должны присутствовать биты выполнения (x).

Символьные и жесткие ссылки

Вместо копирования файлов и каталогов в разные части файловой системы можно создать ссылки для обеспечения доступа к одному и тому же файлу из множества мест. Linux поддерживает как *гибкие ссылки* (обычно называемые *символьными*), так и *жесткие.*

Когда вы попытаетесь открыть символьную ссылку, указывающую на файл, либо перейти по той, которая направляет в каталог, введенная вами команда будет выполняться в отношении файла или каталога, являющегося целью этой ссылки. Эта цель имеет собственный набор прав доступа и владение, которые вы не сможете увидеть посредством символьной ссылки. Символьная ссылка может находиться

в другом разделе диска, нежели цель. Более того, символьная ссылка может существовать даже при отсутствии цели.

Жесткая ссылка, наоборот, может использоваться только в случае с файлами (не каталогами) и, по сути, является способом снабдить один и тот же физический файл несколькими именами. Каждый физический файл обладает как минимум одной жесткой ссылкой, под которой обычно понимается он сам. Любые дополнительные имена (жесткие ссылки), указывающие на этот одиночный физический файл, должны быть в том же разделе, что и оригинальный целевой файл (фактически один из способов узнать, что файлы являются жесткими ссылками, заключается в выяснении, что все они имеют один и тот же номер индексного дескриптора). Изменение прав доступа, владения, даты/временных меток или содержимого любой жесткой ссылки на файл приведет к тому, что изменятся и все остальные. Однако удаление одной ссылки не повлечет за собой удаление файла; он продолжит существовать, пока не будет удалена последняя ссылка на него.

Вот несколько примеров использования команды In для **создания жестких и символьных ссылок:**

```
$ touch myfile
$ In myfile myfile-hard!ink
$ In -s myfile myfile-symlink
$ Is -li myfile*
4460742 -rw-rw-r--    2chris   chris 0 Feb  10 18:01 myfile
4460742 -rw-rw-r--    2chris   chris O'      Feb10  18:01   myfile-hardlink
4460748 Irwxrwxrwx  1chris   chris 6 Feb  10 18:02 myfile-symlink -> myfile
```

Обратите внимание, что после создания жестких и символьных ссылок на файлы я использовал команду Is -li для отображения результатов. Параметр -li обеспечивает вывод на экран индексных дескрипторов, связанных с каждым файлом. Как видите, у шу■file и myfi le-hardl ink есть номер индексного дескриптора 4460742 (который обозначает один и тот же файл на жестком диске). Символьная ссылка myfile-symlink имеет другой номер индексного дескриптора. Хотя жесткая ссылка просто представлена как файл (-), символьная идентифицирована как ссылка (1) с широкими правами доступа. Вы не сможете узнать, есть ли у вас доступ к файлу, на который указывает символьная ссылка, пока не попробуете перейти по ней или не увидите ее цель.

Файлы устройств

Когда приложениям необходимо взаимодействовать с аппаратным обеспечением компьютера, они направляют данные в *файлы устройств.* По соглашению файлы устройств располагаются в каталоге /dev. Устройства обычно делятся на блочные (например, носители информации) и символьные (например, последовательные порты и терминальные устройства).

ПРИМЕЧАНИЕ----------

Файлы устройств часто называют драйверами устройств. Операционные системы Linux и UNIX рассматривают почти все как файлы — отсюда и термин «файлы устройств».

Каждый файл устройства связан со старшим номером (индикатором типа устройства) и младшим номером (индикатором номера экземпляра устройства). Например, терминальные устройства (tty) представляются в виде старшего устройства номер 4, а жесткие диски с интерфейсом SCSI — в виде старшего блочного устройства номер 8. Вот **примеры файлов устройств:**

```
$ Is -1 /dev/ttyO /dev/sdal Показать специальные символьные и блочные устройства
brw-rw---- 1 root disk 8, 1 Feb 3 17:24 /dev/sdal
crw--w---- 1 root tty 4. 0 Feb 3 17:24 /dev/ttyO
```

Список имен и номеров устройств, выделяемых в Linux, доступен в Ubuntu на соответствующей странице интерактивного руководства, посвященной команде MAKEDEV. Большинство файлов устройств создается автоматически во время загрузки либо посредством udev при подключении нового аппаратного обеспечения в процессе работы компьютера (например, когда вы вставляете флеш-диск USB). Большинство пользователей никогда не создает файлы устройств вручную. Однако вы можете **создать собственный файл устройства** с помощью команды mknod. Вот пример:

```
$ sudo mknod /dev/ttyS40 c 4 68 Добавить устройство для последовательного порта 41
$ Is -1 /dev/ttyS40             Показать новый файл устройства
erw-rw---- 1 root dial out 4. 68 Feb 3 17:24 /dev/ttyS4
```

Именованные каналы и сокеты

Если вы захотите позволить одному процессу отправить информацию другому, можете просто передать по каналу (|) вывод от одного процесса в качестве ввода для другого. Кроме того, для обеспечения присутствия в файловой системе, из которой процесс сможет взаимодействовать с другими процессами, вы можете создать *именованные каналы* или *сокеты*. Первые обычно служат для межпроцессного взаимодействия в локальной системе, в то время как вторые можно использовать, чтобы процессы могли взаимодействовать по сети.

Именованные каналы и сокеты часто создаются приложениями в каталоге /tmp. Вот несколько **примеров именованных каналов и сокетов:**

```
$ Is -1 /tmp/.TV-chris/tvtimefifo-local /tmp/.Xll-unix/XO
prw--------- i chris chris 0 Sep 26 2007 /tmp/.TV-chris/tvtimefifo-local
srwxrwxrwx 1 root root 0 Sep 4 01:30 /tmp/.Xll-unix/X0
```

В первом листинге мы видим именованный канат, созданный приложением tvtime TV Card Player (следует отметить, что символ p в начате является индикатором именованного канала), а во втором листинге — сокет, созданный GUI-интерфейсом X для межпроцессного взаимодействия.

Чтобы **создать собственный именованный канал,** используйте команду mkfi fo:

```
$ mkfi fo mypipe
$ Is -1 mypipe
prw-rw-r-- 1 chris chris 0 Sep 26 00:57 mypipe
```

Настройка прав доступа к файлам и папкам

Возможности доступа к файлам, выполнения команд и перехода в тот или иной каталог можно ограничить путем настройки прав доступа для пользователя, группы и остальных пользователей. При выводе на экран длинного списка (Is -1) файлов и каталогов в Linux первые десять отображаемых символов являются индикаторами, что представляет собой соответствующий элемент (файл, каталог, блочное устройство и т. д.), наряду с тем, возможны ли чтение, запись и/или выполнение этого элемента. На рис. 4.1 показаны значения этих десяти символов.

Рис. 4.1. Права доступа с возможностью чтения, записи и выполнения задаются для файлов и каталогов

Следуя примерам, приведенным в этом разделе, создайте каталог с именем /tmp/test, а также файл под названием /tmp/test/hello.txt. Затем выведите на экран длинные листинги этих двух элементов, как показано далее:

```
$ mkdir /tmp/test
$ echo "some text" > /tmp/test/hello.txt
$ Is -Id /tmp/test/ /tmp/test/hello.txt
drwxrwxr-x 2 chris sales 4096 Feb 10 19:24 /tmp/test/
-rw-rw-r-- 1 chris sales 16 Feb 10 19:24 /tmp/test/hello.txt
```

После создания каталога и файла первый символ в соответствующем длинном листинге выступает в роли индикатора, что /tmp/test представляет собой каталог, а hello, txt является файлом (-). К файлам других типов, доступным в мире Linux, которые идентифицировались бы первым символом, относятся символьные устройства (с), блочные устройства (Ь), символьные ссылки (1), именованные каналы (р) и сокеты (s).

Следующие девять символов представляют права доступа, настроенные для файла и каталога. Первый набор символов, rwx, является индикатором того, что владелец (chris) обладает правами доступа к каталогу с возможностью чтения, записи и выполнения. Аналогичным образом группа sales имеет такие же права доступа (rwx). Затем определяется, что все остальные пользователи обладают только правами доступа с возможностью чтения и выполнения (r-x); знак тире является индикатором отсутствующих прав доступа с возможностью записи. Касательно файла hello.txt пользователь и члены группы имеют права доступа с возможностью чтения и записи (rw-), а все остальные — с возможностью только чтения (r--).

Если вы решите изменить права доступа, знайте, что их можно представить посредством восьмеричного числа (где чтение — это 4, запись — 2, а выполнение — 1)

либо букв (rwx). Вообще, права доступа с возможностью чтения позволят вам просматривать содержимое каталога, с возможностью записи — изменять (добавлять или модифицировать) наполнение каталога, а с возможностью выполнения — переходить (иными словами, получать доступ) в каталог.

Если вас не устраивают права доступа в отношении файлов или каталогов, находящихся в вашем владении, вы можете изменить их с помощью команды chmod.

Изменение прав доступа с помощью chmod

Команда chmod позволяет **изменять права доступа к файлам и каталогам.** В табл. 4.1 показаны некоторые командные строки chmod и изменение доступа к каталогам и файлам.

Таблица 4.1. Изменение прав доступа к файлам и каталогам

Команда chmod (восьмеричным числом или буквами)	Исходные права доступа	Новые права доступа	Описание
chmod 0700 *каталог*	**Все**	**drwx.....**	**Владелец каталога может осуществлять чтение и запись файлов в этот каталог, а также переходить в него. У всех остальных пользователей (за исключением суперпользователя) доступ отсутствует**
chmod 0711 *каталог*	**Все**	**drwx-x-x**	**Те же права доступа, что и у владельца. Все остальные пользователи могут переходить в каталог, но не имеют возможности просматривать или изменять файлы в нем. Это может оказаться полезным для усиления защиты сервера, когда вы запрещаете кому-либо просматривать содержимое каталога, но предоставляете доступ к тому или иному файлу в этом каталоге, если кто-то уже знает, что он там**
chmod go+r *каталог*	**drwx------**	**drwxr-r—**	**Добавление прав доступа с возможностью чтения в случае с каталогом может не привести к желаемым результатам. Не имея прав доступа с возможностью выполнения, остальные пользователи не могут просматривать содержимое любых файлов в этом каталоге**
chmod 0777 *каталог* **и chmod a=rwx** *каталог*	**Все**	**drwxrwxrwx**	**Предоставляются все права доступа**

Команда chmod (восьмеричным числом или буквами)	Исходные права доступа	Новые права доступа	Описание
chmod 0000 *каталог* **и chmod a-rwx** *каталог*	**Все**	**d.....**	**Не предоставляется никаких прав доступа. Такой подход обеспечивает хорошую защиту каталога от ошибочных изменений. Однако у программ резервного копирования, выполняемым не от имени суперпользователя, может не получиться осуществить резервное копирование содержимого каталога**
chmod 666 *файл*	**Все**	**-rw-rw-rw-**	**В полном объеме предоставляются права доступа с возможностью чтения/записи в отношении файлов**
chmod go-rw *файл*	**-rw-rw-rw-**	**-rw.....**	**Никто, кроме владельца, не может просматривать, изменять или удалять файл**
chmod 644 *файл*	**Все**	**-rw-r-r-**	**Только владелец может изменять или удалять файл, однако все пользователи имеют возможность просматривать этот файл**

Общие сведения об изменении прав доступа. Об изменении прав доступа с использованием командной строки в целом вам нужно знать следующее.

Первый 0 в строке режима обычно не указывается (поэтому вы можете ввести 777 вместо 0777). Этот заполнитель имеет особое значение. Он представляет собой восьмеричное число, которое можно использовать в случае с командами (выполняемыми файлами) в качестве индикатора того, что команда может выполняться как программа-установщик идентификатора пользователя (UID) (4), программа-установщик идентификатора группы (GID) (2) или стать *битом закрепления в памяти* (1). При использовании setmd и setgid команда выполняется с соответствующими правами доступа, присвоенными пользователю или группе (вместо того чтобы выполняться с правами доступа пользователя или группы, к которой относится запустивший ее выполнение).

ВНИМАНИЕ----------

Не следует использовать suid в случае со сценариями интерпретатора команд. Вот предупреждение из практического руководства Linux Security HOWTO: «suid-сценарии интерпретатора представляют собой серьезную угрозу безопасности, в силу чего для ядра они неприемлемы. Независимо от того, насколько, как вам кажется, безопасен определенный сценарий интерпретатора команд, он может использоваться взломщиками с целью получения прав доступа суперпользователя для работы в интерпретаторе команд».

Задание бита закрепления в памяти для каталога не позволит пользователям удалять или переименовывать файлы в этом каталоге, которыми они не владеют (примером здесь является /tmp). Однако если указать соответствующие настройки прав доступа, то пользователи смогут изменять содержимое файлов, которыми не

владеют, расположенных в каталоге с битом закрепления в памяти. Последним символом, определяющим права доступа, является t вместо x в случае с каталогом с битом закрепления в памяти. Раньше команды с битом закрепления обычно оставались в памяти, даже если при этом не использовались. Это старый UNIX-параметр, который не поддерживается в Linux.

Параметр -R — это полезная особенность команды chmod. Используя -R, вы можете **рекурсивно изменять права доступа ко всем файлам каталога, начиная с той или иной точки в файловой системе.** Вот несколько примеров:

```
$ sudo chmod -R 700 /tmp/test          Предоставить владельцу права
                                       доступа к содержимому /tmp/test
$ sudo chmod -R 000 /tmp/test          Отменить все права доступа
                                       к содержимому /tmp/test
$ sudo chmod -R a+rwx /tmp/test        Предоставить все права доступа
                                       к содержимому /tmp/test
```

Следует отметить, что параметр -R включается в указанный вами каталог. Таким образом, права доступа в приведенном выше коде, к примеру, изменились бы и в случае с самим каталогом /tmp/test, а не только касательно файлов и каталогов, которые в нем содержатся.

Задание значения для umask

Права доступа, определенные для файла или каталога, изначально присваиваются в момент его создания. Эти права настраиваются исходя из текущего значения umask пользователя. Используя команду umask, вы можете **настроить права доступа, задаваемые в случае с файлом или каталогом** при их создании.

```
$ umask 0066    Настроить права доступа drwx--x--x для каталогов и -rw------- для файлов
$ umask 0077    Настроить права доступа drwx------ для каталогов и -rw------- для файлов
$ umask 0022    Настроить права доступа drwxr-xr-x для каталогов и -rw-r--r- для файлов
$ umask 0777    Настроить права доступа d--------- для каталогов ---------- для файлов
```

Изменение владения

При создании файла или каталога ваша учетная запись пользователя присваивается этому файлу или каталогу. То же происходит и в отношении вашей первичной группы. Являясь суперпользователем, **вы можете изменить владение (пользователя) и группу, присвоенную файлу, на другого пользователя и/или группу** с помощью команд chown и chgrp. Вот несколько примеров:

```
$ chown  chris test/              Изменить владельца на chris
$ chown  chris:markettest/ Изменить владельца на chris, а группу - на
                           market
$ chgrp  market test/             Изменить группу на market
$ chown  -R chris test/           Изменить владельца всех файлов в test/ на chris
```

Приведенный выше рекурсивный параметр команды chown (-R) полезен, если вам нужно **изменить владение целой структурой каталога.** Как и chmod, команда chown рекурсивно изменяет права доступа к указанному каталогу и всему его содержимому. Вы можете задействовать chown рекурсивно, когда то или иное лицо будет уходить из компании или прекращать пользоваться вашей веб-службой. Применив chown -R, вы сможете задать другого пользователя в качестве владельца всего его каталога /home.

К родственным командам для изменения присваиваний и паролей в случае с группами относится gpasswd, а также файл /etc/gshadow.

Обход файловой системы

Основные команды для смены каталогов (cd), проверки текущего каталога (pwd) и вывода на экран содержимого каталогов (Is) хорошо известны даже непрофессиональным пользователям интерпретатора команд. В этом разделе мы сосредоточимся на некоторых реже встречающихся параметрах этих команд, а также менее известных параметрах для перемещения по файловой системе. Вот краткие примеры использования cd для **перемещения по файловой системе:**

$ cd	*Перейти в ваш домашний каталог*
$ cd $H0ME	*Перейти в ваш домашний каталог*
$ cd -	*Перейти в ваш домашний каталог*
$ cd -chris	*Перейти в домашний каталог пользователя chris*
$ cd -	*Перейти в предыдущий рабочий каталог*
$ cd $0LDPWD	*Перейти в предыдущий рабочий каталог*
$ cd ~/public_html	*Перейти в public_html в вашем домашнем каталоге*
$ cd ..	*Перейти в родительский каталог текущего каталога*
$ cd /usr/bin	*Перейти в usr/bin из корневого каталога*
$ cd usr/bin	*Перейти в usr/bin в текущем каталоге*

Если вы захотите **узнать, какой каталог является вашим текущим,** то воспользуйтесь pwd (print working directory — «вывести на экран рабочий каталог»):

```
$ pwd
/home/chris
```

Создание символьных ссылок — это способ получить доступ к файлу из других частей файловой системы (дополнительные сведения о символьных и жестких ссылках вы найдете в подразделе «Символьные и жесткие ссылки» ранее в этой главе). Однако символьные ссылки могут вызывать некоторое замешательство относительно способа просмотра родительских каталогов. Приведенные далее команды **создают символьную ссылку** на каталог /tmp из вашего домашнего каталога и показывают, как узнать, где вы ссылаетесь на связанный каталог:

```
$ cd $H0ME
$ in -s /tmp tmp-link
$ Is -1 tmp-link
lrwxrwxrwx 1 chris chris 13 Mar 24 12:41 tmp-link -> /tmp
$ cd tmp-link/
```

```
$ pwd
/home/chris/tmp-1 ink
$ pwd -P
/tmp
$ pwd -L
/home/chris/tmp-1ink
$ cd -L ..
$ pwd
/home/chris
$ cd tmp-link
$ cd -P ..
$ pwd
/
```

Используя параметры -P и -L для pwd и cd, вы можете **работать с символически связанными каталогами соответственно там, где они располагаются постоянно или где находятся ссылки на них.** Например, cd -L .. позволит переместиться на один уровень вверх, в ваш домашний каталог, а с помощью cd -P .. вы сможете перейти на один уровень выше постоянного каталога (/). Кроме того, параметры -P и -L для pwd позволяют отображать постоянное местоположение каталогов и расположение ссылок на них.

bash способен запоминать список рабочих каталогов. Такой перечень окажется полезным, если вы захотите вернуться к ранее посещенным каталогам. Этот список организуется в форме стека. Используйте pushd и popd **для добавления и удаления каталогов:**

```
$ pwd
/home/chris
$ pushd /usr/share/man/
/usr/share/man -
$ pushd /var/1og/
/var/log /usr/share/man -
$ dirs
/var/log /usr/share/man -
$ dirs -v
 0   /var/log
 1   /usr/share/man
 2   -
$ popd
/usr/share/man -
$ pwd
/usr/share/man
$ popd
~
$ pwd
/home/chris
```

Команды di rs, pushd и popd можно использовать для изменения порядка каталогов в стеке. Например, pushd -0 перемещает последний каталог в стеке на его вершину (делая этот каталог текущим). Команда pushd -2 переносит третий каталог с низа стека на его вершину.

Копирование файлов

Если вы обладаете правами доступа с возможностью записи к целевому каталогу, то вы можете копировать файлы и каталоги с помощью достаточно простых команд. Стандартная команда cp **скопирует файл с присвоением ему нового имени или сохранением прежнего названия в новый каталог** с новой временной меткой, связанной с новым файлом. Другие параметры команды cp позволяют вам сохранять дату/временные метки, выполнять рекурсивное копирование и отображать приглашения ввести подтверждение перед перезаписью. Вот несколько примеров:

```
$ cd ; touch index.html
$ mkdir /tmp/html
$ cp -i index.html /tmp/html/
$ cp -il index.html /tmp/html/index2.html
$ mkdir /tmp/back
$ cp -a /tmp/html /tmp/back/
$ cp -R /tmp/html /tmp/back/
```

В приведенных выше примерах продемонстрированы способы копирования файлов. В первом примере с использованием cp, если файл index.html существует в /tmp/html, вы получите приглашение ввести подтверждение перед его перезаписью новым файлом. В следующем примере файл i ndex. html связан посредством жесткой ссылки с файлом, который имеет аналогичное имя и находится в каталоге /tmp/html. В данном случае, поскольку обе жесткие ссылки указывают на один и тот же файл, редактирование файла из любого расположения приведет к изменению содержимого файла и в том и в другом месте (ссылка возможна, только если /tmp/html и ваш домашний каталог находятся в одной и той же файловой системе).

Команда cp -a копирует все файлы, содержащиеся в каталоге /tmp/html, сохраняя все настройки владения и прав доступа. Например, если /tmp/back будет представлять флеш-диск USB, то данная команда окажется способом скопировать содержимое вашего веб-сервера на этот носитель. Кроме того, параметр -R обеспечивает рекурсивное копирование структуры каталога, однако присваивает владение текущему пользователю и добавляет текущую дату/временные метки.

Команда dd представляет собой еще один способ **копирования данных.** Эта команда очень мощная, поскольку в Linux-системах все является файлами, включая периферийные устройства. Вот пример:

```
$ dd if=/dev/zero of=/tmp/mynullfi1e count=l
1+0 records in
1+0 records out
512 bytes (512 B) copied, 4.5447e-05 s, 11.3 MB/s
```

Файл /dev/zero — это специальный файл, который генерирует нулевые символы. В приведенном выше примере команда dd принимает /dev/zero в качестве файла ввода и осуществляет вывод в /tmp/mynul 1 fi 1 e. Значение count — это количество блоков. По умолчанию размер блока составляет 512 байт. Результатом является файл размером 512 байт, полный нулевых символов. Вы можете воспользоваться

less или vi для просмотра содержимого этого файла, однако наилучшим инструментом просмотра файла будет команда od (octal dump — восьмеричный дамп).

$ od -vt xl /tmp/mynul1fi1e *Просмотреть восьмеричный дамп файла*

Вот еще один пример использования команды dd:

$ dd if-/dev/zero of=/tmp/mynunfПe count=10 bs=2
10+0 records in
10+0 records out
20 bytes (20 B) copied, 0.000173996 s. 115 kB/s

На этот раз размер блока задан равным **2** байтам, а копированию подвергаются десять блоков **(20** байт). Приведенная далее строка команд **копирует первый раздел основного жесткого IDE-диска** во второй раздел подчиненного жесткого IDE-диска (прежде чем делать нечто подобное, сделайте резервную копию всех данных):

$ sudo dd if=/dev/hdal of=/dev/hdb9

ВНИМАНИЕ----------

Будьте очень осторожны с этой командой — вы же не хотите слепо перезаписывать части своих жестких дисков.

В следующем примере демонстрируется **создание сжатой резервной копии** первого раздела первичного ведущего диска с IDE-интерфейсом. Обычно раздел следует демонтировать перед резервным копированием.

$ sudo umount /dev/hdal
$ sudo dd if=/dev/hdal | gzip > bootpart.gz

Приведенная далее команда копирует файл ISO-образа с CD или DVD на ваш флеш-диск USB (при этом предполагается, что ваш диск представлен как /dev/sdbl):

$ sudo dd if=whatever.iso of=/dev/sdb9

Следует отметить, что эта команда делает двоичную копию байтов файла, а это, возможно, не то, что вам нужно.

В следующем примере демонстрируется копирование главной загрузочной записи с первичного ведущего жесткого диска с интерфейсом IDE в файл с именем mymbrfile:

$ dd if=/dev/hda of=mymbrfile bs=512 count-1

Если вы захотите сделать копию ISO-образа, записанного на CD или DVD, вставьте этот носитель в привод и (предполагается, что /dev/cdrom ассоциирован с дисковым приводом вашего компьютера) введите следующую команду:

$ dd if-/dev/cdrom of=whatever.iso

ПРИМЕЧАНИЕ----------

Ubuntu также создает файлы /dev/cdrw и /dev/dvd в дополнение к /dev/cdrom.

Изменение атрибутов файлов

Для работы со всеми файлами и каталогами в Linux-системах имеются права доступа с возможностью чтения, записи и выполнения, ассоциированные с пользователем, группой и остальными (пользователями). Однако также есть и другие атрибуты, которые могут быть прикреплены к файлам и каталогам и являются специфичными для файловых систем определенных типов.

Файлы в системах ext2 и ext3 обладают специальными атрибутами, которые вы можете выбирать. Для их просмотра применяется команда lsattr. Большинство атрибутов скрыто и не активизировано по умолчанию. Вот пример использования lsattr для просмотра атрибутов некоторых файлов:

```
$ lsattr /etc/host*
----------e_  /etc/host,conf
----------e_  /etc/hosts
----------e_  /etc/host.allow
----------e-  /etc/host.deny
$ lsattr -aR /tmp/ | less          Рекурсивно отобразить все атрибуты /tmp
```

Знаки тире представляют 13 атрибутов ext2/ext3, которые можно задать. Ни один из них не активизирован по умолчанию. Это следующие атрибуты: a (только добавление), c (сжатый), d (не делать резервной копии), e (формат экстента), i (неизменяемый), j (журналирование данных), s (безопасное удаление), t (запрет слияния в конце файла), и (неудаляемый), A (без обновлений atime), D (синхронные обновления каталогов), S (синхронные обновления) и T (вершина иерархии каталогов). Вы можете **изменять эти атрибуты,** используя команду chattr. Вот несколько примеров:

```
$ sudo chattr +d whatever.iso
$ sudo chattr +A -R /home/chris/images/*
$ sudo chattr +d ubuntu.iso
$ lsattr whatever.iso /home/chris/images/* ubuntu.iso
----i----------- whatever.iso
--------д------ /home/chris/images/einstein.jpg
--------A------ /home/chris/images/goth.jpg
-------d-------- ubuntu.iso
```

Как видно из приведенного выше примера, если использовать параметр +i, то файл whatever. i so станет неизменяемым, то есть его нельзя будет удалить, переименовать, изменить или создать на него ссылку. В данном случае это позволит предотвратить внесение любых произвольных изменений в файл (даже суперпользователь не сможет изменять этот файл, пока атрибут i не будет убран). Вы можете использовать его для усиления защиты системных файлов.

Параметр R в данном примере рекурсивно задает параметр +A, из-за чего для всех файлов, расположенных в каталоге images и его подкаталогах, нельзя изменить время доступа (запись atime). Задание атрибутов A может помочь сэкономить некоторое количество операций ввода/вывода на ноутбуках или при использовании флеш-дисков. Если вы задействуете команду dump для резервного копирования своих файловых систем ext2/ext3, то параметр +d позволит предотвратить резервное

копирование выбранных файлов. При этом вы сможете сделать так, чтобы не создавалась резервная копия большого ISO-образа.

Чтобы **удалить атрибут** с помощью команды chattr, используйте знак минуса (-). Например:

```
$ sudo chattr -i whatever.iso
```

ПРИМЕЧАНИЕ----------

Взломщики, которым удастся проникнуть в тот или иной компьютер, часто будут подменять некоторые системные двоичные файлы (например, Is или ps) искаженными версиями и делать их неизменяемыми. Правильным будет время от времени проверять атрибуты, заданные для ваших выполняемых файлов (например, в /bin, /usr/bin, /sbin и /usr/sbin).

Поиск файлов

В Ubuntu база данных всех файлов в системе (с исключениями, определенными в /etc/updatedb. conf) ведется с использованием функционала пакета ml ocate. Команда locate позволяет искать файлы в этой базе данных (в Ubuntu команда locate является символьной ссылкой на свою безопасную версию под названием ml ocate).

Результаты выполнения 1 ocate возвращаются мгновенно, поскольку поиск осуществляется в базе данных, а не в фактической файловой системе. До появления locate большинство Linux-пользователей применяло команду find для поиска файлов в системе. Далее рассматривается как locate, так и find.

Поиск файлов с помощью locate

База данных содержит названия файлов множества типов из файловой системы, а не только имена команд, вы можете использовать locate для **поиска команд, устройств, MAN-страниц, файлов данных и всего прочего, что идентифицируется посредством имени** в файловой системе. Вот пример:

```
$ locate eMOO
/1i b/modules/2.6.20 -16-generi c/kernel/dri vers/net/elOOO
/1ib/modules/2.6.20-16-generic/kernel/drivers/net/elOOO/elOOO.ko
/1ib/modules/2.6.20-15-generic/kernel/drivers/net/elOOO
/1i b/modules/2.6.20-15-generic/kernel/dri vers/net/elOOO/elOOO.ko
/usr/src/linux-headers-2.6.20-16-generic/include/config/el000
/usr/src/linux-headers-2.6.20-16-generic/include/config/el000/napi.h
/usr/src/1inux-headers-2.6.20-16-generic/include/config/el000.h
/usr/src/linux-headers-2.6.20-15-generic/include/config/el000
/usr/src/1inux-headers-2.6.20-15-generic/include/config/el000/napi,h
/usr/src/1inux-headers-2.6.20-15-generic/include/config/el000.h
/usr/src/1inux-headers-2.6.20-15/include/config/elOOO.h
/usr/src/1inux-headers-2.6.20-15/drivers/net/elOOO
/usr/src/1 inux-headers-2.6.20-15/dnvers/net/elOOO/Makefile
/usr/src/1inux-headers-2.6.20-16/include/config/elOOO.h
/usr/src/linux-headers-2.6.20-16/drivers/net/el000
/usr/src/1inux-headers-2.6.20-16/drivers/net/el000/Makefile
```

В данном примере было найдено две версии модулей ядра el000.ko. Команда locate чувствительна к регистру, если только вы не используете параметр -i. Например:

```
$ locate -i itcowdt
/1 i b/modul es/2.6.20- 16-generi c/kernel/dri vers/char/watchdog/1 TCO wdt. ko
/lib/modules/2.6.20-15-generic/kernel/drivers/char/watchdog/iTCO_wdt.ko
```

Пакет ml ocate включает задание cron, которое выполняет команду updatedb один раз в день для обновления базы данных файлов locate.

Чтобы **немедленно обновить базу данных locate,** вы можете вручную ввести команду updatedb:

```
$ sudo updatedb
```

Как уже отмечалось ранее, файл /etc/updatedb.conf содержит информацию, благодаря которой файлы из конкретных каталогов и файловых систем определенных типов не попадают в обновляемую базу данных. Вот пример информации, содержащейся в файле updatedb.conf:

```
PRUNEPATHS="/tmp /var/spool /media /home/.eeryptfs"
PRUNEFS="NFS nfs nfs4 rpc_pipefs afs binfmt_misc proc smbfs
autofs iso9660 nepfs coda devpts ftpfs devfs mfs shfs sysfs cifs
lustrejite tmpfs usbfs udf fuse.glusterfs fuse.sshfs curlftpfs
ecryptfs fusesmb devtmpfs"
```

Переменная PRUNEPATHS определяет каталоги, файлы и подкаталоги которых будут загружаться в базу данных 1 ocate. Сюда входят места, в которых располагаются временные файлы (например, / var/spool и /tmp). К типам файловых систем, исключаемых строкой PRUNEFS, относятся типы временных и удаленно смонтированных файловых систем (например, nfs и cifs).

Поиск файлов с помощью find

До появления 1 ocate пользователи задействовали команду find для поиска файлов. Хотя locate позволяет быстрее находить нужные файлы, у find имеется много эффективных параметров для поиска файлов по различным атрибутам, а не только по имени.

ПРИМЕЧАНИЕ----------

Поиск по всей файловой системе может занять много времени. Перед тем как запускать такой поиск, попробуйте поискать в подмножестве файловой системы либо исключить определенные каталоги или удаленно смонтированные файловые системы.

В этом примере выполняется рекурсивный поиск файлов с именем eЮО в корневой файловой системе (/):

```
$ find / -name "elOO*" -print
find: /usr/1ib/audit: Permission denied
find: /usr/1ibexec/utempter: Permission denied
/sys/module/elOO
/sys/bus/pci/drivers/elOO
...
```

Выполнение команды от имени обычного пользователя может привести к длинным спискам сообщений Permission denied (Отсутствуют необходимые права доступа), поскольку find будет пытаться войти в каталоги, прав доступа к которым у вас нет. Вы можете **отфильтровать недоступные каталоги:**

```
$ find / -name elOO -print 2>&1 | grep -v "Permission denied"
```

либо **отправить все сообщения об ошибках в битоприемник /dev/null:**

```
$ find / -name elOO -print 2> /dev/null
```

При поиске с помощью find имеет место чувствительность к регистру, а соответствие указанному имени должно быть точным (elOO не будет соответствовать e 100 ko), вы можете **воспользоваться регулярными выражениями, чтобы расширить поиск** (чтобы чувствительности к регистру не было, используйте -i name). Вот пример:

```
$ find / -name 'elOO*' -print
/lib/modules/2.6.20-16-generic/kernel/drivers/net/elOOO
/lib/modules/2.6.20-16-generic/kernel/drivers/net/elOOO/elOOO.ko
/lib/modules/2.6.20-16-generic/kernel/drivers/net/elOO.ko
/1ib/modules/2.6.20-15-generic/kernel/drivers/net/elOOO
/1ib/modules/2.6.20-15-generic/kernel/drivers/net/el000/el000.ko
/1ib/modules/2.6.20-15-generic/kernel/drivers/net/elOO.ko
/usr/src/1inux-headers-2.6.20-16-generic/include/config/el00.h
/usr/src/1inux-headers-2.6.20-16-generic/include/config/el000
/usr/src/1inux-headers-2.6.20 -16-generic/include/config/el000.h
/usr/src/1 inux-headers-2.6.20-15-genenc/include/config/el00.h
/usr/src/1inux-headers-2.6.20-15-generic/include/config/elOOO
/usr/src/1inux-headers-2.6.20-15-generic/include/config/elOOO.h
/usr/src/1inux-headers-2.6.20-15/include/config/elOO.h
/usr/src/1inux-headers-2.6.20-15/include/config/elOOO.h
/usr/src/1inux-headers-2.6.20-15/drivers/net/el000
/usr/src/1 inux-headers-2.6,20-16/incl ude/conf i g/elOO. h
/usr/src/1inux-headers-2.6.20-16/include/config/elOOO.h
/usr/src/1inux-headers-2.6.20-16/drivers/net/el000
```

Вы также можете **искать файлы по временным меткам.** Приведенная далее строка команд ищет в /usr/bin/ файлы, к которым обращались за последние две минуты:

```
$ find /usr/bin/ -amin -2 -print
/usr/bin/
/usr/bin/find
```

Следующая строка ищет в /home/chri s файлы, к которым не было обращения на протяжении более чем 60 дней:

```
$ find /home/chris/ -atime +60
```

Для **поиска каталогов** используйте параметр -type. Приведенная далее строка команд ищет все каталоги в /etc и перенаправляет stderr в битоприемник (/dev/nul 1):

```
$ find /etc -type d -print 2> /dev/null
```

Следующая строка ищет в /sbin файлы **согласно правам доступа,** которые в данном случае должны соответствовать 755:

$ find /sbin/ -perm 755 -print

Параметр exec команды find очень эффективен, поскольку позволяет **воздействовать на файлы, найденные с помощью команды find.** Приведенная далее команда ищет в /var все файлы, которыми владеет пользователь с именем chris (он должен быть допустимым пользователем), и выполняет команду Is -1 в отношении каждого из них:

$ find /var -user chris -exec Is -1 {} \;

Альтернативой параметру exec команды find является xargs:

$ find /var -user chris -print | xargs Is -1

В работе приведенных выше команд есть большие различия, что приводит к значительно отличающейся производительности. Find -exec выполняет команду 1 s в отношении каждого результата, который находит. Команда xarg работает более эффективно, передавая множество результатов в качестве ввода одной команде 1 s (касательно команды Is -1 следует отметить, что вы можете использовать -Is в строке find вместо xargs Is -1).

Чтобы **инвертировать критерий поиска,** поставьте перед ним восклицательный знак (!). Код в следующем примере ищет все файлы, которые не находятся во владении группы суперпользователей и являются обычными файлами, а затем выполняет 1 s -1 в отношении каждого из них:

$ find / ! -group root -type f -print 2> /dev/null | xargs Is -1

Код в приведенном далее примере осуществляет в /sbi п поиск обычных файлов, к которым у остальных пользователей нет прав доступа с возможностью записи, а затем передает их команде Is -1:

$ find /sbin/ -type f ! -perm /o+w **-print | xargs Is -1**
-rwxr-xr-x 1 root root 3056 2007-03-07 15:44 /sbin/acpi_available
-rwxr-xr-x 1 root root 43204 2007-02-18 20:18 /sbin/alsactl

Поиск файлов по размеру — это отличный способ определить, что занимает место на ваших жестких дисках. Следующая строка ищет все файлы, размер которых превышает 10 Мбайт (+10M), генерирует их список, начиная с самого большого файла и заканчивая самым маленьким (Is -IS), и направляет этот список в файл (/tmp/bigfi 1 es. txt):

$ find / -xdev -size +10M -print | xargs Is -IS > /tmp/bigfiles.txt

В этом примере параметр -xdev предотвращает поиск в любых смонтированных файловых системах, кроме корневой. Это хороший способ не позволить команде find осуществлять поиск в каталоге /рrос и любых удаленно смонтированных файловых системах, а также других локально смонтированных файловых системах.

Другие команды для поиска файлов

К командам для поиска файлов относятся также whereis и which. Вот несколько примеров их использования:

```
$ whereis man
man: /usr/ЬIn/man /usr/XllR6/bin/man /usr/bin/Xli/man /usr/iocal/man
   /usr/share/man /usr/share/man/manl/man.l.gz
   /usr/share/man/man7/man.7.gz
$ which Is
/bin/ls
```

Команда whereis является полезным инструментом, поскольку **позволяет искать** не только команды, но и **MAN-страницы и конфигурационные файлы, ассоциированные с той или иной командой.** В примере использования whereis для поиска слова man вы видите, что в результате возвращаются выполняемый файл man, его конфигурационный файл, а также сведения, где располагаются MAN-страницы для команды man. В примере применения which показано, что возвращается информация о расположении выполняемого файла Is (/bin/ls). Команда which окажется полезной, если вы захотите узнать фактическое местонахождение выполняемого файла в своей переменной PATH, как в этом примере:

```
$ dpkg-query -S 'which ps'
procps: /Ьш/ps
```

Получение дополнительной информации о файлах

Теперь, когда вам известно, как искать файлы, вы сможете получить дополнительную информацию о них. С помощью реже использующихся параметров команды 1 s вы можете выводить на экран информацию о файлах, которую не увидели бы, введя команду 1 s без параметров. Такие команды, как fi 1 e, помогут идентифицировать тип файла. Используя md5sum и shalsum, вы можете проверить валидность файла.

Отображение списка файлов

Вы, вероятно, хорошо знаете команду 1 s, однако вам могут быть незнакомы ее многочисленные полезные параметры, которые способны помочь получить больше информации о файлах в вашей системе. Вот некоторые примеры **использования Is для отображения длинных списков** (-1) файлов и каталогов:

$ Is -1	*Показать файлы и каталоги, содержащиеся в текущем каталоге*
$ Is -la	*Включить в список файлы/каталоги, начинающиеся с точки (.)*
$ Is -It	*Упорядочить файлы по времени последнего изменения*
$ **Is -lu**	*Упорядочить файлы по времени последнего обращения*
$ Is -IS	*Упорядочить файлы по размеру*
$ Is -1i	*Показать индексные дескрипторы, ассоциированные со всеми файлами*
$ Is -In	*Показать числовые идентификаторы пользователя/группы вместо имен*
$ Is -lh	*Показать размеры файлов в удобочитаемой форме (Кбайт, Мбайт и т. д.)*
$ Is -1R	*Показать рекурсивно файлы из текущего каталога и подкаталогов*

Кроме того, при выводе на экран списка файлов в нем можно **по-разному отобразить файлы различных типов:**

```
$ Is -F                    Добавить символ как индикатор типа файла
myfile-symlinkfp config/ memo.txt pipefi 1 e| script,sh* xpid.socket=
$ Is --color=always Выделить типы файлов разными цветами
$ Is -C                    Отобразить файлы, расположив их в столбцах
```

В примере с использованием -F в выводе отображаются файлы нескольких типов, myfi 1 e-symlink@ является индикатором символьной ссылки на каталог, confid/ представляет собой обычный каталог, memo.txt — обычный файл (без дополнительных символов), pi pefi 1 e | — именованный канал (созданный с помощью mkfi fo), script, sh* — исполняемый файл, а xpi d. socket= является сокетом. В двух следующих примерах соответственно файлы различных типов выделяются разными цветами и вывод отображается в столбцах.

Проверка файлов

Когда файлы типа программных пакетов и образов CD или DVD распространяются посредством Интернета, с каждым из них часто поставляется файл SHA1SUM или md5sum.txt. Эти файлы содержат контрольные суммы, с помощью которых можно убедиться, что загруженный вами файл в точности соответствует находящемуся в репозитории.

Далее приведены примеры использования команд md5sum и shalsum для **генерирования контрольных сумм файлов:**

```
$ md5sum whatever.iso
d41d8cd98f00b204e9800998ecf8427e whatever.iso
$ shalsum whatever.iso
da39a3ee5e6b4b0d3255bfef95601890afd80709 whatever.iso
```

Выбор команды зависит от того, позаботился ли поставщик проверяемого вами файла о наличии информации md5sum или shalsum. Существуют также другие команды sha*, которые используют большее количество битов для шифрования. Например, вот так выглядят некоторые MD5-xenm ISO-образов дистрибутива Ubuntu Quanta! Quetzal:

```
7b7c56c74008da7d97bd49669c8a045d ubuntu-12.W-desktop-amd64+mac.iso
7ad57cadae955bd04019389d4b9cldcb ubuntu-12.10-desktop-amd64.iso
b4191cldld6fdf358cl54f8bf86b97dd ubuntu-12.10-desktop-i386.iso
b8a4d9513ed53dclbe05576113bll3e8 ubuntu-12.10-server-amd64+mac.iso
4bd3270bde86d7e4e017e3847a4af485 ubuntu-12.10-server-amd64.iso
...
```

Внутри каждого ISO-образа Ubuntu также есть файл md5sum.txt, который расположен в каталоге верхнего уровня этого образа. В данном файле содержатся контрольные суммы MD5 всех файлов в образе CD или DVD.

Сохранив контрольные суммы MD5 в файле, можете использовать его для проверки файлов, которые в нем указаны. Таким образом, у вас будет возможность проводить валидацию контрольных сумм MD5 множества файлов за один раз.

Например, я смонтировал ISO-образ ubuntu-ll.10-desktop-amd64.iso и запустил выполнение команды md5sum -c с верхнего уровня этого образа:

```
$ md5sum -c md5sum.txt
,/casper/initrd.lz: OK
./casper/filesystem.manifest-remove: OK
./casper/fiiesystem.manifest: OK
./casper/filesystem.squashfs: OK
./casper/vmlinuz: OK
./casper/fiiesystem.size: OK
./dists/oneiric/Release: OK
. . .
```

Чтобы **проверить только один из файлов, указанных** в соответствующем файле, вы можете сделать что-то наподобие следующего:

```
$ cat md5sum.txt | grep initrd.lz |md5sum -c
./dists/feisty/Release.gpg: OK
```

Если у вас будет файл SHA1SUM вместо md5sum.txt для проверки на соответствие, вы сможете воспользоваться командой shalsum аналогичным образом. Применяя комбинацию команды find, описанной ранее в этой главе, и команды md5sum, вы можете проверить любую часть своей файловой системы. Вот, например, как **создается контрольная сумма MD5 для каждого файла в каталоге /etc,** чтобы их можно было проверить позднее с целью выяснить, изменился ли какой-либо из этих файлов:

```
$ sudo find /etc -type f -exec md5sum {} \; > /tmp/md5.1st 2> /dev/null
```

Результатом выполнения приведенной выше строки команд будет файл /tmp/md5.1st, содержащий 128-битную контрольную сумму каждого файла в каталоге /etc. Вы могли бы позднее ввести приведенную далее команду, чтобы узнать, были ли внесены изменения в какой-либо из этих файлов:

```
$ cd /etc
$ md5sum -c /tmp/md5.list | grep -v 'OK'
./hosts.allow: FAILED
md5sum: WARNING: 1 of 1668 computed checksums did NOT match
```

Как видите, изменился только один файл (hosts.allow), поэтому следующим шагом будет проверка измененного файла и выяснение, были ли изменения внесены в этот файл намеренно.

Резюме

Существует множество команд для исследования и работы с файлами в Linux. Такие команды, как, например, chmod, позволяют изменять права доступа, связанные с файлами, а команды, к которым относятся lsattr и chattr, могут использоваться для отображения и изменения атрибутов файлов, ассоциированных с файловыми системами ext2 и ext3.

Для перемещения по файловой системе чаще всего используется команда cd. Однако для повторяющегося передвижения по одним и тем же каталогам вы можете прибегнуть к командам pushd и popd, которые позволяют работать со стеком каталогов.

Текст копируется с помощью команды cp. Вместе с тем для копирования файлов (например, образов дисков) с устройств (к примеру, с привода CD-ROM) можно использовать команду dd. Для создания каталогов вы можете задействовать команду `mkdi r.`

Вместо того чтобы хранить множество копий одного и того же файла в системе, вы можете прибегнуть к символьным и жестким ссылкам, чтобы несколько файловых имен указывали на один и тот же файл или каталог. Символьные ссылки могут располагаться где угодно в файловой системе, в то время как жесткие ссылки должны находиться в том же разделе, в котором размещен оригинальный файл.

Для поиска файлов Linux предлагает команды locate и find. Для проверки целостности файлов, загруженных из Интернета, вы можете воспользоваться командами md5sum и shalsum.

5 Манипуляции с текстом

В этой главе:

- сопоставление текста с использованием регулярных выражений;
- редактирование текстовых файлов с помощью vi, JOE или nano;
- использование графических текстовых редакторов;
- отображение текста с помощью команд cat, head и tail;
- постраничный просмотр текста с использованием less и more;
- разбивка текста на страницы с помощью команды pr;
- поиск текста с помощью grep;
- подсчет слов, строк и символов с использованием wc;
- сортировка вывода посредством sort;
- потоковое редактирование с помощью sed, tr, cut и awk;
- поиск текста в двоичных файлах с использованием strings;
- выявление различий между файлами посредством diff;
- конвертирование текстовых файлов с помощью unix2dos/dos2unix.

В первых UNIX-системах (которые выступили в качестве основы Linux) был доступен только интерпретатор команд, в силу чего работа в них подразумевала главным образом использование команд и взаимодействие с обычными текстовыми файлами. Документы, программный код, конфигурационные файлы, электронная почта и почти все, что вы создавали или конфигурировали, представлялось текстовыми файлами. Для манипуляций с ними первые разработчики создали множество инструментов, позволяющих выполнять операции с текстом.

Несмотря на наличие графических инструментов для работы с текстом, большинство опытных Linux-пользователей считает инструменты командной строки более эффективными и удобными. Такие текстовые редакторы, как, например, vi (vim), Emacs, JOE, nano и Pico, доступны в большинстве дистрибутивов Linux. Команды вроде grep, sed и awk можно использовать для поиска и, возможно, изменения фрагментов информации в текстовых файлах.

В этой главе демонстрируется, как использовать множество популярных команд для работы с текстовыми файлами в Ubuntu. В ней также исследуются некоторые не столь распространенные способы использования команд для манипуляций с текстом, которые могут показаться вам интересными.

Сопоставление текста с использованием регулярных выражений

Многие инструменты для работы с текстом позволяют использовать *регулярные выражения*, иногда называемые *regex* (regular expressions — «регулярные выражения»), для выявления текста, который вы ищете на основе того или иного шаблона. Вы можете использовать такие шаблоны для поиска текста в текстовом редакторе либо задействовать их в сочетании с командами поиска для сканирования множества файлов на предмет наличия нужных вам строк текста.

Шаблон поиска с использованием регулярных выражений может включать определенную строку текста (например, такое слово, как Linux) либо местоположение (к примеру, конец строки или начало слова). Кроме того, поиск может быть узким (найти именно слово he! 1 o) либо широким (найти любое слово, начинающееся с h и заканчивающееся на o).

В приложении Б приведена справочная информация о метасимволах интерпретатора команд, которые можно использовать в сочетании с регулярными выражениями для выявления нужных вам точных соответствий. В этом разделе будут рассмотрены примеры применения регулярных выражений наряду с несколькими инструментами, с которыми вы будете сталкиваться на протяжении всей текущей главы.

В приведенном далее списке демонстрируются некоторые примеры, в которых используются основные регулярные выражения для сопоставления текстовых строк (табл. 5.1).

Регулярные выражения многих типов встречаются в примерах по ходу всей этой главы. Имейте в виду, что не каждая команда, которая включает регулярные выражения, одинаково использует их параметры.

Таблица 5.1. Основные регулярные выражения для сопоставления текстовых строк

Выражение	Соответствия
a*	a, ab, abc и aecjejich
^a	Любое a, находящееся в начале строки
*a$	Любое a, расположенное в конце строки
a.c	Трехсимвольные строки, начинающиеся на c и заканчивающиеся на c
[bcfjat	bat, cat или fat
[a-d]at	aat, bat, cat, dat, но не Aat, Bat и т. д.
[A-D]at	Aat, Bat, Cat и Dat, но не aat, bat и т. д.
1[3-5]7	137, 147 и 157
\tHello	Символ табуляции, предшествующий слову Hello
\.[tT][xX][Tt]	.txt, .TXT, .TxT или другие комбинации регистров

Редактирование текстовых файлов

В мире Linux/UNIX есть много текстовых редакторов. Наиболее распространенный — vi, который можно найти практически в любой доступной сегодня UNIX- или Linux-системе. Вот почему умение вносить хотя бы незначительные изменения в файлы в редакторе vi является важнейшим практическим навыком для любого Linux-администратора. Если вы однажды будете работать в минималистской незнакомой Linux-среде, пытаясь вернуть сервер в режим онлайн, vi окажется инструментом, который почти всегда будет под рукой.

Убедитесь, что в Ubuntu у вас установлен пакет vim-enhanced, vim (Vi IMproved) с пакетом vim-enhanced предоставит в ваше распоряжение самый современный, функционально богатый и дружественный по отношению к пользователю редактор vi. Более подробно об использовании vi вы можете узнать из приложения А.

ПРИМЕЧАНИЕ

В Ubuntu по умолчанию устанавливается vim.

По традиции еще одним популярным текстовым UNIX-редактором является Ema.cs и его более графически насыщенный вариант XEMacs. Emacs — это мощный многофункциональный инструмент, который также может выступать в качестве программы для чтения почты/новостей либо в роли интерпретатора команд, а также выполнять другие функции. Кроме того, Emacs известен своими очень сложными клавиатурными сокращениями, для соответствующего использования которых нужно иметь три руки.

В середине 1990-х годов Emacs опережал vi в плане функциональности. Сейчас, когда vim широко доступен, они оба способны предоставить все возможности, связанные с редактированием, которые вам когда-либо потребуются. Если вы еще не знакомы ни с vi, ни с Emacs, рекомендую начать с изучения vi.

Есть много других текстовых редакторов командной строки, а также текстовых редакторов с GUI-интерфейсом, доступных для Linux. К редакторам командной строки, которые могут показаться вам проще по сравнению с vi и Emacs, относятся JEDJOE и nano. Запустите любой из этих редакторов, введя имя соответствующей команды, затем указав (не обязательно) имя файла, который вы желаете отредактировать. В приведенных далее разделах содержится краткое описание того, как использовать каждый из этих редакторов.

Редактор JOE

Если вам доводилось пользоваться классическими текстовыми процессорами, например WordStar, работающими с текстовыми файлами, то, возможно, вы будете чувствовать себя комфортно и в редакторе JOE. Чтобы использовать его, установите пакет JOE. Для применения средства проверки орфографии в JOE убедитесь, что у вас установлен пакет Aspell (он устанавливается в Ubuntu по умолчанию). Чтобы установить JOE, введите следующую команду:

```
$ sudo apt-get install joe
```

При использовании **JOE,** вместо того чтобы переходить в командный или текстовый режим, вы всегда будете готовы сразу начать печатать. Для перемещения по содержимому файла вы можете использовать управляющие символы или клавиши со стрелками. Чтобы **открыть файл для редактирования,** просто введите joe и имя соответствующего файла либо воспользуйтесь одним из следующих параметров:

$ joe memo.txt *Открыть memo.txt для редактирования*
$ joe -wordwrap memo.txt *Включить перенос слов при редактировании*
$ joe -lmargin 5 -tab 5 memo.txt *Задать отступ от левого края и табуляцию как равные 5*
$ joe +25 memo.txt *Начать редактирование со строки 25*

Чтобы **добавить текст,** просто начните печатать. Вы можете **использовать клавиатурные сокращения** для реализации многих функций. Используйте клавиши со стрелками для перемещения курсора влево, вправо, вверх и вниз. Нажмите **Delete** для удаления текста справа от курсора и **Backspace** для удаления текста слева от курсора. Используйте клавишу **Enter,** чтобы добавить разрыв строки. Нажмите комбинацию **Ctrl+K+H,** чтобы увидеть справочный экран. Далее приведен список наиболее часто используемых сочетаний клавиш для редактирования в **JOE** (табл. **5.2).**

Таблица 5.2. Сочетания клавиш при редактировании текста в ЗОЕ

Сочетание клавиш	Результат
Перемещение курсора	
Ctrl+B	Влево
Ctrl+P	Вверх
Ctrl+F	Вправо
Ctrl+N	Вниз
Ctrl+Z	Предыдущее слово
Ctrl+X	Следующее слово
Поиск	
Ctrl+K+F	Найти текст
Ctrl+L	Найти далее
Блок	
Ctrl+K+B	Начало
Ctrl+K+K	Конец
Ctrl+K+M	Переместить блок
Ctrl+K+C	Скопировать блок
Ctrl+K+W	Записать блок в файл
Ctrl+K+Y	Удалить блок
Ctrl+K+/	Применить фильтр

Продолжение &

Таблица 5.2 *(продолжение)*

Сочетание клавиш	Результат
Разное	
Ctrl+K+A	Центрировать строку
Ctrl+T	Параметры
Ctrl+R	Обновить
Файл	
Ctrl+K+E	Открыть новый файл для редактирования
Ctrl+K+R	Вставить файл в позицию курсора
Ctrl+K+D	Сохранить
Переход	
Ctrl+U	Предыдущий экран
Ctrl+V	Следующий экран
Ctrl+A	Начало строки
Ctrl+E	Конец строки
Ctrl+K+U	Начало файла
Ctrl+K+V	Конец файла
Ctrl+K+L	К строке номер
Удаление	
Ctrl+D	Удалить символ
Ctrl+Y	Удалить строку
Ctrl+W	Удалить слово справа
Ctrl+O	Удалить слово слева
Ctrl+J	Удалить часть строки справа
CtrlH—	Отменить операцию
Ctrl+6	Вернуть отмененную операцию
Выход	
Ctrl+K+X	Сохранить и выйти
Ctrl+C	Прервать
Ctrl+K+Z	Интерпретатор команд
Ctrl+[+N	Слово
Ctrl+[+L	Файл

Редакторы Pico и nano

Pico — популярный и очень маленький текстовый редактор, распространяемый как часть клиента электронной почты Pine. Pico относится к свободному программному обеспечению, однако его исходный код не является по-настоящему открытым. В силу этого многие дистрибутивы Linux, включая Ubuntu, не предоставляют Pico.

Вместо него они предлагают клон Pico с открытым исходным кодом под названием nano (сочетание первой буквы слова папо (нано) и первых трех букв от another editor (другой редактор)). В этом разделе описывается редактор папо.

ПРИМЕЧАНИЕ----------

В Ubuntu команда pico связана с программой, которой является редактор папо.

Компактный текстовый редактор папо (представленный командой папо) запускается из интерпретатора команд, однако является экранно-ориентированным (в силу того, что он основан на библиотеке curses), папо популярен среди тех, кто раньше пользовался клиентом электронной почты Pine, поскольку предлагает те же функциональные возможности, связанные с редактированием, что и редактор Pico клиента электронной почты Pine.

В редких случаях, когда у вас не будет редактора vi в Linux-системе (например, если вы установите минимальный дистрибутив Gentoo Linux), папо почти всегда будет доступен.

папо устанавливается в Ubuntu по умолчанию. Для выполнения проверки орфографии в папо вам потребуется команда spel 1, а не aspel 1.

Как и в случае с редактором JOE, вместо перехода в командный режим или режим набора текста вы сможете сразу начать печатать. Чтобы **открыть текстовый файл для редактирования,** просто введите папо и имя соответствующего файла либо воспользуйтесь одним из следующих параметров:

$ nano memo.txt	*Открыть memo.txt для редактирования*
$ папо -B memo.txt	*Сохранить резервную копию предыдущего варианта в -.имя файла*
$ папо -m memo.txt	*Активизировать мышь для перемещения курсора (если это поддерживается)*
$ папо +83 memo.txt	*Начать редактирование со строки 83*

Параметр командной строки -т активизирует поддержку мыши. Вы сможете использовать мышь для выбора положения в тексте, в котором в итоге окажется курсор. Однако после первого щелчка кнопкой мыши в папо выделится блок текста, что может оказаться не тем, что вы хотели сделать.

Как и в случае с JOE, чтобы **добавить текст,** просто начните печатать. Используйте клавиши со стрелками для перемещения курсора влево, вправо, вверх или вниз. Нажмите **Delete,** чтобы удалить текст справа от курсора, и **Backspace** для удаления текста слева от курсора. Чтобы добавить разрыв строки, нажмите клавишу **Enter.** Для отображения справочного текста используйте комбинацию **Ctrl+G.** В приведенном далее списке вы найдете управляющие коды для папо, описание которых содержит соответствующий справочный экран (табл. 5.3).

Таблица 5.3. Управляющие коды для папо

Управляющий код	Функциональная клавиша	Описание
Ctrl+G	F1	Показать справочный текст (для выхода из справки нажмите Ctrl+X)

Продолжение

Таблица 5.3 *(продолжение)*

Управляющий код	Функциональная клавиша	Описание
Ctrl+X	F2	Выйти из nano (или закрыть текущий буфер файла)
Ctrl+O	F3	Сохранить текущий файл
Ctrl+J	F4	Выровнять текущий текст в текущем абзаце
Ctrl+R	F5	Вставить содержимое файла в текущий файл
Ctrl+W	F6	Искать текст
Ctrl+Y	F7	Перейти к предыдущему экрану
Ctrl+V	F8	Перейти к следующему экрану
Ctrl+K	F9	Вырезать (и сохранить) текущую строку или выделенный текст
Ctrl+U	F10	Вернуть (вставить) в файл ранее вырезанную строку
Ctrl+C	FII	Показать текущую позицию курсора
Ctrl+T	F12	Начать проверку орфографии
Ctrl+-		Перейти к строке и столбцу с выбранными номерами
Ctrl+\		Найти и заменить текст
Ctrl+6		Выделить текст, Начиная с позиции курсора (чтобы снять выделение, снова нажмите Ctrl+6)
Ctrl+F		Перейти вперед на один символ
Ctrl+B		Перейти назад на один символ
Ctrl+Пробел		Перейти вперед на одно слово
Alt+Пробел		Перейти назад на одно слово
Ctrl+P		Перейти к предыдущей строке
Ctrl+N		Перейти к следующей строке
Ctrl+A		Перейти в начало текущей строки
Ctrl+E		Перейти в конец текущей строки
Alt+(		Перейти в начало текущего абзаца
Alt+)		Перейти в конец текущего абзаца
Alt+\		Перейти к первой строке файла
Alt+/		Перейти к последней строке файла
Alt+]		Перейти к квадратной скобке, пару которой составляет текущая квадратная скобка
Alt+=		Прокрутить на одну строку вниз
AltH—		Прокрутить на одну строку вверх

Графические текстовые редакторы

Для редактирования текста вы не обязаны использовать редактор командной строки. Основное преимущество графического текстового редактора заключается в том, что вы можете использовать мышь для выбора элементов меню, выделения текста, его вырезания и копирования, а также чтобы задействовать специальные плагины.

Если в вашей Linux-системе установлен рабочий стол GNOME, можете рассчитывать на наличие текстового редактора GNOME (gedit). Его функциональность позволяет проверять орфографию, выводить на экран статистику документов, изменять отображаемые шрифты и цвета, а также распечатывать документы. У рабочего стола KDE также имеется текстовый редактор — KDE (KEdit в пакете `kdeutils`). Он обладает теми же функциональными возможностями, что и текстовый редактор GNOME, а также несколькими дополнительными, например позволяет отправить текущий документ посредством KMail или другого конфигурируемого пользователем компонента KDE.

vim как таковому сопутствует версия, имеющая GUI-интерфейс X. Эта версия запускается с помощью команды `grim`, которая является частью пакета `vim-Xll`. Если вы захотите превратить vim с GUI-интерфейсом в более дружественный по отношению к пользователю текстовый редактор, то можете загрузить стороннюю конфигурацию под названием Cream. Чтобы установить ее, введите `sudo apt-get install cream`.

ПРИМЕЧАНИЕ----------

Чтобы использовать gvim, вам потребуется установить дополнительный пакет — vim-gnome.

К прочим текстовым редакторам, которые вы можете установить, относятся NEdit (с функционалом, позволяющим использовать макросы и выполнять команды интерпретатора команд и рассчитанный на разработчиков программного обеспечения) и Leafpad (схож с текстовым редактором Windows Блокнот). Текстовый редактор Scribes (`scribes`) обладает расширенными функциональными возможностями для автоматического исправления, замены, обеспечения отступов и завершения ввода слов.

Отображение, сортировка и изменение текста

Вместо того чтобы просто редактировать одиночные текстовые файлы, вы можете использовать целый набор Linux-команд для отображения, поиска и манипулирования содержимым нескольких текстовых файлов за раз.

Отображение текстовых файлов

Самый простой способ отобразить содержимое текс тового файла заключается в использовании команды `cat`. Команда `cat` конкатенирует (выводит строку символов)

содержимое текстового файла с вашим экраном (по умолчанию). Кроме того, вы можете использовать различные метасимволы интерпретатора команд, чтобы **направлять содержимое этого файла разными путями.** Например:

```
$ cat myfi1e.txt                  Отправить все содержимое файла на экран
$ cat myfile.txt > copy.txt       Направить содержимое файла в другой файл
$ catmyfile.txt » myotherfile.txt Добавить содержимое файла в другой файл
$ cat    -s myfile.txt            Отобразить последовательные пустые строки как одну
$ cat    -n myfile.txt            Отобразить номера строк в выводе
$ cat    -b myfile.txt            Отобразить номера строк, не являющихся пустыми
```

Однако если ваш блок текста будет простираться на более чем несколько строк, то использование команды cat нецелесообразно. В этом случае вам потребуются инструменты получше, чтобы взглянуть на начало или конец либо постранично просмотреть весь текст целиком.

Для **просмотра начала файла** используйте команду head:

```
$ head myfile.txt
$ cat myfile.txt | head
```

В обеих строках команд для вывода на экран первых десяти строк файла используется head. Вы можете указать в качестве параметра любое число, чтобы отобразить это количество строк с самого начала файла. Например:

```
$ head -п     50 myfile.txt   Отобразить первые пятьдесят строк файла
$ ps auwx     | head -n 15    Отобразить первые 15 строк вывода ps
```

Это также можно сделать с помощью устаревшего (но более короткого) синтаксиса:

```
$ head -50 myfile.txt
$ ps auwx | head -15
```

Вы можете использовать команду `tail` аналогичным образом для **просмотра конца файла:**

```
$ tail -п      15 myfile.txt  Отобразить последние 15 строк файла
$ tail -15 myfile.txt         Отобразить последние 15 строк файла
$ ps auwx      | tail -n 15   Отобразить последние 15 строк вывода ps
```

Команду `tail` также можно применять для **непрерывного просмотра конца файла** по мере того, как этот файл записывается другой программой. Очень полезно читать файлы журналов, создаваемые в реальном времени, при устранении неполадок в случае с Apache, Sendmail и многими другими системными службами (некоторые из этих журналов не появятся, пока не установлено соответствующее приложение):

```
# tail -f /var/1og/messages      Просмотр системных сообщений в реальном
                                 времени
# tail -f /var/1og/mai1.log      Просмотр сообщений почтового сервера
                                 в реальном времени
# tail -f /var/1 og/httpd/accessJog  Просмотр сообщений веб-сервера в реальном
                                 времени
```

Вы можете нажать комбинацию клавиш **Ctrl+C,** чтобы завершить выполнение команды tail -t.

Постраничный просмотр текста

Если у вас имеется большой текстовый фрагмент и вы желаете взглянуть не только на его начало или конец, потребуется инструмент для **постраничного просмотра текста.** Оригинальным системным UNIX-инструментом постраничного вывода была команда more:

```
$ ps auwx | more Постраничный просмотр вывода ps (нажмите клавишу пробела)
$ more myfile.txt Постраничный просмотр содержимого файла
```

Однако у more имеются некоторые ограничения. Например, в строке с ps, приведенной выше, more не разрешает прокрутку вверх. Команда 1 ess была создана как более мощный и дружественный по отношению к пользователю аналог more. При описании команды less обычно говорят: «What is less? less is more!» (дословно: «Что такое меньше (less)? Меньше — это больше (more)»). Рекомендую вам больше не использовать команду more, а применять вместо нее less.

ПРИМЕЧАНИЕ

У команды less есть еще одно преимущество. В отличие от текстовых редакторов, например vi, она не осуществляет чтение всего файла при его открытии. Благодаря этому сокращается время открытия больших файлов для их просмотра.

Команду 1 ess можно использовать в сочетании с тем же синтаксисом, что и more в приведенных выше примерах:

```
$ ps auwx | less          Постраничный просмотр вывода ps
$ cat myfile.txt | less   Постраничный просмотр содержимого файла
$ less myfile.txt         Постраничный просмотр текстового файла
```

Команда less позволяет **осуществлять навигацию** с использованием таких клавиш, как T, 4-, **Page Up, Page Down** и **Пробел.** Если вы примените команду less в отношении файла (речь не идет о стандартном вводе), то нажмите клавишу **V,** чтобы открыть текущий файл в редакторе. Какой именно редактор запустится, будет зависеть от переменных среды, которые определены для вашей учетной записи. Имя редактора будет взято из переменной среды VISUAL, если она определена, либо из EDITOR, если VISUAL не определена. Если ни одна из этих переменных не задана, то less запустит редактор JOE в Ubuntu.

ПРИМЕЧАНИЕ

В других версиях Linux в качестве редактора по умолчанию в таких ситуациях используется vi.

Нажмите комбинацию клавиш **Ctrl+C,** чтобы выйти из этого режима. Как и в vi, при просмотре файла с использованием 1 ess вы можете **выполнить поиск строки,** для чего нужно нажать клавишу / (прямой слеш), затем указать искомую строку и щелкнуть **Enter.** Для поиска других случаев употребления этой строки повторно нажимайте клавиши / и **Enter.**

Для **прокрутки вперед и назад** при применении 1 ess используйте соответственно клавиши F и B. Например, нажатие F десять раз позволит прокрутить текст вперед на десять строк, а нажав **15** раз клавишу **B,** вы сможете прокрутить его назад на 15 строк. Нажмите клавишу D, чтобы прокрутить вниз на половину экрана, и U, чтобы прокрутить на половину экрана вверх.

Разбивка текста на страницы с помощью pr

Команда pr позволяет быстро отформатировать фрагмент текста так, чтобы его можно было распечатать. Она может оказаться очень полезной, если вы захотите распечатать результаты выполнения некоторых команд без необходимости открывать текстовый процессор или редактор. С помощью pr вы можете **отформатировать текст, разбив его на страницы с заголовочной информацией,** такой как дата, время, имя файла и номер страницы. Вот пример:

```
$ dpkg-query -1 | sort ] pr --co1umn=2 | less        Разбить на страницы список из 2 столбцов
```

В этом примере команда dpkg-uery -1 отображает список всех программных пакетов, установленных в вашей системе, и передает его по каналу команде sort для сортировки в алфавитном порядке. Затем этот список поступает по каналу команде pr, которая преобразует одностолбцовый список в двухстолбцовый (--co1umns=2) и разбивает его на страницы. Наконец, команда less позволяет вам постранично просмотреть текст.

Вместо постраничного просмотра вывода вы можете **отправить его в файл или на принтер.** Вот примеры:

```
$ dpkg-query -1 | sort | pr --column=2 > pkg.txt      Отправить вывод pr в файл
$ dpkg-query -1 | sort | pr --column=2 | 1pr          Распечатать вывод pr
```

К другим **манипуляциям с текстом,** которые вы можете осуществлять с помощью команды pr, относятся установка двойных интервалов между строками (-d), отображение управляющих символов (-c) и смещение текста на определенное количество пробелов относительно левого края (например, -o 5 обеспечивает отступ на пять пробелов слева).

Поиск текста с помощью grep

Команда grep придется кстати, когда вам потребуется расширенный поиск строк в файле. Фактически выражение to grep вошло в компьютерный жаргон как глагол подобно тому, как выражение to Google («гуглить») стало частью разговорного языка. Вот примеры использования команды grep:

```
$ grep Remote /etc/services          Отобразить строки, содержащие слово Remote
# grep sudo /var/1og/auth.log        Отобразить строки, содержащие число 404
$ ps auwx | grep init                Отобразить строки init из вывода ps
$ ps auwx | grep "\[*\]"             Отобразить заключенные в квадратные скобки команды
$ dmesg | grep "[ ]ata\|*ata"        Отобразить информацию об ATA-устройствах ядра
```

Эти строки команд имеют конкретное применение, помимо того что являются примерами использования grep. Если вы выполните поиск слова sudo в auth.log, то узнаете, когда и с какими параметрами выполнялась команда sudo. Отображение заключенных в квадратные скобки команд, содержащихся в выводе ps, представляет собой способ увидеть команды, параметры которых ps не может показать. Последняя команда проверяет кольцевой буфер ядра на наличие информации о любых устройствах с ATA-интерфейсом, таких, например, как жесткие диски и приводы CD-ROM.

Команда grep также позволяет **выполнять одновременный рекурсивный поиск в нескольких или сразу многочисленных файлах.** Если у вас установлен пакет apache2, то с помощью приведенной далее команды вы сможете осуществить поиск строки Vi rtualHost в кататогах /'etc/apache2/sites-enabled и /etc/apache2/conf.d:

```
$ grep -R VirtualHost /etc/apache2/conf.d /etc/apache2/sites-enabled
```

Добавьте номера строк (-n) в свою команду, чтобы **найти именно те строки,** в которых присутствуют нужные элементы:

```
#  grep -Rn VirtualHost /etc/apache2/*conf*
```

По умолчанию искомые элементы отображаются в цвете в каждой из найденных строк. Чтобы явно дать указание окрасить нужный элемент в результатах поиска, добавьте параметр - -color:

```
#  grep --color -Rn VirtualHost /etc/apache2/*conf*
```

По умолчанию при многофайловом поиске название соответствующего файла отображается в каждом результате поиска. Используйте параметр -h, чтобы **деактивизировать отображение имен файлов.** В следующем примере демонстрируется поиск строки sshd в файле auth. 1 од:

```
#  grep -h sshd /var/1og/auth.log
```

Чтобы при поиске в сообщениях **игнорировался регистр,** задействуйте параметр - i:

```
$ grep -i acpi /var/1од/dmesg          Искать в файле слово acpi (независимо
                                       от регистра)
```

Чтобы **отобразить только имя файла,** содержащего искомый элемент, добавьте параметр -1:

```
$ grep -R1 VirtualHost /etc/apache2
```

Для **отображения всех строк, в которых не обнаружено соответствия указанной строке,** добавьте параметр -v:

```
$ grep -v " 200 " /var/1og/apache2/access_* Отобразить строки, в которых
                                            нет " 200 "
```

ПРИМЕЧАНИЕ

При передаче по каналу вывода ps команде grep вы можете прибегнуть к следующему трюку, чтобы предотвратить появление процесса grep в результатах grep: # ps auwx | grep "[ijnit".

Подсчет количества слов с помощью wc

Возможны ситуации, в которых вам потребуется узнать количество строк, содержащих соответствие строке поиска. Команду wc можно использовать для **подсчета строк,** которые она получит. Например, приведенная далее команда показывает, сколько в файле журната Apache зафиксировано обращений с определенного IP-адреса:

```
$ grep 192.198.1.1 /var/1og/apache2/access.log | wc -1
```

Команда wc также может использоваться для других целей. По умолчанию **wc отображает количество строк, слов и байтов в файле:**

```
$ wc /var/1 og/dmesg          Отобразить количество для одиночного файла
436 3847 27984 /var/1og/dmesg
$ wc /var/log/*.log           Отобразить отдельные значения для каждого файла
                                и общие для всех файлов
   305    3764   25772  /var/1og/auth.1og
   780    3517   36647  /var/log/bootstrap.log
   350    4405   39042  /var/log/daemon.log
 10109   60654  669687  /var/1og/dpkg.log
    71     419    4095  /var/1og/fontconfig.log
  1451   19860  135252  /var/1og/kern.log
     0       0       0  /var/1og/1pr.1og
     0       0       0  /var/1og/mai1.log
     0       0       0  /var/1og/pycentral.log
     0       0       0  /var/1og/scrol1 keeper.log
   108    1610   13864  /var/log/user.log
     0       0       0  /var/log/uucp.log
    12      43     308  /var/1og/wvdialconf.log
   890    6717   46110  /var/log/Xorg.0.1og
 14076  100989  970777  total
```

Сортировка вывода с помощью sort

В определенных случаях целесообразна **сортировка содержимого файла или вывода команды.** Это поможет навести порядок в неаккуратном выводе. **В** приведенных далее примерах демонстрируются имена служб и соответствующие номера из файла /etc/servi ces и выполняется сортировка результатов в алфавитно-числовом порядке (прямом и обратном):

```
$ cat /etc/services | sort  Сортировать в алфавитно-числовом порядке
$ cat /etc/services | sort -r  Сортировать в обратном алфавитно-числовом порядке
```

Следующая команда **сортирует процессы по уровню использования оперативной памяти в убывающем порядке** (четвертое поле вывода команды ps). Параметр - к определяет ключевое поле для использования при сортировке; 4,4 является индикатором того, что именно четвертое поле — ключевое.

```
$ ps auwx | sort -r -k 4,4 j less
```

Приведенная далее строка команд **сортирует загруженные модули ядра по размеру в возрастающем порядке.** Параметр n дает sort указание рассматривать второе поле как номер, а не строку:

```
$ lsmod | sort -k 2,2n
```

Поиск текста в двоичных файлах с помощью strings

Иногда возникает необходимость прочитать ASCII-текст, содержащийся в двоичном файле. Часто таким путем можно многое узнать о выполняемом файле. В подобных ситуациях используйте strings для **извлечения всего удобочитаемого ASCII-текста.** Команда strings является частью пакета bi nut i I s и по умолчанию устанавливается в Ubuntu. Вот несколько примеров:

```
$ strings /bin/ls | grep -i libc          Найти случаи употребления libc в Is
$ cat /bin/ls | strings                   Показать весь ASCII-текст в Is
$ strings /bin/ls                         Показать весь ASCII-текст в Is
$ strings /usr/sbin/sshd | grep libwrap Показать библиотеку TCP Wrapper
```

Замена текста с помощью sed

Поиск текста в файле иногда бывает первым шагом на пути к замене текста. Потоки текста редактируются с помощью команды sed. На самом деле она является полноценным языком сценариев. В приведенных в этой главе примерах рассматриваются принципы замены текста посредством sed.

Если вам знакомы команды для замены текста в vi, вы заметите, что у sed имеются сходства с ними. В приведенном далее примере **в каждой строке произошла бы замена только первого появления** tcp на TEST. Здесь sed принимает вывод из канала, в то же время отправляя свой вывод на stdout (ваш экран):

```
$ cat /etc/services | sed s/tcp/TEST/ | head -n 20
TESTmux     l/t;cp      # TCP port service multiplexer
Echo        7/TEST
Echo        7/udp
discard     9/TEST      sink null
```

Добавление g в конец строки замены, как показано в следующем примере, приводит к тому, что в каждом случае применения tcp она заменяется на TEST. Кроме того, в данном примере ввод направляется из файла myfile.txt, а вывод — в файл mynewfile.txt:

```
$ sed s/tcp/TEST/д < /etc/services > mynewfile.txt
$ head -20 mynewfile.txt
TESTmux     1/TEST      # TCP port service multiplexer
Echo        7/TEST
Echo        7/udp
discard     9/TEST      sink null
```

Чтобы сделать команду sed чувствительной к регистру, добавьте в строку команд параметр i:

```
$ cat /etc/services | sed s/tcp/TEST/gi | head -n 20
TESTmux 1/TEST # TEST port service multiplexer
```

В следующем примере демонстрируется замена первых случаев появления текста /home/chris на /home2/chris из файла /etc/passwd (следует отметить, что эта команда не изменяет соответствующий файл, а выводит измененный текст). Это может пригодиться при переносе учетных записей пользователей в новый каталог (возможно, расположенный на новом диске), которому было весьма предусмотрительно присвоено имя home2. Здесь приходится использовать кавычки и обратные слеши для экранирования прямых слешей, чтобы они не интерпретировались как разделители:

```
$ sed 's/\/home\/chris/\/home2\/chris/g' < /etc/passwd | grep chris
chris:x:1000:1000:Chris Negus..,:/home2/chris:/bin/bash
```

Несмотря на то что прямой слеш является разделителем по умолчанию в случае с командой sed, вы можете **заменить его** на любой другой символ на ваш выбор. Замена разделителя облегчит вам работу, когда в строке будут слеши. Например, предыдущую строку команд, в которой содержится путь, можно было бы заменить на эту:

```
$ sed 's./home/chris./home2/chris.g' < /etc/passwd | grep chris
chris:x:1000:1000:Chris Negus,,,:/home2/chris:/bin/bash
```

В этой строке в качестве разделителя используется точка (.).

Команду sed можно использовать для **выполнения множественных замен за один раз,** вставив перед каждой из них -e. Здесь в тексте, исходящем из /etc/services, каждый раз tcp заменяется на LOWER, а TCP — на UPPER:

```
$ sed -e s/tcp/LOWER/g -e s/TCP/UPPER/g /etc/services | head -n 20
LOWERmux 1/LOWER          # UPPER port service multiplexer
Echo     7/LOWER
Echo     7/udp
discard 9/LOWER           sink null
```

Вы можете использовать sed для **добавления символов новой строки в поток текста.** Нажмите клавишу **Enter,** когда увидите слово Enter. Символ > во второй строке генерируется интерпретатором команд, то есть его вводить не нужно.

```
$ echo aaabccc | sed 's/b/\Enter
> Г
aaa
ccc
```

Только что показанный трюк не работает слева от команды замены sed. Чтобы заменить символы новой строки, лучше воспользоваться командой tr.

Преобразование и удаление символов с помощью tr

Использование команды tr — легкий способ **выполнения простых преобразований символов на лету.** В приведенном далее примере новые строки заменяются пробелами, благодаря чему все файлы, отображаемые из текущего каталога, выводятся в одной строке:

```
$ Is | tr '\n' 1 ' Заменить символы новой строки пробелами
```

Команду tr можно использовать для **замены одного символа другим,** однако она не работает со строками, в отличие от sed. Приведенная далее команда заменяет все экземпляры буквы f (в нижнем регистре) на F (в верхнем):

```
$ tr f F < /etc/services Заменить каждую букву f в файле на F
```

Вы также можете использовать команду tr для **удаления символов.** Вот два примера:

```
$ Is | tr -d '\л'          Удалить новые строки (чтобы в результате получилась
                           одна строка)
$ tr -d f < /etc/services Удалить все буквы f из файла
```

Команда tr позволяет проделывать изящные трюки при **определении диапазонов символов для работы.** Далее приведен пример изменения регистра букв с нижнего на верхний:

```
$ echo chris | tr a-z A-Z Преобразовать chris в CHRIS
CHRIS
```

Аналогичного результата можно добиться посредством следующего синтаксиса:

```
$ echo chris | tr '[:1ower:]1 '[:upper:3' Преобразовать chris в CHRIS
```

Выявление различий между двумя файлами с помощью diff

Если у вас есть две версии одного файла, может быть полезным **знать, какие различия имеются между этими двумя файлами.** Например, при обновлении программного пакета вы сохранили свой старый конфигурационный файл под новым именем, скажем config.old или config. bak, чтобы сберечь свою конфигурацию. Если возникнет такая ситуация, можете воспользоваться командой di f f для выяснения, какими строками отличаются ваша старая и новая конфигурации, чтобы произвести их слияние. Например:

```
$ diff config conf ig. old
```

Вы можете перевести вывод di ff в так называемый *унифицированный формат,* который может оказаться проще для восприятия. Он предполагает добавление трех строк контекста до и после каждого блока измененных строк, о которых сообщает,

с последующим использованием символов + и - для демонстрации разницы между файлами. Приведенный далее набор команд создает файл (fl.txt), содержащий последовательность чисел (1-7), и файл (f2.txt) с одним из этих чисел, которое было изменено (с использованием sed), а затем сравнивает эти два файла посредством команды diff:

```
$ seq 1 7 > fl.txt                    Отправить последовательность чисел в fl.txt
$ cat fl.txt                          Отобразить содержимое fl.txt
1
2
3
4
5
6
7
$ sed s/4/FOUR/ < fl.txt > f2.txt Заменить 4 на FOUR и отправить в f2.txt
$ diff fl.txt f2.txt
4c4                                   Показывает, что в файле была изменена строка 4
< 4
---
> FOUR
$ diff -u fl.txt f2.txt               Отобразить унифицированный вывод diff
--- fl.txt 2007-09-07 18:26:06.000000000 -0500
+++ f2.txt 2007-09-07 18:26:39.000000000 -0500
@<a -1.7 +1,7 @@
1
2
3
-4
+FOUR
5
6
7
```

В этом выводе diff -и дополнительно отображается такая информация, как даты и время изменений, если сравнивать его с обычным выводом diff. Команду sdiff можно использовать для создания еще одного визуального представления. Она позволяет производить **слияние вывода двух** файлов интерактивно, как показано в следующем выводе:

```
$ sdiff fl.txt f2.txt
1                                     1
2                                     2
3                                     3
4                                   | FOUR
5                                     5
6                                     6
7                                     7
```

Другой вариацией на тему diff является команда vimdiff, которая открывает одновременно два файла в редакторе vim и выделяет цветом различия между ними. Аналогичным образом gvi mdi ff открывает два файла в gvim.

ПРИМЕЧАНИЕ

Для запуска программы gvim или gvimdiff вам потребуется установить пакет vim-gnome.

Вывод diff -и можно передать команде patch. Она принимает как ввод старый файл и файл от diff и генерирует в качестве вывода файл, снабженный патчем. Следуя предыдущему примеру, я использую команду diff в случае с двумя файлами для создания патча с последующим его применением к первому файлу:

```
$ diff -u fl.txt f2.txt > patchfile.txt
$ patch fl.txt < patchfile.txt
patching file fl.txt
$ cat fl.txt
1
2
3
FOUR
5
6
7
```

Именно так многие OSS-разработчики (включая разработчиков ядра) распространяют свои программные патчи. Команды patch и di ff также можно выполнять для целого дерева каталогов. Например, приведенный далее набор команд рекурсивно копирует каталог /etc/pam.d в /tmp/newpam.d и /tmp/oldpam.d, а затем вносит изменения в два файла в каталоге newpam. d:

```
# cp -r /etc/pam.d /tmp/oldpam.d
# cp -r /etc/pam.d /tmp/newpam.d
# echo hello » /tmp/newpam.d/atd
# echo goodbye » /tmp/newpam.d/sshd
# diff -r /tmp/newpam.d/ /tmp/oldpam.d/
diff -r /tmp/newpam.d/atd /tmp/oldpam.d/atd
10d9
< hello
diff -r /tmp/newpam.d/sshd /tmp/oldpam.d/sshd
40d39
< goodbye
```

Выполняя рекурсивно di ff (diff -r) для каталогов, вы можете видеть, что hello присутствует в файле /tmp/newpam.d/atd, но не в файле /tmp/oldpam.d/atd (< hello). Аналогичным образом слово goodbye наличествует только в файле /tmp/newpam.d/sshd.

Если вывод получается слишком детализированным, а вы хотите узнать, какие файлы изменились, но не как именно, можете добавить параметр -q:

```
# diff -rq /tmp/newpam.d/ /tmp/oldpam.d/
Files /tmp/newpam.d/atd and /tmp/oldpam.d/atd differ
Files /tmp/newpam.d/sshd and /tmp/oldpam.d/sshd differ
```

Использование diff -r — хороший способ отслеживания изменений в конфигурационных файлах или программных проектах. Например, если бы вы скопировали все свои файлы из /etc в другой каталог, а затем выполнили команду diff -rq для этих двух каталогов, то увидели бы, какие конфигурационные файлы изменились с тех пор, как вы скопировали их.

Команды awk и cut для обработки столбцов

Еще один солидный инструмент для обработки текста — команда awk. Она представляет собой полноценный язык программирования. В приведенных далее примерах демонстрируется лишь несколько трюков, связанных с **извлечением столбцов текста.** На самом деле с помощью awk вы можете сделать намного больше.

```
$ ps auwx | awk '{print $1.$11}'[1]   Показать столбцы 1. 11 вывода ps
$ ps auwx | awk '/chris/ {print $11}' Показать процессы chris
$ ps auwx I grep chris | awk '{print $11}'   То же, что и выше
```

В первом примере отображается содержимое первого столбца (имя пользователя) и 11-го столбца (имя команды) из вывода команды ps касательно запущенных на данный момент процессов. Следующие две команды генерируют тот же вывод, только в одном случае используется команда awk, а в другом grep для поиска всех процессов, владельцем которых является пользователь chris. В каждом случае при обнаружении процессов, которыми владеет chris, 11-й столбец (имя команды) отображается для каждого из них.

По умолчанию команда awk предполагает, что разделителем между столбцами является пробел. Вы можете **определить другой разделитель** посредством параметра -F.

```
$ awk -F: '{print $1,$5}' /etc/passwd   Использовать двоеточие как разделитель
                                        при отображении столбцов
```

Аналогичного результата можно добиться с помощью команды cut. Как и в предыдущем примере, вы можете указать двоеточие (:) в качестве разделителя столбцов для обработки информации из файла /etc/passwd:

```
$ cut -d: -f1,5 /etc/passwd   Использовать двоеточие как разделитель
                              при отображении столбцов
```

Команду cut также можно **использовать в сочетании с диапазонами полей.** Следующая команда выводит на экран столбцы с первого по пятый из файла /etc/passwd:

```
$ cut -d: -f1-5 /etc/passwd   Показать столбцы с 1-го по 5-й
```

Вместо использования тире (-) как индикатора диапазона номеров вы можете задействовать его для **вывода на экран всех столбцов, начиная со столбца с опре-**

деленным номером и далее. Следующая команда отображает все столбцы, начиная со столбца 5 и далее из файла /etc/passwd:

```
$ cut -d: -f5- /etc/passwd          Показать столбцы начиная с 5-го и далее
```

Я предпочитаю использовать команду awk, когда столбцы разделены разным количеством пробелов, как, например, в выводе ps. Если приходится иметь дело с файлами, для которых в качестве разделителей используются запятые (.) или двоеточия (:), как, например, в ситуации с файлом /etc/passwd, я выбираю команду cut.

Конвертирование текстовых файлов в другие форматы

В текстовых файлах в мире UNIX используются символы конца строки (\n), отличающиеся от тех, которые применяются в DOS/Windows (\r\n). Вы можете просмотреть эти специальные символы, содержащиеся в текстовом файле, с помощью команды od:

```
$ echo hello > myunixfile.txt
$ od -c -t xl myunixfile.txt
0000000 h       e 1         1 o \n
          68 65 6c 6c 6f 0a
0000006
```

Чтобы содержимое скопированных из одной среды в другую файлов отображалось корректно, необходимо **конвертировать эти файлы.** Пакет dos2unix включает набор инструментов, позволяющих конвертировать файлы из одного формата в другой (для установки этого пакета введите apt-get install dos2unix). Вот примеры:

```
$ unix2dos < myunixfile.txt > mydosfile.txt
$ cat mydosfile.txt | dos2unix > myunixfile.txt
```

В данном примере команда unix2dos конвертирует обычный текстовый файл Linux или UNIX (myuni xf i 1 e. txt) в текстовый файл DOS или Windows (mydosfi 1 e. txt). В примере использования dos2unix делается противоположное путем конвертирования файла DOS/Windows в файл Linux/UNIX.

Резюме

В системах Linux и UNIX традиционно используются обычные текстовые файлы для конфигурации системы, документации, вывода команд, а также хранимой информации во многих других формах. В силу этого появилось множество команд для поиска, редактирования и прочих манипуляций с текстовыми файлами. Даже при современных GUI-интерфейсах умение выполнять различные операции с обычными текстовыми файлами необходимо, чтобы стать продвинутым Linux-пользователем.

В этой главе были описаны некоторые наиболее популярные команды для работы с обычными текстовыми файлами в Linux. К ним относятся команды текстовых редакторов (например, vi, nano или JOE), а также команды, позволяющие редактировать потоковые данные (к примеру, команды sed и awk). Были также рассмотрены команды для сортировки текста (sort), подсчета слов, строк и символов (wc) и преобразования символов в тексте (tr).

6 Использование мультимедийных данных

В этой главе:

- проигрывание музыки с помощью команд oggl23 и mpg321;
- регулирование уровня громкости звука с применением команд alsamixer и aumix;
- копирование музыкальных CD с использованием cdparanoia;
- кодирование музыки с помощью команд oggenc, f 1 ac и 1 ate;
- потоковая передача музыки посредством icecast и ices;
- конвертирование аудиофайлов с использованием пакета sox;
- преобразование цифровых изображений с помощью convert;
- проигрывание видео с DVD.

Нет необходимости прибегать к GUI-инструменту, если вы хотите только проиграть песню либо конвертировать изображение или аудиофайл в другой формат. Есть команды для взаимодействия с мультимедийными файлами (аудиофайлами и изображениями), которые быстры и эффективны, если работать из интерпретатора команд. Для пакетных манипуляций с мультимедийными файлами вы можете добавить команду, которую используете для преобразования одного файла, в сценарий для повторения соответствующего процесса для множества файлов.

Внимание в этой главе сосредоточено на инструментах для работы с аудио- и видеофайлами, а также цифровыми изображениями из интерпретатора команд.

Работа с аудиоданными

Для Linux-систем доступны команды, позволяющие манипулировать файлами множества аудиоформатов. Такие команды, как oggl23, mpg321 и pi ay, можно использовать для прослушивания аудиофайлов. Есть также команды для копирования песен с музыкальных CD и их кодирования для удобного хранения. Есть даже команды, которые позволяют осуществлять потоковую передачу аудиоданных, чтобы любой пользователь в вашей сети мог прослушать содержимое вашего списка воспроизведения.

Проигрывание музыки

В зависимости от аудиоформата файла, который нужно воспроизвести, вы можете выбрать подходящий вариант из нескольких проигрывателей командной строки для Linux. Команда `pi ay` (на основе инструмента sox, о котором мы поговорим позднее) позволяет воспроизводить аудиофайлы множества свободно доступных форматов. Вы можете использовать `ogg123` для проигрывания музыки, записанной в популярных форматах с открытым исходным кодом, включая файлы Ogg Vorbis, Free Lossless Audio Codec (FLAC) и Speex. Проигрыватель `mpg321`, доступный в сторонних репозиториях, — популярное средство воспроизведения музыкальных файлов формата MP3.

Команда `play` требует наличия пакета sox. Установите его следующим образом:

$ sudo apt-get install sox

Введите `sox -h`, чтобы **увидеть аудиоформаты и эффекты,** доступные для использования в сочетании с `play`:

$ sox -h

```
...
Supported file formats: 8svx aif aifc aiff aiffc al alsa au auto avr cdda cdr cvs
cvsd dat dvms fssd gsm hcom ima ircam la lu maud nist nul null ogg ossdsp pro raw
s3 sb sf si smp snd sndt sou sph sw txw u3 u4 ub ul uw vms voc vorbis vox wav wve
xa
Supported effects: all pass band bandpass bandreject bass chorus compand deshift
deemph dither earwax echo echos equalizer fade filter flanger highpass lowpass
mcompand mixer noiseprof noisered pad pan phaser pitch polyphase repeat resample
reverb reverse silence speed stat stretch swap synth treble tremolo trim vibro
vol
```

Команда pi ay использует код sox для воспроизведения звуков.

Вот несколько примеров проигрывания файлов с помощью play:

$ play inconceivable.wav *Проиграть UAV-файл (возможно, скопированный с CD)*

$ play *.wav *Проиграть все UAI-файлы, расположенные в каталоге (до 32)*

$ play hi .au vol .6 *Проиграть AU-файл, уменьшив громкость звука (что может снизить уровень искажений)*

$ play -r 14000 short.aiff *Проиграть AIFF-файл с частотой дискретизации 14000 Герц*

Для проигрывания файлов Ogg Vorbis установите пакет vorbis-tools, введя следующее:

$ sudo apt-get install vorbis-tools

Вот примеры воспроизведения файлов Ogg Vorbis **(www.vorbis.com/)** с помощью `одд123`:

$ oggl23 mysong.ogg *Проиграть Ogg-файл*

$ cd /usr/share/example-content/Ubuntu_Free_Culture_Showcase/

$ oggl23 How\ fast.ogg *Проиграть файл-образец*

```
$ ogg123 http://vorbis.com/music/Lumme-BacJoop.ogg Проиграть файл по веб-адресу
$ ogg123 -z *.ogg                                  Проиграть файлы в псевдослучайном порядке
$ ogg123 /var/music/                               Проиграть песни из /var/music и подкаталогов
$ ogg123 @playlist                                 Проиграть песни из списка воспроизведения
```

Список воспроизведения — это просто перечень каталогов или отдельных Ogg-файлов для проигрывания. Если каталог присутствует в списке, то будут воспроизведены все Ogg-файлы, которые располагаются в этом каталоге или любом его подкаталоге. При проигрывании множества файлов нажмите комбинацию клавиш **Ctrl+C,** чтобы **перейти к следующей песне.** Дважды нажмите сочетание **Ctrl+C для выхода.**

Чтобы использовать проигрыватель шрд321 для воспроизведения MP3-файлов, потребуется установить пакет mpg321. В силу патентных притязаний, связанных с MP3-кодеками, последние не включаются автоматически в большинство дистрибутивов Linux. Вам придется выбрать установку M P3-кодеков и проигрывателей по отдельности. Для этого введите следующее:

```
$ sudo apt-get install mpg321
```

Вот примеры воспроизведения аудиофайлов в формате M P3 с помощью шрд321:

```
$ шрд321 yoursong.mp3          Проиграть MP3-файл
$ mpg321 @mp31ist              Проиграть песни из списка воспроизведения MP3-файлов
$ cat mp31ist | трд321 -0 -    Передать по каналу список воспроизведения mpg321
$ трд321 -z *.трЗ              Проиграть файлы в псевдослучайном порядке
$ трд321 -Z *.трЗ              То же, что и с -г, только повторять воспроизведение
                               бесконечно
```

Список тра321 — это просто перечень файлов. Вы можете создать список воспроизведения с помощью простой команды **1 s** и направить вывод в файл. Используйте полные пути к файлам, если только не планируете задействовать список из расположения, откуда имеет смысл указать относительные пути к файлам.

Регулирование уровня громкости звука

Выбор аудиоинструментов командной строки, которые потребуется использовать для активизации аудиоустройств и регулирования уровня громкости звука, зависит от типа вашей звуковой системы. ALSA (Advanced Linux Sound Architecture — продвинутая звуковая архитектура Linux) является звуковой системой, используемой сегодня в большинстве Linux-систем. OSS (Open Source Sound System — звуковая система с открытым исходным кодом) существует дольше и все еще применяется для более старого аппаратного обеспечения. В общем, вы можете использовать команду **alsamixer** для регулирования уровня громкости звука, если задействуется ALSA, и **aumix,** если применяется OSS.

ALSA является звуковой системой по умолчанию для большинства Linux-систем. Если добавить загружаемые модули, которые активизируют интерфейсы устройств OSS, чтобы они тоже функционировали, то аудиоприложения, которым требуется интерфейс устройства OSS, смогут работать и с ALSA. Узнать, **загружены ли модули OSS,** например **snd-pcm-oss** (эмулирует **/dev/dsp** и **/dev/audio),**

snd-mixer-oss (эмулирует /dev/mixer) и snd-seq-oss (эмулирует /dev/sequencer), можно, введя следующее:

 # lsmod | grep snd

Если окажется, что они не загружены, то вы можете **установить и загрузить нужные модули OSS:**

$ sudo apt-get install oss4
$ sudo modprobe snd-pcm-oss
$ sudo modprobe snd-mixer-oss
$ sudo modprobe snd-seq-oss

Если выяснится, что модули загружены, можете использовать alsamixer для регулирования уровня громкости звука для аудиоприложений OSS. **Запуск alsamixer** осуществляется следующим образом:

$ alsamixer	*Отобразить экран alsamixer и отразить процесс воспроизведения*
$ alsamixer -V playback	*Отобразить только каналы воспроизведения (используемые по умолчанию)*
$ alsamixer -V all	*Отобразить процесс воспроизведения и вид ввода*
$ alsamixer -c 1	*Использовать alsamixer на второй (1) звуковой карте*

Полосы громкости звука отображаются для всех звуковых каналов:

О используйте клавиши <- и -> для **выделения разных каналов (Master** (Основной), **PCM** (Pulse Code Modulation — импульсно-кодовая модуляция), **Headphone** (Е1аушники) и т. д.);

О нажимайте Т и *I* для увеличения или уменьшения громкости звука в каждом канале;

О выделив канал, используйте клавишу **М,** чтобы **выключить или включить** звук в этом канале;

О нажмите **Пробел** при выделенном входном канате **(Mic** (Микрофон), **Line** (Линейный вход) и т. д.), чтобы **назначить этот канал в качестве канала захвата** (для записи с аудиовхода);

О для **выхода из alsamixer** используйте комбинацию **Alt+Q** либо клавишу **Esc;**

О нажимайте **Tab** для циклического прохода по настройкам для **Playback** (Воспроизведение), **Capture** (Захват) и **АН** (Все).

Приложение для микширования звука aumix (для использования которого вам потребуется установить пакет aumix) может работать в экранно-ориентированном или обычном командном режиме. В обычном текстовом режиме для **изменения или отображения настроек** используются параметры. Вот примеры строк команды aumix:

$ sudo apt-get install aumix
$ aumix -q *Показать уровень громкости звука в левом/правом канале и типы всех каналов*

vol 62. 62
pcm 100.100
igain 53.53

$ aumix -1 q -m q	*Отобразить текущие настройки только для линейного входа и микрофона*
$ aumix -v 80 -m 0	*Задать уровень громкости звука как равный 70 %, а для микрофона - значение 0*
$ aumix -m 80 -m R -m q	*Задать для микрофона значение 80 %, использовать его для записи, показать настройки микрофона*
$ aumix	*При отсутствии параметров aumix запускается в экранно-ориентированном режиме*

При запуске в экранно-ориентированном режиме aumix отображает все доступные звуковые каналы. При работе в этом режиме **используйте клавиши для выделения и изменения отображаемых настроек звука:**

- нажимайте клавиши **Page Up, Page Down,** T и *I* для выбора каналов;
- используйте <- и -> для увеличения или уменьшения громкости звука;
- нажмите клавишу M, чтобы выключить звук в текущем канале;
- используйте пробел, чтобы выбрать текущий канат в качестве канала записи;
- если доступно использование мыши, можете задействовать ее для выбора уровней громкости звука, а также уровней баланса и текущего канала записи.

Копирование музыки с CD

Чтобы иметь возможность проигрывать свою личную музыкальную коллекцию в Linux, вы можете воспользоваться такими инструментами, как, например, cdparanoia, который позволяет копировать треки с музыкальных компакт-дисков, конвертируя их при этом в WAV-файлы, на ваш жесткий диск. Скопированные таким образом файлы затем можно кодировать, чтобы они занимали меньше дискового пространства, с использованием таких инструментов, как oggenc (Ogg Vorbis), flac (FLAC) или lame (MP3).

ПРИМЕЧАНИЕ

Есть отличные графические инструменты для копирования и кодирования музыки с CD, например grip и sound-juicer. Они поддерживают CDDB, поэтому также могут использовать информацию о музыке на компакт-диске для присваивания имен выходным файлам (исполнитель, альбом, песня и т. д.). Однако в этом разделе описывается, как применять некоторые базовые команды для копирования и кодирования музыки с CD вручную.

Используя cdparanoia, вы можете проверить, поддерживает ли ваш CD-привод компакт-диски CDDA (Compact Disc Digital Audio — звуковой компакт-диск), извлекать аудиотреки с использованием CD-привода и копировать их на жесткий диск. Если инструмент cdparanoia не установлен, инсталлируйте его с помощью apt-get. Затем начните с того, что вставьте музыкальный CD в привод и введите следующее:

```
$ sudo apt-get install cdparanoia
$ cdparanoia -vsQ
...
Checking /dev/cdrom for cdrom...
Checking for SCSI emulation...
```

```
Checking for MMC style command set..
Verifying CDDA command set...
...
Table of contents (audio tracks only):
track          length               begin         copy pre ch
==========================================================
  1.     18295 [04:03.70]            0 [00:00.00]    no   no  2
  2.     16872 [03:44.72]        18295 [04:03.70]    no   no  2
    ...
 11.     17908 [03:58.58]       174587 [38:47.62]    no   no  2
 12.     17342 [03:51.17]       192495 [42:46.45]    no   no  2
TOTAL   209837 [46:37.62]      (audio only)
```

В этом сокращенном выводе показано, что cdparanoia проверяет возможности /dev/cdrom путем выяснения, имеется ли поддержка эмуляций SCSI и набора команд MMC и способен ли привод обрабатывать информацию CDDA. Наконец, на экран выводятся сведения о каждом треке. Вот примеры строк команды cdparanoi a для **оцифровки CD и записи их на жесткий диск:**

$ cdparanoia -B	*Скопировать треки, выполнив их конвертирование в WAV-файлы с присваиванием им имен треков*
$ cdparanoia -B -- "5-7"	*Скопировать треки 5-7 в отдельные файлы*
$ cdparanoia -- "3-8" abc.wav	*Скопировать треки 3-8 в один файл (abc.wav)*
$ cdparanoia -- "1:[40]-"	*Скопировать трек 1 с 40-й секунды и до конца CD*
$ cdparanoia -f -- "3"	*Скопировать трек 3 и сохранить его в формате A IFF*
$ cdparanoia -a -- "5"	*Скопировать трек 5 и сохранить его в формате AIFC*
$ cdparanoia-w -- "1"my.wav	*Скопировать трек 1 и присвоить ему имя my.wav*

Кодирование музыки

После копирования с CD музыкального файла его обычно кодируют, чтобы он занимал меньше дискового пространства. К популярным кодировщикам относятся oggenc, flac и lame для кодирования соответственно в формат Ogg Vorbis, FLAC и MP3.

Используя oggenc, вы можете начать с аудиофайлов или потоков в формате WAV, AIFF, FLAC или RAW и конвертировать их в формат Ogg Vorbis. Хотя Ogg Vorbis является форматом с потерями, кодирование по умолчанию WAV-файлов в него все же позволяет получить звук очень хорошего качества и может привести к тому, что размер файла уменьшится примерно в десять раз. Вот несколько примеров использования oggenc:

$ oggenc ab.wav	*Кодирует WW в Ogg (ab.ogg)*
$ oggenc ab.flac -o new.ogg	*Кодирует FLAC в Ogg (new.ogg)*
$ oggenc ab.wav -q 9	*Повышает качество сжатия до 9*

По умолчанию качество (-q) генерируемого инструментом oggenc результата установлено на уровень 3. Однако вы можете **устанавливать любое его значение:** от -1 до 10 (включая дроби, например 5.5).

```
$ oggenc NewSong.wav -o NewSong.ogg \
-a Bernstein -G Classical \
-d 06/15/1972 -t "Simple Song" \
-1 "Bernsteins Mass"          \
-c info="From Kennedy Center"
```

Приведенная выше команда конвертирует MySong.wav в MySong.ogg. Именем исполнителя в данном случае является Bernstein, а стилем музыки — Classical. Дата помечена как 06/15/1972, название песни — Simple Song, а название альбома — Bernsteins Hass. Комментарием является From Kennedy Center. Обратные слеши не нужны, если вы уместите всю команду в одну строку. Однако если вы все же поставите обратные слеши, убедитесь, что после них нет пробелов.

В предыдущем примере в заголовок итогового Ogg-файла была добавлена информация. Вы можете **просмотреть заголовочную информацию** наряду с другими сведениями о файле, воспользовавшись командой ogginfo:

```
$ ogginfo NewSong.ogg
Processing file "NewSong.ogg"...
. . .
Channels: 2
Rate: 44100
Nominal bitrate: 112.000000 kb/s
User comments section follows...
info=From Kennedy Center
title=Simple Song
artist=Bernstein
genre=Classical
date=06/15/1972
a1bum=Bernsteins Hass
Vorbis stream 1:
Total data length: 3039484 bytes
Playback length: 3m:25.240s
Average bitrate: 118.475307 kb/s
Logical stream 1 ended
```

Как видите, во время кодирования были добавлены комментарии. Параметр -c использовался с целью задания произвольного поля (в данном случае info) со значением для заголовка. Помимо информации в виде комментариев, вы также можете видеть, что в файле имеется два канала и он был записан с частотой дискретизации, равной 44 100 Герц. Кроме того, вы видите объем данных, длительность воспроизведения и средний битрейт.

Команда Л ас — это кодировщик, аналогичный oggenc, за исключением того, что файлы в формате WAV, AIFF, RAW, FLAC или Ogg кодируются в файлы в формате FLAC. Кодировщик flac представляет собой свободный аудиокодек *без потерь*, поэтому он популярен среди пользователей, которые хотят сэкономить место на диске, но при этом получить звук высшего качества в формате с открытым исходным кодом. Наше кодирование WAV в FLAC с использованием значений по умолчанию привело к тому, что получились наполовину меньшие по размеру файлы в противоположность десятикратному уменьшению объема файлов в случае

применения oggenc. Установите пакет flac, чтобы иметь возможность задействовать команду Лас. Далее показано, как установить пакет flac, после чего следуют примеры использования команды Лас:

```
$ sudo apt-get install Лас
$ flac now.wav                  Кодирует WAV в FLAC (now. flac)
$ sox now.wav now.aiff          Кодирует WAV в AIFF (now.aiff)
$ flac now.aiff -o now2.flac    Кодирует AIFF в FLAC (now. flac)
$ flac -8 top.wav -o top.flac   Увеличивает значение уровня сжатия до 8
```

Значение уровня сжатия задается как равное -5 по умолчанию. Можно использовать значения в диапазоне от -0 до -8, при этом наибольшее значение обеспечивает самое сильное сжатие, а наименьшее значение — более быстрое сжатие. Для **конвертирования файлов в формат MP3** с использованием команды lame вам потребуется установить пакет lame. Вот несколько примеров использования команды 1 аше для кодирования файлов в форматах WAV и AIFF:

```
$ sudo apt-get install lame
$ lame in.wav                   Кодирует WAV в MP3 (in.wav.mp3)
$ lame in.wav --preset standard Кодирует в MP3 с использованием
                                стандартных предварительных настроек
$ lame tune.aiff -o tune.mp3    Кодирует AIFF в MP3 (tune.mp3)
$ lame -h -b 64 -m m in.wav out.mp3  Высокое качество, 64 бита, режим моно
$ lame -q 0 in.wav -o abcHQ.mp3 Кодирует со значением качества, равным О
```

Задействуя lame, вы можете задать значение качества от 0 до 9 (5 является значением по умолчанию). Указание значения качества равным 0 обеспечивает использование наилучших алгоритмов кодирования, в то время как задание значения качества, равного 9, деактивизирует большинство алгоритмов (однако процесс кодирования протекает намного быстрее). Как и в случае применения oggenc, вы можете **добавить теговую информацию в свой MP3-файл,** которую можно использовать позднее при воспроизведении этого файла. Вот пример:

```
$ lame NewSong.wav NewSong.mp3 \
--ta Bernstein --tg Classical \
--ty 1972 --tt "Simple Song" \
--tl "Bernsteins Mass"         \
--tc "From Kennedy Center"
```

Как и в примере кодирования WAV в Ogg, приведенном ранее в этой главе, только что продемонстрированная команда конвертирует MySong.wav в MySong.mp3. Именем исполнителя опять же является Bernstein, а стилем музыки — Classical. Год представлен как 1972, название песни — Simple Song, а название альбома — Bernsteins Mass. Комментарием является From Kennedy Center. Обратные слеши не нужны, если вы уместите всю команду в одну строку. Однако, если вы все же добавите обратные слеши, убедитесь, что после них нет пробелов.

Теговая информация отображается на экране в графических MP3-проигрывателях (например, Rhythmbox и Totem, когда в них была добавлена поддержка воспроизведения файлов в формате MP3). Вы также можете увидеть теговую ин-

формацию при использовании проигрывателей командной строки, в частности mpg321, как показано в следующем примере:

```
$ mpgl23 NewSong.mp3
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layer 1, 2. and 3.
...
Title : Simple Song                    Artist: Bernstein
Album : Bernsteins Mass                Year : 1972
Comment: From Kennedy Center           Genre : Classical
Playing MPEG stream from NewSong.mp3 ...
MPEG 1.0 layer III, 128 kbit/s, 44100 Hz joint-stereo
```

Потоковая передача музыки

Если вата музыка находится на одном компьютере, а вы работаете на другом, то **конфигурирование сервера потоковой передачи музыки** — это быстрый способ наладить ее вещание, чтобы ее можно было прослушать на одном или нескольких компьютерах в вашей сети. Потоковый медиасервер Icecast (пакет icecast2) и клиент источника звука Ices (пакет ices2) в Ubuntu можно установить, введя следующее:

```
$ sudo apt-get install icecast2 ices2
```

При установке icecast2 у вас будет возможность сконфигурировать его во время инсталляции либо после нее (как описывается далее).

Ниже приведена быстрая процедура настройки Icecast и Ices для потоковой передачи вашей музыки. Выполните эти действия на компьютере, где находится музыка, которую вы хотите предоставить другим пользователям, выбрав вариант, предполагающий конфигурирование службы позднее.

1. Отредактируйте файл /etc/icecast2/icecast.xml, чтобы изменить все представленные пароли. Выполните поиск hackme, чтобы найти текущие пароли. Вы, вероятно, пожелаете задать другие пароли пользователя и администратора, особенно если решите позволить другим осуществлять потоковую передачу музыки на сервер. Запомните пароли, которые укажете, чтобы использовать их позднее. Вам, возможно, захочется изменить и другие настройки в этом файле, например имя хоста:

```
$ sudo vi /etc/icecast2/icecast.xml
```

2. Если у вас установлен брандмауэр, убедитесь в доступности TCP-порта 8000.
3. Отредактируйте файл /etc/default/icecast2, изменив строку ENABLE на ENABLE=true.
4. Запустите сервер icecast2 от имени суперпользователя, набрав приведенное далее (на самом деле сервер запустится от имени пользователя **icecast2),** и выполните проверку посредством команды netstat:

```
$ sudo /etc/init.d/icecast2 start
$ sudo netstat -topnavel | grep 8000
Tcp        0      0 0.0.0.0:8000          0.0.0.0:*               LISTEN
115        35790      21494/-icecast          off (0.00/0/0)
```

5. Создайте каталоги, необходимые программе ices2, которая обеспечивает список воспроизведения и музыку для сервера icecast2. Выполните следующие команды:

```
$ sudo mkdi r /var/log/ices
$ sudo mkdir /etc/ices2
$ sudo mkdir /etc/ices2/music
```

6. Создайте список воспроизведения с использованием любого текстового редактора или направив список своей музыки в файл. Если, к примеру, все ваши музыкальные Ogg-файлы будут располагаться в подкаталогах **/var/music,** введите следующее:

```
$ find /var/music -name *.одд > playlist.txt
```

7. Файл **play1ist.txt** должен содержать полные пути ко всем файлам, при этом файлы должны быть доступны серверу icecast2. Далее скопируйте файл со списком воспроизведения в каталог **/etc/ices2:**

```
$ sudo cp playlist.txt /etc/ices2
```

Создав файл со списком воспроизведения, откройте любой текстовый редактор, чтобы удалить либо добавить файлы или каталоги с. целью сделать свой список воспроизведения таким, каким вы хотели бы его видеть (если хотите опробовать несколько файлов для списка воспроизведения, можете загрузить их, зайдя по адресу http://vorbis.com/music).

8. Отредактируйте от имени суперпользователя файл **/etc/ i ces2/i ces - pi ayl** i **st. xml,** чтобы обеспечить проигрывание музыки из списка воспроизведения и ее передачу на выполняющийся сервер icecast2. Начните с конфигурационного файла-образца и отредактируйте его. Выполните следующие команды:

```
$ sudo cp /usr/share/doc/ices2/examples/ices-playlist.xml /etc/ices2
$ sudo vi /etc/ices2/ices-playlist.xml
```

9. В частности, вам потребуется модифицировать модули **metadata, input** и **instance** (обязательно замените /**etc/ices2/pl ayl i st. txt** на путь, ведущий **к** вашему файлу **play! 1st .txt):**

```
<metadata>
    <name>My Music Server</name>
    <genre>Oifferent music styles</genre>
    <description>Mix of my personal music</description>
</metadata>
<input>
    <module>playli st</module>
    <param name='’type''>basic</param>
    <param name="fi1e">/etc/ices2/playli st.txt</param>
    <!-- воспроизведение в случайном порядке -->
    <param name="random">l</param>
    ...
</input>
<instance>
```

```
<hostname>loca1host</hostname>
<port>8000</port>
<password>MIcePw</password>
<mount>/mymusic.ogg</rnount>
...
</instance>
```

Из только что приведенных значений (выделенных жирным шрифтом) критически важные — местоположение вашего списка воспроизведения и информация об экземпляре вашего сервера icecast2. Пароль должен соответствовать исходному паролю, который вы добавили в свой файл `/etc/icecast2/icecast.xml`.

10. Запустите передачу аудиоданных ices, введя следующее:

$ sudo ices2 /etc/ices2/ices-playlist.xml &

11. Проверьте, можете ли вы проигрывать музыку с локального компьютера, как показано далее:

$ oggl23 http://localhost:8000/raymusic.ogg

12. Если эта проверка даст положительный результат, попробуйте воспроизвести поток icecast.2 с другого компьютера в вашей сети, заменив `1 ocal host` на IP-адрес сервера и имя хоста.
13. Если возникнут проблемы, загляните в файлы журналов /var/1 og/icecast2 и /var/1 од/ices. Перепроверьте свои пароли и местоположение конфигурационных файлов.
14. Когда вы все сделаете, завершите работу службы icecast2:

$ sudo /etc/init.d/icecast2 stop

Во время активности серверов icecast и ices у вас должен быть доступ к передаваемой в потоковом режиме музыке с любого компьютера, у которого есть доступ к вашему компьютеру-серверу. Используйте любой музыкальный проигрыватель, который может воспроизводить музыку с того или иного HTTP-адреса (oggl23, Rhythmbox, XMMS и т. д.). Музыкальные проигрыватели для Windows, поддерживающие тип предоставляемого вами содержимого, тоже должны работать.

ПРИМЕЧАНИЕ ----------

Если вы захотите пропустить песню, выполните на сервере следующую команду: killall -HUP ices.

Конвертирование аудиофайлов

Утилита sox является в высшей степени универсальным инструментом для работы с аудиофайлами разных свободно доступных форматов. В приведенном далее материале содержатся примеры того, что можно сделать с помощью sox.

Следующая команда **конкатенирует два WAV-файла в один выходной файл:**

$ sox head.wav tai1.wav output.wav

Данная команда **смешивает два WAV-файла:**

$ soxmix soundl.wav sound2.wav output.wav

Чтобы использовать sox для **отображения информации о файле,** задействуйте инструмент `st at` следующим образом:

$ sox soundl.wav -e stat

```
Samples read:                208512
Length (seconds):          9.456327
Scaled by:             2147483647.0
Maximum   amplitude:       0.200592
Minimum   amplitude:      -0.224701
Midi me ampl itude:       -0.012054
Mean norm:                 0.030373
Mean     amplitude:        0.000054
RMS      amplitude:        0.040391
Maximum delta:             0.060852
Minimum delta:             0.000000
Mean         delta:        0.006643
RMS          delta:        0.009028
Rough    frequency:             784
Volume  adjustment:           4.450
```

Используйте `trim` для удаления сегментов аудиофайла, имеющих длительность в несколько секунд:

$ sox soundl.wav output.wav trim 4 *Уделить первые 4 секунды*
$ sox soundl.wav output.wav trim 2 6 *Оставить в файле только сегмент со 2 по 6 секунду*

В первом примере удаляются первые четыре секунды из файла `soundl .wav` и результат записывается в файл `output. wav.` Во втором примере берется файл `soundl. wav,` оставляется сегмент со второй по шестую секунду, остальное удаляется, а результат записывается в файл `output.wav`.

Преобразование изображений

Когда каталоги заполнены цифровыми изображениями, возможность манипуляций с этими изображениями из командной строки может сэкономить много времени. 1 lакет ImageMagick (для установки его в Ubuntu используйте apt-`get install imagemagick`) поставляется с очень полезными инструментами для преобразования цифровых изображений в формы, с которыми вы сможете работать. В этом разделе приведены команды для манипуляций с цифровыми изображениями, а также примеры простых сценариев для внесения соответствующих изменений в пакетном режиме.

Получение информации об изображениях

Чтобы **получить информацию об изображении,** воспользуйтесь командой identi fy, как показано далее:

```
$ identify p2090142.jpg
p2090142.jpg JPEG 2048x1536+0+0 Di+ectClass 8-bit 4-02.037kb
$ identify -verbose p2090142.jpg j less
Standard deviation: 61.1665 (0.239869)
Colors: 205713
Rendering intent: Undefined
Resolution: 72x72
Units: PIxelsPerlnch
Filesize: 402.037kb
Interlace: None
Background color: white
Border color: rgb(223.223,223)
Matte color: grey74
Transparent color: black
Page geometry: 2048x1536+0+0
Compression: JPEG
Quality: 44
...
```

Первая команда в данном примере выводит основную информацию об изображении (его файловое имя, формат, геометрию, класс, глубину канала и размер файла). Вторая команда показывает всю информацию, которую может извлечь из изображения. Помимо сведений, представленных в этом примере, в подробном выводе также содержатся данные о времени создания файла, типе использованной камеры, значении диафрагмы и светочувствительности ISO.

Конвертирование изображений

Команда convert является швейцарским армейским ножом конвертера файлов. Далее приведены некоторые способы манипуляций с изображениями с использованием convert. В следующих примерах **выполняется конвертирование файлов изображений из одного формата в другой:**

```
$ convert  tree.jpg tree.png      Конвертировать  файл JPEG в PNG
$ convert  icon.gif icon.bmp      Конвертировать  файл GIF в BMP
$ convert  photo.tiff photo.pcx   Конвертировать  файл TIFF в PCX
```

К типам изображений, поддерживаемых командой `convert`, относятся .jpg, . `bmp`, .pcx, .gif, .png, .tiff, .xpmn ,xwd. Вот примеры использования convert для **изменения размеров изображений:**

```
$ convert -resize 1024x768 hat.jpg hat-sm.jpg
$ convert -sample 50Жx50Ж dog.jpg dog-half.jpg
```

В первом примере создается изображение (hat-sm. jpg) размером 1024 x 768 пикселов. Во втором примере изображение dog. jpg уменьшается наполовину (50Жx50%) и сохраняется как dog-half.jpg.

Вы также можете **поворачивать изображения** на величину от 0 до 360°. Вот примеры:

```
$ convert -rotate 270 sky.jpg sky-final.jpg      Повернуть изображение на 270°
$ convert -rotate 90 house.jpg house-final.jpg   Повернуть изображение на 90°
```

Вы можете **добавить текст к изображению** с использованием параметра **-draw:**

```
$ convert -fill black -pointsize 60 -font helvetica \
 -draw 'text 10,80 "Copyright NegusNet Inc."' \
plO.jpg plO-cp.jpg
```

В приведенном выше примере к изображению была добавлена информация об авторских правах с использованием черного шрифта Helvetica размером 60 пунктов для нанесения надписи на это изображение. Текст был размещен на 10 пунктов правее и на 80 пунктов ниже верхнего левого угла. Имя нового изображения рЮ-ср.jpg, выступающее как индикатор того, что к этому новому изображению была добавлена информация об авторских правах.

Вот несколько интересных способов **создания миниатюр** с помощью команды **convert:**

```
$ convert -thumbnail 120x120 a.jpg a-a.png
$ convert -thumbnail 120x120 -border 8 a.jpg a-b.png
$ convert -thumbnail 120x120 -border 8 -rotate 8 a.jpg a-c.png
```

Во всех трех примерах создается миниатюра размером 120 x120 пикселов. Во втором примере используется параметр **-border** для добавления рамки вокруг миниатюры, благодаря чему она выглядит как снимок, сделанный с помощью фотоаппарата Polaroid. В последнем примере изображение поворачивается. На рис. 6.1 показаны результаты применения команд в этих трех примерах.

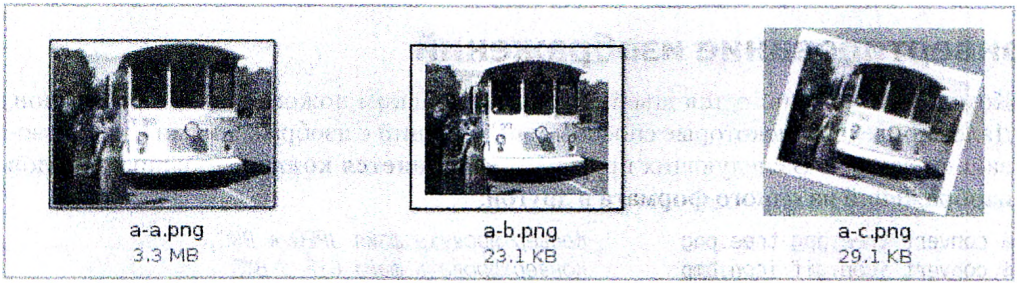

Рис. 6.1. Используйте convert для создания миниатюр, добавления рамок и поворота изображений

Помимо манипуляций, с помощью которых можно сделать картинки практичными и управляемыми, есть также способы **превратить ваши изображения в забавные и даже причудливые.** Вот несколько примеров:

```
$ convert -sepia-tone 75* house.jpg oldhouse.png
$ convert -charcoal 5 house.jpg char-house.png
$ convert -colorize 175 house.jpg col or-house.png
```

Параметр **-sepia-tone** придает изображению вид фотоснимка, сделанного на Старом Западе. Параметр **-charcoal** делает картинку такой, как будто она нарисовано от руки углем. При использовании параметра **-colori ze** цвет каждого пиксела изображения модифицируется согласно заданному значению (в данном случае —175).

В верхнем левом углу на рис. 6.2 показано оригинальное изображение дома, в верхнем правом — изображение дома после применения -sepia-tone, в нижнем левом — изображение дома после использования -charcoal, а в нижнем правом — изображение дома после применения -colorize.

Если вы захотите взглянуть на еще один пример причудливого преобразования изображения, то попробуйте придать картинке эффект завихрения. Например:

$ convert -swirl 300 photo.pcx weird.pcx

Пакетное конвертирование изображений

Большинство описанных в этой главе преобразований картинок можно легко осуществить с помощью графического инструмента, позволяющего производить манипуляции с изображениями, например GIMP. Однако команда convert, примеры использования которой были приведены выше, по-настоящему блистает, если задействовать ее в сценариях. Вместо того чтобы выполнять операции, связанные с изменением размеров, поворотом, нанесением надписей или окраской в отношении одного файла изображения, вы можете выполнить любую (или сразу все) в отношении целого каталога.

Возможно, вы захотите создать миниатюры для коллекции изображений муляжей уток, или уменьшить размеры всех своих свадебных фотографий, чтобы они уместились на экране цифровой фоторамки, или добавить информацию об авторских правах к каждому изображению в каталоге, перед тем как поделиться ими в Интернете. Все это легко сделать посредством приведенных выше примеров использования команды convert и простых сценариев интерпретатора команд.

Вот пример сценария, который вы можете выполнить для **изменения размеров всех фотографий в каталоге** на 1024 x 768 пикселов, чтобы воспроизводить их на экране цифровой фоторамки:

```
$ cd $H0ME/myimages
$ mkdir small
$ for pic in 'Is *.png'
do
echo "converting $pic"
convert -resize 1024x768 $pic small/sm-Spic
done
```

Перед выполнением сценария эта процедура переходит в каталог $H0ME/myimages (в котором в данном случае содержится набор изображений с высоким разрешением). Затем она создает подкаталог sma 11 для размещения уменьшенных изображений. Сценарий как таковой запускается с цикла for, который отображает каждый файл из текущего каталога, имеющий расширение . png (вам, возможно, потребуется указать .jpg или другое расширение). После этого размеры каждого файла изменяются на 1024 x 768 пикселов, затем файлы копируются в каталог small, при этом к имени каждого добавляется -sm.

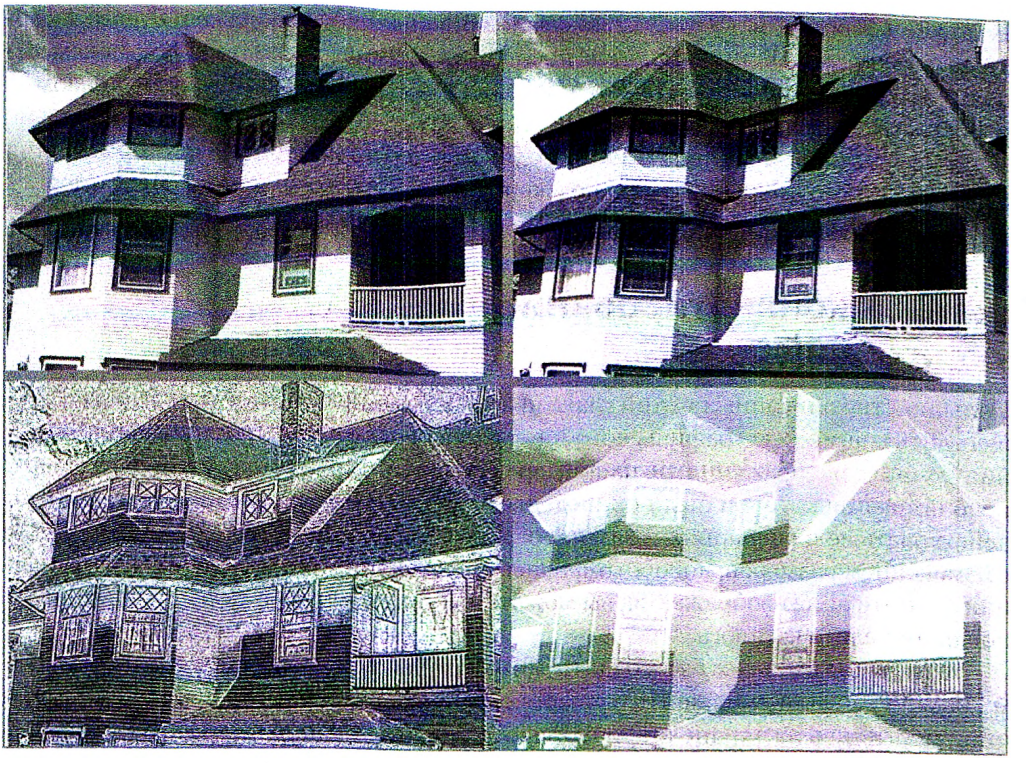

Рис. 6.2. **Начните с нормального изображения и задействуете параметры -sepia-tone, -charcoal и -colorize**

Используя аналогичный базовый сценарий, вы можете задействовать любую из строк команды convert, приведенных ранее, либо создать собственные, которые будут отвечать вашим нуждам. Вы сможете конвертировать все изображения, располагающиеся в том или ином каталоге, за пару минут, вместо того, чтобы несколько часов щелкать кнопкой мыши в GUI-интерфейсе.

Манипуляции с видео

Проигрывание, запись, копирование и прочая работа с видео уже долгое время является спорным вопросом в Linux. На раннем этапе существовало много инструментов с открытым исходным кодом для работы с видео, однако те, кто создавал коммерческие видеокодеки и видеосодержимое (в частности, коммерческие фильмы), не всегда страстно желали, чтобы другие свободно пользовались их собственностью.

Возможность воспроизводить и копировать коммерческие DVD-фильмы в Linux базируется на специальном программном обеспечении для доступа и конвертирования видео в подходящий для просмотра формат. Первоначально ПО для дешифрования под названием DeCSS было получено из взломанного DVD-проигрывателя. После этого был создан инструмент libdvdcss, благодаря чему Linux-

пользователи получили возможность дешифровывать коммерческое содержимое DVD, чтобы его можно было воспроизводить или копировать.

Несмотря на то что libdvdcss не столкнулся с правовыми проблемами, которые имели место в случае с DeCSS, большинство дистрибутивов Linux возлагает на пользователя обязанность оценивать законность использования libdvdcss для дешифрования фильмов. Все остальное программное обеспечение, необходимое /ыя управления и проигрывания коммерческих DVD-фильмов, легкодоступно и способно воспроизводить фильмы после того, как вы добавите в систему libdvdcss.

Для развлечения в следующем подразделе описывается, как вы можете обзавестись программным обеспечением, необходимым для воспроизведения DVD-фильмов.

Проигрывание видеофайлов

Существует набор высококачественных видеопроигрывателей, доступных для Ubuntu. Далее приведен список наиболее популярных Linux-видеопроигрывателей.

- Totem — это проигрыватель фильмов по умолчанию, доступный вместе с рабочим столом GNOME. Он основан на проекте GStreamer, участники которого создают множество плагинов, необходимых для обеспечения аудио- и видеокодеков. В Ubuntu значок Movie Player (Проигрыватель фильмов) по умолчанию запускает Totem. Таким образом, если вы установите настольную систему, то Totem инсталлируется автоматически.
- Mplayer — проигрыватель, который является одним из старейших видеопроектов в Linux. Он поддерживает множество аудио- и видеоформатов. Он особенно хорош в проигрывании поврежденных медиафайлов, которые не будут воспроизводиться проигрывателями Windows. Чтобы обзавестись базовым вариантом Mplayer, вам потребуется установить пакет mplayer.

```
$ sudo apt-get install mplayer
```

- Xine — проект, который на самом деле представляет собой движок медиапроигрывателей. Его можно использовать с разными внешними интерфейсами, связанными с аудио-/видеопроигрывателями. Чтобы получить в свое распоряжение базовый внешний интерфейс для воспроизведения видео с использованием движка Xine, необходимо установить пакет xine-ui.

```
$ sudo apt-get install xine-ui
```

- VLC — еще один популярный медиапроигрыватель. Он поддерживает целый набор аудио- и видеокодеков (www.videolan.org/vlc/features.html). Чтобы добавить его в Ubuntu, установите пакет vie:

```
$ sudo apt-get install vie
```

Во избежание правовых проблем, связанных с воспроизведением фильмов в Linux, вы можете приобрести проигрыватель Fluendo DVD Player от компании, которая ведет проект GStreamer (www.fluendo.com/shop/product/fluendo-dvd-player/).

Он стоит примерно €20 (это около $26). В следующем разделе описывается, как установить программное обеспечение, необходимое для видеопроигрывателей с открытым исходным кодом.

Установка программного обеспечения для проигрывания видео

Когда вы установите любой из видеопроигрывателей, описанных в предыдущем разделе, вам понадобится дополнительное программное обеспечение, чтобы воспроизводить файлы разных аудио- и видеоформатов. Для проигрывания содержимого, поддержка воспроизведения которого еще не встроена в проигрыватель, вы можете дополнительно установить кодеки.

Помимо обеспечения своего собственного коммерческого DVD-проигрывателя, участники проекта GStreamer также создали набор пакетов плагинов GStreamer, которые поддерживают различные аудио-/видеоформаты. Чтобы узнать, какие кодеки поддерживаются каждым пакетом плагинов GStreamer, обратитесь к документации по плагинам GStreamer: http://gstreamer.freedesktop.org/documentation/plugins.html.

Описание наборов кодеков Good, Bad и Ugly, доступных в случае с GStreamer, можно найти на соответствующей странице в «Википедии» с характеристикой GStreamer (http://en.wikipedia.org/wiki/GStreamer). Если, к примеру, вы захотите установить плагины GStreamer из набора Good, то введите следующее:

```
$ sudo apt-get install gstreamerO.10-plugins-good
```

Вместе с тем, как я уже отмечая ранее, главный (и самый спорный) программный инструмент для чтения коммерческих DVD-фильмов — это libdvdcss. На справочных страницах, созданных сообществом Ubuntu, описано, как получить и установить libdvdcss без активизации каких-либо специальных сторонних репозиториев. Вы увидите предупреждение, что вам следует удостовериться в законности использования этого инструмента в вашей стране: https://help.ubuntu.com/community/RestrictedFormats/PlayingDVDs.

Пакет libdvdcss2 обеспечивает для Ubuntu программное обеспечение, которое позволяет воспроизводить коммерческие DVD-фильмы. Вы можете обзавестись им, установив пакет libdvdcss4 и выполнив следующий сценарий:

```
$ sudo /usr/share/doc/Tibdvdread4/instaU-css.sh
```

На данном этапе вы уже должны уметь производить чтение и проигрывание коммерческих DVD-фильмов в Linux-системе. Если вам потребуется помощь в определении кода региона в вашей системе Ubuntu, обратитесь к странице PlayingDVDs, созданной сообществом Ubuntu (https://help.ubuntu.com/community/RestrictedFormats/PlayingDVDs).

Запуск DVD-проигрывателя

Вы можете запустить любой видеопроигрыватель, описанный ранее в этой главе, с рабочего стола Ubuntu, просто вставив DVD с фильмом в привод и выбрав Movie

Player (Проигрыватель фильмов). По умолчанию запустится проигрыватель фильмов Totem, но если хотите, выберите вместо него Xine или другой.

В качестве альтернативы вы можете запускать видеопроигрыватели из командной строки. Например, чтобы включить проигрыватель фильмов Mplayer, сделайте следующее в интерпретаторе команд:

```
$ mplayer dvd://
```

После установки соответствующего программного обеспечения проигрывание DVD-фильмов будет интуитивно понятным. Однако если вы захотите внести какие-либо улучшения касательно воспроизведения DVD или воспроизводить фильмы неким особым образом, обращайтесь к MAN-страницам, посвященным используемому вами DVD-проигрывателю. Например:

```
$ man mplayer
```

Допустим, вы установили Ubuntu, но не инсталлировали рабочий стол, из-за чего не можете воспроизвести свое видео с помощью видеопроигрывателя Mplayer в ASCII-арте. Далее приведен пример строки команды для направления видеовывода в ваш локальный интерпретатор команд для просмотра фильма в ASCII-арте:

```
$ mplayer dvd:// -vo aa -monitorpixelaspect 0.5
```

Нажимайте **Пробел,** чтобы поставить фильм на паузу и снять с нее. Закончив просмотр, нажмите клавишу **Esc,** чтобы закрыть окно с фильмом.

Вам может потребоваться откорректировать значение параметрamonitorpi xel aspect в соответствии с монитором. Загляните на MAN-страницу, посвященную мрТayer, чтобы узнать о других полезных настройках для регулирования различных параметров воспроизведения.

Резюме

Интерпретатор команд способен обеспечить оперативную и эффективную среду для работы с аудиоданными и файлами цифровых изображений. В этой главе были описаны способы проигрывания, копирования, кодирования, конвертирования и потоковой передачи аудиофайлов посредством командной строки. Касательно цифровых изображений было приведено много примеров использования команды convert для изменения размеров, поворота, конвертирования, нанесения надписей и прочих манипуляций с этими изображениями. Для воспроизведения видео, как только вы установите соответствующее программное обеспечение, у вас будет несколько вариантов на выбор, включая отображение видео в ASCII-арте в интерпретаторе команд.

7 Администрирование файловых систем

В этой главе:

- понятие типов файловых систем Linux;
- разбивка дисков на разделы с помощью fdisk и parted;
- работа с метками посредством e21 abel и findfs;
- создание файловых систем с помощью mkfs;
- просмотр информации о файловых системах с использованием tune2fs/ dumpe2fs;
- использование областей подкачки посредством rakswap, swapon и swapoff;
- применение fstab, mount и umount для монтирования и демонтирования файловых систем;
- проверка файловых систем с помощью badblocks и fsck;
- создание зашифрованных файловых систем;
- просмотр информации о RAI D-массивах посредством mdadm;
- проверка дискового пространства с использованием du и df;
- программа управления логическими томами (LVM).

Файловые системы — это структуры, посредством которых в Linux осуществляется доступ к файлам, каталогам, устройствам и прочим элементам системы. Linux поддерживает множество типов файловых систем (ext4, VFAT, ISO9660, NTFS и т. д.) и видов носителей, на которых могут находиться файловые системы (жесткие диски, CD, флеш-диски USB, ZIP-диски, сетевые диски и т. д.).

Создание дисковых разделов и файловых систем в этих разделах и управление ими относятся к главным задачам администрирования Linux-системы. Причина заключается в том, что если вы повредите файловую систему, то, весьма вероятно, потеряете критически важные данные, хранящиеся на жестком диске вашего компьютера или съемном носителе.

В этой главе приведены команды для разбивки на разделы носителей информации, создания файловых систем, монтирования и демонтирования разделов, проверки файловых систем на предмет ошибок, а также свободного дискового пространства.

Изучение основ файловых систем

Несмотря на то что в случае с Linux доступно множество типов файловых систем, вы, вероятно, задействуете лишь некоторые из них в той или иной Linux-системе. В базовой Linux-системе, установленной на жестком диске вашего компьютера, может содержаться только три раздела: раздел подкачки (используется для размещения информации, переполняющей оперативную память), загрузочный раздел, содержащий загрузчик и ядро, и раздел корневой файловой системы. В случае с загрузочным разделом, а также разделом корневой файловой системы обычно используется такой тип файловой системы, как ext4.

В большинстве примеров, приведенных в этой главе, рассматривается файловая система ext4, чтобы проиллюстрировать, как происходит создание и управление файловой системой. Тип файловых систем ext традиционно используется как тип файловых систем по умолчанию для дистрибутивов Linux. Однако бывают ситуации, когда вам могут потребоваться другие виды файловых систем. Далее приведен список разных типов файловых систем и описание, когда есть необходимость использовать их.

- ext4 — более производительная и надежная файловая система по сравнению с ext3.
- ext3 — поддерживает журналирование для обеспечения большей сохранности данных и быстрой перезагрузки после внезапного отключения.
- ext2 — предшественница ext3, не поддерживающая журналирование.
- iso9660 — развилась из файловой системы High Sierra (которая была оригинальным стандартом, использовавшимся в случае с CD-ROM). Файловая система iso9660 может содержать расширения Rock Ridge для обеспечения поддержки длинных имен файлов и другой информации (права доступа к файлам, владение и ссылки).
- Jffs2 — Journaling Flash Filesystem version 2 (журналируемая файловая система для флеш-носителей версии 2), предназначенная для эффективной работы с флеш-дисками USB. UpeeMHHnaJFS.
- jfs — файловая система, которая использовалась компанией IBM в OS/2 Warp. Приспособлена для организации большой файловой системы и пригодна для использования в высокопроизводительных средах.
- msdos — файловая система MS-DOS. Может использоваться для монтирования более старых файловых систем MS-DOS, например использующихся для устаревших дискет.
- ntfs — Microsoft New Technology Filesystem (Файловая система новой технологии Microsoft). Полезна, когда необходимо, чтобы система использовала файлы совместно с более новыми Windows-системами (например, с двойной загрузкой или на съемных носителях).
- reiserfs — журналируемая файловая система, которая раньше применялась по умолчанию в SUSE, Slackware и других Linux-системах. Не очень хорошо поддерживается в Ubuntu.

- squashfs — сжатая файловая система «только для чтения», используемая на многих Live CD с Linux.
- swap — применяется в разделах подкачки для временного размещения данных, когда оперативная память недоступна в текущий момент.
- ufs — популярная файловая система в операционных системах Solaris и SunOS от компании Sun Microsystems.
- vfat — расширенная файловая система FAT. Полезна, когда файловой системе необходимо совместно использовать файлы вместе с более старыми Windows-системами (например, с двойной загрузкой или на съемных носителях).
- xfs — журналируемая файловая система для высокопроизводительных сред. Может масштабироваться до уровня системы, которая включает много терабайтов данных и передает информацию со скоростью в несколько гигабайтов в секунду.

Помимо файловых систем вышеперечисленных типов, также существуют так называемые сетевые *совместно используемые* файловые системы. Локальной сетевой совместно используемой файловой системой может быть ext4, ntfs или обычная файловая система другого типа. Вместе с тем все эти системы целиком или их части могут совместно использоваться посредством сетевых протоколов, например Samba (тип файловой системы smbfs или cifs), NFS (nfs) и NetWare (ncpfs).

Многие доступные типы файловых систем либо не подходят для создания новых систем, либо не полностью поддерживаются в любой из версий Linux. Например, такие типы, как minix (для Minix-систем), befs (для систем BeOS) и affs (для Amiga-систем), в основном полезны, когда необходимо осуществить монтирование и получить доступ к информации на очень старых носителях резервных копий, записанной с использованием файловых систем этих типов. Даже популярные файловые системы могут не полностью поддерживаться. Например, на момент написания этой книги файловая система reiserfs не обладала полной поддержкой со стороны Kubuntu-варианта Ubuntu.

Создание файловых систем и управление ими

Ubuntu позволяет выбрать, должен ли установщик создать разделы на диске и использовать схему файловой системы по умолчанию либо вы сделаете все это сами вручную при первой установке Linux. Установщик дает возможность выбрать полное удаление всего с жесткого диска, удатение лишь разделов Linux или использование только свободного дискового пространства для создания разделов. Чтобы указать вариант, который предполагает осуществление действий вручную, вам потребуется выбрать выполнение пользовательской разметки.

При выборе действий вручную инструменты разбивки диска на разделы позволят поделить жесткий диск на разделы согласно вашим пожеланиям. Позднее вы сможете воспользоваться множеством утилит командной строки для изменения и работы с разделами своего диска и файловыми системами, созданными в них.

Разбиение жестких дисков на разделы

Исторически сложилось, что жесткие диски персональных компьютеров использовали 32-битную таблицу разделов PC-BIOS с главной загрузочной записью (Master Boot Record, MBR). Это ограничивало размер разделов величиной 2 Тбайт и допускало наличие только четырех первичных разделов на одном диске. Использование расширенных разделов является способом преодолеть ограничение в четыре первичных. Для обхода ограничения в 2 Тбайт таблицы разделов PC-BIOS заменяются на OPT (GUID Partition Table — таблица разделов GUID).

Старая стандартная команда для работы с дисковыми разделами — fdisk. Однако, поскольку fdisk не может работать с разделами GPT, она потихоньку переходит в категорию устаревших средств. Более мощным и активно поддерживаемым инструментом является команда parted.

Чтобы опробовать примеры использования fdisk и parted, приведенные в этом разделе, рекомендую использовать флеш-диск USB, в случае с которым вам будет все равно, будут ли с него удалены данные. Когда вы подключите его, он определится как доступное устройство sd, например /dev/sdb или /dev/sdc. Безопасная команда fdisk -l поможет вам найти нужный диск, благодаря чему вы не будете рисковать повредить данные на основном жестком диске.

ПРИМЕЧАНИЕ ----------

Если вы предпочитаете использовать графические инструменты для разбивки диска на разделы, изменения размера и других манипуляций со своим жестким диском, можете попробовать инструмент разбиения диска на разделы gparted. Чтобы получить его в свое распоряжение, установите пакет gparted, который не инсталлируется по умолчанию.

Изменение разделов диска с помощью команды fdisk

Команда fdisk является полезным Linux-инструментом для просмотра информации о дисковых разделах и их изменения. Не забывайте, что модификация или удаление разделов может привести к уничтожению ценных данных, поэтому убедитесь в необходимости вносимых вами изменений, прежде чем записывать их на диск. Чтобы использовать команду fdisk для отображения информации о разделах вашего жесткого диска, введите от имени суперпользователя следующее:

```
$ sudo fdisk -l                    Показать сведения о разделах каждого диска
Disk /dev/sda: 82.3 GB, 82348277760 bytes
255 heads, 63 sectors/track, 10011 cylinders
Units = cylinders     of 16065  * 512 =  8225280 bytes
   Device Boot      Start         End      Blocks   Id  System
/dev/sdal *           1          13      104391   83  Linux
/dev/sda2            14        9881    79264710   83  Linux
/dev/sda3          9882       10011     1044225   82  Linux swap
```

В этом примере приведена информация о жестком диске объемом 80 Гбайт, который разбит на три раздела. Первый (/dev/sdal) — небольшой раздел /boot, который сконфигурирован как файловая система Linux (Id 83) и подходит для любой файловой системы ext. Следует отметить, что звездочка (*) выступает в роли индикатора того, что первый раздел — загрузочный. Следующий раздел закреплен за корневой файловой системой и тоже является разделом Linux. Последний раздел — это Linux swap.

ПРИМЕЧАНИЕ

Начиная с версии 2.6.20 ядра Linux, для дисков как с IDE-, так и SCSI-интерфейсом используются имена устройств /dev/sd?, где вместо знака ? указывается буква (а, Ь или с и т. д.). В более старых версиях Ubuntu имена /dev/sd? использовались только для SCSI-дисков и флеш-дисков USB. Для жестких дисков с интерфейсом ШЕ указывались имена /dev/hd?.

Если в системе присутствует несколько дисков, то fdisk -1 покажет информацию о них всех, если только вы **не укажете, о каком именно диске** хотите увидеть сведения:

```
$ sudo fdisk -1 /dev/sdb          Показать разделы определенного диска
```

Для **работы с определенным диском** с помощью команды fdisk необходимо просто указать нужный диск без каких-либо дополнительных параметров:

```
$ sudo fdisk /dev/sda             Начать интерактивный сеанс fdisk
Command (m for help): m           Нажмите клавишу М для вывода на экран
                                  справочного текста, как показано

Command action
a toggle a bootable flag
b edit bsd disklabel
c toggle the dos compatibility flag
d delete a partition
1 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):
```

Опираясь на отображенную подсказку, воспользуйтесь любой из показанных команд для работы с жестким диском. В частности, можете задействовать p (для вывода на экран той же информации, которую показывает fdisk -1), п (для создания нового раздела), d (для удаления существующего раздела), 1 (для отображения известных типов файловых систем) или t (для изменения типа файловой системы

для раздела). В приведенных далее примерах демонстрируются некоторые варианты использования команды fdisk на практике:

```
Command (m for help): d                    Отправить запрос на удаление раздела
Partition number (1-4): 4                  Ввести номер раздела для удаления
...
Command (m for help): n                    Создать новый раздел на диске
First cylinder (1-4983. default 1): 1      Выбрать начало (или нажмите Enter)
Last cylinder ... (default 4983): 4983     Выбрать конец (или нажмите Enter)
...
Command (гл for help): a                   Сделать раздел загрузочным
Partition number (1-3): 1                  Ввести номер загрузочного раздела
...
Command (гл for help): t                   Выбрать тип файловой системы
Partition number (1-3): 3                  Выбрать раздел для изменения
Hex code (type L to list codes): 82 Назначить раздел как Linux swap
```

Если вы не укажете иное, `fdi sk` будет предполагать, что новым разделом является раздел Linux (`83`). Вы можете ввести `L`, чтобы увидеть такой же список типов файловых систем и шестнадцатеричных кодов, который генерирует команда 1. Как уже отмечалось ранее, 82 позволяет назначить раздел как swap. Среди других типов разделов Linux, которые могут вас заинтересовать, можно назвать Linux-расширенный (85), Linux-LVM (8e), Linux-программный массив (fd) и EFI/GTP (ee).

Для разделов Windows вы можете назначать разделы HPFS/NTFS (`7`), Windows 95 FAT32 (`b`), FAT16 (`6`) или Windows 95 FAT32 LBA (`c`). Среди других подобных типов файловых систем UNIX можно назвать Minix (`be` или `bf`), BSD/OS (`e4`), FreeBSD (`ee`), OpenBSD (`ef`), NeXTSTEP (`f0`), Darwin UFS (`f1`) и NetBSD (`f4`). Любой из этих типов файловых систем может оказаться полезным, если у вас будут старые носители резервных копий с записанной с использованием этих файловых систем информацией, которую вы захотите восстановить.

До сих пор вы не вносили никаких перманентных изменений в таблицу разделов. Если вы сейчас *полностью уверены*, что ваши новые настройки корректны, введите w, чтобы записать соответствующие изменения в таблицу разделов. Для отмены ваших изменений (или выхода после их записи) введите q, чтобы завершить сеанс `fdisk`.

Копирование таблиц разделов с помощью sfdisk

Чтобы **сделать резервную копию или дублировать таблицу разделов диска,** используйте sfdisk:

```
$ sudo sfdisk -d /dev/sda > sda-table          Произвести резервное копирование
                                               таблицы разделов в файл
$ sudo sfdisk /dev/sda < sda-table             Восстановить таблицу разделов
                                               из файла
$ sudo sfdisk -d /dev/sda | sfdisk /dev/sdb Скопировать таблицу разделов
                                               с диска a на диск b
```

Изменение разделов диска с помощью parted

Как и fdi sic, команду parted можно использовать для просмотра информации о дисковых разделах и их изменения. Однако parted обладает и другими полезными возможностями. Вот как можно **отобразить информацию о разделах определенного диска,** которым в данном случае является /dev/sda, используя parted:

```
$ sudo parted /dev/sda print
Model: ATA ST3160815AS (scsi)
Disk /dev/sda: 160GB
Sector size (logical/physical): 512B/512B
Partition Table: msdos
Number  Start   End     Size    Type      File system  Flags
 1      1049kB  256MB   255MB   primary   ext 2        boot
 2      257MB   160GB   160GB   extended
 5      257MB   160GB   160GB   logical                1 vm
```

Из этого вывода можно узнать, имеет ли диск классическую метку msdos (таблицу разделов) либо gpt. В данном случае таблицей разделов является msdos.

Для *выполнения parted, в интерактивном режиме* введите parted, указав далее имя устройства для хранения, с которым желаете поработать (например, /dev/sda), либо, если у вас только одно устройство для хранения, просто введите parted. Если у вас есть флеш-диск USB, с которым вам все равно, будут ли с него удалены данные, то рекомендую использовать его, когда вы будете применять parted в первый раз. Здесь USB-диск находится в /dev/sdc:

```
$ sudo parted /dev/sdc
GNU Parted 2.3
Using /dev/sdc
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted)
```

Для интерактивного использования parted либо вводите команды целиком, либо набирайте первые буквы и используйте клавишу **Tab** для завершения ввода команды (как, например, при работе в bash). Если вы захотите действовать по-настоящему рационально, просто вводите количество букв, достаточное, чтобы позволить parted догадаться, какую именно команду вы хотите выполнить, как, например, при работе в Cisco IOS: p для ввода print, mkl для ввода mkl abel и т. д.

ВНИМАНИЕ----------

В отличие от fdisk, команда parted незамедлительно применяет изменения, вносимые вами в разделы, не записывая эти изменения на диск явным образом, поэтому не стоит рассчитывать на то, что вы сможете отменить любые изменения, просто прекратив выполнение parted.

Для каждой команды во время сеанса parted у вас также есть выбор — ввести ее со всеми аргументами (например, mkpart logical ext4 1MB 2GB) или набрать только команду (mkpart) и parted будет направлять вас в интерактивном режиме.

```
(parted) mkpart                                    Создать новый раздел
Partition type? [logical]? primary
File system type? [ext2]? ext4
Start? 1MB
End? 2GB
```

ВНИМАНИЕ----------

Приведенные далее команды изменят конфигурацию вашего диска. Именно поэтому я использую флеш-диск USB, не содержащий никаких данных, которые я боюсь потерять. Если вы допустите ошибку, работая с первичным диском, то можете сделать так, что ваши данные окажутся недоступными, а система перестанет загружаться.

Команду `parted` не рекомендуется использовать для изменения размера разделов, поскольку она может сделать файловые системы на дисках непригодными для применения. Если вы все же воспользуетесь parted для изменения размера раздела, то она также попытается изменить размер файловой системы, расположенной в нем. Команда `parted` не поддерживает журналирование и другие функциональные возможности, необходимые файловым системам ext3 и ext4, поэтому операции по изменению размеров будут терпеть неудачу в большинстве соответствующих случаев.

Изменение размера — рискованное мероприятие при работе со стандартными дисковыми разделами, поэтому если в какой-то момент вам покажется, что необходимо изменить размер раздела, рассмотрите возможность использования программы-менеджера логических томов (Logical Volume Manager, LVM) для создания логических разделов и управления ими. Она рассматривается позднее в текущей главе.

Для **изменения размеров разделов NTFS** вы можете использовать команду `ntfsresize`. В Ubuntu она входит в пакет ntfsprogs. В этом пакете также есть команды для создания (`mkfs.ntfs`), исправления (`ntfsfix`) и получения информации (`ntfsinfo`) о разделах NTFS.

Работа с метками, используемыми с файловыми системами

Термин *«метка»* в контексте дисковых разделов может иметь отношение к двум вещам. *Метку диска* можно использовать в качестве другого имени для таблицы разделов, как видно из вывода, генерируемого командой parted. *Метка раздела* тоже может играть роль имени, но для отдельного раздела, содержащего файловую систему ext.

Метки обеспечивают следующие преимущества.

- Вы можете использовать метку как постоянное имя при монтировании файловой системы. Возможно, `/dev/sda` определится как `/dev/sdb` после следующей перезагрузки системы. Монтирование с использованием метки позволяет вам присвоить разделу уникальное имя.

О Если метка расположена на съемном USB-диске, то, когда вы подключите его, Linux будет использовать ее имя в качестве названия каталога, в который будет выполняться монтирование. Например, если меткой окажется mydocs, то USB-диск будет монтирован в каталог /media/mydocs.

Чтобы **увидеть метку раздела** (если она у него имеется), используйте команду e21abel:

```
$ sudo e21abel /dev/sda2
/home
```

Чтобы **задать метку для раздела,** введите следующее:

```
$ sudo e21abel /dev/sda2 mypartition
```

Имейте в виду, что /etc/fstab можно настроить на использование метки раздела для монтирования соответствующего раздела, как показано в следующем примере. Изменение этой метки может привести к тому, что система перестанет загружаться.

```
l_ABEL=/boot          /boot ext4 defaults 1 2
```

Чтобы **найти раздел, если вам известна только его метка,** введите следующее:

```
$ sudo findfs LABEL=mypartition
/dev/sdc2
```

Форматирование файловой системы

Разбив диск на разделы, вы сможете создать файловую систему по своему выбору в каждом из них. В большинстве Linux-систем есть команды, необходимые для создания и проверки файловых систем, которые широко используются в Linux. Командами для **форматирования и проверки** файловых систем являются соответственно mkfs и fsck.

Команда mkfs служит в качестве внешнего интерфейса для многих команд, предназначенных для форматирования файловых систем определенных типов, таких, например, как mkfs. ext2, mkfs. ext3, mkfs. ext4, mkfs. cramfs, mkfs. msdos, mk fs. ntfs и mkfs. vfat. Если добавить пакеты, поддерживающие другие файловые системы, то станут доступными дополнительные команды mkfs для эффективной работы с mkfs. К ним относятся mkfs.bfs, mkfs .mini x, mkfs.xfs и mkfs.xiafs. Используйте каждую команду напрямую (например, mkfs.vfat /dev/sdbl) или посредством mkfs (к примеру, mkfs -t vfat /dev/sdbl).

Создание файловой системы в разделе диска

Среди основных программных пакетов, необходимых для создания и проверки файловой системы Ubuntu, можно назвать util-linux (включает в себя команду mkfs и другие приложения общего назначения) и e2fsprogs (включает в себя специальные приложения файловых систем ext2/ext.3). Специальные команды mkfs для

различных типов файловых систем входят в программные пакеты ntfsprogs (предназначен для работы в NTFS), dosfstools (MS-DOS и VFAT), xfsprogs (XFS), jfsutils (JFS), mtd-utils (JFFS и JFFS2) и reiserfs-utils (reiserfs). Основные приложения устанавливаются вместе с Ubuntu.

Вот примеры **использования команды mkfs для создания файловых систем** (сначала обязательно указывать параметр -t):

```
$ sudo mkfs -1: ext4 /dev/sdbl                    Создать файловую систему ext4 в разделе
                                                  sbal
$ sudo mkfs -t ext4 -v -c /dev/sdbl Сгенерировать подробный вывод/сканировать
                                                  на предмет поврежденных блоков
$ sudo mkfs.ext4 -c /dev/sdbl                     Дает тот же результат, что и предыдущая
                                                  команда
```

Если вы захотите добавить метку для нового раздела при создании файловой системы, используйте параметр -L:

```
$ sudo mkfs.ext4 -c -L mypartition /dev/sdbl Добавить метку mypartition
```

Создание виртуальной файловой системы

Если вы хотите лишь ознакомиться с различными типами файловых систем или сделать файловую систему более гибкой (другими словами, не привязанной к физическому диску), то можете создать *виртуальную файловую* систему. Виртуальная файловая система — это та, которая располагается в файле в существующей файловой системе. Вы можете отформатировать ее как файловую систему любого типа на свое усмотрение, переносить и использовать на разных компьютерах.

Виртуальные файловые системы полезны для таких операций, как создание Live CD или запуск специализированных виртуальных операционных систем. В приведенных далее примерах демонстрируется создание файла образа диска размером 1 Гбайт и его форматирование как файловой системы с последующим монтированием для обеспечения доступа к данным в файловой системе:

```
$ dd if=/dev/zero of=mydisk count=2048000          Создать заполненный нулями файл
                                                   размером 1 Гбайт
$ du -sh mydisk                                    Проверить размер виртуального
                                                   файла
1001M mydisk
$ mkfs -t ext4 mydisk                              Создать файловую систему в mydisk
mydisk is not a block special device.
Continue (y/n): y
$ sudo mkdir /mnt/image                            Создать точку монтирования
$ sudo mount -o loop mydisk /mnt/image             Монтировать mydisk в /mnt/image
```

В приведенной процедуре команда dd создает пустой файл образа диска объемом 2 048 000 блоков (примерно 1 Гбайт). Команда mkfs позволяет создать файловую систему любого типа на ваш выбор (в данном случае создается ext4). Файл не является файлом специального блочного устройства, как это имеет место при форматировании дисковых разделов, поэтому mkfs предупредит вас,

прежде чем приступит к созданию файловой системы. После создания точки монтирования остается лишь указать, что вы монтируете файл (mydi sk) как петлевое устройство (-o loop). Следует отметить, что mount — единственная команда, показанная в приведенном выше коде, которая требует наличия привилегий суперпользователя.

Выполнив монтирование виртуальной файловой системы, которое в этом примере производится в /mnt/image, вы сможете осуществлять к ней доступ как к любой системе. Закончив работать с файловой системой, выйдите и демонтируйте ее:

```
$ sudo cd /mnt/image        Перейти в точку монтирования
$ sudo mkdir test           Создать каталог в файловой системе
$ sudo cp /etc/hosts .      Скопировать файл в файловую систему
$ cd                        Выйти из файловой системы
$ sudo umount /mnt/image    Демонтировать файловую систему
```

Демонтировав виртуальную файловую систему, вы можете перенести ее в другую систему или записать на CD, чтобы использовать в другом месте. Если файловая система вам больше не нужна, просто удалите соответствующий файл.

Просмотр и изменение атрибутов файловых систем

Используя команду tune2fs или dumpe2fs, вы можете **просматривать атрибуты файловых систем ext2, ext3 и ext4.** Команду tune2fs также можно применять для **изменения атрибутов файловых систем.** Для **создания раздела подкачки** используйте команду swapfs. Вот примеры (обе команды генерируют одинаковый вывод):

```
$ sudo tune2fs -1 /dev/sdal | less    Показать настраиваемые атрибуты файловой системы
$ sudo dumpe2fs -h /dev/sdal          Генерирует тот же вывод, что и tune2fs
dumpe2fs 1.42 (29-Nov-2011)
Filesystem volume name:   <none>
Last mounted on:          <not aval 1able>
Filesystem UUID:          f5f261d3- 3879-41d6-8245-f2153b003204
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      filetype sparse_super
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              7914368
Block count:              7907988
Reserved block count:     395399
Free blocks:              5916863
Free inodes:              7752077
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      1022
```

```
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         32704
Inode blocks per group:   1022
Filesystem created:       Fri Jun 15 12:13:17 2007
Last mount time:          Sat Mar 12:13:14 2013
Last write time:          Sat Mar 12:13:14 2013
Mount count:              2
Maximum mount count:      29
```

В данном выводе приведено множество информации о файловой системе. Например, если у вас имеется файловая система, которой необходимо создавать много небольших файлов (к примеру, новостной сервер), то можете проверить, не исчерпали ли вы индексные дескрипторы. Задание значения для Maximum mount count гарантирует, что файловая система будет проверена на предмет ошибок после ее монтирования выбранное количество раз. Вы также можете найти даты и время создания системы, ее последнего монтирования и последней записи в нее.

Чтобы **изменить настройки существующей файловой системы ext2, ext3 или ext4,** можно воспользоваться командой tune2fs. Приведенная далее команда изменяет количество монтирований, по достижении которого запускается принудительная проверка файловой системы:

```
$ sudo tune2fs -c 31 /dev/sdal  Задает количество монтирований,
                                по достижении которого будет запущена
                                принудительная проверка
tune2fs 1.42 (29-Nov-2011)
Setting maximal mount count to 31
```

Если вы захотите обеспечить запуск принудительной **проверки файловой системы через определенный интервал времени,** а не по достижении определенного количества монтирований, деактивизируйте соответствующую проверку, задав отрицательное значение в виде единицы с минусом (-1):

```
$ sudo tune2fs -c -1 /dev/sdal
t.une2fs 1.42 (29-Nov-2011)
Setting maximal mount count to -1
```

Используйте параметр - i, чтобы активизировать **проверку, зависящую от времени.** Примеры:

```
$ sudo tune2fs -i 10 /dev/sdal   Запустить проверку через 10 дней
$ sudo tune2fs -i Id /dev/sdal   Запустить проверку через 1 день
$ sudo tune2fs -i 3w /dev/sdal   Запустить проверку через 3 недели
$ sudo tune2fs -i 6m /dev/sdal   Запустить проверку через 6 месяцев
$ sudo tune2fs -i 0 /dev/sdal    Деактивизировать проверку, зависящую от времени
```

У вас всегда должна быть активизирована либо проверка, запускаемая по достижении определенного количества монтирований, либо проверка, зависящая от времени. Используйте параметр - j, чтобы **преобразовать файловую систему ext2 в ext3** (задействовав при этом журналирование):

```
$ sudo tune2fs -j /dev/sdal  Обеспечить журналирование преобразования ext2 в ext3
```

Создание и использование разделов подкачки

Разделы подкачки необходимы в Linux-системах для размещения данных, переполняющих оперативную память. Если вы не создали раздел подкачки при установке Linux, вы сможете создать его позднее с помощью команды mkswap. Можете создать раздел подкачки либо в обычном дисковом разделе, либо в файле, отформатированном как раздел подкачки. Вот примеры:

```
$ sudo mkswap /dev/sda6                    Отформатировать sda6 как раздел подкачки
Setting up swapspace version 1, size = 205594 kB
```

Для **проверки своей области подкачки на предмет поврежденных блоков** используйте команду mkswap с параметром -c:

```
$ sudo mkswap -c /dev/sdal
```

Если у вас нет запасного раздела, можете **создать область подкачки в файле:**

```
$ sudo dd if=/dev/zero of=/mnt/swapfile count=65536
65536+0 records in
65536+0 records out
33554432 bytes (34 MB) copied, 1.56578 s. 21.4 MB/s
$ sudo chmod 600 /mnt/swapfile
$ sudo mkswap /mnt/swapfile
Setting up swapspace version 1. size = 67104 kB
```

Команда dd в приведенном выше коде создает файл размером 32 Мбайт с именем swapfi 1 e. Команда chmod блокирует права доступа к нему во избежание генерирования командой swapon предупреждения впоследствии. Команда mkswap форматирует файл /mnt/swapfile, чтобы сделать его разделом подкачки.

Когда вы создадите раздел подкачки или файл подкачки, вам потребуется **дать системе указание использовать область подкачки,** созданную вами с помощью команды swapon. Это похоже на происходящее во время загрузки. Взгляните на следующие примеры:

```
$ sudo swapon /dev/sdal                    Использовать /dev/sdal как раздел подкачки
$ sudo swapon -v /dev/sdal                 Генерировать более подробный вывод
                                           при использовании раздела подкачки
swapon on /dev/sdal
$ sudo swapon -v /mnt/swapfile Использовать /mnt/swapfile как файл подкачки
swapon on /mnt/swapfile
```

Вы также можете использовать команду swapon для **просмотра списка своих файлов подкачки и разделов подкачки:**

```
$ swapon —s        Показать все используемые файлы подкачки и разделы подкачки
Filename                Type            Size    Used    Priority
/dev/sda5               partition       1020088 142764  -1
/mnt/swapfile           file            65528   0       -6
```

До завершения использования области подкачки убедитесь, что с ней не задействуется никакого пространства (ищите значение 0 в столбце Used). Затем, чтобы

прекратить использование области подкачки, можете прибегнуть к команде swapoff:

```
$ sudo swapoff -v /mnt/swapfile
swapoff on /mnt/swapfile
```

Области подкачки обладают разными приоритетами. Ядро будет в первую очередь использовать области подкачки с высоким приоритетом, а затем те, которые имеют более низкий приоритет. Области с одинаковым приоритетом используются попеременно. Вы можете **указать приоритет для своей области подкачки,** когда задействуете ее, воспользовавшись параметром -p:

```
$ sudo swapon -v -p 1 /dev/sdal Присвоить sdal высший приоритет
```

Монтирование и демонтирование файловых систем

Прежде чем использовать обычную файловую систему без подкачки, потребуется добавить ее в каталог в дереве файловых систем на вашем компьютере путем ее *монтирования.* Корневая (/) и другие файловые системы, используемые вами на постоянной основе, обычно монтируются автоматически, исходя из записей в файле /etc/fstab. Монтировать другие файловые системы можно вручную по мере необходимости, используя команду mount.

Монтирование файловых систем из файла fstab

При первой установке Linux файл /etc/fstab обычно конфигурируется таким образом, чтобы он содержал информацию о ваших корневой и других файловых системах. Затем можно сделать так, чтобы эти файловые системы монтировались во время загрузки либо были готовы к монтированию вручную (вместе с точками монтирования и другими параметрами, готовыми к использованию при монтировании вручную).

Вот пример файла /etc/fstab:

```
Proc                    /proc           proc nodev.noexec.nosuid    0 0
/dev/mapper/abc-root    /               ext4errors=remount-ro       0 1
LABEL=/boot             /boot           ext2 defaults               0 2
/dev/mapper/abc-swap    swap            swap defaults               0 0
/dev/sdbl               /mnt/windows    vfat noauto                 0 0
```

ПРИМЕЧАНИЕ----------

Для ясности UUID-данные для файловой системы /boot не были приведены в продемонстрированном выше примере. В случае с файловой системой /boot вы обычно будете видеть такую запись, как, например, UUID=da2dbc48-862e-4fbe-9529-a88b57bl5bac, а не просто UUID. Для уникальной идентификации дискового раздела можно использовать либо UUID, либо LABEL.

Все файловые системы в приведенном выше примере монтируются автоматически, за исключением /dev/sdbl (о чем свидетельствует параметр noauto). Корневой

раздел диска (/), а также раздел диска swap конфигурируются как тома LVM (Logical Volume Management — «управление логическими томами»). Тома LVM облегчают увеличение и уменьшение файловых систем с одновременным сохранением прежних имен устройств. Дисковый раздел /dev/sdbl был добавлен вручную в приведенном выше примере для монтирования раздела Windows, расположенного на втором жестком диске.

Файл /etc/fstab обычно уже не содержит информации о съемных носителях. Это потому, что инструмент HAL (Hardware Abstraction Layer — «слой аппаратных абстракций») автоматически обнаруживает съемные носители и монтирует их в соответствующих точках монтирования в каталоге /medi a (в зависимости от таких вещей, как, например, идентификатор тома).

Далее приведено описание каждого поля в файле /etc/fstab.

- 1 — имя устройства, которое представляет файловую систему. Первоначально это поле содержало название устройства раздела для монтирования (например, /dev/sdal). Теперь оно также может содержать LABEL, универсальный уникальный идентификатор (UUID) или удаленную файловую систему (NFS либо CIFS) вместо имени устройства.
- 2 — точка монтирования в файловой системе. Файловая система содержит все данные, начиная с точки монтирования и далее вниз по структуре дерева каталогов до тех пор, пока в какой-то момент вслед за ней не будет смонтирована другая файловая система.
- 3 — тип файловой системы. Сведения о многих распространенных типах файловых систем содержатся в разделе «Изучение основ файловых систем» данной главы.
- 4 — параметры команды mount. К примерам параметров команды mount относятся noauto (для предотвращения монтирования файловой системы во время загрузки) и го (для монтирования файловой системы только для чтения). Чтобы дать любому пользователю возможность монтировать файловую систему, можете добавить параметр user или owner в это поле. Параметры должны быть разделены запятыми. Информацию о других поддерживаемых параметрах вы найдете на MAN-странице, посвященной команде mount (с использованием параметра -o).
- 5 — использовать ли dump в отношении файловой системы. Это поле существенно, если вы будете выполнять резервное копирование с помощью dump (что бывает редко). Значение 1 говорит о необходимости резервного копирования файловой системы, а значение 0 подразумевает обратное.
- 6 — необходима ли проверка файловой системы. Значение в этом поле является индикатором того, нужна ли проверка файловой системы с помощью fsck. Значение 0 свидетельствует о необходимости проверки файловой системы. Значение 1 будет индикатором, что файловую систему нужно проверить в первую очередь (такой подход применяется для корневой файловой системы). Значение 2 говорит о том, что файловая система может быть проверена в любой момент после проверки корневой файловой системы.

Вы можете создавать собственные записи в файле /etc/fstab для разделов любого жесткого диска или съемного носителя. В удаленных файловых системах (NFS, Samba и др.) также могут содержаться записи в файле /etc/fstab для автоматического монтирования этих файловых систем во время загрузки или позднее вручную.

Монтирование файловых систем с помощью команды mount

Команда mount используется для просмотра смонтированных файловых систем, а также непосредственного монтирования любых локальных (на жестком диске, USB-диске, CD, DVD и т. д.) или удаленных (NFS, Samba и т. п.) файловых систем. Вот пример выполнения команды mount для **отображения смонтированных файловых систем:**

```
$ mount          Показать смонтированные удаленные и локальные файловые системы
proc on /proc type proc (rw.noexec.nosuid.nodev)
/dev/mapper/atc-root on / type ext4 (rw,errors=remount-ro)
/dev/sdal on /boot type ext2 (rw)
192.1.0.9:/volumel/books on /mnt/books type nfs (rw,addr=192.1.1.9)
/dev/sdbl on /media/myusb type ext4 (rw,nosuid,nodev,uhelper=udisks)
```

Используйте параметр -t для **отображения только смонтированных файловых систем опред еленного типа:**

```
$ mount -t ext4 Показать смонтированные файловые системы ext4
/dev/mapper/afcc-root on / type ext4 (rw,errors=remount-ro)
/dev/sdbl on
    /media/myusb type ext4 (rw,nosuid,nodev.uhelper=udisks)
```

Для **отображения меток разделов с информацией о монтировании** используйте параметр -':

```
$ mount -t ext4 -1      Показать смонтированные файловые системы ext4
                        и соответствующие метки
/dev/sda7 on / type ext4 (rw) [/123]
/dev/sda6 on /mnt/debian type ext4 (rw) [/mnt/debian]
/dev/sda3 on /mnt/slackware type ext4 (rw) [/mnt/slackware]
```

Вот простой пример использования команды mount для монтирования устройства /dev/sdbl в существующий каталог /mnt/mymount:

```
$ sudo mount /dev/sdbl /mnt/mymount/      Монтировать локальную файловую
                                          систему
$ sudo mount -v /dev/sdbl /mnt/mymount/   Монтировать /dev/sdbl с выводом
                                          более подробной информации
mount: you dicn't specify a filesystem type for /dev/sdbl
I will try type ext4
/dev/sdbl on /mnt/mymount type ext4 (rw)
```

В приведенных выше примерах команда mount будет искать запись для dev/sdbl в файле /etc/ 'stab либо пытаться угадать тип файловой системы.

Используйте -t, чтобы **явно указать тип файловой системы для монтирования:**

```
$ sudo mount -v -t ext4 /dev/sdbl /mnt/mymount/ Монтировать файловую систему ext4
/dev/sdbl on /mnt/mymount type ext4 (rw)
```

Вы также можете **отобразить метку/имя монтируемого раздела:**

```
$ sudo mount -vl -t ext4 /dev/sdbl /mnt/mymount/ Монтировать и показать метку
```

Если вы монтируете что-то, информация о чем уже содержится в файле fstab, то потребуется указать только один элемент — точку монтирования или устройство. Например, в случае со следующей записью в fstab:

```
/dev/sdbl /mnt/mymount ext4 defaults 1 2
```

вы можете прибегнуть к одному из двух следующих вариантов:

```
$ sudo mount -v /dev/sdbl Монтировать файловую систему с использованием
только имени устройства
/dev/sdbl on /mnt/mymount type ext4 (rw)
$ sudo mount -v /mnt/mymount/ Монтировать файловую систему
с использованием только точки монтирования
/dev/sdbl on /mnt/mymount type ext4 (rw)
```

Вы можете **указать параметры команды mount,** добавив -o и список параметров, разделенных запятыми. Эти параметры аналогичны тем, которые можно добавить в поле 4 файла /etc/fstab. По умолчанию разделы монтируются с доступом с правом чтения /записи. Вы можете явно указать, следует ли **монтировать файловую систему с правом чтения/записи (rw) или только для чтения (ro):**

```
$ sudo mount -v -t ext4 -o rw /dev/sdbl /mnt/mymount/ Монтировать с правом
чтения/записи
/dev/sdbl on /mnt/mymount type ext4 (rw)
$ sudo mount -v -t ext4 -o ro /dev/sdbl /mnt/mymount/ Монтировать только
для чтения
/dev/sdbl on /mnt/mymount type ext4 (ro)
```

Далее приведены другие полезные параметры команды mount.

- noatime — не обновляет время доступа к файлам. Подходит для файловых систем с большим количеством операций ввода/вывода, которые имеют место, например, в случае с почтовыми серверами и журналами.
- noexec — предотвращает выполнение двоичных файлов, расположенных в соответствующей файловой системе. Может использоваться с целью повышения безопасности, например, для /tmp в средах с неавторизованными пользователями.
- remount — изменяет параметры смонтированной файловой системы. Используя его, вы можете демонтировать файловую систему, а затем снова ее смонтировать, задействовав новые параметры в одной команде. В данном примере мы изменяем приведенное ранее монтирование с правом чтения/записи на вариант «только для чтения»:

```
$ sudo mount -v -o remount.ro /dev/sdbl
/dev/sdbl on /mnt/mymount type ext4 (ro)
```

- --bind — монтирует существующую файловую систему в другое расположение в дереве. При условий, что /dev/sdbl уже смонтирован в /mnt/mymount, введите следующее:

```
$ sudo mount --bind -v /mnt/mymount/ /tmp/mydir/
/mnt/mymount on /tmp/mydir type none (rw.bind)
```

  Теперь одна и та же файловая система доступна в двух местах. Новая точка монтирования обладает теми же параметрами монтирования, что и оригинальная.

- -move — перемещает файловую систему из одной точки монтирования в другую. При условии, что /dev/sdbl уже смонтирован в /mnt/mymount, приведенная далее строка переместит файловую систему в /tmp/mydir:

```
$ sudo mount -v --move /mnt/mymount/ /tmp/mydir/
/mnt/mymount on /tmp/mydir type none (rw)
```

Подобно тому как для подкачки используется файл, вы также можете создать файловую систему в файле, а затем выполнить его *петлевое* монтирование. Создание и монтирование такого файла было описано в подразделе «Создание виртуальной файловой системы» ранее в этой главе. Распространенной ситуацией, в которой вам может потребоваться **петлевое монтирование файла,** является имеющая место после загрузки установочного CD или Live CD с Linux. Произведя петлевое монтирование образа этого компакт-диска, вы сможете просматривать его содержимое и копировать файлы из образа на свой жесткий диск.

В приведенном далее примере команде mount разрешается автоматически выбирать существующее петлевое устройство при монтировании файла образа CD (тип файловой системы — iso9660). Из командного вывода видно, что было выбрано устройство /dev/loop0:

```
$ sudo mount -v -t iso9660 -o loop /tmp/myimage.iso /mnt/mymount/
mount: going to use the loop device /dev/1oop0
/tmp/myimage.iso on /mnt/mymount type ext4 (rw,1oop=/dev/loop0)
```

В следующем примере я загрузил образ USB-диска с Linux, который называется diskboot. img, в /tmp. Вот пример **монтирования загрузочного образа:**

```
$ sudo mount -v -o loop /tmp/diskboot.img /mnt/mymount
mount: going to use the loop device /dev/1oop0
mount: you didn't specify a filesystem type for /dev/1oop0
       I will try type vfat
/tmp/diskboot.img on /mnt/mymount type vfat (rw,loop=/dev/loop0)
```

Чтобы **узнать статус петлевых устройств,** используйте команду losetup:

```
$ sudo losetup /dev/1oopO                    Показать смонтированные петлевые устройства
/dev/1oopO: [0807]4009045 (/tmp/diskboot.img)
```

Если петлевое монтирование зависнет или у вас возникнут проблемы во время демонтирования, попробуйте выполнить соответствующую операцию, как показано далее:

```
$ sudo losetup -d /dev/1oopl                 Принудительно демонтировать смонтированное
                                             петлевое устройство
```

ПРИМЕЧАНИЕ----------

Команду mount также можно использовать для присоединения к совместно используемым ресурсам NFS или Samba/Windows CIFS. Информацию о монтировании файловых систем этих типов вы найдете в гл. 12.

Демонтирование файловых систем с помощью unmount

Чтобы демонтировать файловую систему, задействуйте команду unmount. Вы можете демонтировать файловую систему, используя имя устройства или точку монтирования. Предпочтительнее демонтирование с использованием точки монтирования во избежание путаницы при связанном монтировании (одно устройство, несколько точек монтирования). Вот пример использования обоих упомянутых вариантов с генерированием при этом подробного вывода:

```
$ sudo umount -v /dev/sdbl          Демонтировать с использованием имени устройства
/dev/sdbl umounted
$ sudo umount -v /mnt/mymount/      Демонтировать с использованием точки монтирования
/tmp/diskboot.img umounted
```

Если устройство окажется занятым, то демонтирование потерпит неудачу. Распространенной причиной провала операции демонтирования является открытый интерпретатор команд с текущим каталогом, который располагается в каталоге в смонтированной файловой системе:

```
$ sudo umount -v /mnt/mymount/
umount: /mnt/mymount: device is busy
umount: /mnt/mymount: device is busy
```

Причина, по которой устройство оказывается занятым, не всегда очевидна. Вы можете использовать 1 sof, чтобы вывести на экран список открытых файлов, а затем **найти в нем точки монтирования,** которые вас интересуют:

```
$ sudo Isof | grep mymount          Найти открытые файлы в разделе mymount
bash 11224 chris cwd DIR 8,17          4096     2 /mnt/mymount
```

Как видите, процесс bash, запущенный пользователем **chris** с PID 9341, препятствует демонтированию раздела mymount.

Другой причиной, по которой файловая система может быть занята, является **выполнение отложенного демонтирования:**

```
$ sudo umount -vl /mnt/mymount/  Выполнить отложенное демонтирование
```

При *отложенном демонтировании* файловая система демонтируется с дерева сразу же, но с ожиданием, пока устройство не освободится, прежде чем все убирать. Демонтирование съемных носителей также можно выполнять с помощью eject. Приведенная далее строка **обеспечивает демонтирование CD и извлечение его** из привода:

```
$ sudo eject /dev/cdrom             Демонтировать и извлечь CD
```

Проверка файловых систем

В Linux вместо утилиты scandisk, используемой в Windows, вы можете прибегнуть к команде badblocks для сканирования устройства на предмет поврежденных блоков на физическом уровне или к команде f sck для сканирования файловой системы на наличие ошибок на логическом уровне. Вот как выполняется **сканирование на предмет поврежденных блоков:**

$ sudo badblocks /dev/sdbl *Физически сканировать диск на предмет поврежденных блоков*

$ sudo badblocks -v /dev/sdbl *Обеспечить генерирование подробного вывода при сканировании жесткого диска*

```
Checking block; 0 to 200781
Checking for bod blocks (read-only test): done
Pass completed. 0 bad blocks found.
```

По умолчанию badbl ocks производит безопасное тестирование блоков, при котором осуществляется только чтение. Вы также можете выполнять безопасное тестирование с чтением/записью. Это самое продолжительное по времени тестирование, однако и лучшее, которое вы можете произвести, не уничтожив данные на устройстве. Добавьте -s, чтобы **наблюдать за процессом:**

$ sudo badblocks -vsn /dev/sdbl *Выполнить безопасную проверку на предмет поврежденных блоков*

```
Checking for bad blocks in non-destructive read-write mode
From block 0 to 200781
Testing with random pattern: Pass completed, 0 bad blocks found.
```

Для тестирования диска, не содержащего данных, которые необходимо сохранить, используйте приведенную далее команду, производящую **более быстрое тестирование с чтением/записью, при котором данные уничтожаются:**

ВНИМАНИЕ ----------

Использование этой команды приведет к удалению всех данных, содержащихся в разделе.

$ sudo badblocks -vsw /dev/sdal *Выполнить проверку на предмет поврежденных блоков с уничтожением данных*

```
Checking for bad blocks in read-write mode
From block 0 to 200781
Testing with pattern Oxaa: done
Reading and comparing: done
. . .
Pass completed. 0 bad blocks found.
```

Вы можете сделать **несколько заходов с использованием badblocks;** например, приведенная далее строка команды может использоваться для тестирования жесткого диска с последующим выводом на экран информации о его возможном выходе из строя в начальный период эксплуатации:

$ sudo badblocks -vswp 2 /dev/sdal

Как и mkfs, команда fsck является внешним интерфейсом в случае с утилитами, специфичными для файловой системы. Вы можете **проверить файловую систему ext,** просто добавив имя устройства дискового раздела, который хотите проверить с помощью команды fsck:

```
$ sudo fsck /dev/sdbl
fsck from uti1-1inux 2.20.1
mypart was not cleanly unmounted, check forced.
fsck 1.42 (29 - Nov-2011)
e2fsck 1.42 (29-Nov-2011)
Pass 1: Checking Inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3; Checking directory connectivity
Pass 4: Checking reference counts
Pass 5; Checking group summary information
mypart: 11/50200 files (9.1% non-contiguous), 12002/200780 blocks
```

Вы можете **снабдить fsck другими параметрами,** например -T (чтобы на экран не выводилась бесполезная информация о номере версии fsck) и -V (для генерирования более подробного вывода, содержащего сведения о фактических действиях fsck):

```
$ sudo fsck -TV /dev/sdbl Проверить файловую систему с генерированием
                          подробного вывода и без отображения номера версии
[/sbin/fsck.ext4 (1) -- /dev/sdbl] fsck.ext4 /dev/sdbl
e2fsck 1.42 (29-Nov-2011)
mypart: clean, 11/50200 files, 12002/200780 blocks
```

При обнаружении командой fsck любой проблемы вам будет задан вопрос, желаете ли вы ее устранить:

```
$ sudo fsck -TV /dev/sdbl Выводить предложение устранить обнаруженные проблемы
[/sbin/fsck.ext4 (1) -- /mnt/mymount] fsck.ext4 /dev/sdal
e2fsck 1.39 (29-Nov-2011)
Couldn't find ext2 superblock, trying backup blocks...
Resize inode not valid. Recreate<y>? y
```

Если вы не обладаете глубокими познаниями в области файловых систем, лучше ответить «да». Это можно сделать автоматически, посредством использования параметра -y:

```
$ sudo fsck -TVy /dev/sdbl
[/sbin/fsck.ext4 (1) -- /mnt/mymount] fsck.ext4 -y /dev/sdbl
e2fsck 1.42 (29 - Nov - 2011)
Couldn't find ext2 superblock, trying backup blocks...
Resize inode not valid. Recreate? yes
mypart was not cleanly unmounted, check forced.
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
Free blocks count wrong for group #0 (3552, counted=3553).
```

```
Fix? yes
Free blocks count wrong (188777, counted=188778).
Fix? yes
mypart: ***** FILE SYSTEM WAS MODIFIED *****
mypart: 11/50200 files (O.OK non-contiguous), 12002/200780 blocks
```

Создание зашифрованных файловых систем

Используя компоненты пакета cryptsetup, вы можете создавать зашифрованные файловые системы для защиты данных на жестких дисках или съемных носителях. Распространенная практика — шифрование данных в отдельном разделе /home на ноутбуке или USB-устройстве, которое вы носите с собой. Если эти устройства будут утеряны и украдены, никто не сможет получить доступ к вашим данным без фразы-пароля.

Шифрование раздела /home в вашей системе Ubuntu означает выбор соответствующих параметров во время разбивки диска на разделы при установке Ubuntu. Шифрование дискового раздела позднее с использованием командной строки потребует выполнения пошаговых действий, перечисленных ниже.

Для **шифрования USB-диска** так, чтобы только знающий фразу-пароль пользователь мог получить доступ к зашифрованным данным на нем, сделайте следующее.

1. Изучите файл /var/1 og/messages, используя команду tail. Введя ее, подключите имеющийся USB-диск к компьютеру, и вы увидите сообщения, связанные с обнаружением этого USB-диска:

```
$ sudo tail -f /var/log/messages
Apr 11 05:54:09 myubuntu kernel: [4486653.002219] sd 11:0:0:0:
    Attached scsi generic sg3 type 0
Apr 11 05:54:09 myubuntu kernel: [4486653.004329] sd 11:0:0:0:
    [sdc] 7831552 512-byte logical blocks: (4.00 GB/3.73 GiB)
Apr 11 05:54:09 myubuntu kernel: [4486653.005707] sd 11:0:0:0:
    [sdc] Write Protect is off
Apr 11 05:54:09 myubuntu kernel: [4486653.014322] sd 11:0:0:0:
    [sdc] Attached SCSI removable disk
```

2. Обратите внимание на информацию об устройстве. В этом примере USB-устройство для хранения закреплено за sdc (/dev/sdc). Это устройство объемом 4 Гбайт, не защищенное от записи. Важно точно знать имя устройства, поскольку в противном случае вы можете удалить свою систему Ubuntu или другие критически важные данные на следующем этапе!

3. Если хотите **удалить все старые данные, находящиеся на устройстве,** воспользуйтесь командой dd:

```
$ sudo dd if=/dev/zero of=/dev/sdc
dd: writing to '/dev/sdc': No space left on device
7831553+0 records in
7831552+0 records out
4009754624 bytes (4.0 GB) copied, 1074.47 s, 3.7 MB/s
```

4. **Разбейте на разделы USB-диск,** чтобы подготовить его для размещения зашифрованной файловой системы. Диск будет чистым, поэтому сначала просто сохраните на нем пустую таблицу разделов:

```
$ sudo fdisk /dev/sdc
Command (m for help): w
```

5. Создайте один раздел на USB-диске, как показано далее:

```
$ sudo fdisk /dev/sdc
Command (m for help): n
Partition type:
   p primary (0 primary. 0 extended. 4 free)
   e extended
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-78315, default 2048): <нажмите ENTER (ВВОД)>
Using default value 2048
Last sector, +sectors or +size{K,M.G} (2048-78315, default 78315):
<нажмите ENTER (ВВ0Д>
Using default value 78315
Command (m for help): w
```

6. **Установите программное обеспечение для шифрования** в своей системе Ubuntu:

```
$ sudo apt-get install cryptsetup
```

7. **Загрузите модули,** необходимые для шифрования вашего USB-диска:

```
$ sudo modprobe dm-crypt
$ sudo modprobe aes
$ sudo modprobe sha512
```

8. **Зашифруйте раздел USB-диска** с помощью команды cryptsetup. Введите YES (Да), когда получите соответствующее приглашение. Затем обязательно запомните фразу-пароль, так как в противном случае позднее вы не сможете получить доступ к данным, которые разместите на диске:

```
$ sudo cryptsetup luksFormat /dev/sdcl
WARNING!
========
This will overwrite data on /dev/sdcl irrevocably.
Are you sure? (Type uppercase yes): YES
Enter LUKS passphrase: *********
Verify passphrase: *********
```

9. **Откройте зашифрованный раздел,** как показано далее (я присвоил устройству имя mycrypt, однако вы можете указать любое на ваше усмотрение):

```
$ sudo cryptsetup luksOpen /dev/sdcl mycrypt
Enter passphrase for /dev/sdcl: *********
```

10. **Отформатируйте раздел** как любой другой дисковый раздел (в данном случае используется тип файловой системы ext4). Имя устройства нового раздела

файловой системы будет включать название, присвоенное вами в строке 1 uksOpen в каталоге /dev/mapper (/dev/mapper/mycrypt в этом примере):

```
$ sudo mkfs -t ext4 /dev/mapper/mycrypt
```

11. Теперь вы можете **монтировать файловую систему.** Затем измените владение на пользователя, которого вы хотите сделать владельцем раздела USB-диска (в данном случае это пользователь с именем **chris** и группа **chris):**

```
$ sudo mkdir /mnt/testing
$ sudo mount /dev/mapper/mycrypt /mnt/testing/
$ sudo chown chris:chris /mnt/testing
```

12. Вы можете скопировать файлы в новый раздел, чтобы убедиться в его работоспособности. Закончив, демонтируйте этот раздел. После этого вам потребуется закрыть раздел, воспользовавшись cryptsetup с параметром luksClose и именем в каталоге / dev/mapper, представляющем зашифрованный раздел:

```
$ sudo umount /dev/mapper/mycrypt
$ sudo cryptsetup luksClose \
  udi sks-1uks-uui d-9f8cb640-8a9b-4ae0-ac31-48aff59bf167 -ui dl3597
```

Теперь вы можете безопасно извлечь USB-диск. В следующий раз, когда вы подключите его при использовании любой Linux-системы, поддерживающей этот тип шифрования, происходящее далее будет зависеть от того, запущен ли у вас рабочий стол.

- При использовании рабочего стола на нем появится всплывающее окно, в котором вас попросят ввести фразу-пароль. Когда вы введете ее, нешифрованная файловая система смонтируется в каталог /media и откроется окно файлового менеджера, из которого вы получите доступ к данным в разделе. Закончив работу с данными, щелкните правой кнопкой мыши на значке, представляющем раздел на рабочем столе, и выберите **Safely Remove** (Безопасное извлечение). После этого вы сможете безопасно извлечь USB-диск.
- При использовании консоли интерпретатора команд вы получите приглашение ввести фразу-пароль, которую указали при создании шифрованного раздела. При этом нам, возможно, потребуется смонтировать файловую систему вручную (с именем вроде /dev/mapper/mycrypt), чтобы использовать ее. Закончив работу с файловой системой, демонтируйте ее (с помощью umount) и выполните команду cryptsetup luksClose для зашифрованного устройства, как было описано выше в соответствующей процедуре. После этого вы сможете безопасно извлечь диск.

Проверка дисков в RAID-массивах

Избыточные массивы независимых дисков (Redundant Array of Independent Drives, RAID) позволяют дублировать или распределять данные по нескольким жестким дискам. Использование RAID-массива может повысить надежность и производительность вашей системы хранения. Команду mdadm, которая является частью

пакета mdadm, можно использовать для **проверки устройств в программном RAID-массиве** на вашем компьютере. Взгляните на следующий пример:

```
$ sudo apt-get install mdadm
$ sudo mdadm -Q /dev/mdl
/dev/mdl: 1498.13MiB raidl 2 devices, 0 spares.
Use mdadm --detail for more detail.
/dev/mdl: No md super block found, not an md component.
```

Сообщение в последней строке попросту означает, что /dev/mdl не является членом RAID-массива. Это нормально, поскольку mdl представляет собой массив как таковой. Аналогичным образом, если вы запросите информацию о том или ином члене RAID-массива, полученный вами вывод будет выглядеть так:

```
$ sudo mdadm -Q /dev/sdb3
/dev/sdb3: is not an md array
/dev/sdb3: device 1 in 4 device active raid6 mdO.
Use mdadm --examine for more detail.
```

Для получения более подробного вывода добавьте параметр --detail:

```
$ sudo mdadm -Q-detail /dev/mdl
/dev/mdl:
        Version : 00.90.01
  Creation Time : Fri Apr 5 16:32:12 2013
     Raid Level : raidl
     Array Size : 1534080 (1498.38 Mi B 1570.90 MB)
    Device Size : 1534080 (1498.38 MiB 1570.90 MB)
   Raid Devices : 2
  Total Devices : 2
Preferred Minor : 1
    Persistence : Superblock is persistent
    Update Time : Mon Apr 15 02:06:01 2013
          State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0
           UUID : 49c564cc:2d3c9al4:d93celc9:070663ca
         Events : 0.42
    Number Maj or Minor RaidDevice State
       0       3       2        0      active sync   /dev/hda2
       1       3       66       1      active sync   /dev/hdb2
```

Команду mdadm также можно использовать для управления вашими устройствами в программном RAID-массиве. Для получения дополнительной информации введите следующее:

```
$ sudo mdadm --manage -help
$ man mdadm
```

Убедитесь, что у вас есть дисковый RAID-массив, прежде чем устанавливать пакет mdadm. При инсталляции mdadm программа установки попытается сконфи-

гурировать ваши диски, входящие в RAID-массив, как часть инсталляции. Кроме того, если вы деинсталлируете mdadm, то, вероятно, после этого в /etc/mdadm останется файл mdadm.conf. Его присутствие может привести к проблемам с совместимостью для пакета lvm2, о котором мы поговорим позднее.

Получение информации об использовании файловой системы

Израсходование свободного дискового пространства может оказаться досадным происшествием для настольной системы, однако оно потенциально способно стать катастрофой, если речь идет о серверах. Для выяснения, сколько доступно дискового пространства и какая его часть используется на данный момент, вы можете прибегнуть к команде df. Чтобы проверить, как много дискового пространства занимают определенные файлы и каталоги, используйте команду du.

Команда df предоставляет **суммарную информацию об использовании дискового пространства вашими смонтированными системами.** Задействовав параметр - h, вы можете перевести единицы измерения объема данных (который по умолчанию отображается в байтах) в мегабайты (И) и гигабайты (G), чтобы сделать вывод удобочитаемым:

$ df -h *Отобразить в удобочитаемой форме информацию о дисковом пространстве, используемом файловыми системами*

Filesystem	Size	Used	Avail	Use2	Mounted on
/dev/sda2	7.6G	3.4G	3.9G	472	/
/dev/sdal	99M	14M	80M	151	/boot
Tmpfs	501M	0	501M	02	/dev/shm
/dev/sda5	352G	197G	137G	592	/home
//thompson/chris	9204796	5722608	3007068	662	/mnt/mymount

В файловых системах ext есть ровно столько индексных дескрипторов, сколько было создано во время выполнения mkfs, поэтому если у вас будет много небольших файлов, то вы, вероятно, исчерпаете индексные дескрипторы еще до того, как израсходуете все фактически доступное пространство. Чтобы **проверить объем файлов inode,** задействуйте параметр - i:

$ df -hi

Filesystem	Inodes	Iused	Ifree	IUse2	Mounted on
/dev/sda2	2.0M	108K	1.9M	62	/

При наличии смонтированных сетевых файловых систем (например, **Samba** или **NFS)** касающиеся их сведения также будут показаны в выводе df. Чтобы в **выводе df отображалась информация только о локальных файловых системах,** введите следующее:

$ df -hi *Отобразить информацию о дисковом пространстве, занимаемом только локальными файловыми системами*

Чтобы **добавить информацию о типе файловой системы** в список, используйте параметр -T:

```
$ df -hT          Добавить информацию о типе файловой системы в список
Filesystem     Type     Size     Used     Avail     Use&     Mounted on
/dev/sda7      ext4     8.8G     5.5G      2.9G     66%      /
```

Чтобы **узнать, сколько дискового пространства занимают определенные файлы или ка талоги в файловой системе,** используйте du. Приведенная далее команда была выполнена от имени пользователя **chris:**

```
$ du -h /home/ Показать, сколько дискового пространства занимает каталог /home
du: '/home/joe': Permission denied
4.0K    /home/chris/Mai1
52K     /home/chris
64K     /home/
```

В этом выводе отобразилось, что доступ к информации о дисковом пространстве, занимаемом домашним каталогом другого пользователя (в данном случае /home/joe), был запрещен из соображений безопасности. В следующих примерах показано, как **избежать проблем с правами доступа и получить корректную суммарную информацию, прибегнув к учетной записи суперпользователя.** Это четко видно при использовании параметра -s для суммирования показателей:

```
$ du -sh /home      Обычному пользователю запрещен доступ к суммарным
                    данным о дисковом пространстве, занимаемом домашними
                    каталогами других пользователей
du: '/home/joe': Permission denied
du: '/home/horatiol99': Permission denied
64K     /home
$ sudo du -sh /home С помощью учетной записи суперпользователя вы можете
                    вывести отчет об использовании места на жестком диске
1.6G /home
```

Вы можете **указать сразу несколько каталогов** посредством параметра -c и вывести на экран информацию об их суммарном размере:

```
$ sudo du -sch /home /var Показать размер каждого каталога в отдельности
                          и их суммарный размер
1.6G /home
HIM /var
1.7G    total
```

Можно **исключить файлы, соответствующие тому или иному шаблону,** при вычислении объема занимаемого дискового пространства, воспользовавшись параметром excl ude. В приведенном далее примере файлы образов дисков (имеющие расширение .iso) не учитываются при определении суммарного пространства, занимаемого на диске:

```
$ sudo du -sh --exclude='*.iso' /home/chris Исключить файлы ISO-образов
                                            при вычислении суммарного объема
                                            пространства, занимаемого на диске
588M /home/chris
```

Вы можете **указать, на какое количество уровней следует углубиться в дереве** при суммировании показателей. Задайте для - **-max-depth** величину, большую чем приведенное здесь значение 1, чтобы углубиться на большее количество уровней при вычислении занимаемого дискового пространства:

```
$ sudo du -h ■-max-depth=l /home    Показать объем занимаемого дискового пространства. углубившись на один уровень
1.6G    /home/chris
52K     /home/joe
1.6G    /home
$ sudo du -h --max-depth=2 /home    Показать объем занимаемого дискового пространства. углубившись на два уровня
4.OK    /home/joe/Mail
52K     /home/joe
1.6G    /home
```

Программа управления логическими томами (LVM)

Программа управления логическими томами (LVM) — это инструмент, который поможет вам справиться с меняющимися потребностями в дисковом пространстве в Linux-системах. Сконфигурировав жесткие диски как тома LVM, вы получите потрясающую гибкость в увеличении, уменьшении и перемещении места для хранения по мере изменения собственных потребностей. Программа управления логическими томами также позволяет делать моментальные снимки, что является функциональностью, обычно встречающейся в дорогостоящих корпоративных сетях устройств для хранения данных (Storage-Area Network, SAN).

Ubuntu поставляется с LVM2, с помощью которой вы можете получать информацию о размещении разделов жесткого диска уже при первой установке Ubuntu. Используя LVM2, вы можете определять группы томов (vg), логических томов (lv) и физических томов (pv), а также управлять ими. Каждый логический и физический том разделяется на логические и физические составляющие соответственно.

Основной момент при использовании программы управления логическими томами заключается в том, чтобы создать группу томов и логические тома, которые вам нужны, а затем назначить экстенты (небольшие фрагменты дискового пространства) для областей, где они необходимы. В отличие от более старых схем разбивки диска на разделы, при которых нужно было делать резервную копию данных, модифицировать дисковые разделы, а затем возвращать данные в разделы измененного размера, теперь вы можете просто добавить неиспользованные экстенты туда, где они необходимы. Кроме того, если вы исчерпаете экстенты в группе томов, можете просто добавить дополнительные физические тома, чтобы стало доступно больше пространства.

Для использования программы управления логическими томами необходимо установить пакет lvm2.

Программе управления логическими томами сопутствует набор команд, которые можно задействовать для работы с томами LVM. Пошагово пройдите процедуру, описанную в следующем разделе, чтобы узнать о многих из них.

ВНИМАНИЕ —;——

Во избежание повреждения данных на жестких дисках, обеспечивающих работу вашего компьютера, во время изучения программы управления логическими томами, когда вы решите опробовать приведенные далее примеры, рекомендую использовать устройство для хранения, не содержащее критически важных данных. Я взял недорогой флеш-диск объемом 4 Гбайт (в /dev/sdc) для выполнения команд, демонстрируемых в этом разделе.

Создание томов LVM

Для начала воспользуйтесь командой fdisk, чтобы **создать физические разделы на устройстве для хранения,** на котором вы хотите создать логические разделы. Как я уже упомянул, у меня есть USB-диск объемом 4 Гбайт, который расположен в /dev/sdc, представляющем устройство:

```
$ sudo fdisk /dev/sdc          Запустить выполнение команды для управления
                               дисковыми разделами
Command (лп for help): p       Вывести на экран информацию о текущих разделах
                               (таковых нет)
Disk /dev/sdc: 4009 MB. 4009754624 bytes
84 heads, 22 sectors/track, 4237 cylinders
Units = cylinders of 1848 * 512 = 946176 bytes
Sector size ("logical/physica 1): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier:       0xdba43801 Device Boot Start End
Blocks Id System
Command (m for help): n        Создать новый раздел
Command action
e extended
p primary partition (1-4)
p                              Сделать этот раздел первичным
Partition number (1-4): 1      Этот номер присвоен первому разделу
First cylinder (2-4237. default. 2): <Enter>
Using default value 2
Last cylinder or +size or +sizeM or +sizeK
(2-4237, default 4237): <Enter>
Using default value 4237
Command (m for help): t        Задать тип раздела
Selected partition 1
Hex code (type L to list codes): 8E Использовать 8E как индикатор
                                    (раздел LVM)
Changed system type of partition 1 to 8e (Linux LVM)
Command (m for help): p        Нажмите P. чтобы увидеть информацию
                               о новом разделе
. . .
   Device Boot      Start         End      Blocks   Id  System
/dev/sdcl               1        4237     3914977   8e  Linux LVM
```

Прежде чем двигаться дальше, убедитесь, что внесли правильные изменения в соответствующий раздел! Если все будет верно, выполните запись новой таблицы разделов, как показано далее:

```
Command (m for help): w
The partition table has been altered!
Calling ioctlO to re-read partition table.
Syncing disks.
```

Вернитесь к приглашению интерпретатора команд и воспользуйтесь sfdisk, чтобы увидеть информацию о разделах диска:

```
$ sudo sfdisk -1 /dev/sdc          Показать информацию о разделах LVM
Disk /dev/sdc: 4237 cylinders, 84 heads, 22 sectors/track
Units = cylinders of 946176 bytes, blocks of 1024 bytes, count from 0
   Device Boot  Start     End   #cyls    #blocks   Id  System
/dev/sdcl           0+    4236   4237-   3914977   8e  Linux LVM
/dev/sdc2           0       -      0         0    0   Empty
/dev/sdc3           0       -      0         0    0   Empty
/dev/sdc4           0       -      0         0    0   Empty
```

Далее сделайте /dev/sdcl новым физическим разделом LVM и выполните команду pvs для просмотра информации о физических томах LVM:

```
$ sudo pvcreate /dev/sdcl          Сделать sdbl физическим томом LVM
  Physical volume "/dev/sdcl" successfully created
$ sudo pvs                         Показать информацию о физических разделах LVM
  PV                VG              Fmt Attr PSize PFree
  /dev/sdcl                         lvm2 a--  3.73g  3.73g
```

Затем воспользуйтесь vgcreate для создания группы томов vgusb и отображения текущих групп активных томов:

```
$ sudo vgcreate vgusb /dev/sdcl    Создать группу томов vgusb
  Volume group "vgusb" successfully created
$ sudo vgs                         Показать текущие группы томов
  VG      #PV #LV #SN Attr      VSize VFree
  Vgusb 100 wz--n-              3.73g 3.73g
```

Задействуйте 1 vcreate для создания нового раздела LVM размером 1 Гбайт из группы томов vgusb. Затем примените 1 vs для просмотра сведений о логическом томе, и vgs, чтобы увидеть, сколько осталось свободного дискового пространства после внесения изменений:

```
$ sudo 1 vcreate --size IG --name lvm_ul vgusb
  logical volume "lvm_ul" created
$ sudo lvs                         Показать информацию о логическом томе
 LVVG Attr LSize Pool Origin Data% Move Log Copy& Convert
lvm_ul vgusb -wi-a--- l.OOg
$ sudo vgs                         Вы увидите, что все еще свободно 2.73 Гбайт
  VG     #PV #LV #SN Attr      VSize VFree
Vgusb    1    1    0 wz--n- 3.73g 2.73g
```

Чтобы создать файловую систему ext4 в разделе LVM, воспользуйтесь командой mkfs. ext4, как показано далее:

```
$ sudo mkfs.ext4 /dev/mapper/vgusb-1vmul
mke2fs 1.41.12 (17-May-2010)
```

```
Filesystem label-
OS type: Linux
Block size=4096 (log=2)
Fragment size-4096 (log=2)
. . .
Writing superblocks and filesystem accounting information: done
This filesystem will be automatically checked every 23 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

Файловая система ext4 создана, а том LVM готов к использованию.

Использование томов LVM

Чтобы воспользоваться только что созданным новым томом, который представляет /dev/mapper/vgusb-1 vm_ul, **создайте точку монтирования (/mnt/ui) и выполните монтирование этого тома.** Затем примените df для проверки доступного пространства:

```
$ sudo mkdir /mnt/ul                          Создать точку монтирования
$ sudo mount -t ext4 /dev/mapper/vgusb-lvm_ul /mnt/ul Монтировать том
$ df -m /mnt/ul                               Проверить дисковое пространство
Filesystem              lM-blocks Used Available Use% Mounted on
      /dev/mapper/vgusb-lvm_ul
                        1008              34          924       43> /mnt/ul
```

На данном этапе файловая система содержит только каталог 1 ost+found:

```
$ Is /mnt/ul
1ost+found
```

Скопируйте файл в новую файловую систему. Например, выберите один из файлов ядра в каталоге /boot и скопируйте его в /mnt/ul:

```
$ sudo cp /boot/vmlinuz-* /mnt/ul/ Скопировать файл в /mnt/ul
$ df -m /mnt/ul                               Вы видите, что размер /mnt/ul составляет
                                              45 Мбайт
Filesystem              lM-blocks Used Available Use% Mounted on
/dev/mapper/vgusb-lvm_ul
                              1008      45          913     5% /mnt/ul
```

Выполните команду md5sum для скопированного вами файла и сохраните получившуюся контрольную сумму для использования позднее. Например:

```
$ sudo md5sum /mnt/ul/vmlinuz-*               Проверить контрольную сумму MD5
56b9ca81bdfba6e563f407caf9a52a2b /boot/vmlinuz-3.2.0-38-genenic
```

Увеличение тома LVM

Допустим, вы исчерпали дисковое пространство и хотите **добавить больше пространства в свой том LVM.** Для этого необходимо демонтировать том и воспользоваться командой 1 vresi ze (вообще для увеличения тома не требуется демонтировать его, однако здесь это сделано в качестве дополнительной меры предосторожности).

Затем вам придется проверить файловую систему с помощью e2fsck и выполнить команду resize2fs для изменения размера файловой системы ext4 для соответствующего тома:

```
$ sudo umount /mnt/ul                                   Демонтировать том
$ sudo lvresize --size 2G /dev/vgusb/lvm_ul             Изменить размер тома
  Extending logical volume lvm_ul to 2.00 GiB
  Logical volume lvm_ul successfully resized
$ sudo e2fsck -f /dev/vgusb/l vmul
e2fsck 1.41.12 (17-May-2010)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/vgusb/lvm_ul: 14/65536 files (0.0% non-contiguous),
15546/262144 blocks$ sudo resize2fs /dev/vgusb/lvm_ul 2G Изменить размер lvm_ul
resize2fs 1.41.12 (17-May-2010)
Resizing the filesystem on /dev/vgusb/lvm_ul to 524288 (4k) blocks.
The filesystem on /dev/vgusb/lvm_ul is now 524288 blocks long.
```

В приведенном выше примере изменяется размер как тома, так и файловой системы, становясь равным 2 Гбайт. Далее снова выполните монтирование и проверьте дисковое пространство, а также контрольную сумму MD5, сгенерированную вами ранее:

```
$ sudo mount -t ext4 /dev/mapper/vgusb-1 vmul /mnt/ul Снова смонтировать том
$ df -m /mnt/ul Вы увидите, что использовано 45 Мбайт из 2016 Мбайт
Filesystem           lM-blocks Used Available Use* Mounted on
/dev/mapper/vgusb-1vm_ul
                          2016   45      1869   3* /mnt/ul
$ sudo md5sum /mnt/ul/vmlinuz-*          Перепроверить контрольную сумму MD5
56b9ca81bdfba6e563f407caf9a52a2b /boot/vmlinuz-3.2.0-38-generic
```

Заново смонтированный том теперь имеет размер 2 Гбайт, а не 1 Гбайт.

Уменьшение тома LVM

Вы также можете использовать lvresize, если захотите **изъять ненужное пространство из существующего тома LVM.** Как и ранее, демонтируйте том, прежде чем изменять его размер, и выполните команды e2fsck (для проверки файловой системы) и resize2fs (чтобы сделать ее меньше по размеру):

```
$ sudo umount /mnt/ul
$ sudo e2fsck -f /dev/vgusb/lvm ul
e2fsck 1.41.12 (17-May-2010)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
...
/dev/vgusb/lvm_ul: 14/131072 files (0.0* non-contiguous),
  19723/524288 blocks
```

```
$ sudo resize2fs /dev/vgusb/lvmul IG          Изменить размер файловой системы
resize2fs 1.41.12 (17-May-2010)
Resizing the filesystem on /dev/vgusb/lvm_ul to 262144 (4k) blocks.
The filesystem on /dev/vgusb/lvm_ul is now 262144 blocks long.
$ sudo lvresize --size IG /dev/vgusb/lvm ul
  WARNING: Reducing active logical volume to 1.00 GiB
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce lvm_ul? [y/n]: y
  Reducing logical volume lvm_ul to 1.00 GI B
  Logical volume lvm_ul successfully resized
$ sudo mount -t ext4 /dev/mapper/vgusb-lvm_ul Imtlul Снова смонтировать том
$ df -m /mnt/ul                  Вы видите, что использовано 4 Мбайт из 12 Мбайт
Filesystem           IM-blocks Used Available Use% Mounted on
/dev/mapper/vgusb-1vm_ul
                        1008          45          913          5% /mnt/ul
```

Заново смонтированный том теперь имеет размер 1008 Мбайт, а не 2016 Мбайт.

Удаление логических томов LVM и соответствующих групп

Чтобы **удалить логический том LVM из группы томов,** демонтируйте его, а затем выполните команду 1 vremove, как показано далее:

```
$ sudo umount /dev/vgusb/l vm ul
$ sudo 1 vremove /dev/vgusb/l vm ul
Do you really want to remove active logical volume "lvm_ul"? [y/n]: y
  Logical volume "lvm_ul" successfully removed
```

Чтобы удалить существующую группу томов LVM, используйте команду vgremove:

```
$ sudo vgremove vgusb
  Volume group "vgusb" successfully removed
```

Есть множество других способов работы с программой управления логическими томами. Дополнительную информацию вы сможете найти в практическом руководстве LVM HOWTO (http://tldp.org/HOWTO/LVM-HOWTO/).

Резюме

Создание и управление файловыми системами в Linux — одни из самых важных задач системного администрирования Linux. Ubuntu поддерживает большое количество стандартных типов файловых систем Linux (ext2, ext3, ext4, xfs, reiserfs и др.). Кроме того, Ubuntu позволяет создавать файловые системы Windows (VFAT, NTFS и is д.), а также унаследованные и специализированные файловые системы Linux и UNIX (например, minix, jfs и xfs) и управлять ими.

Вы можете разбивать на разделы жесткие диски с помощью таких команд, как fdisk и parted. К инструментам для работы с файловыми системами относятся те, которые позволяют создавать файловые системы (mkfs), просматривать и изменять атрибуты файловых систем (tuneffs и dumpe2fs), монтировать/демонтировать файловые системы (mount и umount) и осуществлять проверку на предмет проблем (badblocks и fsck). Чтобы узнать, сколько пространства используется файловыми системами, задействуйте команды df и du.

8 Резервное копирование и съемные носители

В этой главе:

- создание резервных архивов с помощью команды tar;
- сжатие резервных копий посредством команд gzi p, bzip2 и Izop;
- резервное копирование по сети с использованием ssh;
- выполнение резервного копирования по сети с помощью команды rsync;
- создание резервных ISO-образов посредством mkisofs;
- запись резервных образов на CD или DVD с использованием команд cdrecord и growisofs.

Резервное копирование данных в Linux первоначально осуществлялось путем выполнения команд для архивации и сжатия файлов, подлежащих ему, с последующей записью резервного архива на ленту. Ассортимент инструментов архивации, методик сжатия и носителей резервных копий чрезвычайно расширился за последние годы. Для многих пользователей на смену архивированию с использованием ленты пришли методики резервного копирования данных по сети на другие жесткие диски, CD, DVD и иные недорогие съемные носители.

В этой главе подробно рассматриваются некоторые полезные инструменты для резервного копирования и восстановления ваших критически важных данных. В первой части текущей главы приведены детальные инструкции по использованию для резервного копирования таких базовых инструментов, как tar, gzip и rsync.

Резервное копирование данных с размещением их в сжатых архивах

Если у вас есть опыт работы в Windows, то, возможно, вам доводилось использовать инструменты типа WinZip и PKZIP, которые позволяют архивировать и сжимать группы файлов в рамках одного приложения. Linux предлагает отдельные инструменты для сбора групп файлов в один архив (например, tar) и сжатия этого архива с целью эффективного хранения (gzip, bzip2 и lzop). При этом вы также можете выполнять эти две операции вместе, используя команду tar с дополнительными параметрами.

Создание резервных архивов с помощью tar

Команда tar, имя которой является сокращением от tape archiver *(ленточный архиватор),* берет начало в ранних UNIX-системах. Хотя магнитная лента была распространенным носителем, на который первоначально записывались данные с использованием tar, сегодня эта команда чаще всего применяется для создания архивных фай лов, которые можно разместить на различных носителях.

Команда ta *г* обладает широкими возможностями, что находит отражение в множестве параметров, доступных для использования с ней. Базовые операции, выполняемые посредством tar, — это создание резервных архивов (-c), извлечение файлов из архивов (- x), выявление различий между архивами путем их сравнения (-d) и обновление файлов в архивах (-и). Вы также можете добавлять файлы в существующие архивы (- г или -A) или удалять файлы из них (-d) и выводить на экран содержимое архивов (t).

ПРИМЕЧАНИЕ----------

Несмотря на то что команда tar доступна почти во всех UNIX- и Linux-системах, она работает поразному во многих из них. Например, Solaris не поддерживает использование -z для управления TAR-архивами, сжатыми в формате gzip. Команда Star (ess-tar) поддерживает списки контроля доступа (Access Control List, ACL) и флаги файлов (для расширенных прав доступа, используемых в Samba). В Red Hat Enterprise Linux tar позволяет осуществлять резервное копирование атрибутов файловой системы SELinux.

Вы можете добавлять параметры, обеспечивающие сжатие итогового архива, как часть процесса создания TAR-архива. Например, используйте - j, чтобы сжать архив в формат е bzip2, либо -*г* для сжатия в формате gzip. По соглашению обычные TAR-файлы имеют расширение .tar, а сжатые — расширение ,tar.bz2 (сжатые с помощью bzip2) либо .tar.gz (сжатые посредством gzip). Если вы произведете сжатие файла вручную с использованием Izop (www.lzop.org), то сжатый TAR-файл должен будет иметь расширение .tar.Izo.

Помимо использования для резервного копирования, TAR-файлы являются популярным средством распространения исходного кода и двоичных файлов программных проектов, так как в любой Linux- и UNIX-подобной системе имеются инструменты для работы с TAR-файлами.

ПРИМЕЧАНИЕ ----------

Одна из особенностей работы с tar обусловлена тем, что эта команда была создала до появления стандартов, касающихся ввода параметров. Хотя у вас есть возможность задействовать в качестве префикса для tar параметры с использованием тире, это не всегда обязательно, поэтому вы можете встретить команду, начинающуюся с tar xvf без тире для индикации параметров.

В качестве классического примера использования команды tar можно привести комбинирование устаревших параметров и каналов для сжатия вывода:

```
$ tar c *.txt | gzip -c > myfiles.tar.gz
```

Создать архив, окать его и вывести

В данном гримере показан двухэтапный процесс, описание которого вы можете найти в документации к старым UNIX-системам. Команда tar создает (c) архив

из всех файлов с расширением .txt в текущем каталоге. Затем вывод передается по каналу команде gzi p и отправляется на stdout (-c), после чего перенаправляется в файл myfiles.tar.gz. Следует отметить, что tar является одной из немногих команд, которые не требуют указания тире (-) перед параметрами.

Новые версии tar в современных Linux-системах позволяют **создавать архивы и сжимать вывод** в один шаг:

```
$ tar czf myfiles.tar.gz *.txt Создать TAR-файл, сжатый с помощью gzip,
                               из файлов с расширением .txt
$ tar czvf myfiles.tar.gz *.txt Сгенерировать более подробный вывод
                                при создании архива
textfi1 el.txt
textfile2.txt
```

В этом примере нужно отметить, что название нового архива (myfiles.tar.gz) должно следовать непосредственно за параметром f команды tar (который является индикатором имени этого архива). В противном случае вывод tar будет отправлен на stdout (другими словами, на ваш экран). Параметр z говорит о необходимости применения gzip-сжатия, а v обеспечивает генерирование подробного описания процесса обработки.

Вернуть файлы в файловую систему (с распаковкой и извлечением из TAR-файла) вы можете либо в один, либо в два этапа, используя команду tar и (не обязательно) команду gunzi p:

```
$ gunzip -c myfiles.tar.gz | tar x    Распаковывает архив и извлекает
                                      содержимое TAR-файла
```

Как вариант попробуйте воспользоваться приведенной далее строкой команд:

```
$ gunzip myfiles.tar.gz ; tar xf myfiles.tar    Распаковывает архив, а затем
                                                извлекает содержимое TAR-файла
```

Чтобы выполнить аналогичную процедуру в один шаг, можете прибегнуть к следующей команде:

```
$ tar xzvf myfiles.tar.gz
textfilel.txt
textfile2.txt
```

В результате выполнения приведенных выше команд находящиеся в архиве файлы с расширением txt будут скопированы из него в текущий каталог. Параметр x обеспечит извлечение файлов, z — распаковку (разархивирование) файлов, v — генерирование вывода, а f будет выступать в качестве индикатора, что следующий параметр является именем архивного файла (myfi les.tar.gz).

Инструменты сжатия

Сжатие — важный аспект работы с резервными файлами. Для хранения сжатых файлов требуется меньше дискового пространства на носителях резервных копий (CD, DVD, лентах и т. д.) или сервере. На перенос архивов на другие носители либо загрузку файлов по сети также требуется меньше времени.

Сжатие способно сэкономить дисковое пространство и сократить время переноса данных на другие носители, однако оно может значительно увеличить нагрузку на центральный процессор компьютера. Как вариант вы можете рассмотреть использование аппаратного сжатия на ленточном носителе (www.amanda.org/docs/faq.html#id346016).

В примерах, приведенных в предыдущем разделе, tar вызывает команду gzip. Однако tar может работать с многими инструментами сжатия. Стандартно в Ubuntu tar будет работать с gzip и bzip2. Третьей утилитой сжатия, которую вы можете добавить в инструментарий, является команда 1 zop, которую можно использовать по-другому в сочетании с tar. Вот порядок расположения этих инструментов, начиная с обеспечивающего самое быстрое, но самое слабое сжатие и заканчивая дающим в результате самое медленное, но самое сильное: lzop, gzip и bzip2.

Архивирование и сжатие больших объемов данных может занять продолжительное время, поэтому вам следует знать, что при использовании bzip2 на сжатие данных может потребоваться примерно в десять раз больше времени, чем при применении инструмента lzop, который обеспечивает только в два раза лучшее сжатие. Однако при выполнении любой команды для сжатия вы можете выбирать разные его уровни, чтобы найти баланс между необходимостью в более сильном сжатии и временем, которое на него потребуется.

Для использования команды tar в сочетании с **Ьг1р2-сжатием** задействуйте параметр -j:

```
$ tar cjvf myfiles.tar.bz2 *.txt
```
Создать архив, сжатый посредством bzip2

Вы также можете **распаковать (-j) сжатый посредством bzip2 файл** при извлечении файлов (-x) с использованием команды tar:

```
$ tar xjvf myfiles.tar.bz2
```
Извлечь файлы с распаковкой архива, сжатого посредством bzip2

Утилита сжатия lzop немного менее интегрирована с tar. Прежде чем использовать ее, вам может потребоваться установить пакет lzop. Чтобы выполнить сжатие с помощью lzop, вам также понадобится параметр --use-compress-program:

```
$ sudo apt-get install lzop
$ tar --use-compress-progranHzop -cf myfiles.tar.lzo *.txt
$ tar --use-compress-progranHzop -xf myfiles.tar.lzo
```

В строках команд в приведенных выше примерах реверсируется старый синтаксис tar с переключателем перед командой. При обычном использовании и в других примерах я задействую современный синтаксис tar без переключателя.

ПРИМЕЧАНИЕ

Вы можете столкнуться со сжатыми файлами, имеющими расширение .rar и, соответственно, формат RAR. Этот формат — проприетарный, в силу чего для работы с ним нет никаких широко распространенных инструментов сжатия. В Ubuntu вы можете установить пакеты unrar и rar, чтобы получить в свое распоряжение команды для работы с файлами формата RAR.

gzip

Как уже отмечалось ранее, можно **отдельно использовать любую команду, обеспечивающую сжатие** (в отличие от строки команды tar). Далее приведены примеры использования gzip для создания и работы с файлами, сжатыми посредством gzip:

```
$ gzip myfile Сжимает myfile с помощью gzip и переименовывает его в myfile.gz
```

Следующая команда дает аналогичный результат наряду с подробным выводом:

```
S gzip -v myfile Сжимает myfile посредством gzip и генерирует подробный вывод
myfile; 86.0% -- replaced with myfile.gz
$ gzip -tv myfile.gz Проверяет целостность файла, сжатого с помощью gzip
myfile.gz:          OK
$ gzip -lv myfile.gz Отображает подробную информацию о файле, сжатом
                     посредством gzip
method crc date time compressed uncompressed ratio uc_name
defla Of27d9e4 Jul 10 04:48          46785          334045          86.0% myfile
```

Используйте любую из следующих команд для **сжатия всех файлов в каталоге:**

```
$ gzip -rv mydir Сжать все файлы в каталоге
mydir/filel: 39.1% -- replaced with mydir/filel.gz
mydir/fi1e2: 39.5% -- replaced with mydir/file2.gz
$ gzip -1 myfile          Самое быстрое, но самое слабое сжатие
S gzip -9 myfile          Самое медленное, но самое сильное сжатие
```

Добавьте тире перед цифрой в диапазоне от 1 до 9, чтобы задать уровень сжатия. Как показано в примере, -1 обеспечивает самое быстрое (и самое слабое) сжатие, а -9 дает самое медленное (но самое сильное). Значением уровня сжатия по умолчанию для gzip является 6. Команда lzop предусматривает меньше уровней сжатия: 1,3 (по умолчанию), 7, 8 и 9. Уровни сжатия в случае с bzip2 работают по-другому.

Чтобы **распаковать файл, сжатый с помощью gzip,** вы можете прибегнуть к команде gunzi p. Используйте любой из следующих примеров:

```
$ gunzip -v myfile.gz Распаковывает myfile.gz и переименовывает его в myfile
myfile.gz:          86.0% -- replaced with myfile
$ gzip -dv myfile.gz Делает то же. что и предыдущая строка команды
```

Хотя данные примеры связаны со сжатием обычных файлов, аналогичные параметры можно использовать для сжатия TAR-архивов.

bzip2

Команда bzip2 считается средством, обеспечивающим самое сильное сжатие среди инструментов, описываемых в этой главе. Вот несколько примеров ее использования:

```
$ bzip2 myfile          Сжимает файл и переименовывает его в myfile.bz2
$ bzip2 -v myfile       Делает то же. что и предыдущая команда,
                         но с генерированием более подробного вывода
```

```
  myfile: 9.529:1, 0.840 bits/byte. 89.51* saved, 334045 in, 35056 out.
$ bunzip2 myfile.bz2          Распаковывает файл и переименовывает его в myfile
$ bzip2 -d myfile.bz2         Делает то же. что и предыдущая команда
$ bunzip2 -v myfile.bz2       Делает то же, что и предыдущая команда,
                               но с генерированием более подробного вывода
  myfile.bz2: done
```

lzop

Команда lzop работает иначе, чем gzip и bzip2. Она является самым подходящим инструментом в ситуациях, в которых скорость сжатия важнее, чем итоговая степень сжатия. При сжатой содержимого файла команда lzop оставляет оригинальный файл нетронутым (если только вы не задействуете -U) и создает новый файл с расширением . 1 zo. Вы можете воспользоваться любым из следующих примеров использования команды 1 zop **для сжатия файла с именем myfile:**

```
$ lzop -v myfile Оставить myfile. создать сжатый myfile. Izo
compressing myfile into myfile.Izo
$ lzop -U myfile Удалить myfile, создать сжатый myfile. Izo
```

Создав myfi 1 e. 1 zo, выберите *любую* из приведенных далее команд для проверки, отображения информации или сжатия этого файла:

```
$ lzop -t myfile.Izo          Проверить целостность сжатого файла
$ lzop  --info myfile.Izo     Показать  внутренний заголовок для каждого  файла
$ lzop  -1 myfile.Izo         Показать  информациюо сжатии по каждому файлу
Method compressed uncompr. ratio uncompressed_name
LZ01X-1      59008       99468   59.31 myfile
$ lzop  --Is myfile.Izo       Показать  содержимое сжатого файла подобно  Is -1
$ cat myfile | lzop > x.lzo   Сжать соответствующий файл и отправить на  stdout
$ lzop -dv myfile.Izo Оставить myfile. Izo, создать сжатый myfile
```

В отличие от gzip и bzip2, у lzop нет соответствующей команды для распаковки файлов, сжатых с ее помощью. Всегда используйте команду lzop с параметром -d для распаковки файлов. Если передать ей список имен файлов и каталогов, то команда lzop выполнит сжатие всех файлов, но проигнорирует каталоги. Оригинальное файловое имя, режимы прав доступа и временные метки будут у сжатого файла такие же, как у оригинального.

Просмотр, объединение и добавление файлов в TAR-архивы

До сих пор вы, используя tar, лишь создавали и распаковывали архивы. Однако также есть параметры для просмотра содержимого архивов, объединения архивов, добавления файлов в существующие архивы и удаления файлов из архивов.

Для **просмотра содержимого архива** используйте параметр -t:

```
$ tar tvf myfiles.tar Показать файлы из несжатого архива
-rw-r--r-- rooz/root 9584 2007-07-05 11:20:33 textfilel.txt
-rw-r--r-- root/root 9584 2007-07-09 10:23:44 textfile2.txt
$ tar tzvf myfiles.tgz        Показать файлы из архива, сжатого с помощью gzip
```

Если архив будет представлять собой TAR-архив, который сжат посредством 1 zop и называется myfi 1 e. tar Л zo, вы сможете **просмотреть содержимое этого tar/lzop-файла** следующим образом:

```
$ tar --use-compress-progranHzop -tf myfiles.tar.lzo Показать содержимое LZO-архива
```

Чтобы **конкатенировать один TAR-файл с другим,** используйте параметр -A. Следующая команда добавляет содержимое archive2.tar в архив archive!..tar:

```
$ tar -Af archivel.tar archive2.tar
```

Задействуйте параметр -г, чтобы **добавить один или более файлов в существующий архив. В** приведенном далее примере myfile добавляется в архивный файл archive.tar:

```
$ tar rvf archive.tar myfile Добавить файл в TAR-архив
```

Вы можете использовать подстановочные символы для **выбора нескольких соответствующих файлов** для добавления в свой архив:

```
$ tar rvf archive.tar *.txt Добавить несколько файлов в TAR-архив
```

Удаление файлов из TAR-архивов

Если на жестком диске есть архивный TAR-файл, вы можете удалить файлы из этого архива. Следует отметить, что данной методикой нельзя воспользоваться для удаления файлов из TAR-архивов, записанных на магнитную ленту. Вот пример **удаления файлов из TAR-архива:**

```
$ tar --delete filel.txt -f myfile.tar Удалить filel.txt из my file.tar
```

Резервное копирование по сети

Выполнив резервное копирование своих файлов и собрав их в один TAR-архив, что вы будете делать с этим архивом? Резервное копирование данных необходимо в первую очередь на случай, если что-то произойдет (например, сломается жесткий диск) и вам потребуется восстановить файлы из резервной копии. К методам, которыми вы можете воспользоваться для того, чтобы сохранить резервные копии, относятся:

О **копирование резервных копий на съемные носители,** например ленту, CD или DVD (будет описано позднее в этой главе);

О копирование их на другой компьютер по сети.

Быстрые и надежные сети, недорогие жесткие диски большой емкости и сохранность, которая обеспечивается благодаря размещению копий ваших данных за пределами накопителя компьютера, сделали резервное копирование по сети популярной практикой. Для человека, выполняющего резервное копирование личных данных, или для небольшого офиса комбинация из нескольких простых команд может оказаться всем, что потребуется для создания эффективных и защищенных резервных копий. Этот подход является прямым применением философии UNIX,

суть которой заключается в объединении простых программ, делающих что-то одно, для выполнения более сложных действий.

Хотя для переноса ваших резервных данных на удаленный компьютер можно использовать почти любую команду, позволяющую копировать файлы по сети, некоторые утилиты особенно подходят для этой операции. Используя такие OpenSSH-инструменты, как ssh и scp, вы можете наладить безопасную передачу резервных архивов без использования паролей, а также их зашифрованную передачу.

Инструменты наподобие команды rsync позволяют сэкономить ресурсы путем резервного копирования только тех файлов (или их частей), которые изменились со времени предыдущего резервного копирования. Используя такие инструменты, как unison, вы можете выполнять резервное копирование файлов по сети из систем Windows и Linux.

В следующих разделах этой главы описываются методики резервного копирования данных на другие компьютеры по сети.

ПРИМЕЧАНИЕ----------

Похожий инструмент, который может заинтересовать вас, — это команда rsnapshot (www.rsnapshot.org). Она способна работать вместе с rsync для создания конфигурируемых ежечасных, ежедневных или ежемесячных моментальных снимков файловой системы. Эта команда использует жесткие ссылки для сохранения моментального снимка файловой системы, который затем сможет синхронизировать с измененными файлами.

Установите этот инструмент, прибегнув к следующим командам:

```
$ sudo apt-get install rsnapshot
$ sudo apt-get install sshfs
```

Резервное копирование TAR-архивов с использованием ssh

OpenSSH **(wvyw.openssh.org)** обеспечивает инструменты для безопасного удаленного входа в систему, удаленного выполнения и удаленного копирования файлов посредством сетевых интерфейсов. Настроив два компьютера на совместное использование ключей шифрования, вы сможете передавать файлы между ними без необходимости каждый раз вводить пароли. Благодаря этому у вас есть возможность создавать сценарии для резервного копирования своих данных из SSH-клиента на SSH-сервер без операции вручную.

Работая в центральной Linux-системе, вы можете **собирать резервные копии с множества клиентских компьютеров** с помощью OpenSSH-команд. В приведенном далее примере на удаленном узле выполняется команда tar (для архивирования и сжатия файлов), осуществляется передача по каналу потока tar на stdout, а также используется команда ssh для перехвата резервной копии локально (посредством SSH) с применением tar:

```
$ mkdir mybackup : cd mybackup
$ ssh chris@serverl 'tar cf - myfile*' | tar xvf -
chris@serverl s password: ******
myfi lei
myfile2
```

В данном примере все файлы, начинающиеся с myfi 1 e, копируются из домашнего каталога пользователя **chris** на **serverl** и размещаются **в** текущем каталоге. Обратите внимание, что на левом конце канала создается архив, а на правом из архива извлекаются файлы в текущий каталог (имейте в виду, что ssh перезапишет локальные файлы при наличии таковых, в силу чего в данном примере был создан пустой каталог).

Чтобы реверсировать процесс и **скопировать файлы из локальной системы в удаленную,** вы можете сначала локально выполнить команду tar. Однако в этом примере была добавлена команда cd для размещения файлов в каталоге /home/chris/myfolder на удаленном компьютере:

```
$ tar cf - myfile* | ssh chris@serverl \
        'cd /home/chris/myfolder; tar xvf  - '
chris@serverl's password: ******
myfi Tel
myfiie2
```

В приведенном далее примере мы не собираемся извлекать файлы из TAR-архива на принимающей стороне, а вместо этого **запишем результаты в tgz-файлы:**

```
$ ssh chris@serverl 'tar czf - myfile*' | cat > myfiles.tgz
$ tar cvzf - myfile* | ssh chris@serverl 'cat > myfiles.tgz'
```

В первом примере берутся все файлы, начинающиеся с myfile, из домашнего каталога пользователя **chris** на **serverl** и сжимаются, после чего эти сжатые файлы направляются в файл myfi 1 es. tgz в локальной системе. Во втором примере делается обратное, для чего берутся все файлы, начинающиеся с myfile, из локального каталога и отправляются в файл myfi 1 es. tgz в удаленной системе.

Данные примеры подходят для копирования файлов по сети. Помимо обеспечения сжатия, они также позволяют использовать любые параметры tar на ваш выбор, например связанные с инкрементным резервным копированием.

Резервное копирование файлов с помощью rsync

Более функционально богатой командой для резервного копирования является rsync. Столь уникальной эту команду делает ее алгоритм, который сравнивает локальные и удаленные файлы по одному небольшому блоку за раз с использованием контрольных сумм и передает только отличающиеся блоки. Этот алгоритм настолько эффективен, что повторно используется во многих продуктах, применяемых для резервного копирования.

Команда rsync может работать либо в сочетании с удаленным интерпретатором команд (ssh), либо путем запуска демона rsyncd на стороне сервера. В приведенном далее примере rsync используется посредством ssh для **зеркалирования каталога:**

```
$ rsync -avz --delete chris@serverl:/home/chris/pics/ chrispics/
```

Приведенная выше команда предназначена для зеркалирования структуры удаленного каталога (/home/chris/pics/) в локальную систему. Параметр -a обеспечивает выполнение в режиме архивации (с рекурсивным копированием всех

файлов из удаленного каталога), параметр *-z* приводит к сжатию файлов, а благодаря -v генерируется подробный вывод. Параметр - -delete дает rsync указание удалить все файлы в локальной системе, которых больше нет в удаленной системе.

Для обеспечения непрерывного резервного копирования вы можете сделать так, чтобы команда rsync выполняла инкрементное резервное копирование семь дней в неделю. Вот пример:

```
# mkdir /var/backups
# rsync -delete -backup                                  \
    --backup-dir=/var/backups/backup-'date +ЯА' \
    -avz chrisi?serverl:/home/chris/Personal/       \
    /var/backups/current-backup/
```

При выполнении приведенной выше команды все файлы, расположенные в /home/chris/Personal на serverl удаленной системы, будут скопированы в локальный каталог /var/backups/current-backup. Все файлы, модифицированные сегодня, будут скопированы в каталог с именем, которое соответствует текущему дню недели, например /var/backups/backup-Monday. По прошествии недели будет создано семь каталогов, отражающих изменения за каждый из последних семи дней.

Другой хитростью для выполнения циклического резервного копирования является **использование вместо многочисленных копий файлов жестких ссылок.** Этот двухэтапный процесс предполагает ротацию файлов с последующим выполнением rsync:

```
# rm -rf /var/backups/backup-old/
# mv /var/backups/backup-current/ /var/backups/backup-old/
# rsync --delete --link-dest=/var/backups/backup-old -avz \
    chri s@serverl:/home/chri s/Personal/ /var/backups/backup-current/
```

В приведенной выше процедуре существующий каталог backup-current заменяет содержимое каталога backup-old, при этом удаляется полная резервная копия двухнедельной давности, а на замену ей приходит такая же, но за последнюю неделю. При создании новой полной резервной копии посредством rsync с параметром --link-dest, если окажется, что какие-либо файлы, подвергаемые резервному копированию из удаленного каталога Personal на serverl, присутствовали во время предыдущего резервного копирования (а теперь располагаются в backup-old), будет создана жесткая ссылка, связывающая файлы в каталогах backup-current и backup-old.

Вы можете сэкономить много пространства благодаря наличию жестких ссылок, связывающих файлы в ваших каталогах backup-old и backup-current. Например, если бы у вас в обоих каталогах имелся файл с именем filel.txt, то вы могли бы убедиться, что оба этих файла являются одним и тем же физическим файлом, отобразив их индексные дескрипторы, как показано далее:

```
$ Is -i /var/backups/backup*/filel.txt
260761 /var/backups/backup-current/filel.txt
260761 /var/backups/backup-old/filel.txt
```

Резервное копирование с помощью unison

Команда rsync подходит для резервного копирования данных с одного компьютера на другой, однако она предполагает, что компьютер, на котором расположены подвергаемые резервному копированию данные, — единственный, на котором эти данные модифицируются. Однако что, если у вас два компьютера, на которых модифицируется один и тот же файл, и вы захотите синхронизировать эти изменяемые файлы? Инструмент, который позволит вам сделать это., называется unison.

Людям часто требуется работать с одними и теми же документами на своих ноутбуках и настольных системах. Эти компьютеры даже могут функционировать под управлением разных операционных систем, unison является кросс-платформенным приложением, поэтому позволяет **синхронизировать файлы,** расположенные как в Linux-, так и Windows-системах. Чтобы использовать unison в Ubuntu, вам потребуется установить одноименный пакет (для этого введите команду sudo apt-get instal 1 unison). В обеих синхронизируемых вами системах должна использоваться одна и та же версия unison.

Задействуя unison, вы можете определить двоих суперпользователей, которые представляют два пути для синхронизации. Эти суперпользователи могут быть локальными или удаленными. Например:

```
$ unison /home/chris ssh://chris@serverl//home/cnegus
$ unison /home/chris /mnt/backups/chris-homedir
```

unison включает как графический инструмент, так и инструмент командной строки для резервного копирования посредством unison. По умолчанию используется графический инструмент. Однако попытка задействовать его может потерпеть неудачу, если у вас не запущен рабочий стол или вы решите выполнить unison из screen. Чтобы **форсировать выполнение unison в режиме командной строки,** добавьте параметр -ui text, как показано далее:

```
$ unison /home/chris ssh://chris@serverl//home/cnegus -ui text
Contacting server...
chris@serverl's password:
Looking for changes
    Waiting for changes from server
Reconciling changes
local                serverl
newfile ------> memo.txt [f] y
propagating updates
    . . .
```

После этого утилита unison сравнит двоих суперпользователей и по каждому изменению, которое имело место с прошлого раза, спросит вас, какие действия вы хотите выполнить. В приведенном выше примере показан новый файл memo. txt в локальной системе. Вам задается вопрос, желаете ли вы приступить к обновлению (в данном случае файл memo.txt будет скопирован с локального компьютера на **serverl).** Нажмите клавишу **Y,** чтобы произвести обновление.

Если вы доверяете **unison,** добавьте -auto, чтобы утилита **выполняла действия по умолчанию, не отображая при этом для вас приглашение ввести подтверждение:**

$ unison /home/chris ssh://chris@serverl//home/cnegus -auto

Для получения дополнительной информации загляните на MAN-страницу, посвященную команде unison. Кроме того, вы можете просмотреть параметры unison, воспользовавшись параметром -help. Вы также можете вывести на экран и постранично просмотреть руководство по этой команде, применив параметр -doc all, как показано далее:

$ unison -help *Показать параметры unison*
$ unison -doc all | less *Отобразить руководство по unison*

Если вы будете часто осуществлять синхронизацию двоих суперпользователей, то можете **создать профиль,** представляющий собой набор предварительных настроек. В графическом режиме экран по умолчанию предполагает создание профилей. Профили хранятся в текстовых файлах .prf в каталоге -/.unison/. Они могут быть вот такими простыми:

root = /home/chris
root = ssh://chris@serverl//home/cnegus

Если сохранить эти данные в профиле с именем cn-home.prf, то вы сможете легко вызвать его, воспользовавшись следующей строкой команды:

$ unison cn-home

Резервное копирование на съемные носители

Емкость CD и DVD, а также низкая стоимость этих накопителей сделали их привлекательным и в качестве носителей резервных копий данных с компьютеров. Используя инструменты, которые обычно входят в состав Linux-систем, вы можете собирать файлы, подлежащие резервному копированию, в образы CD или DVD и записывать эти образы на соответствующие носители.

Такие инструменты командной строки, как mkisofs (для создания образов CD) и cdrecord (для записи образов на CD или DVD), когда-то обеспечивали наиболее популярные интерфейсы для резервного копирования на CD и DVD. Сегодня существует множество графических внешних интерфейсов для инструментов такого типа, использование которых вы можете рассмотреть как вариант. Например, к GUI-инструментам для мастеринга и записи CD/DVD относятся K3b (оптимизированное для KDE приложение, позволяющее записывать CD и DVD) и Nautilus (файловый менеджер для GNOME с возможностью записи CD). К GUI-инструментам, используемым для записи CD, можно отнести gcombust, X-CD-Roast и graveman.

В этом разделе описываются команды для создания образов файловых систем с целью их последующего резервного копирования на CD или DVD, а также команды для записи этих образов.

Создание резервных образов с помощью mkisofs

Доступ к содержимому большинства CD и DVD с данными можно получить как в Windows-, так и в Linux-системах, поскольку они создаются с использованием стандарта ISO9660, применяемого для форматирования данных на этих дисках. Большинству современных операционных систем приходится сохранять больше информации о файлах и каталогах, чем предусматривает базовый стандарт ISO9660, поэтому с целью включения соответствующей информации были добавлены расширения для этого стандарта.

Используя команду mkisofs, вы можете выполнять резервное копирование структур файлов и каталогов из любой точки в файловой системе Linux и создавать образы 1SO9660. Эти образы могут содержать расширения следующих типов.

- Расширения **System Use Sharing Protocol** (SUSP) (протокол совместного использования систем) являются записями, идентифицируемыми в Rock Ridge interchange Protocol (протокол обмена Rock Ridge). SUSP-записи могут включать атрибуты в стиле UNIX, например владение, длинные имена файлов и специальные файлы (такие как файлы символьных устройств и символьные ссылки).
- В записях каталогов **Joliet** хранятся более длинные имена файлов в форме, которая делает их пригодными для использования в Windows-системах.
- Расширения **Hierarchical File System** (HFS) (иерархическая файловая система) позволяют ISO-образу иметь вид файловой системы HFS, нативной для компьютеров Macintosh. Кроме того, можно разными способами добавить ветви данных и ресурсов для чтения на компьютерах Macintosh.

Если вы решите создать собственный ISO-образ, обдумайте, где вам в конечном счете потребуется получать доступ к файлам, резервное копирование которых вы выполните с использованием mkisofs (в Linux, Windows или Mac). Создав образ, вы сможете использовать его различными путями, самый очевидный из которых — его запись на CD или DVD.

Помимо того, что команда mkisofs применима для создания целой файловой системы Linux или ее частей с целью использования на портативных носителях, ее также можно задействовать для создания Live CD/DVD. Для этого она добавляет в образ зафузочную информацию, с помощью которой можно запустить ядро Linux или другую операционную систему без использования жесткого диска компьютера.

ПРИМЕЧАНИЕ

Хотя вы по-прежнему можете использовать команду mkisofs в Ubuntu, теперь она представляет собой указатель на genisoimage. Команда genisoimage является производной от mkisofs, которая была частью пакета cdrtools (http://cdrecord.berlios.de). genisoimage была разработана в рамках проекта CDRkit (www.cdrkit.org). Введите apt-get install genisoimage, чтобы установить соответствующий пакет.

Большинство Linux-пользователей хранит свои персональные файлы в собственных домашних каталогах, поэтому распространенный способ применения mkisofs для резервного копирования файлов — это создание резервной копии всего, что находится в каталоге /home. Вот несколько примеров использования mkisofs для создания ISO-образа из всех файлов и каталогов, содержащихся в каталоге /home:

```
$ cd /tmp
$ sudo mkisofs -o home.iso /home                    Создать базовый образ IS09660
$ sudo mkisofs -o home2.iso -J -R /home             Добавить расширения Joliet
                                                    и Rock Ridge
$ sudo mkisofs -o home3.iso -J -R -hfs /home        Добавить также расширения HFS
```

При выполнении последней команды вы увидите предупреждающее сообщение наподобие следующего:

```
genisoimage: Warning: no Apple/Unix files will be decoded/mapped
```

В каждом из трех приведенных выше примеров все файлы и каталоги, располагающиеся в каталоге /home, добавляются в ISO-образ (home.iso). В первом случае нет никаких расширений, в силу чего все имена файлов конвертируются в названия в стиле DOS (в формате 8.3). Во втором примере используются расширения Joliet и Rock Ridge, поэтому когда вы откроете ISO-образ в Linux- или Windows-системе, имена файлов и права доступа должны быть такими же, как и в оригинальной систем Linux. В последнем случае файлы в образе также делаются пригодными для чтения при использовании файловой системы, которая используется на компьютерах Macintosh.

ПРИМЕЧАНИЕ----------

Mac OS X тоже поддерживает чтение расширений Rock Bridge и Joliet.

Вы можете **добавить в образ различные источники.** Вот несколько примеров:

```
$ mkisofs -o home.iso -R -J music/ docs/ \          Различные каталоги/файлы
        chris.pdf /var/spool/mail
$ mkisofs -o home.iso -J -R \                       Добавить файлы в образ
    -graft-points Pictures/=/usr/share/pixmaps/ \
    /home/chris
```

В первом примере в приведенном выше коде показано комбинирование и размещение в корне ISO-образа различных файлов и каталогов, во втором — содержимое каталога/usr/share/pi xmaps добавляется в каталог /home/chris/Pictures. В результате в образе CD каталог / Pictures будет включать все содержимое каталога /usr/share/pi xmaps.

Добавление информации в заголовок ISO-образа может помочь вам идентифицировать содержимое этого образа позднее, что особенно полезно, если образ будет сохраняться или распространяться посредством Интернета без использования физического диска, на который можно записать данные. Вот несколько примеров:

```
$ mkisofs -o /tmp/home.iso -R -J                    \ Добавитьзаголовочную
                                                    информацию в ISO-образ
    -p www.handsonhistory.com                   \
    -publisher "Swan Bay Folk Art Center" \
    -V "WebBackup"                              \
    -A "mkisofs"                                \
    -volset "1 of 4 backups, July 30, 2013" \
    /home/chris
```

В данном примере -p используется для обозначения идентификатора создателя ISO-образа, который может включать телефонный номер, почтовый адрес или адрес сайта, посредством которого можно связаться с создателем ISO-образа. Используя параметр -publisher, вы можете обозначить 128-символьное описание создателя ISO-образа (например, название компании или организации).

Параметр -V используется для обозначения идентификатора тома. Идентификатор тома важен потому, что во многих Linux-системах применяется для монтирования CD, вставленного в привод. К примеру, в случае применения приведенной выше строки команды CD был бы смонтирован в /medi a/WebBackup в Ubuntu и других Linux-системах. Параметр -A можно задействовать для обозначения приложения, использованного для создания ISO-образа, а -vol set включает строку информации о наборе ISO-образов.

Создав собственный ISO-образ, перед его записью на диск вы **можете проверить образ** и убедиться, что у вас есть доступ к содержащимся в нем файлам. Вот способы сделать это:

```
$ volname home.iso                    Отобразить имя тома
WebBackup
$ isoinfo -d -i home.iso              Отобразить заголовочную информацию
CD-ROM is in ISO 9660 format
System id: LINUX
Volume id: WebBackup
Volume set id: All Website material on November 2, 2013
Publisher id: Swan Bay Folk Art Center
Data preparer id: www.handsonhistory.com
Application id: mkisofs
Copyright File id:
Abstract File id:
Bibliographic File id:
Volume set size is: 1
Volume set sequence number is: 1
Logical block size is: 2048
Volume size is: 23805
Joliet with UCS level 3 found
Rock Ridge signatures version 1 found
```

Вы можете видеть большое количество информации, введенной в строку команды mkisofs при создании образа. Если бы это был образ, предназначенный для публикации, то вам, возможно, также потребовалось бы обозначить местоположение на CD файла с информацией об авторских правах (-copyright), абстрактного файла (-abstract:), и библиографического файла (-biЫ io). При условии, что с заголовком все в порядке, далее вы можете попробовать **получить доступ к файлам в ISO-образе, смонтировав его:**

```
$ sudo mkdir /mnt/myimage                          Создать точку монтирования
$ sudo mount -o loop home.iso /mnt/myimage         Выполнить петлевое монтирование
                                                   ISO-образа
$ Is -1 /mnt/myimage                               Проверить содержимое ISO-образа
$ sudo umount /mnt/myimage                         Демонтировать образ по окончании
                                                   работы
```

ПРИМЕЧАНИЕ

ISO-образ монтируется только для чтения. Если вы захотите поработать с его содержимым, придется скопировать нужные файлы и каталоги в другой каталог.

Помимо проверки наличия доступа к файлам и каталогам в ISO-образе, убедитесь, что дата/временные метки, владение и права доступа такие, как вам надо. Эта информация может оказаться полезной, если потребуется восстановить данные позднее.

Запись резервных образов на оптические диски с помощью cdrecord

Команда cdrecord — самый популярный инструмент командной строки Linux для записи образов CD и DVD. После того как вы создадите ISO-образ (как было описано выше) либо получите его иным способом (например, загрузите установочный CD или Live CD из Интернета), cdrecord позволит вам с легкостью записать этот образ на диск.

ПРИМЕЧАНИЕ

В Ubuntu команда cdrecord заменена командой wodim. Команда wodim создана на основе кодовой базы cdrecord и при этом поддерживает большинство тех же самых параметров. Если вы введете cdrecord, то на самом деле в этом выпуске Ubuntu запустится выполнение wodim. Если возникнут проблемы с этой утилитой, свяжитесь с участниками проекта CDRkit (http://cdrkit.org). Введите apt-get install wodem, чтобы установить соответствующий пакет, если он еще не инсталлирован.

ISO-образы CD и DVD создаются одинаково, за исключением того, что образ DVD может, конечно же, оказаться намного большего размера, чем образ CD, поэтому проверьте емкость ваших носителей. CD обычно имеют объем 650, 700 или 800 Мбайт, а мини-CD — 50, 180, 185 или 193 Мбайт. Объем однослойных DVD составляет 4,7 Гбайт, а двухслойных — 8,4 Гбайт.

ПРИМЕЧАНИЕ

Однако имейте в виду, что производители CD/DVD указывают емкость выпускаемых ими носителей, считая 1 Мбайт равным 1000 Кбайт, а не 1024. Введите du -si home.iso, чтобы узнать размер своего ISO-образа, а не du -sh, как вы поступили бы для проверки, поместится ли ваш ISO-образ на имеющийся носитель.

Прежде чем: вы приступите к записи образа на CD или DV, **убедитесь, что ваш привод поддерживает запись CD/DVD,** и определите адрес привода. Для этого воспользуйтесь cdrecord с параметром --scanbus:

```
$ cdrecord --scanbus          Показывает привод, не поддерживающий запись
scsibus0:
        0,0,0 0) 'SAMSUNG ' 'DVD-ROM SD-616E ' 'F503' Removable CD-ROM'
        0.0,0 1)*
        0.0,0 2)*
...
```

```
$ cdrecord -scanbus        Показывает привод, который позволяет записывать CD
                           или DVD
scsibusO:
        0.0,0 0) 'LITE-ON ' 'DVDRW S0HW-1633S' 'BSOC Removable CD-ROM
        0,0,0 1)*
        0,0,0 2)*
        ...
```

В первом приведенном выше примере показан CD/DVD-привод, который поддерживает только чтение и не позволяет записывать CD (может читать DVD- и CD-ROM). Во втором примере вы видите привод, поддерживающий запись CD или DVD (DVD-RW). Вставьте в привод носитель, на который хотите записать образ. Если привод способен записать данные на имеющийся у вас носитель, вот простые примеры использования команды cdrecord для **записи образов CD или DVD:**

```
$ cdrecord -dummy home.iso      Провести тест записи без ее осуществления
                                на самом деле
$ cdrecord -v home.iso          Записать CD (используя настройки по умолчанию)
                                с генерированием подробного вывода
$ cdrecord -v speed=24 home.iso Установить определенную скорость
$ cdrecord -pad home.iso        Если невозможно выполнить чтение дорожки, то
                                добавить 15 секторов, заполненных нулями
$ cdrecord -eject home.iso      Извлечь CD/DVD по окончании записи
$ cdrecord /dev/cdrw home.iso   Идентифицировать привод по имени устройства
                                (может отличаться)
$ cdrecord dev=0,2,0 home.iso   Идентифицировать привод по имени SCSI
```

Команда cdrecord также позволяет **записывать мультисессионные CD/DVD.** Вот пример:

```
$ cdrecord -multi home.iso      Открыть мультисессию записи
$ cdrecord -msinfo              Проверить смещение сессии для следующей
                                процедуры записи
Using /dev/cdrom of unknown capabilities
0.93041
$ mkisofs -J -R -o new.iso \    Создать второй ISO-образ для записи
    -C 0,93041 /home/chris/more Обозначить начальную точку и новые
                                данные для ISO-образа
$ cdrecord new.iso              Записать новые данные на имеющийся CD
```

Вы можете выполнять запись с параметром -multi, пока CD не будет заполнен. Во время последней процедуры записи не используйте -multi, чтобы CD был закрыт.

Создание и запись DVD с помощью growisofs

Используя команду growisofs, вы можете **объединить два этапа, которые заключаются соответственно в сборе файлов в ISO-образ (mkisofs) и записи этого**

образа на DVD (cdrecord). Помимо объединения двух операций в одну, команда growisofs также обеспечивает преимущество в виде возможности сохранения сессии, открытой по умолчанию до тех пор, пока вы не закроете ее, благодаря чему вам не придется делать ничего особенного при проведении мультисессий записи.

Вот пример использования growisofs при проведении **мультисессий записи:**

```
$ growisofs -Z /dev/dvd -R -J /home/chris   Осуществить мастеринг и запись
                                            на DVD
$ growisofs -Z /dev/dvd -R -J /home/chris   Добавить данные для записи
$ growisofs -M /dev/dvd=/dev/zero           Закрыть сессию записи
```

Если вы захотите применить дополнительные параметры при создании ISO-образа, можете просто добавить параметры mkisofs в строку команды (взгляните, к примеру, на то, как в приведенных ранее примерах использовались параметры -R и -J).

Чтобы **записать DVD-образ посредством growisofs,** можете воспользоваться параметром -dvd-compat:

```
$ growisofs -dvd-compat -Z /dev/dvd-=image.iso   Записать ISO-образ на DVD
```

Параметр -dvd-compat позволяет обеспечить лучшую совместимость с разными DVD-приводами во время процедур записи на мультисессионные DVD.

Резюме

В системе Linux и предшествующих ей UNIX-системах резервное копирование данных выполнялось путем комбинирования команд, каждая из которых отвечала за решение отдельного набора задач. Резервное копирование ваших критически важных данных по-прежнему можно осуществлять таким путем. Однако многие инструменты, которые вы можете использовать, позволяют действовать безопаснее и эффективнее.

Задачи, для решения которых применяется утилита, являющаяся ленточным архиватором (команда tar), вышли далеко за пределы того, для чего она первоначально предназначалась — для записи резервных копий файлов данных на магнитную ленту. Почти каждая система Linux и UNIX включает tar, поэтому данная команда превратилась в стандартную утилиту для упаковки программного обеспечения и резервного копирования данных в сжатые архивы. Эти архивы затем можно переда:зать и хранить различными способами.

Для передачи резервных копий данных на другие компьютеры по сети вы можете использовать функции удаленного исполнения, которыми обладают OpenSSH-инструменты (например, ssh). Вы также можете прибегнуть к отличной утилите rsync, используя которую вы сэкономите ресурсы, подвергая резервному копированию только изменившиеся файлы (или их части).

Благодаря невысокой стоимости CD и DVD стали популярными носителями для размещения резервных копий личных данных или информации, принадлежащей небольшим офисам. Команда mkisofs позволяет создавать файловые системы данных, подвергнутых резервному копированию, в формате ISO9660, которые можно восстановить в различных операционных системах (Linux, Windows или Mac). После создания ISO-образа с помощью mkisofs его можно записать на CD или DVD, воспользовавшись командой cdrecord или growisofs.

9 Проверка запущенных процессов и управление ими

В этой главе:

- просмотр информации об активных процессах с помощью команд ps и top;
- поиск процессов с использованием pgrep;
- изменение приоритетов процессов касательно использования ресурсов центрального процессора посредством nice и renice;
- перевод процессов в фоновый или приоритетный режим;
- завершение процессов и отправка им сигналов с помощью kill и killall;
- использование at и batch для выполнения команд;
- планирование повторного выполнения команд посредством cron.

После запуска выполняемой программы она работает как процесс, который управляется посредством таблицы процессов вашей Linux-системы. Linux обеспечивает все инструменты, необходимые для просмотра и изменения активных процессов.

Команды ps и top отлично подходят для просмотра информации о ваших запущенных процессах. Для ps и top есть множество параметров, которые помогают отображать информацию о процессах именно так, как вам надо. Команда pgrep может дополнительно помочь при поиске нужного процесса.

Существуют такие команды, как nice и renice, для повышения и понижения приоритета того или иного процесса. Вы можете переводить запущенные процессы в фоновый режим (с помощью команды bg) и обратно в приоритетный (посредством команды fg). В редких случаях можно использовать команду chrt для выполнения процессов в реальном времени.

Отправка сигналов процессу — это способ изменить его поведение или вообще завершить его. Используя команды kill и killall, вы можете отправлять сигналы процессам согласно их соответственно PID или именам. Вы также можете посылать другие сигналы процессам с целью, например, повторного чтения конфигурационных файлов или возобновления остановленного процесса.

Для выполнения команд в запланированное время или так, чтобы они не были привязаны к сеансу вашего интерпретатора команд, вы можете использовать at и batch. Для повторного выполнения команд в заданное время существуют такие

инструменты, как cron и anacron. Вы также можете разместить сценарии (или символические ссылки на них) в /etc/cron.hourly (или cron.daily, cron.weekly либо cron, monthly).

Отображение информации об активных процессах

Чтобы увидеть, какие процессы запущены в системе на данный момент, большинство пользователей выполняет команды ps и top. Команда ps позволяет сделать моментальный снимок (в виде простого списка) процессов, запущенных в настоящее время. Команда top обеспечивает экранно-ориентированное, постоянно обновляемое представление запущенных процессов с сортировкой на ваш выбор (в зависимости от уровня использования ресурсов центрального процессора, оперативной памяти, UID и т. д,).

Просмотр информации об активных процессах с помощью ps

Каждая Linux-система (а также любая производная от UNIX, например BSD, Mac OS X и др.) включает команду ps. Однако с годами появилось множество немного различных версий ps, позволяющих использовать немного отличающиеся параметры. Команда ps берет начало в первых UNIX-системах, поэтому поддерживает и нестандартные способы ввода некоторых параметров (к примеру, позволяя поставить тире перед тем или иным параметром в определенных случаях).

Примеры использования команды ps, приведенные в этой главе, сработают в Ubuntu и большинстве других Linux-систем. Далее показаны примеры, к которым вы можете прибегнуть для **отображения информации о запущенных процессах текущего пользователя** (описание столбцов вывода ps приведено в табл. 9.1).

```
$ ps                    Показать информацию о процессах текущего пользователя
                        в текущем интерпретаторе команд
 PID TTY TIME CMD
2552 pts/0     00:00:00 bash
3438 pts/0     00:00:00 ps
$ ps -u chris           Показать информацию о запущенных процессах пользователя
                        chris (простой вывод)
PID TTY TIME COMMAND
2678 ttyl 0:00 startx
2689 ttyl 0:00 xinit
2710 ttyl 0:06 gnome-session
...
$ ps -u chris u         Показать информацию обо всех запущенных процессах
                        пользователя chris (с CPU/MEM)
USER PID 3SCPU &MEM VSZ RSS TTY STAT START TIME COMMAND
chris 2678 0.0 0.0 4328 852 ttyl S+ Augl4 0:00 /bin/sh startx
```

```
chris 2689 0.0 0.1 2408 488 ttyl S+ Augl4 0:00 xinit
chris 2710 0.0 1.1 22016 5496 ttyl S        Augl4 0:06 gnome-session
...
```

$ ps -fu chris *Показать информацию обо всех запущенных процессах пользователя chris (с PP1D)*

```
UID   PID   PPID   C STINE TTY          TIME CMD
chris  2678  2645   0 Aug 14 ttyl 00:00:00 /bin/sh /usr/XHR6/bin/startx
chris  2689  2673   0 Augl4 ttyl 00:00:00 xinit /etc/Xll/xinit/xinitrc
chris 2710 2681     0 Augl4 ttyl 00:00:09 /usr/bin/gnome-session
...
```

$ ps -Fu chris *Показать информацию о запущенных процессах пользователя chris (с SZ и PSR)*

```
UID   PID   PPID   C SZ RSS PSR STIME TTY          TIME CMD
chris  2678  2645   0  1082  852   0  Augl4 ttyl  00:00:00 /bin/sh startx
chris  2689  2678   0   602  488   0  Augl4 ttyl  00:00:00 xinit
chris 2710 2689     0  5504 5440   0  Augl4 ttyl  00:00:09 gnome-session
```

В этих примерах показаны некоторые из процессов пользователя во время сеанса рабочего стола GNOME. В первом примере демонстрируется команда ps, выполнение которой было запущено из окна терминала, в силу чего вы видите информацию только о процессах касательно текущего интерпретатора команд в этом окне. В остальных примерах показано, как вывести на экран различные сведения о каждом процессе (позднее вы узнаете, как обеспечить генерирование пользовательского вывода).

Вот примеры использования ps для генерирования вывода по каждому процессу, запущенному в системе на данный момент:

$ ps -e *Показать информацию о каждом запущенном процессе*

```
  PID TTY          TIME CMD
    1 ?        CO:00:01 init
    2 ?        CO:00:00 migration/0
    3 ?        00:00:00 ksofti rqd/0
...
```

$ ps -el *Показать информацию о каждом запущенном процессе в виде длинного списка*

```
F S UID          PID PPID C PRI NI ADDR SZ WCHAN TTY          TIME CMD
4 S 0          I 0 0 75 0 - 534 -                ?        00:00:01 init
1 S 0          2 1 0 -40 - - 0 -                 ?        00:00:00 migration/0
1 S 0          3 1 0 94 19 - 0 -                 ?        00:00:00 ksoftirqd/0
...
```

$ ps -ef *Показать информацию о каждом запущенном процессе в виде полноформатного списка*

```
UID         PID  PPID   C  STIME  TTY          TIME CMD
root          1     0   0  Aug05  ?        00:00:01 init [5]
root          2     1   0  Aug 05 ?        00:00:00 [migration/0]
root          3     1   0  Aug05  ?        00:00:00 [ksoftirqd/0]
...
```

$ ps -eF *Показать информацию о каждом запущенном процессе в виде особого полноформатного списка*

```
DID PID PPID C SZ RSS PSR STIME TTY TIME CMD
```

```
Root         1    0  0  534    556    0 Aug05 ?        00:00:01 init [5]
Root         2    1  0    0      0    0 Aug05 ?        00:00:00 [migration/0]
Root         3    1  0    0      0    0 Aug05 ?        00:00:00 [ksoftirqd/0]
  ...
$ ps ax          Показать информацию о каждом запущенном процессе в сжатом
                 BSD-стиле
  PID TTY      STAT TIME COMMAND
    1 ?          Ss 0:01 init [5]
    2 ?        S    0:00 [migration/0]
    3 ?        SN   0:00 [ksoftirqd/0]
  ...
$ ps aux         Показать информацию о каждом запущенном процессе
                 в подробном BSD-стиле
USER  PID «CPU«MEMVSZ RSS  TTYSTAT START  TIME COMMAND
Root    1  0.0   0.0  2136 556  ?   Ss   Aug05   0:01 init [5]
Root    2  0.0   0.0     0 0    ?   S    Aug05   0:00 [migration/0]
Root    3  0.0   0.0     0 0    ?   SN   Aug05   0:00 [ksoftirqd/0]
  ...
$ ps auwx Показать информацию о каждом запущенном процессе
                 в BSD-стиле в широком формате
$ ps auwwx Показать информацию о каждом запущенном процессе
                 в BSD-стиле с неограниченной шириной
```

Некоторые процессы активизируют другие процессы. Например, веб-сервер (демон httpd) запустит множество демонов httpd, которые будут ожидать поступления запросов к веб-серверу. Вы можете просматривать иерархию процессов (в виде дерева), используя ps в сочетании с разными параметрами:

```
$ ps -ejH            Показать иерархию процессов с идентификаторами
                     процессов/сеансов.
  PID PGID    SID TTY          TIME CMD
16267 16267 16267 ?        00:00:00      sshd
16462 16267 16267 ?        00:00:00      sshd
16463 16463 16463 pts/2    00:00:00        bash
16563 16563 16463 pts/2    00:00:00          f i refox
16606 16606 16463 pts/2    00:00:00          sudo
16607 16606 16463 pts/2    00:00:00            su
16615 16615 16463 pts/2    00:00:00              bash
16673 16673 16463 pts/2    00:00:00                    ps
$ ps axjf                Показать иерархию процессов в виде вывода в BSD-стиле
 PPID PID    PGID SID TTYTPGID STAT UID  TIME COMMAND
    1  957    957   957 ?    -1 Ss       0 0:00 /usr/sbin/sshd -D
  957 16267 16267 16267 ?    -1 Ss       0 0:00 \_ sshd: chris [priv]
16267 16462 16267 16267 ?    -1 S     1000 0:00 \_ sshd: chrisOpts/
16462 16463 16463 16463 pts/2 16688 Ss 1000 0:00 \_ -bash
16463 16563 16563 16463 pts/2 16688 SI 1000 0:02   \_ /usr/1ib
16463 16606 16606 16463 pts/2 16688 S  0    0:00     \_ sudo su
$ ps -ef --forest Показать иерархию процессов в виде леса
UID        PIDPPID  CSTIME      TTY   TIMECMD
root       957    1  0 Febl8 ?        00:00:00 /usr/sbin/sshd -D
root     16267 957  0 19:03 ?        00:00:00   \_ sshd: chris [priv]
```

```
chns  16462  16267  O  19:03 ?        00:00:       \_ sshd: chris@pts/2
chris 16463  16462  O  19:03 pts/2    00:00:00       \_ -bash
chris 16563  16463  O  19:03 pts/2    00:00:02          _  \ /usr/1ib/firefox/
root  16606  16463  O  19:03 pts/2    00:00:00            \_ sudo su -
root  16607  16606  0  19:03 pts/2    00:00:00                    \_su -
$ pstree                                   Показать процессы в алфавитном порядке в виде дерева
 l-rsysl ogd— 3*[{rsyslogd} ]
      l-rtki t-daemon—2*[ {rtki t-daemon} ]
      l-sound-juicer—3*[    {sound-juicer}]
      l-sshd—sshd— sshd—bash—fi refox—25*[{fi refox}]
      |                              L-ssudo—su— bash— pstree
  . . .
```

В этих примерах древовидных представлений показаны разные способы отображения иерархии процессов. Вывод был сокращен для удобства сравнения нескольких одинаковых процессов, для которых на экране отображаются отличающиеся сведения. Следует отметить, что PPID (Parent Process ID — идентификатор родительского процесса) является идентификатором процесса, запустившего каждый из показанных дочерних. Процессы `sshd` демонстрируют выполняющийся Secure Shell Daemon (демон безопасного интерпретатора команд) с пользователем, вошедшим в систему по сети, в результате чего запускается интерпретатор команд bash. Из последнего пользователь выполняет команды `fi refox` в фоновом режиме. Затем пользователь открывает интерпретатор команд от имени суперпользователя (`sudo su`). В последнем примере показана команда `pstree`, которая, в частности, используется для отображения древовидных представлений процессов.

Если вы предпочитаете персонализированные представления вывода ps, можете использовать параметр `-o` для выбора, какие именно столбцы данных должны отображаться на экране посредством ps. Затем вы можете воспользоваться параметром `--sort` для сортировки вывода по любым из этих данных. В табл. 9.1 приведены возможные варианты вывода в виде столбцов, а также параметры, которые необходимо добавить к `-o`, чтобы команда ps отобразила на экране каждый столбец.

Таблица 9.1. Выбор и просмотр вывода ps в виде столбцов

Параметр	Заголовок столбца	Описание
%cpu	%CPU	Уровень использования ресурсов центрального процессора во время существования процесса в формате 00.0
%mem	%MEM	Уровень использования физической памяти системы в процентах (размер резидентного набора)
args	COMMAND	Команда со всеми аргументами
bsdstart	START	Время запуска команды в формате ЧЧ:ММ или Месяц:День
bsdtime	TIME	Общее (пользовательское и системное) время центрального процессора
comm	COMMAND	Только имя команды (без отображения аргументов)

Продолжение ^

Таблица 9.1 *(продолжение)*

Параметр	Заголовок столбца	Описание
cp	CP	Уровень использования ресурсов центрального процессора с точностью до десятой доли процента
cputime	TIME	Общее время центрального процессора в формате [Число-]Часы:Минуты:Секунды
eg id	EGID	Эффективный идентификатор группы процесса (в виде целого числа)
egroup	EGROUP	Эффективный идентификатор группы процесса (в виде имени)
etime	ELAPSED	Количество времени, прошедшего с момента запуска процесса, в формате [[Число-]Часы:]Минуты:Секунды
euid	EUID	Эффективный идентификатор пользователя процесса (в виде целого числа)
euser	EUSER	Эффективный идентификатор пользователя процесса (в виде имени)
fgid	FGID	Идентификатор группы с доступом к файловой системе (в виде номера)
fgroup	FGROUP	Идентификатор группы с доступом к файловой системе (в виде имени)
fname	COMMAND	Первые восемь символов имени команды
fuid	FUID	Идентификатор пользователя с доступом к файловой системе (в виде номера)
fuser	FUSER	Идентификатор пользователя с доступом к файловой системе (в виде имени)
Istart	STARTED	Дата и время запуска выполнения команды
nice	NI	Значение nice, которое может находиться в диапазоне от 19 (самый низкий приоритет процесса) до -20 (наивысший приоритет процесса, позволяющий ему захватывать ресурсы центрального процессора)
pgid	PGID	Идентификатор группы процесса
pid	PID	Идентификационный номер процесса
ppid	PPID	Идентификатор процесса, являющегося родительским по отношению к соответствующему
psr	PSR	Значение, присвоенное процессу процессора (в случае с первым центральным процессором оно равно 0)
rgid	RGID	Реальный идентификатор группы (в виде номера)
rgroup	RGROUP	Реальный идентификатор группы (в виде имени)
rss	RSS	Физическая память без подкачки (размер резидентного набора)в килобайтах
rtprio	RTPRIO	Приоритет реального времени

Параметр	Заголовок столбца	Описание
ruid	RUID	Реальный идентификатор пользователя (в виде номера)
ruser	RUSER	Реальный идентификатор пользователя (в виде имени)
s	S	Односимвольный индикатор состояния (D — спящий, не может быть прерван; R — выполняющийся; S — спящий, может быть прерван; T — остановленный; W — «слушает»; X — неактивный; Z — «зомби»)
sess	SESS	Идентификатор сеанса лидера сеанса
sgLp	P	Процессор, с использованием которого процесс выполняется на данный момент
size	SZ	Приблизительный объем пространства подкачки, который необходим, если процесс потребуется выгрузить
start	STARTED	Время запуска команды в формате ЧЧ:ММ:СС или Месяц День
start_time	START	Время запуска выполнения команды: ЧЧ:ММ или Месяц День
stat	STAT	Многосимвольный индикатор состояния: односимвольный индикатор состояния s плюс другие символы для обозначения состояния (< — высокий приоритет, N — низкий приоритет, L — обладает страницами, заблокированными в памяти, s — является лидером сеанса, 1 — многопоточный, + — находится в группе фоновых процессов)
SZ	SZ	Размер образа ядра процесса (физические страницы)
tname	TTY	Управляющая консоль tty (терминал)
user	USER	Эффективный идентификатор пользователя процесса (в виде имени)
vsize	VSZ	Объем виртуальной памяти процесса (1024-байтовые блоки)

Следует отметить, что некоторые значения, предназначенные для отображения имен пользователей, могут показывать вместо них номера (UID), если имена окажутся слишком длинными, чтобы уместиться в имеющемся пространстве.

Задействовав разделенный запятыми список параметров для отображения столбцов, вы сможете сгенерировать собственный пользовательский вывод. Вот несколько примеров **пользовательских представлений запущенных процессов:**

$ ps -eo ppid,userДтеш,size,vsize,comm --sort=-size *Сортировать по уровню использования памяти*

```
 PPID USER ЖМЕМ SIZE VSZ COMMAND
    1 root     0.0   1004292 1043060 console-kit-dae
 6901 chris 2.3       713248 1125172 compiz
```

$ ps -eo ppid,user,bsdstart.bsdtime,Xcpu.args --sort=-*cpu *Сортировать по уровню использования ресурсов центрального процессора*

```
 PPID USER START TIME £CPU COMMAND
 6901 chris Feb 23 11:23 0.1 compiz
16463 chris 19:03 0:03 0.0 /usr/Iib/firefox/firefox
 1101 root Feb 18 14:31 0.0 /usr/bin/X :0 -auth ...
```

$ ps -eo ppid,user,nice,cputime,args --sort=-nice *Сортировать по приоритету, начиная с низкого и далее*

```
PPIDUSER     NI      TIMECOMMAND
  2     root     19  00:00:00[khugepaged]
  1 chris       10  00:00:31   /usr/bin/python /usr/bin/update-manager
  2   root 500:00:00 [ksmd]
```

$ ps -eo ppid,user,stat.tname,sess,cputime,args --sort=user *Сортировать по имени пользователя*

```
PPID USER      STAT TTYSESS      TIMECOMMAND
   1 avahi     S    ?    851   00: 00: 09 avahi-daemon: running
1640 chris     Ssl  ?   6901   00: 00: 00 gnome-session --session=ubuntu
   1 daemon    Ss   ?   1088   00: 00: 00 atd
 846 Ip        S    ?    846   00: 00: 00 /usr/Iib/cups/notifier/dbus
   0 root      Ss   ?      1   00: 00: 00 /sbin/init
```

Вот другие **примеры использования команды ps:**

```
$ ps -C httpd          Отобразить информацию о запущенных процессах httpd
  PID TTY          TIME CMD
 1493 ?        00:00:00 httpd
 1495 ?        00:00:00 httpd
```

Следует отметить, что вам потребуется установить HTTP-сервер, например Apache, для запуска процессов httpd.

```
$ ps -p 16563 -o pid,ppid,bsdtime,  args        Отобразить информацию о процессе с PID 16565

  PID PPID TIME COMMAND
16563 16463 0:04 /usr/1ib/firefox/firefox
$ ps -U chris,avahi -o pid,ruser,tty,stat,args        Отобразить информацию о двух пользователях

  PID RUSER TT       STAT COMMAND
  852 avahi ?          S    avahi-daemon: running [ubuntutb.local]
  853 avahi 7          S    avahi-daemon: chroot helper
 6890 chris 7          SI   /usr/bi'n/gnome-keyring-daemon
 6901 chris 7          Ssl  gnome-session --session=ubuntu
```

Наблюдение за активными процессами с помощью top

Для *просмотра* **информации о процессах, запущенных в вашей системе, на постоянной основе** можете использовать команду top. Она обеспечивает экранно-ориентированное представление активных процессов, которое постоянно обновляется. Если вы запустите выполнение команды top без параметров, она выведет на экран информацию о том, сколько времени работает ваша система, о задачах, уровне использования ресурсов центрального процессора и памяти, а затем представит список запущенных процессов, отсортированный по уровню использования ресурсов центрального процессора. Пример:

```
$ top
top - 2:37:18 up 2 days, 14:00, 3 users, load average: 0.00, 0.01, 0.05
Tasks: 178 total, 1 running, 177 sleeping, 0 stopped, 0 zombie
```

```
Cpu(s): O.O^us, 0.2%sy, 0.0&ni, 99.7£id, O.O&wa, O.OXhi, 0.2%si, O.O^st
Mem: 4014504k total, 2208972k used, 1805532k free, 161628k buffers
Swap: 4157436k total.            Ok used,  4157436k free, 1388516k cached
  PID USER      PR  NI  VIRT  RES  SHR S %CPUЯMEM     TIME+COMMAND
 6967 chris     20   0 1213m  93m  29m S      0  2.4   11:24.95compiz
16748 root      20   0 17340 1332  932 R      0  0.0    0:00.09top
```

Вот примеры других параметров, которые вы можете использовать для **выполнения top с целью постоянного отображения информации о запущенных процессах:**

$ top -d 5	*Изменить время задержки обновления, сделав его равным 5 секундам (вместо 3 секунд по умолчанию)*
`$ top -u chris`	*Показать информацию только о процессах эффективного пользователя с именем chris*
$ top -p 190,2690	*Отобразить информацию только о процессах 190 и 2690*
$ top -n 10	*Обновить экран 10 раз перед выходом*
$ top -b	*Запустить выполнение в пакетном режиме*

В последнем примере (top -b) вывод top форматируется так, чтобы он подходил для направления в файл, вместо повторного отображения одного и того же экрана для интерактивного просмотра. Эту команду можно использовать для создания журнала процессов, например при отслеживании процесса-беглеца, пожирающего все ваши ресурсы среди ночи. Вот как **запустить выполнение top и занести вывод в журнал** 50 раз:

$ top -b -п 50 > myprocesslog

При выполнении top вы можете нажимать следующие клавиши и их комбинации с целью обновления и сортировки списка процессов различными способами:

- **Пробел или Enter** — немедленно обновляет список процессов;
- **Shift+N** — сортирует по PID;
- **Shift+P** — сортирует по уровню использования ресурсов центрального процессора;
- **Shift+M** — сортирует по уровню использования памяти;
- **Shift+T-** сортирует по количеству затраченного времени центрального процессора;
- **клавиша** < — сортирует столбец слева;
- **клавиша** > — сортирует столбец справа;
- **F+клавиша с буквой столбца** — сортирует после отображения списка столбцов.

Существует несколько способов изменить поведение команды top во время выполнения:

- **D+клавиша с цифрой** — обеспечивает задержку между обновлениями на количество секунд, соответствующее цифре на нажатой клавише;
- **U+ввод имени пользователя** — отображает информацию только о процессах выбранного пользователя;

- **N+клавиша с цифрой, означающей количество процессов, информацию о которых вы желаете увидеть,** — отображает только выбранное количество процессов;
- **клавиша** = — обеспечивает возврат на исходный экран top (можете сделать это в любой момент).

Вы можете по-разному воздействовать на любой из запущенных процессов:

- **К+ввод** PID — отправляет сигнал запущенному процессу (завершает его);
- **клавиша с цифрой 9 (после нажатия К)** — завершает процесс;
- **ввод номера другого сигнала (после нажатия клавиши К)** — отправляет соответствующий сигнал процессу;
- 0 **N** — присваивает процессу более высокий или более низкий приоритет выполнения; можете нажать клавишу N, а затем ввести отрицательное число (чтобы повысить приоритет) или положительное (чтобы понизить приоритет).

Чтобы **найти дополнительную информацию об использовании top,** введите ? во время сеанса top. На соответствующей MAN-странице также есть много информации касательно способов применения top:

```
$ man top Показать MAN-страницу, посвященную top
```

Закончив использовать top, нажмите клавишу Q для выхода.

Поиск процессов и управление ими

Изменение запущенного процесса в первую очередь означает, что нужно найти выбранный вами процесс, а затем модифицировать его приоритет либо отправить ему сигнал для изменения его поведения. При поиске процесса иногда сложно определить его местоположение в большом списке, сгенерированном командой ps или top.

Команда рдгер предлагает способы поиска среди активных процессов тех, которые вам нужны. Команда renice позволяет изменять приоритеты запущенных процессов. Благодаря командам kill, pkill и killall вы можете отправлять сигналы активным процессам (включая сигналы для их завершения).

Команда рдгер для поиска процессов

В ее самой простой форме команду рдгер можно использовать для поиска имени команды (или его части) и отображения идентификатора любого процесса, который включает это название. Например:

```
$ рдгер init Показать PID любого процесса, имя которого включает строку 'Init'
1
2689
```

Вам известно, что выполняется только одна команда init, поэтому далее используйте параметр -1, чтобы увидеть имя команды каждого процесса (с целью узнать, почему обнаружилось два процесса):

```
$ рдгер -I i n i t  Показать PID и имена процессов, включающие строку init
1 init
2689 xinit
```

Вы также можете **осуществлять поиск процессов, ассоциированных с определенным пользователем:**

```
$ рдгер -1u chris  Показать все процессы, владельцем которых
                   является пользователь chris
16462 sshd
16463 bash
16563 firefox
```

Вероятно, самым эффективным способом использования команды рдгер является ее применение для **поиска идентификаторов запущенных процессов и передачи этих PID по каналу другой команде** для генерирования вывода. Вот несколько примеров (ищите другие команды, если окажется, что sshd или firefox не выполняется):

```
$ ps -p 'рдгер sshd'        Осуществить поиск sshd и выполнить run ps
                            (краткий вывод)
  PID TTY      STAT   TIME COMMAND
  957 ?        Ss     0:00 /usr/sbin/sshd -D
16267 ?        Ss     0:00 sshd: chris [priv]
16462 ?        R      0:00 sshd: chrisOpts/2
$ ps -fp $(рдгер firefox)  Осуществить поиск firefox и выполнить ps
                           (полный вывод)
UID PID PPID C STIME TTY TIME              CMD
chris 16563 16463 0 19:03 pts/2 00:00:05 /usr/Tib/fi refox/fi refox
$ sudo renice -5 $(pgrep firefox)   Осуществить поиск firefox,
                                    повысить приоритет
16563 (process ID) old priority 0. new priority -5
```

Любую команду, которая способна принимать идентификатор процесса в качестве ввода, можно комбинировать с рдгер показанными выше способами. Как видно из приведенного ранее примера использования рдгер, вы можете задействовать такие команды, как renice, для изменения поведения процесса во время его выполнения.

Команда fuser для поиска процессов

Еще один способ найти определенный процесс заключается в выполнении поиска, ориентируясь на то, к чему процесс осуществляет доступ. Команду fuser можно использовать для выяснения, у каких процессов в данный момент есть открытый файл или сокет. Когда эти процессы найдены, fuser можно применять для отправки им сигналов.

Команда fuser наиболее полезна для выяснения, располагаются ли файлы, которые процессы держат открытыми, в смонтированных файловых системах (как, например, на локальных жестких дисках или совместно используемых ресурсах Samba). Найдя эти процессы, вы сможете закрыть их надлежащим образом (или

просто завершить, если потребуется), чтобы аккуратно демонтировать файловую систему (если вы не обнаружите команду fuser, установите пакет psmisc).

Вот несколько примеров использования `fuser` **для отображения процессов с файлами, открытыми в выбранной файловой системе:**

```
$ sudo fuser -uiauv /boot      Подробный вывод касательно процессов с файлами,
                               открытыми в /boot
                     USER        PID ACCESS COMMAND
/boot:               root     kernel mount (root)/boot
                     root      16615 . .c.. (root)bash
                     Root      17289 . .c.. (root)v1
```

В данном примере отображаются идентификаторы запущенных процессов, которые ассоциированы с `/boot`. У них может быть открыт файл, интерпретатор команд, либо они могут оказаться дочерними процессами интерпретатора команд с текущим каталогом в `/boot`. Конкретно в этом примере открыт интерпретатор команд bash в файловой системе `/boot` и выполняется одна команда vi с файлами, открытыми в `/boot`. Параметр `-a` показывает все процессы, `-и` указывает, какой пользователь владеет каждым из процессов, а `-V` обеспечивает генерирование подробного вывода.

Вот другие примеры использования `fuser для` **отображения процессов с открытыми файлами:**

```
$ fuser -m /boot               Показать PID всех процессов с файлами, открытыми
                               в /boot
/boot:          16615c 17289c 17312c
$ sudo fuser -um /boot Показать PID/пользователя для этого интерпретатора
                               команд, открытого в /boot
/boot:         16615c(root) 17289c(^oot) 17312c(root)
```

Узнав, у каких процессов есть открытые файлы, вы сможете вручную закрыть эти процессы или завершить их. По возможности всегда вручную останавливайте процессы, поскольку при уничтожении после них могут оставаться нежелательные файлы. Вот примеры использования `fuser` для **уничтожения или отправки иных сигналов всем процессам с файлами, открытыми в файловой системе:**

```
$ sudo fuser -k /boot          Завершить процессы с файлами, открытыми в /boot.
$ fuser -1                     Показать поддерживаемые сигналы.
HUP INT QUIT ILL TRAP ABRT IOT BUS FPE KILL USR1 SEGV USR2 PIPE ALRM
TERM
 STKFLT CHLD CONT STOP TSTP TTIN TTOU URG XCPU XFSZ VTALRM PROF
WINCH 10
PWR SYS UNUSED
$ sudo fuser -k -HUP /boot Отправить сигнал HUP процессам с файлами,
                               открытыми в /boot.
```

Изменение запущенных процессов

Даже после запуска процесса его поведение можно изменять разными способами. Используя команду ren ice, описанную ранее, вы можете модифицировать приори-

тет запущенного процесса в планировщике системы. С помощью nice можно определить приоритет по умолчанию, а также присвоить более высокий или более низкий приоритет в момент запуска процесса.

Другой способ изменить поведение процесса заключается в отправке ему сигнала. Команды kill и ki П а 11 можно использовать для отправки сигналов запущенным процессам. Команда pki 11 тоже позволяет посылать сигналы процессам.

Изменение приоритетов процессов в плане использования ресурсов центрального процессора с помощью nice

У каждого запущенного процесса есть значение nice, которое можно использовать для указания планировщику процессов Linux, какой приоритет следует присвоить этому процессу. Положительные значения nice в действительности наделяют ваш процесс более низким приоритетом. Соответствующая концепция возникла во времена больших многопользовательских UNIX-систем, в которых можно было проявить тактичность, запустив несрочный процесс с более низким приоритетом, чтобы другие пользователи смогли задействовать ресурсы центрального процессора.

Значение nice не навязывает приоритет при планировании, являясь лишь рекомендацией для планировщика. Чтобы узнать, **каким у вас является текущее значение nice по умолчанию,** выполните команду nice без параметров:

```
$ nice          Выполнить команду nice для определения текущего значения nice
```

Значением nice по умолчанию является 0. Вы можете использовать команду nice для запуска процесса с более высоким или более низким приоритетом по сравнению с присвоенным по умолчанию. Значение приоритета находится в диапазоне от -20 (наивысший приоритет при планировании) до 19 (самый низкий приоритет при планировании). Обладатель прав суперпользователя может увеличивать или уменьшать значение nice в случае с любым пользователем, а у обычного пользователя есть возможность только понижать приоритет того или иного процесса (путем задания более высокого значения для nice).

ВНИМАНИЕ

Действуйте осторожно, задавая отрицательные значения nice для процессов. В работе операционной системы, установленной на вашем компьютере, может произойти серьезный сбой, если критически важные системные процессы лишатся своего высокого приоритета.

Вот примеры запуска выполнения команды nice для **изменения значения nice команды:**

```
$ nice -n 12 nroff -man a.roff | less   Форматировать MAN-страницы. присвоив соответствующему процессу низкий приоритет
$ sudo nice -n -10 gimp                 Запустить выполнение gimp с более высоким приоритетом
```

Когда процесс уже запущен, вы можете **изменить его значение nice, воспользовавшись командой renice.** Примеры:

```
$ renice +2 -u chris          Изменить значения nice процессов пользователя chris, увеличив их на 2
```

```
$ renice +5 4737                          Изменить значение nice процесса
                                          с PID 4737, увеличив его на 5
$ sudo renice -3 'pgrep -u chris spamd'   Изменить значения nice процессов
                                          spamd пользователя chris
9688: old priority -1, new priority -3
20279: old priority -1, new priority -3
20282: old priority -1. new priority -3
```

В приведенной выше строке команд обратные кавычки используются в качестве индикатора того, что вывод команды pgrep (предполагается, что им будут PID демонов spamd, запущенных пользователем chris) передается команде renice.

Значения nice ваших процессов отображаются по умолчанию при выполнении top. Вы также можете увидеть значения nice, используя -o nice, когда будете генерировать пользовательский вывод с помощью команды ps.

Запуск процессов в фоновом или приоритетном режиме

При запуске процесса из интерпретатора команд он по умолчанию выполняется в приоритетном режиме. Это означает, что вы не сможете ввести другую команду, пока не закончится выполнение первой. Добавив амперсанд (&) в конец строки, вы получите возможность запустить ее выполнение в фоновом режиме. Задействовав команды fg, bg и jobs наряду с. различными управляющими кодами, вы сможете переводить команды в фоновый или приоритетный режим.

Используя соответствующую команду из показанной далее последовательности, можно запустить графический редактор GIMP из окна терминала. Следом приведены сочетания клавиш и команды для *приостановки и запуска процесса, а также перевода его в приоритетный или фоновый режим*:

```
$ gimp                                    Запустить выполнение gimp в фоновом режиме
<Ctrl+Z>                                  Приостановить процесс и перевести его
                                          в фоновый режим
[1]+ Stopped gimp
$ bg 1                                    Снова запустить процесс в фоновом режиме
$ fg 1                                    Продолжить выполнение процесса в приоритетном
                                          режиме
gimp
<Ctrl+C>                                  Завершить процесс
```

Следует отметить, что процессу, переведенному в фоновый режим, присваивается идентификационный номер задания (в данном случае он равен 1). Указав перед этим номером знак процента (например, II) либо просто введя номер вместе с командой (например, fg 1), вы обозначите определенный фоновый процесс для команд bg и fg. При выполнении нескольких фоновых заданий в текущем интерпретаторе команд можно **использовать команду jobs для управления фоновыми заданиями:**

```
$ jobs                                    Отобразить фоновые задания в случае с текущим интерпретатором
                                          команд
[1]     Running          gimp &
[2]     Running          xmms &
```

```
[3] -   Running          gedit &
[4] +   Stopped          gtal i
$ jobs -1                Отобразить PID и информацию о каждом задании
[1]   31676 Running          gimp &
[2]   31677 Running          xmrns &
[3] - 31683 Running          gedit &
[4] + 31688 Stopped          gtali
$ jobs -1 %2             Отобразить информацию только о задании %2
[2] 31677 Running            xmms &
```

Процессы, выполняющиеся в примерах использования jobs, могут быть активизированы после того, как вы зайдете (с использованием ssh) в удаленную систему. Некоторые из них может потребоваться **запустить как удаленные GUI-приложения на локальном рабочем столе.** Активизировав эти процессы в фоновом режиме, вы сможете сделать так, что у вас будет выполняться сразу несколько приложений, при этом ассоциированных с текущим интерпретатором команд. Как только процесс запустится, можно **обособить его от текущего интерпретатора команд с помощью команды disown:**

```
$ disown £3 Обособить задание %3 от текущего интерпретатора команд
$ disown -a Обособить все задания от текущего интерпретатора команд
$ disown -h Оградить все задания от сигнала HUP, отправленного
            текущему интерпретатору команд
```

Обособив процесс посредством disown, вы сможете закрыть интерпретатор команд, не завершив этот процесс.

ПРИМЕЧАНИЕ----------

Если при использовании fg, bg или disown вы не укажете, на какой процесс необходимо воздействовать, то команда повлияет на текущее задание. Рядом с текущим заданием стоит знак «плюса» (+).

Команды fg и bg позволяют манипулировать запущенными процессами, переводя их в фоновый или приоритетный режим. Другой подход к манипулированию выполняющимися командами заключается в отправке сигналов непосредственно соответствующим процессам. Распространенный способ отправки сигналов запущенным процессам — использование команд kill и kiHall.

Завершение процессов и отправки им сигналов

Вы можете приостанавливать или вносить изменения касательно запущенных процессов путем отправки им сигналов. Такие команды, как kill и kill all, позволяют отправлять выбранные вами сигналы активным процессам, которые, как видно из названий этих команд, часто оказываются сигналами для завершения процессов.

Сигналы представлены номерами (9, 15 и т. д.) и строками (SIGKILL, SIGTERM и т. п.). В табл. 9.2 перечислены стандартные сигналы, которые вы можете отправлять процессам в Linux.

Таблица 9.2. Стандартные сигналы для отправки процессам

Номер сигнала	Имя сигнала	Описание
1	SIGHUP	Отключение от терминала или завершение управляющего процесса
2	SIGINT	Прерывание посредством клавиатуры
3	SIGQUIT	Завершение посредством клавиатуры
4	SIGILL	Недопустимая инструкция
6	SIGABRT	Сигнал аварийного завершения, отправляемый при выполнении функции abort
8	SIGFPE	Исключение в операции с плавающей точкой
9	SIGKILL	Сигнал завершения
11	SIGSEGV	Недопустимое обращение к памяти
13	SIGPIPE	Нерабочий канал (ни один процесс не осуществляет чтение из канала)
14	SIGALRM	Сигнал таймера, отправляемый системным вызовом alarm
15	SIGTERM	Сигнал завершения
30, 10, 16	SIGUSR1	Определяемый пользователем сигнал 1
31, 12, 17	SIGUSR2	Определяемый пользователем сигнал 2
20, 17, 18	SIGCHLD	Завершение или приостановка дочернего процесса
19, 18, 25	SIGCONT	Возобновление выполнения процесса, если он был приостановлен
17, 19, 23	SIGSTOP	Приостановка процесса
18, 20, 24	SIGTSTP	Сигнал приостановки, введенный в терминале
21, 21, 26	SIGTTIN	Ввод с терминала для фонового процесса
22, 22, 27	SIGTTOU	Вывод на терминал для фонового процесса

Команда kill позволяет отправлять сигналы процесса согласно их идентификаторам или номерам заданий, а команда kill all дает возможность посылать сигналы процессам в соответствии с именами команд. Примеры:

```
$ kill    28665                 Отправить сигнал SIGTERM процессу с PID 28665
$ kill       -9 4895            Отправить сигнал SIGKILL процессу с PID 4895
$ kill -SIGC0NT 5254            Возобновить выполнение приостановленного
                                процесса (с PID 5254)
$ kill       13                 Завершить процесс, представленный заданием 23
$ kill all       spamd          Завершить все запущенные на данный момент
                                демоны spamd
$ kill all -SIGHUP sendmail     Сделать так. чтобы процессы sendmail
                                осуществили повторное чтение конфигурацион-
                                ных файлов
```

К сигналу SIGKILL (-9), часто используемому безрассудными администраторами-новичками, следует прибегать как к последнему средству. Он не позволяет

целевому процессу аккуратно завершиться, а заставляет его резко оборваться, что может привести к потере или повреждению данных, обрабатываемых этим процессом. Сигнал SIGHUP первоначально использовался в UNIX-системах как индикатор того, что терминал отключался от мэйнфрейма (например, если модем, используемый для подключения по телефонной линии путем дозвона, разрывал соединение). Вместе с тем процессы-демоны, к примеру sendmail и httpd, реализовывались для перехвата сигналов SIGHUP с целью индикации того, что эти процессы должны выполнить повторное чтение конфигурационных файлов.

Обособление процессов от текущего интерпретатора команд

Если хотите, чтобы процесс продолжил выполняться даже после того, как вы обособите его от сеанса текущего интерпретатора команд, вы можете добиться этого несколькими способами. Можно воспользоваться командой nohup для **запуска процесса таким образом, чтобы он не реагировал на сигнал отбоя:**

```
$ nohup updatedb &                    Выполнить updatedb без возможности прерывания
# nohup nice -9 gcc hello.c &  Запустить непрерывное выполнение дсс с более
                                      высоким приоритетом
```

Применение nohup отличается от выполнения команды с использованием одного амперсанда, поскольку nohup продолжит работать, даже если вы выйдете из интерпретатора команд, с помощью которого было запущено ее выполнение.

Команда nohup широко использовалась в эпоху медленных процессоров и соединений по телефонной линии через модем (благодаря ее применению не приходилось поддерживать дорогостоящее подключение, дожидаясь, например, окончания длительного компилирования). Кроме того, сегодня, используя такие инструменты, как screen (его описание вы найдете в гл. 13), вы можете сохранить сеанс интерпретатора команд активным даже если будет разорвано сетевое подключение к этому интерпретатору команд, поэтому nohup уже не применяется так часто, как раньше.

Планирование запуска процессов

Команды, ассоциированные с инструментом cron, можно использовать для обеспечения запуска выполнения той или иной команды в определенное время (включая текущий момент), чтобы она при этом была обособлена от текущего интерпретатора команд. Команда at **запускает выполнение команды в заданное вами время:**

```
$ at now +1 min          Запустить выполнение команды через 1 минуту
at> updatedb
at> <Ctrl+D> <E0T>
job 5 at Sat Mar 2 20:37:00 2013
$ at  teatime          Запустить выполнение      командысегодня  в 4 часа дня
$ at  now +5 days      Запустить выполнение      командычерез 5  дней
$ at  06/25/13         Запустить выполнение      командысейчас,  25 июня 2013года
```

Другой способ запустить выполнение команды, которая будет обособлена от текущего интерпретатора команд, заключается в использовании batch. С помощью команды batch вы можете **обеспечить запуск выполнения команды, как только процессор будет готов** (средняя нагрузка окажется ниже . 8):

```
$ batch                    Немедленно запустить выполнение команды
at> find /mnt/isos | grep jpg$ > /tmp/mypics
at> <Ctrl+D> <E0T>
```

Обратите внимание, что за командами at и batch следует вторичное приглашение at>. Укажите команду, которую нужно выполнить, в приглашении и нажмите клавишу Enter. После этого вы сможете продолжить вводить команды. Закончив, нажмите комбинацию Ctrl+D, находясь на строке, чтобы поместить в очередь команды, введенные вами для выполнения.

Когда команды будут указаны, вы сможете **проверить очередь заданий at, сконфигурированных** на выполнение, воспользовавшись командой atq:

```
$ atq
11          Thu Sep 5 21:10:00 2013 a chris
10          Sat Aug 24 21:10:00 2013 a chris
8           Fri Aug 23 20:53:00 2013 a chris
```

Обычные пользователи могут просматривать только свои задания at, помещенные в очередь. Обладатель прав суперпользователя имеет возможность просматривать задания at любого пользователя. Если вы захотите **удалить задание at из очереди,** воспользуйтесь командой atrm:

```
$ atrm 11                  Удалить задание номер.11
```

Команды at и batch предназначены для помещения в очередь команд, которые должны выполниться однократно. Вы можете использовать инструмент cron для **обеспечения повторного выполнения команд.** Эти команды добавляются к заданиям cron, запуск выполнения которых запланирован в файлах crontab. Есть только один системный файл crontab (/etc/crontab). Кроме того, любой пользователь может создать персональный файл crontab, который обеспечит запуск выполнения команд в выбранное этим пользователем время. Чтобы **создать персональный файл crontab,** введите следующее:

```
$ crontab -e               Создать персональный файл crontab
```

Команда crontab -e откроет ваш файл crontab (или создаст новый), используя текстовый редактор vi. Вот примеры различных записей, которые вы можете добавить в файл crontab:

```
15 8 * * Mon,Tue,Wed.Thu,Fri mail chris < /var/project/stats.txt
* * 1 1,4,7.10 * find / | grep .doc$ > /var/sales/documents.txt
```

В первом примере записей в файле crontab пользователю chris посылается почтовое сообщение, в которое было направлено содержимое / var/project/stats.txt. Команда mail выполняется с понедельника по пятницу в 8:15 утра. Во втором примере в первый день января, апреля, июля и октября выполняется команда find для поиска всех файлов с расширением .doc, после чего получившийся список направляется в/var/sales/documents.txt.

В последней части каждой записи в файле crontab содержится выполняемая команда. В первых пяти полях представлены время и дата запуска выполнения команды. Эти поля, слева направо, таковы: минуты (от 0 до 59), часы (от 0 до 23), день месяца (от 0 до 31), месяц (от 0 до 12 или Jan (январь), Feb (февраль), Маг (март), Apr (апрель), May (май), Jun (июнь), Jul (июль), Aug (август), Sep (сентябрь), Oct (октябрь), Nov (ноябрь) или Dec (декабрь)) и день недели (от 0 до 7 или Sun (воскресенье), Моп (понедельник), Тие (вторник), Wed (среда), Thu (четверг), Fri (пятница) или Sat (суббота)). Наличие звездочки (*) в поле означает, что в нем можно указать любое соответствующее значение.

Если выполненная команда сгенерирует вывод и вы не направите его в файл или другой команде, то по умолчанию этот вывод будет послан пользователю, выполнившему команду cron. Чтобы указать определенный адрес электронной почты для приема вывода, можно добавить строку MAILT0 в файл crontab. Например:

MAI LT0=chri s@exampl e. corn

Вот другие параметры, которые можно использовать в сочетании с командой crontab:

```
# crontab -eu chris          Редактировать файл crontab другого пользователя
                             (только для суперпользователя)
$ crontab -1                 Показать содержимое вашего файла crontab
15 8 * * Mon,Tue,Wed.Thu,Fri mail chris < /var/project/stats.txt
* * 1 1,4,7,10 * find / | grep ,doc$ > /var/sales/documents.txt
$ crontab -r                 Удалить ваш файл crontab
```

Традиционный способ конфигурирования системных заданий cron заключался в добавлении их в системный файл crontab. Несмотря на то что это по-прежнему является вариантом, к которому можно прибегнуть, Ubuntu обеспечивает лучший способ создания ежечасно, ежедневно, еженедельно или ежемесячно выполняемых заданий cron путем **ассоциирования команды, выполнение которой необходимо запустить, с каталогом cron.** Просто создайте сценарий, который хотите выполнить. Затем скопируйте его в каталог /etc/cron .hourly, /etc/cron. daily, /etc/cron .weekly или /etc/cron, monthly. После этого выполнение нужной команды будет запускаться с периодичностью, соответствующей имени каталога (/etc/cron. hour! у — ежечасно, /etc/cron. dai 1 у — ежедневно, /etc/cron. weekly — еженедельно, /etc/cron.monthly — ежемесячно).

В качестве альтернативы cron можно использовать anacron. С помощью anacron вы можете, как и при использовании cron, настраивать периодичность выполнения команд, однако anacron чаще всего применяется на компьютерах, которые не постоянно включены. Если команда не была запущена, поскольку компьютер был выключен во время предполагающегося выполнения команды, то при следующем включении компьютера утилита anacron выполнит заданную команду.

Планирование запуска процессов реального времени

Порядок, в котором выполняются процессы, а также количество внимания, уделяемого им центральными процессорами, определяется планировщиком процессов.

В большинстве случаев модификация значений nice обычных процессов (тех, которые выполняются в соответствии со стандартной политикой SCHED_OTHER) является единственным изменением, которое вам необходимо внести, чтобы присвоить процессу более высокий или более низкий приоритет в плане использования ресурсов центрального процессора (см. подраздел «Изменение приоритетов процессов в плане использования ресурсов центрального процессора с помощью nice» данной главы).

Однако в некоторых редких случаях вам может потребоваться назначить для процесса специальную политику. Одни политики приводят к уменьшению количества внимания, которое центральные процессоры уделяют процессам (например, некоторым долго выполняющимся пакетным заданиям, не являющимся критическими по времени), а другие, называемые политиками реального времени, ставят тот или иной процесс выше остальных, выполняющихся в системе в соответствии с политикой SCHED_OTHER.

ПРИМЕЧАНИЕ----------

Не следует присваивать процессам приоритеты реального времени в производственной системе, если только вы не уверены в том, что делаете. Эти процессы получат преимущество перед системными и могут сделать вашу систему нестабильной или даже непригодной для использования, если вы допустите ошибку, присваивая приоритеты реального времени.

К процессорным политикам, используемым в Linux, относятся следующие.

- SCHEDJ3THER — при этой политике запуск всех процессов планируется с использованием базовых систем с разделением времени Linux, если только не задан другой тип планирования. Порядок обработки определяется исходя из значений nice.
- SCHED_BATCH — процессам, для которых назначена данная политика, отдается меньше предпочтения при планировании исходя из того, что они потребуют много ресурсов центрального процессора, но смогут подождать, пока им будет уделено внимание. Запуск процессов при использовании этой политики планируется с учетом их значений nice (хотя им все равно будет уделено меньше внимания, чем процессам, выполняющимся в соответствии с SCHED_OTHER).
- SCHED_IDLE — процессам при использовании этой политики присваивается очень низкий приоритет (даже если для процессов, которые будут выполняться в соответствии с SCHED_OTHER или SCHED_BATCH, заданы значения nice, равные 19).
- SCHEDJ4F0 — любой процесс, для которого назначена эта политика реального времени, получит преимущество перед любым другим запущенным процессом, выполняющимся в соответствии с SCHED_OTHER, SCHED_BATCH или SCHED_IDLE. Мри зовании SCHED_FIFO запуск процессов планируется по принципу очередности.
- SCHED_RR — то же самое, что политика SCHED_FIFO для процессов реального времени, за исключением того, что процессы, для которых она назначена, совместно используют ресурсы центрального процессора. Другими словами, нет необходимости, чтобы процесс, выполняющийся в соответствии с SCHED_RR, завершился перед тем, как другой процесс, для которого назначена политика SCHED_RR, сможет запуститься.

Используя команду chrt, можно узнать, какая из пяти политик планирования Linux назначена для той или иной команды. Вы также можете задействовать chrt с целью смены политики планирования для запущенного процесса либо запуска процесса с использованием определенной политики планирования.

Дополнительную информацию о планировании запуска процессов можно найти на MAN-страницах, посвященных chrt и sched_setschedule.

Превращение процесса в процесс реального времени

Приведенные далее команды позволяют найти идентификатор обычного процесса, который вам необходимо превратить в процесс реального времени, вывести на экран сведения о назначенной для него политике планирования и сменить ее:

```
$ pidof firefox                 Найти идентификатор процесса firefox
16563
$ chrt -p 16563                 Показать сведения о политике планирования,
                                назначенной для процесса
pid 16563[1]s current scheduling policy: SCHED_OTHER
pid 16563's current scheduling priority: 0
$ chrt -m                       Показать сведения о политиках планирования
                                и диапазоны значений
SCHED_OTHER min/max priority    : 0/0
SCHED_FIFO min/шах priority     : 1/99
SCHED_RR min/max priority       : 1/99
SCHED_BATCH min/max priority    : 0/0
SCHED_IDLE min/max priority     : 0/0
$ sudo chrt -r -p 20 16563 Превратить в процесс реального времени с SCHED_RR
$ chrt -p 16563                 Отобразить информацию о новом процессе реального
                                времени с SCHED_RR
pid 16563’s current scheduling policy: SCHEDJRR
pid 16563's current scheduling priority: 20
$ sudo chrt -f -p 10 16563 Превратить в процесс реального времени с SCHED_FIFO
```

Запуск процесса как процесса реального времени

Чтобы запустить процесс, наделив его при этом определенным приоритетом, вы можете воспользоваться командой chrt, когда будете запускать этот процесс в первый раз.

```
$ sudo chrt -r 50 firefox       Запустить процесс, назначив для него
                                SCHED_RR и присвоив приоритет со значением 50
$ sudo chrt -f 99 gcc hell o.c  Запустить процесс, назначив для него
                                SCHED_FIFO и присвоив приоритет со значением 99
```

Резюме

Наблюдение и работа с процессами, запущенными в вашей Linux-системе, — важные действия, позволяющие убедиться, что ваша система функционирует эффективно. Используя такие команды, как ps и top, вы можете просматривать информацию

о запущенных процессах. Вы также можете прибегнуть к команде рдгер для поиска и вывода на экран определенных процессов.

С помощью таких команд, как nice и renice, можно изменять рекомендованные приоритеты, которыми наделены выбранные активные процессы. Если процесс запущен, то вы можете изменить то, как он выполняется, либо завершить его, отправив ему сигнал посредством команды kill или ki 11 all.

После запуска выполнения команды из текущего интерпретатора команд вы сможете перевести процесс этой команды в фоновый (bg) или приоритетный (fg) режим. Можно также приостанавливать и повторно запускать процесс с использованием различных управляющих кодов.

Планировать запуск выполнения команды позднее вы можете с помощью команды at или batch. Чтобы обеспечить повторное выполнение команды через заданные интервалы времени, можно прибегнуть к инструменту cron или anacron. Для выполнения команд, наделяемых разными приоритетами при планировании, используйте chrt.

10 Управление системой

В этой главе:

- проверка уровня использования памяти с помощью команд free, top, vmstat и slabtop;
- просмотр уровня использования ресурсов центрального процессора посредством iostat, dstat и top;
- мониторинг устройств для хранения посредством iostat, vmstat и 1 sof;
- модификация даты/времени с помощью date, hwclock, cal и NTP;
- изменение поведения загрузчика GRUB;
- монтирование диска в оперативной памяти для начальной инициализации;
- управление уровнями выполнения с использованием runlevel и init;
- О' добавление, удаление и просмотр информации о службах посредством chkconf i g и service;
- остановка работы системы с помощью reboot, halt и shutdown;
- проверка и изменение настроек драйвера ядра с использованием 1 smod, modi nfo и modprobe;
- просмотр настроек аппаратного обеспечения посредством Ispci, dmidecode и hdparm.

Без четкого управления потребности Linux-системы иногда превышают имеющиеся ресурсы. Возможность мониторинга активности системы (уровня использования памяти, ресурсов центрального процессора и других устройств) в динамике по времени может помочь убедиться, что у компьютера достаточно ресурсов для выполнения необходимых вам операций. Аналогичным образом управление другими аспектами системы, например драйверами устройств, которые она использует, и процессом загрузки, позволит избежать проблем с производительностью и сбоев.

Эта глава разбита на разделы, связанные со способами управления Ubuntu и другими Linux-системами. В первом разделе вы найдете сведения, которые помогут в мониторинге ресурсов (вычислительной мощности, устройств и памяти). В следующем описываются проверка и настройка системных часов. Далее рассматривается процесс загрузки и последующих уровней выполнения. В последних разделах содержится описание, как работать с ядром и соответствующими

драйверами устройств, а также способы просмотра информации об аппаратных компонентах компьютера.

Мониторинг ресурсов

Ubuntu, Debian и другие Linux-системы выполняют полезную работу, которая заключается в отслеживании, что они делают. Вы, если захотите, найдете массу информации об использовании центрального процессора, жестких дисков, виртуальной памяти и других ресурсов компьютера.

Вы можете заглянуть туда, куда ядро Linux сохраняет поступающие в реальном времени данные о системе, открыв файлы в файловой системе /proc для непосредственного просмотра (приложение B). Однако в качестве альтернативы для просмотра информации о том, как в вашей системе используются виртуальная память, процессор, устройства для хранения и сетевые интерфейсы, можно задействовать команды.

Существуют команды, которые позволяют осуществлять мониторинг вашей системы. Эта книга не является MAN-страницей, поэтому я разбил приведенные далее разделы по темам (мониторинг уровня использования памяти, ресурсов центрального процессора, а также устройств для хранения), а не командам для выполнения соответствующих операций (top, vmstat и iostat).

ПРИМЕЧАНИЕ

Некоторые приложения, описанные в далее, устанавливаются в Ubuntu по умолчанию, располагаясь в таких пакетах, как, например, procps. Однако, чтобы использовать iostat или sar, вам потребуется установить пакет sysstat посредством следующей команды:

```
$ sudo apt-get install sysstat
```

Память

Мало что способно свести на нет уровень производительности системы быстрее, чем исчерпание памяти. Такие команды, как free и top, дают вам возможность просматривать основную информацию о том, как используются оперативную память и пространство подкачки. Команда vmstat позволяет получить подробные сведения об использовании памяти и может выполняться постоянно. Команда slabtop способна выводить на экран информацию, сколько памяти потребляет ядро (slab-кэш).

Команда free обеспечивает самый быстрый способ **узнать, какое количество памяти** используется в вашей системе. Она показывает общий объем оперативной памяти (Mem:) и пространства подкачки (Swap:) наряду с объемом, который задействуется на данный момент. Далее приведены примеры использования команды free:

```
$ free
```

Показать объем используемой памяти в килобайтах (-k по умолчанию)

```
$ free -m                 Показать объем используемой памяти в мегабайтах
             total       used       free shared buffers     cached
Hem:          3920       2105       1815 0 550                 942
-/+ buffers/cache:        612       3307
Swap:         4059          0       4059
$ free -b                 Показать объем используемой памяти в блоках
                Total        used                  free shared buffers cached
Hem: 4110852096  2207576064          1903276032 0 576901120 987951104
-/+ buffers/cache:  642723840     3468128256
Swap:4257214464           0     4257214464
$ free -mt                Показать объем используемой памяти И общий объем памяти
                          (Swap + Чет)
             total       used       free     shared    buffers     cached
       Mem:  3920        2075       1845          0        550        942
-/+ buffers/cache:        582       3337
       Swap: 4059           0       4059
       Total: 7980       2075       5905
$ free -g                 Показать объем используемой памяти в гигабайтах
                          (с округлением в меньшую сторону)
             total       used       free     shared    buffers     cached
Mem: 3                       2          1          0          0          0
-/+ buffers/cache:           0          3
Swap: 3                      0          3
$ free -s 5               Постоянно отображать объем используемой памяти каждые
                          5 секунд
```

Чтобы избежать напрасной траты оперативной памяти и ускорить работу приложений, Linux задействует как можно больше неиспользуемой иным образом оперативной памяти для дискового кэша, поэтому первая строка вывода команды free, в которой часто отображается небольшой объем свободной оперативной памяти, может вводить в заблуждение. Рекомендую уделять больше внимания второй строке вывода, в которой демонстрируется объем оперативной памяти, действительно доступный приложениям. В этом примере он равен 3307 Мбайт:

```
-/+ buffers/cache:        612       3307
```

Вот один из способов приблизительно определить, сколько памяти потребуется в системе. Сядьте за другой компьютер, работающий под управлением Ubuntu, и откройте все приложения, которые, как вам кажется, вы будете запускать одновременно. Введите команду free с параметром -t (free -t), чтобы увидеть значение количества используемой памяти. Затем убедитесь, что в новой системе общий объем памяти по крайней мере не меньше этого значения (желательно, чтобы бóльшая его часть или он весь приходился на оперативную память).

Команда top обеспечивает возможность наблюдения за запущенными на данный момент процессами с сортировкой их по уровню использования ресурсов центрального процессора или памяти (применение top для наблюдения за запущенными процессами рассматривалось выше (см. гл. 9)). Однако вы также можете

использовать top для **наблюдения за уровнем потребления памяти экранно-ориентированным образом.** Вот пример:

```
$ top
top - 8:43:04 up 3 days, 5:00. 3 users,
        load average: 0.21, 0,18, 0.11
Tasks: 188 total, 1 running. 187 sleeping 0 stopped, 0 zombie
Cpu(s): 17.4iSus. 3.3£sy, 0.0Ы, 79.3ftid,
O.Oftwa, O.O&hi, O.OXsi, 0.0%st
Mem: 4014504k total, 1950936k used, 2063568k free, 248148k buffers
Swap: 4157436k total,            0k used, 4157436k free, 1016924k cached
  PID USER  PR NI  VIRTRES  SHR        SIfcCPU&MEMTIME+   COMMAND
11062 chris 20  0  1636m47m  23mS    26   1.2 0:20.19 totem
 2646 chris 20  0  1198m88m  28mS     7   2.3 6:36.98 compiz
 1750 root  20  0   209m 29m  12m S    4   0.8 4:33.56 Xorg
```

Для завершения выполнения top нажмите клавишу Q. Как и free, команда top генерирует вывод, содержащий сведения об общем объеме используемой оперативной памяти (Мет:) и пространства подкачки (Swap:). Команда top является экранно-ориентированной и обеспечивает постоянный мониторинг, поэтому вы можете наблюдать за изменением уровня использования памяти с обновлением сведений каждые три секунды (по умолчанию). Нажмите комбинацию клавиш **Shift+M** во время выполнения top, чтобы вывести на экран информацию о запущенных процессах, которые будут рассортированы по уровню использования памяти (благодаря чему вы сможете узнать, какие процессы потребляют наибольшее количество памяти). Самым полезным столбцом для анализа уровня использования памяти тем или иным процессом является RES, в котором отображается фактически потребляемый процессом объем физической памяти, также называемый *резидентным размером.* Информация в столбце ЯМЕМ базируется на этом резидентном размере.

Сегодня на большинстве новых компьютеров устанавливаются многоядерные центральные процессоры; из строки Cpu(s): можно узнать средний уровень использования их ресурсов. Чтобы увидеть уровень потребления ресурсов каждого ядра центрального процессора по отдельности, задействуйте клавишу с цифрой 1. Строки Cpи будут выглядеть примерно так:

```
CpuO : 17.7£us, 5.3$sy, 0.0$ni, 44.7%id.
        31.7&wa, 0.0%hi, 0.7£si. O.O&st
Cpul : 16.4%us, 4.4£sy, O.Obi, 46.4%id,
        32.8ftwa, O.O&hi. O.Ksi, O.O^st
```

Для получения более детального представления статистики по виртуальной памяти задействуйте команду vmstat. С ее помощью вы можете **узнать уровень использования памяти за определенный период времени,** например с момента последней перезагрузки или за другой выборочный временной отрезок. В приведенном далее примере показано использование vmstat для вывода на экран статистики с обновлением ее каждые три секунды:

```
$ vmstat 3
procs -..... .....memory-------  --swap----------io-----  --system--  --cpu--
 r   b swpd free buff cache        si    so bi      bo     in cs us sy id wa st
 1   0 7740 32488 3196 148360     0      00       1      26 3876 85 15 0 0 0
```

```
 1  1 8388   7428 3204 151472   0 216     0 333    30 3200 82 18  0  0  0
 1  0 13316  8148 2980 146968   0 4980    4 5121   79 3846 77 23  0  0  0
 2  0 32648  7472 2904 148488   0 6455    3 6455   90 3644 83 17  0  0  0
 2  0 47892  8088 2732 144208   0 5085    9 5220   79 3468 84 16  0  0  0
 1  0 57948  7680 2308 134812   0 3272   12 3296   69 3174 77 23  0  0  0
 3  0 58348  7944 1100 123888  21 144    25 275    26 3178 86 14  0  1  0
 2  0 66116  7320 568  120280  11 2401   20 2403   51 3175 84 16  0  0  0
 3  0 81048  7708 648  119452  53 4852  796 4984  123 1783 86 13  0  1  0
```

Для завершения выполнения vmstat нажмите сочетание клавиш **Ctrl+C.** В приведенном выше примере использования vmstat охвачен 30-секундный интервал времени, за который было запущено более 100 приложений. Обратите внимание: когда объем свободной памяти уменьшается с 32 488 до 7428 Кбайт (оперативная память заполняется), данные начинают направляться в область подкачки (216 в столбце so). Область подкачки располагается на жестком диске, поэтому, как видите, количество блоков, записываемых на дисковое устройство (Ьо), растет по мере увеличения выгрузки. Вы также можете наблюдать, что объем используемого пространства подкачки возрастает в столбце swpd.

Центральный процессор тоже подвергается нагрузке в приведенном выше примере, при этом время простоя не отображается (id 0). Следует отметить, что, когда какому-либо приложению требуется обратиться к области подкачки (см. последние три строки вывода), процессору приходится дожидаться окончания двух операций ввода/вывода (wa 1).

Вот еще некоторые параметры для использования в сочетании с vmstat.

$ vmstat -S m *Отобразить вывод в мегабайтах из расчета. что 1 Мбайт = 1000 Кбайт*

$ vmstat -S M *Отобразить вывод в мегабайтах из расчета, что 1 Мбайт = 1024 Кбайт*

$ vmstat -S k *Отобразить вывод в килобайтах из расчета, что 1 Мбайт = 1000 Кбайт*

$ vmstat -S К *Отобразить вывод в килобайтах из расчета, что 1 Мбайт = 1024 Кбайт*

$ vmstat -n 2 10 *Генерировать вывод каждые две секунды с повторением операции 10 раз*

$ vmstat -s | less *Отобразить счетчики событий и статистику по памяти*

$ vmstat -S M -s | less *Отобразить статистику в мегабайтах*

```
      3920 M total memory
      1953 M used memory
       972 M active memory
       726 M inactive memory
      1966 M free memory
       261 M buffer memory
       993 M swap cache
      4059 M total swap
         0 M used swap
      4059 M free swap
    137803 non-nice user cpu ticks
      9412 nice user cpu ticks
     52746 system cpu ticks
  62828527 idle cpu ticks
...
```

В предыдущем, примере показан вывод, содержащий различную статистику (-s) касательно памяти в мегабайтах (-S M), который покажется вам удобнее для получения общего представления об уровне использования памяти. Другие примеры демонстрируют, как отобразить вывод vmstat в мегабайтах и килобайтах (как в маркетинговом, так и в техническом исчислении). Далее параметр -п 2 10 дает vmstat указание повторять операцию через заданные интервалы времени в секундах (2) ограниченное количество раз (10).

С помощью таких команд, как ps и top, можно узнать, сколько памяти потребляет каждое приложение в вашей системе. Однако у ядра имеется собственный кэш для отслеживания своих ресурсов, называемый *slab-кэшем ядра.* Вы можете воспользоваться командой vmstat для **отображения статистики касательно slab-кэша ядра** (из /proc/slabinfo):

```
# vmstat -m | less          Постраничный просмотр информации о slab-кэше ядра
Cache                       Num  Total  Size  Pages
udf_inode_cache \           120    120   664     24
dm_crypt_io                   0      0   152     26
nf_eonntrack_expect           0      0   240     17
nfsd4_openowners              0      0   392     20
nfs_read_data                 0      0   768     21
nfs_inode_cache               0      0  1008     16
fscache_cookie_jar           51     51    80     51
rpc_inode_cache              19     19   832     19
ext2_inode_cache            105    105   752     21
ip6_dst cache                50     50   320     25
UDPLITEv6                     0      0  1024     16
...
```

Информация о slab-кэшах включает имя каждого кэша; количество объектов, активных в кэше этого типа; общее количество объектов, доступных в кэше данного типа; размер кэша (в байтах); количество страниц для каждого кэша. Вы можете **отобразить информацию о slab-кэше ядра в виде экранно-ориентированного представления** (как при использовании команды top), прибегнув с slabtop:

```
# slabtop
      Active / Total Objects IX used) : 562520 / 566736 (99.3%)
 Active / Total Slabs (% used) : 20103 / 20103 (100.0%)
 Active / Total Caches (% used) : 74 / 110 (67.3%)
 Active / Total Size (% used) : 152005.08K / 153123.88K (99.3%)
 Minimum / Average / Maximum Object: 0.01K / 0.27K / 8.00K
0BJS ACTIVE USE OBJ SIZE SLABS OBJ/SLAB CACHE SIZE NAME
144768 144400  99%    0.10K  3712      39     14848K buffer_head
142170 141903  99%    0.19K  6770      21      7080K dentry
 57168  57168 100%    0.86K  3176      18     50816K ext4_inode_cache
 40898  40726  99%    0.61K  1573      26     25168K proc_inode_cache
 23713  23611  99%    0.17K  1031      23      4124K vm_area_struct
 19824  19824 100%    0.14K   708      28      2832K sysfs_dir_cache
```

Вывод slabtop обновляется каждые три секунды. По умолчанию slab-кэши сортируются по количеству объектов (первый столбец), содержащихся в каждом из

кэшей. Нажав клавишу C, можно отсортировать по размеру кэша (как было показано в предыдущем примере).

Центральный процессор

Перегрузка центрального процессора — еще один очевидный фактор, на который следует обратить внимание при поиске причин проблем с производительностью системы. Рассмотренная выше команда vmstat позволяет выводить на экран базовую статистику уровня использования ресурсов центрального процессора (активность пользователя и системы, время простоя и ожидания завершения выполнения запросов ввода/вывода, а также время, отнятое у виртуальной машины). Однако команда iostat (из пакета sysstat) позволяет генерировать более подробные отчеты об уровне использования ресурсов центрального процессора.

Вот два примера выполнения iostat для **вывода на экран отчета об уровне использования ресурсов центрального процессора:**

```
$ iostat ■ •c 3          Генерировать статистику касательно
                         центрального процессора каждые 3 секунды
                         (запуск приложений)
Linux 3.2 . 0-38-genenc (ubuntutb) 05/20/2013      _x86_64_ (2 CPU)
avg-cpu:  luser   Inice Isystem liowait Isteal      lidle
           0.26    0.01 0.09 0.05 0.00              99.58
avg-cpu:  luser   Inice Isystem liowait Isteal      lidle
          22.64    0.00 8.18 2.36 0.00              66.82
avg-cpu:  luser   Inice Isystem liowait Isteal      lidle
          32.01    0.00 4.14 19.90 0.00             43.95
$ iostat -c 3            Генерировать статистику касательно центрального
                         процессора каждые 3 секунды
                         (копирование файлов)
Linux 3.2.0-38-generic (ubuntutb)          05/20/2013     _x86_64_ (2 CPU)
avg-cpu:  luser   Inice Isystem liowait     Isteal   lidle
           0.37    0.01 0.11 0.15           0.00     99.35
avg-cpu:  luser   Inice Isystem liowait     Isteal   lidle
          21.69    0.00 6.10 65.42           0.00      6.78
avg-cpu:  luser   Inice Isystem liowait     Isteal   lidle
          21.48    0.00 5.87 63.76           0.00      8.89
avg-cpu:  luser   Inice Isystem liowait     Isteal   lidle
          19.35    0.00 7.30 68.76           0.00      4.58
```

В первом из приведенных выше примеров использования iostat система сначала находится в состоянии покоя, но затем запускается несколько приложений. Как видите, после первой строки, в которой отображается средний уровень использования ресурсов центрального процессора с момента последней загрузки системы, показано, что бóльшая часть обработки, связанной с запуском приложений, выполняется в пространстве пользователя.

Во втором примере файлы большого размера копируются с одного жесткого диска на другой. В результате на ожидание завершения выполнения запросов ввода/вывода (%iowa it) тратится больший процент времени.

В приведенных далее примерах команда iostat задействуется для **вывода на экран отчетов об уровне использования ресурсов центрального процессора с временными метками:**

```
$ iostat -c -t          Отобразить временную метку в отчете об уровне
                        использования ресурсов центрального процессора
Linux 3.2.0-38-generic (ubuntutb) 05/20/2013 _x86_64_ (2 CPU)
05/20/2013 07:08:20 PM
avg-cpu: luser      Inice  Isystem liowait      Isteal      lidle
           0.40     0.01     0.12     0.21        0.00      99.26
$ iostat -c -t 2 10 Генерировать статистику каждые 2 секунды
                        с повторением операции 10 раз
```

Команда dstat (из пакета dstat) доступна в качестве альтернативы iostat для **просмотра информации об уровне использования ресурсов центрального процессора** (а также других сведений, связанных с производительностью). Одним из преимуществ команды dstat перед другими инструментами является то, что она дает более точные результаты в единицах измерения, используемых ею при выводе сведений на экран (например, в килобайтах или мегабайтах), а также задействует цвета для дифференцирования данных. Чтобы установить пакет dstat, введите следующее:

```
$ sudo apt-get install dstat
```

Вот пример применения dstat для отображения информации об уровне использования ресурсов центрального процессора:

```
$ dstat -t -c 3 Постоянно отображать информацию об уровне использования
                ресурсов центрального процессора с временными метками
---- system.....--total -cpu-usage-------
          time | usr  sys  idl  wai  hiq siq
 20-05 19:14:38|  27    5   59    9    0   1
20-05 19:14:411  34    7   55    3    0   1
20-05 19:14:441 35    6   54    4    0   0
20-05 19:14:47|  21    4   74    1    0   0
20-05 19:14:50|  19    3   76    1    0   0
20-05 19:14:53|  19    4   77    0    0   0
20-05 19:14:561  18    3   79    0    0   0
```

В этом примере вывод включает значения даты/времени, основанные на начале отсчета времени (-t) с момента составления отчета об уровне использования ресурсов центрального процессора (.-c), который генерируется каждые три секунды (3). Этот отчет будет постоянно составляться, пока вы не остановите его (нажав комбинацию клавиш **Ctrl+C).**

Чтобы выяснить, обработка каких именно процессов занимает бóльшую часть времени, можно воспользоваться командой top. Введите top, а затем нажмите **Shift+P** для сортировки процессов по уровню использования ресурсов центрального процессора (эго порядок сортировки по умолчанию):

```
$ top                   Отобразить информацию о запущенных процессах с сортировкой
                        по уровню использования ресурсов центрального процессора
top - 9:18:29 up 3 days, 16:36, 3 users, load avg: 0.29, 1.05, 1.73
Tasks: 192 total, 1 running, 191 sleeping, 0 stopped, 0 zombie
```

```
Cpu(s): 14.6&us,2.2lsy.0.0&ni.82.7%id.0.3Xwa,0.0ЖЫ .0.2%si ,0.021st
Mem: 4014504k total, 3305876k used, 708628k free, 360896k buffers
Swap: 4157436k total, 264k used 4157172k free,2229972k cached
  PID USER   PR NI VIRT RES SHR SSCPU&MEMTIME+  COMMAND
11338 chris  20  01637m 48m  24m S   22   1,2 10:48.97 totem
 2646 chris  20  01214m 88m  28m S    6   2.3 17:55.98 compiz
 1750 root   20  0 206m 32m  13m S    4   0.8  8:53.63 Xorg
```

В полном выводе отображалось бы намного больше процессов, отсортированных по текущему уровню использования ресурсов центрального процессора (столбец XCPU). В приведенном выше примере проигрыватель Totem (22&), оконный менеджер combiz (6£) и графический сервер X (4Я) потребляют бо́льшую часть ресурсов центрального процессора. Если вы решите завершить процесс Totem, нажмите клавишу K, а затем укажите идентификатор этого процесса (11338) и сигнал с номером 9 (если по какой-то причине у вас не получится закрыть окно Totem обычным образом).

Информацию о самом центральном процессоре вы сможете получить, заглянув прямо в файл /proc/cpuinfo. Пример:

```
$ cat /proc/cpuinfo          Показать информацию о центральном процессоре из /proc
Processor       : 0
vendor_id       : Genuinelntel
cpu family      : 6
model           : 23
model name      : Pentium(R) Dual-Core CPU E6300 @ 2.80GHz
...
Processor       : 1
vendor_id       : Genuinelntel
cpu family      : 6
model           : 23
model name      : Pentium(R) Dual-Core CPU E6300 @ 2.80GHz
...
Flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 cl flush dts acpi mmx fxsr sse sse2 ss ht
tm pbe syscall nx lm constant_tsc arch_perfmon pebs bts rep_good
nop! aperfmperf pni dtes64 monitor ds_cpl vmx est tm2 ssse3 cxl6
xtpr pdcm xsave lahfjm dtherm tpr_shadow vnmi flexpriority
bogomi ps       : 5585.95
cl flush size   : 64
...
```

Конкретно в этом компьютере установлен центральный процессор с двумя ядрами, в силу чего информация повторяется в случае с каждым из них. В данной ситуации относительно каждого ядра центрального процессора вы можете видеть название его поставщика и модели, а также тактовую частоту (2,8 Ггерц). Любопытной деталью касательно центрального процессора являются флаги, представляющие параметры, которые он поддерживает. Для использования некоторых функциональных возможностей Ubuntu необходимо, чтобы определенные расширения центрального процессора, ассоциированные с этими флагами, были активизированы. Например, для работы паравиртуализированных гостевых операционных

систем на основе технологии виртуализации Хеп должен быть задан флаг рае. Для запуска полностью виртуализированных гостевых операционных систем центральному процессору необходимо поддерживать либо флаговое расширение vmx (для процессоров Intel), либо svm (для процессоров AMD).

Аналогичная информация о центральном процессоре (-ах) собирается системой в самом начале загрузки, а увидеть эти сведения можно, взглянув на начало вывода команды dmesg.

Мониторинг устройств для хранения

Основную информацию о пространстве для хранения, доступном файловым системам Linux, можно просмотреть с помощью команд du и df (см. гл. 7). Однако если вы захотите увидеть подробные сведения о том, как работают устройства для хранения, помогут такие команды, как vmstat и iostat.

Некоторую часть приводившегося ранее вывода i ostat аналогичного рода можно использовать для **выяснения, есть ли узкие места при чтении с диска и записи на него.** Вот пример:

```
$ iostat 3                    Провести тест чтения/записи для каждого диска
Linux 3 .2.0-38-generic (ubuntutb)          05/20/2013 x86_64_ (2 CPU)
avg-cpu: luser Inice Isystem liowait Osteal          lidle
                0.48 0.    02 0.13        0.23 0.00   99.15
Device:          tps    кВ read/s     кВ wrtn/s    кВ read    кВ wrtn
Sda             0.45         4.38        "26.63    1400105    8517944
dm-0            0.68         4.37         26.62    1396897    8517196
dm-1            0.00         0.00          0.00       1152        736
scdO            0.39        24.94          0.00    7978782          0
sdb             0.01         0.01          1.49       2073     476784
Device:          tps     кВ read/s     кВ wrtn/s    кВ read    кВ wrtn
Sda            17.00      1909.33          0.00       5728          0
dm-0           17.00      1909.33          0.00       5728          0
dm-1            0.00         0.00          0.00          0          0
scdO            0.00         0.00          0.00          0          0
sdb            46.33         0.00       5462.67          0      16388
avg-cpu: luser Inice Isystem          liowait Osteal          lidle
                0.00 0.    00 0.34      74.41 0.00          25.25
Device:          tps     кВ read/s     кВ wrtn/s    кВ read    кВ wrtn
Sda            11.00      1257.33          0.00       3772          0
dm-0           11.00      1257.33          0.00       3772          0
dm-1            0.00         0.00          0.00          0          0
scdO            0.00         0.00          0.00          0          0
sdb            42.33         0.00       5021.33          0      15064
```

В первой части вывода iostat приведен средний уровень использования ресурсов центрального процессора с момента последней перезагрузки. В следующей части отражена обработка, выполненная при копировании большого объема данных с первого диска (sda) на второй (sdb). Устройство dm-0 в действительности представляет собой логический том LVM на sda, связанный с корневой файловой системой.

Большие значения iowait указывают, что узким местом в системе является скорость диска при операциях ввода/вывода. Другими словами, установка диска, обеспечивающего более высокую скорость записи, могла бы сильнее повысить производительность, чем установка более мощного центрального процессора.

Команда vmstat также позволяет выводить на экран статистику касательно дисков. Вот пример использования vmstat для **отображения информации о скорости чтения с диска и записи на него:**

```
$ vmstat -d  Отобразить статистику по операциям чтения,
             записи и ввода/вывода для дисков
disk- ---------- reads---------- ---------- writes---------- --I0---
         total  merged    sectors      ms   total  merged sectors     ms  cur  sec
...
sda      80736   21757  9133194 1075864  101513   67798 27543352  97896    0  702
dm-0    101740       0  9127290 1568064  153265       0 27522840  24100    0  702
dm-1       288       0     2304    1680    2561       0    20488   6688    0    3
srO     195504  697318  2496238 7629216       0       0        0      0    0 2129
sdb        457     126     4714     848   31786     307  7492448 075688    0  767
```

В системе Linux в этом примере имеется два жестких диска (sda и sdb). Отображается количество секторов, успешно прочитанных и записанных при их использовании. Видно также, сколько секунд было потрачено на операции ввода/вывода (10) во время работы с этими дисками. Кроме того, вы можете узнать, выполняются ли какие-либо операции ввода/вывода, а также просмотреть информацию об операциях чтения/записи для выбранных дисковых разделов. Пример:

```
$ vmstat -p sdal  Отобразить статистику по операциям чтения/записи в случае
                  с дисковым разделом
sdal              reads read sectors writes requested writes
                    296         2172       10            24
```

К сожалению, приведенная выше команда не работает с разделами md устройств в программных RAID-массивах, разделами lvm и некоторыми устройствами в аппаратных RAID-массивах, зависящими от драйверов.

Чтобы выяснить, **сколько файлов и каталогов открыто на данный момент на устройствах для хранения,** можете воспользоваться командой 1 sof. Она особенно полезна, когда вы пытаетесь демонтировать файловую систему, которая будет продолжать твердить, что занята. Вы сможете проверить, какой файл препятствует демонтированию, и решить, следует ли завершить соответствующий процесс, который держит этот файл открытым, и форсировать демонтирование файловой системы. Вот пример использования 1 sof:

```
$ 1sof | less          Показать процессы, которые держат открытыми файлы/каталоги
COMMAND  PID   USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
Init       1   root  cwd    DIR  252.0     4096    2 /
init       1   root  rtd    DIR  252,0     4096    2 /
init       1   root  txt    REG  252,0    16719 3445 /sbin/init
...
bash     166   chris cwd    DIR    8,17    4096    2 /mnt/a
```

Первые показанные в примере файлы не позволяют завершить процесс init (первый процесс, запущенный в системе). Файлы, закрыть которые не позволяют системные процессы (например, udevd) и демоны (например, sshd и syslogd), привязаны к процессу init. В конце списка отображаются файлы, закрыть которые не позволяют отдельные пользователи (заинтересованные, возможно, в том, чтобы вы не могли размонтировать раздел диска).

ПРИМЕЧАНИЕ----------

Вы можете столкнуться с ограничениями прав доступа, если сначала не введете команду sudo:

```
$ sudo lsof I less
```

После отображения вывода lsof на экране вам потребуется узнать название файла или каталога, который открыт (NAME), и имя команды, держащей его открытым (COMMAND), а также идентификатор процесса этой выполняющейся команды (PID). В приведенном выше примере текущий каталог запущенного интерпретатора команд bash содержит каталог, открытый в файловой системе, которую нужно было демонтировать (/mnt/a). Когда не получается демонтировать файловую систему, причиной этого часто оказывается интерпретатор команд bash со своим текущим каталогом в файловой системе, который препятствует демонтированию. Вместо передачи по канату вывода lsof команде less или grep вы можете воспользоваться несколькими другими способами для получения посредством вывода 1 sof необходимых сведений:

$ lsof -c bash	*Показать файлы, открытые интерпретатором команд bash*
$ lsof -d cwd	*Показать каталоги, открытые как текущие каталоги*
$ lsof -u chris	*Показать файлы и каталоги, открытые пользователем chris*
$ lsof /mnt/sdal	*Показать все, что открыто в файловой системе /mnt/sdal*
$ lsof +d /mnt/sdal/dx/	*Показать все. что открыто и располагается в /mnt/sdal/dx/*

Как уже отмечалось ранее, вам может потребоваться задействовать команду sudo с целью получения прав доступа суперпользователя для просмотра всего вывода команды lsof.

Управление временем

Поддержка точного времени в вашей системе Linux критически важна, чтобы система функционировала должным образом. Компьютер, работающий под управлением Linux, поддерживает время двумя путями — посредством использования системных часов (по ним Linux следит за временем) и аппаратных часов (применяются для установки системного времени при загрузке Linux).

Системное время используется для задания временных меток, когда речь идет о создании файлов, времени выполнения процессов и обо всем остальном, где дата и время имеют значение. Системное время можно просматривать и устанавливать вручную (с помощью команды date) либо автоматически (посредством службы ntpd).

Аппаратные часы соединены с CMOS-батарейкой, расположенной на материнской плате, и питаются от нее, когда компьютер выключен. На аппаратных часах время устанавливается с помощью команды hwclock.

Есть множество других инструментов, которые можно использовать для работы с временем в Linux-системах, например инструменты для проверки времени разными способами наподобие cl ockdi ff (дает возможность определить разницу между временем на часах компьютеров) и uptime (позволяет узнать, сколько времени работает ваша система).

Изменение даты/времени с помощью графических инструментов

К имеющимся в Ubuntu и других Linux-системах графическим инструментам для изменения даты, времени и часового пояса, которые вы можете использовать в своей системе, относится окно **Time & Date Properties** (Свойства времени и даты) (чтобы открыть его, выберите **Time & Date Properties** (Свойства времени и даты) на панели управления Ubuntu). Это окно также можно использовать, чтобы задействовать *NTP* (Network Time Protocol — сетевой протокол времени) с целью автоматической синхронизации даты и времени системы Linux по сети.

Окно **Time & Date Properties** (Свойства времени и даты) позволяет сохранить сделанные вами настройки и выбранные варианты. Во время загрузки Ubuntu система выполнит чтение этих настроек, чтобы задать часовой пояс в зависимости от того, используется ли UTC-время.

Часовой пояс для системы Linux задается на основе содержимого файла /etc/ local time. Вы можете немедленно задать новый часовой пояс, скопировав файл, представляющий нужный пояс, из подкаталога /usr/share/zoneinfo. Например, чтобы сменить текущий часовой пояс на использующийся для Америки/Чикаго, введите следующее:

$ sudo cp /usr/share/zoneinfo/America/CIricago /etc/Toca!time

Это также можно сделать путем создания символьной ссылки:

$ sudo In -s /usr/share/zoneinfo/America/Chicago /etc/localtime

Для перманентной смены часового пояса задайте в окне **Time & Date Properties** (Свойства времени и даты) нужный вам, например New York. United States.

Отображение и настройка ваших системных часов

Команда date является первичным командным интерфейсом, который можно использовать для просмотра и изменения настроек даты и времени, если вы решите не делать этого автоматически с помощью NTP. Вот примеры использования команды date для **отображения даты и времени** разными способами:

$ date *Отобразить текущую дату/время/часовой пояс*
Sun Jul 7 20:13:00 EDT 2013

$ date '+IA IB Id IG' *Отобразить день, месяц, число, год*
Sunday July 7 2013
$ date '+The date today is IF.' *Добавить текст к выводимой на экран дате*
The date today is 2013-07-07.
$ date --date='4 weeks' *Отобразить дату, которая наступит через четыре недели от сегодняшнего дня*

Sun Aug 4 20:16:09 EDT 2013
$ date --date='8 months 3 days' *Отобразить дату, которая наступит через восемь месяцев и 3 дня от сегодняшнего дня*

Mon Mar 10 20:27:46 EDT 2014
$ date --date='4 JuV +IA *Отобразить день, на который выпадает 4 июля*
Thursday

Главный интерес в этом разделе для нас представляет время, но пока мы ведем речь о датах, рассмотрим, как команда cal может использоваться в качестве инструмента, позволяющего быстро **отобразить даты по месяцам.** Примеры:

$ cal *Показать календарь, который охватывает текущий месяц (выделив сегодняшнее число)*

```
     July 2013
Su Mo Tu We Th Fr Sa
      1  2  3  4  5  6
 7    8  9 10 11 12 13
14   15 16 17 18 19 20
21   22 23 24 25 26 27
28   29 30 31
```

$ cal 2013 *Показать календарь на весь год*

```
                                2013
      January               February               March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
          1  2  3  4  5                  1  2                  1  2
 6  7  8  9 10 11 12   3  4  5  6  7  8  9   3  4  5  6  7  8  9
13 14 15 16 17 18 19  10 11 12 13 14 15 16  10 11 12 13 14 15 16
20 21 22 23 24 25 26  17 18 19 20 21 22 23  17 18 19 20 21 22 23
27 28 29 30 31        24 25 26 27 28        24 25 26 27 28 29 30
                                            31
```

. . .

$ cal -j *Показать юлианский календарь (с нумерацией с 1 января)*

```
          July 2013
 Su  Mo  Tu  We  Th  Fr  Sa
    182 183 184 185 186 187
188 189 190 191 192 193 194
195 196 197 198 199 200 201
202 203 204 205 206 207 208
209 210 211 212
```

Команду date также можно использовать для **изменения системной даты и времени.** Вот пример:

$ sudo date 081215212013 *Установить дату и время: 12 августа, 2:21 ночи, 2013 год*

Mon Aug 12 15:21:00 EDT 2013

$ sudo date --set='+7 minutes' *Перевести часы на семь минут вперед*
Mon Aug 12 15:28:39 EDT 2013
$ sudo date --set='-l month' *Установить дату/время предыдущего месяца*
Fri Jul 12 15:29:11 EDT 2013

При следующей загрузке Ubuntu системное время сбросится до значения согласно времени на ваших аппаратных часах (или данным сервера NTP, если активизирована служба NTP). При следующем выключении компьютера время на аппаратных часах сбросится до значения системного времени с целью сохранить его, пока компьютер выключен. Для изменения времени на системных часах вы можете использовать команду hwclock.

Отображение и настройка аппаратных часов

Любой пользователь может просматривать настройки аппаратных часов посредством команды hwcl ock, однако для изменения этих настроек необходимо обладать привилегиями суперпользователя. Чтобы задействовать hwclock для **просмотра текущего времени на аппаратных часах вашего компьютера,** введите следующее:

$ hwclock -г *Отобразить текущие настройки аппаратных часов*
Моп 20 May 2013 08:14:55 PM EDT -0.312931 seconds

Даже если аппаратные часы будут настроены согласно UTC-времени, команда hwclock по умолчанию отобразит местное время. Если системное время будет отличаться от времени на аппаратных часах (например, если вы прибегли к некоторым из примеров использования команды date, приведенных выше), можете **сбросить время на системных часах до значения на аппаратных** следующим способом:

$ sudo hwclock -hctosys *Сбросить время на системных часах до значения на аппаратных*

Аналогичным образом, если время на ваших аппаратных часах окажется неточным (например, если вы заменили CMOS-батарейку на материнской плате), можно **сбросить время на аппаратных часах до значения на системных,** как показано далее:

 # hwclock -systohc *Сбросить время на аппаратных часах до значения на системных*

Со временем ваши аппаратные часы могут сбиваться. Эти часы имеют тенденцию сбиваться на одинаковое количество времени каждый день, поэтому команда hwcl ock в состоянии отслеживать данное отклонение (что она и делает в файле /etc/adjtime). Вы можете **откорректировать время на аппаратных часах,** взяв за основу файл adjtime, следующим образом:

$ sudo hwclock -adjust *Откорректировать время на аппаратных часах с учетом отклонения*

Чтобы **установить на аппаратных часах определенное время,** воспользуйтесь параметром --set. Вот пример:

$ sudo hwclock --set --date="8/12/13 18:22:00" *Установить новую дату/время*

В данном примере для аппаратных часов задаются следующие настройки: 12 августа 2003 года, 6:22 вечера. Это обновление не сразу повлияет на системные часы.

Использование NTP для установки даты/времени

Если вы не сконфигурируете NTP для установки времени в своей системе при первой инсталляции системы Linux, то сможете сделать это позднее, *активизировав службу* ntpd. Чтобы установить и сконфигурировать службу ntpd (демон ntpd), вам потребуется всего лишь несколько команд.

Для активизации соответствующей службы из интерпретатора команд воспользуйтесь приведенной далее строкой:

$ sudo apt-get install ntp *Установить пакет ntp при необходимости и запустить службу*

Служба ntpd использует информацию из файла /etc/ntpd. conf. Вы можете указать, например, имя и IP-адрес сервера времени, к которому следует обращаться с запросами касательно времени.

После установки пакета ntp компьютер превратится в сервер времени, прослушивающий UDP-порт 123. Если только у вас нет специфических потребностей (и собственного GPS-устройства или атомных часов), то запуск ntpd на локальном компьютере может оказаться как напрасной тратой ресурсов, так и угрозой безопасности, поэтому некоторые системные администраторы предпочитают использовать ntpdate (часто как ежедневно выполняемое задание стоп) для установки системного времени посредством NTP:

```
$ sudo ntpdate pool.ntp.org
08 July 20:21:19 ntpdate[16121]:
step time server 72.8.140.200 offset 31565336.613857 sec
```

Если вы попытаетесь задействовать ntpdate во время выполнения ntpd, то увидите следующее сообщение об ошибке:

```
$ sudo ntpdate pool.ntp.org
08 July 00:37:00 ntpdate[9695]: the NTP socket is in use, exiting
```

Необходимо отметить, что команда ntpdate была признана устаревшей и в будущем исчезнет.

Управление процессом загрузки

При первом включении компьютера BIOS (Basic Input/Output System — базовая система ввода/вывода) заглядывает в его настройки порядка загрузки для определения, где можно найти операционную систему для загрузки. Обычно, если загру-

зонный носитель не вставлен в привод для работы со съемными носителями (CD, DVD и т. д.), то базовая система ввода/вывода осуществляет поиск MBR (Master Boot Record — главная загрузочная запись) на первом загрузочном жестком диске. На данном этапе в большинстве Linux-систем управление процессом загрузки передается *загрузчику*.

Сегодня в Ubuntu и фактически в большинстве Linux-систем загрузчиком, используемым по умолчанию, является *GRUB (GRand Unified Boot Loader — главный унифицированный загрузчик)*. GRUB можно настроить на загрузку не только вашей системы Linux, но и любых других операционных систем, установленных на жестких дисках (Windows, BSD и пр.). GRUB также может включать загрузочные параметры в случае с каждой запускаемой операционной системой для настройки процесса загрузки вроде активизации или деактивизации поддержки аппаратного обеспечения определенного типа.

В версии Ubuntu 9.10 и выше в качестве загрузчика используется вторая версия GRUB (GRUB 2). GRUB 2 намного более конфигурируемый и быстрый, чем предыдущая версия (часто называемая GRUB Legacy). Однако GRUB 2 сложнее освоить.

После выбора в загрузчике системы Linux для запуска он загружает ядро. Затем возникает затруднение: ядру необходимо смонтировать корневую файловую систему на жесткий диск. Для этого требуются соответствующие драйверы устройств для хранения (модули ядра блочных устройств), а эти драйверы находятся на самом жестком диске!

Чтобы выйти из замкнутого круга, загрузчик монтирует небольшой диск в оперативной памяти для начальной инициализации (initrd), содержащий модули блочных устройств (наряду с различными инструментами для чтения корневой файловой системы). После этого в дело вступает процесс init, который запускает системные службы, исходя из *уровня выполнения,* заданного для системы.

В следующих разделах описываются команды для изменения конфигурации загрузчика, а также сценарии запуска и уровни выполнения, ассоциированные с вашей системой Linux.

Понятие загрузчика GRUB

Если загрузчик GRUB был сконфигурирован при первой установке Ubuntu, то настройки для него находятся в файле /boot/grub/grub.cfg. В отличие от основных конфигурационных файлов в случае с GRUB Legacy (menu. 1st или grub.cfg), вам не следует напрямую редактировать файл grub.cfg.

Содержимое grub.cfg перезаписывается содержимым файлов из каталога /etc/ grub. d при каждой установке нового ядра или выполнении вами команды update-grub.

Файл /boot/grub/grub. cfg содержит настройки для управления поведением загрузчика GRUB и тем, как должны загружаться модули, а также другую информацию, необходимую для запуска операционной системы. Однако для нас наибольший интерес представляют записи, которые обеспечивают загрузку ядра для запуска Ubuntu (или другой операционной системы). В приведенном далее при-

мере показаны записи в файле grub.cfg, которые используются для создания элементов в меню загрузки GRUB:

```
menuentry 'Ubuntu. with Linux 3.2.0-38-generic' --class ubuntu
                --class gnu-linux --class gnu --class os
{
  recordfail
  gfxmode $1i nux_gfx_mode
  insmod gzio
  insmod partjnsdos
  insmod ext2
  set root='(hdO,msdos1)'
  search --no-floppy --fs-uuid
              --set=root 0d2494ad-62e6-43d4-8aff-780630dd65d
  linux /vmlinuz-3.2.0-38-generic root=/dev/mapper/ubuntutb-root ro
  initrd /initrd.img-3.2.0-38-generic
}
```

В этом примере продемонстрирована строка menuentry для загружаемой операционной системы ('Ubuntu, with Linux 3.2.0-38-generic'), используемая как идентификатор версии ядра, которое загружается при выборе соответствующего элемента меню. Информация в строке menuentry идентифицирует запись как операционную систему Ubuntu Linux. В этой секции также задается разрешение экрана окна терминала (gfxmode) и загружаются модули для распаковки сжатых файлов (gzio), обработки структуры разделов DOS (part_msdos) и поддержки файловых систем типа ext2, необходимых для использования раздела /boot (ext2).

Корневой раздел располагается на первом жестком диске (hdO), а его таблица разделов отформатирована как раздел DOS (part msdos). Однако критически важными данными являются идентификатор ядра (/vml i nuz-3.2.0-38-generi c), информация об устройстве, представляющем корневую (/) файловую систему (/dev/mapper/ubuntub-root), и сведения о местоположении диска в оперативной памяти для начальной инициализации (/initrd.img-3.2.0-38-generic).

Изменение конфигурации загрузчика GRUB

Файл grub. cfg создается с использованием файлов из каталога /etc/grub. d, поэтому для изменения конфигурации загрузчика GRU B вы можете выполнить одну из приведенных далее операций вместо того, чтобы напрямую редактировать данный файл:

- **файл /etc/grub.d/40_custom** — добавьте в него записи, которые хотите включить в свой файл grub.cfg;
- **файл /etc/default/grub** — в нем содержатся настройки, например паузу какой длительности должен выдержать GRUB перед тем, как запускать процесс загрузки (вы задаете значение времени для его прерывания).

Модифицировав файлы /etc/grub.d/40_custom и /etc/default/grub, вы сможете сделать так, чтобы соответствующие изменения вступили в силу, либо установив новое ядро, либо выполнив команду update-grub.

Действия, которые не обеспечивались ранее при использовании загрузчика GRUB Legacy, теперь автоматически совершаются при применении GRUB 2. Например, если GRUB обнаружит Windows или другую операционную систему на любом из ваших дисков, он добавит запись в меню GRUB, чтобы при перезагрузке у вас была возможность выбрать операционную систему, которую вы хотите запустить.

Управление запуском и уровнями выполнения

После запуска ядра оно передает управление системой процессу init. Он оказывается первым из запущенных в системе (PID 1) и управляет запуском других процессов на основе содержимого каталога /etc/init/, уровнем выполнения по умолчанию и сценариями init, которые заданы для запуска на этом уровне.

Значением уровня выполнения по умолчанию в системе Ubuntu обычно является 2 (на основе строки DEFAULT_RUNLEVEL в файле /etc/init/rc-sysini t. conf). Эту величину можно изменить путем добавления нужного значения в строку env RUNLEVEL= в файле rc-sysinit.conf.

Ubuntu 12.04 LTS поддерживает как устаревшие сценарии init System V (расположенные в каталоге /etc/init.d/), так и новые конфигурационные файлы в Upstart-стиле (которые находятся в каталоге /etc/init). Будучи администратором, вы можете изменять значение уровня выполнения по умолчанию, текущего уровня, а также то, какие службы запускаются и останавливаются на каждом уровне выполнения.

Большинство Linux-администраторов не трогает основные параметры запуска и сосредотачивается на том, какие службы активизированы или деактивизированы на выбранном уровне выполнения. В этом разделе приведены команды для работы со сценариями инициализации системы и изменения значений уровней выполнения.

Являясь суперпользователем, вы сможете прибегнуть к команде run! evel, чтобы выяснить значение текущего уровня выполнения:

```
$ runlevel Отобразить значения текущего и предыдущего уровней выполнения
N 2
```

Система в этом примере загружалась непосредственно на уровне выполнения 2, поэтому предыдущий уровень отсутствует (N). Чтобы изменить значение текущего уровня выполнения, можете воспользоваться командой init. Например, если вы захотите остановить работу GUI-интерфейса и сети, а затем перейти в однопользовательский режим для сопровождения системы, введите следующее:

```
$ sudo init 1 Изменить значение уровня выполнения
              на 1 (однопользовательский режим)
```

В данном примере значение текущего уровня выполнения изменяется со значения предыдущего (которое в данном случае было равно 2) на 1 (благодаря чему система переходит в однопользовательский режим). Для управления службами

можно задействовать команды update-rc.d и service. Например, чтобы немедленно запустить службу NTP, вы можете ввести следующее:

```
$ sudo /etc/init.d/ntp start Немедленно запустить службу NTP
Starting NTP server ntpd                    [ OK ]
```

Каждой службе System V сопутствует сценарий интерпретатора команд в каталоге /etc/init.d. Вы можете передавать параметр start или stop любой из служб. Формат таков:

```
sudo /etc/init.d/service_to_control start
sudo /etc/init.d/service_to_control stop
```

Большинство сценариев в каталоге /etc/init. d поддерживает параметры start или stop, при этом некоторые из них также принимают и другие параметры. Вот как можно использовать соответствующие команды для запуска и остановки служб:

```
$ /etc/init.d/ntp Показать сведения об использовании (без параметров)
Usage: /etc/init.d/ntp {start|stop|restart[try-restart|
    force-re1oad|status}
$ sudo /etc/init.d/ntp restart              Перезапустить службу NTP
 * Stopping NTP server ntpd                 [ OK]
 * Starting NTP server ntpd                 [ OK]
$ sudo /etc/init.d/ntp try-restart Перезапустить службу NTP. если она
                                   уже запущена
* Stopping NTP server ntpd                  [ OK]
* Starting NTP server ntpd                  [ OK]
$ sudo /etc/init.d/ntp force-reload Перезагрузить конфигурационный файл
                                   в случае с демоном
* Stopping NTP server ntpd                  [ OK ]
* Starting NTP server ntpd                  [ OK ]
$ sudo /etc/init.d/ntp status               Проверить, запущена ли служба NTP
* NTP server is running.
$ sudo /etc/init.d/ntp stop                 Остановить службу NTP
* Stopping NTP server ntpd                  [ OK ]
```

Любой из сценариев init, содержащихся в /etc/init .d, можно запустить таким способом, однако не все сценарии поддерживают любые из продемонстрированных выше параметров. Вместе с тем большинство сценариев init будет выводить на экран сведения об их использовании без параметров (как показано в первом примере в приведенном выше коде).

Предыдущие команды немедленно запускают службу сценария уровня выполнения, но для автоматического запуска службы во время загрузки или изменения значения уровня выполнения можно использовать и команду update-rc.d. Кроме того, большинство установочных сценариев для служб автоматически активизирует службу при следующей загрузке. Используя update-rc.d, вы можете активизировать и деактивизировать службы, исходя из уровня выполнения в каждом случае. Для службы, которая вообще не сконфигурирована на запуск, можно использовать параметр defaults:

```
$ sudo update-rc.d ntp defaults             Активизировать службу NTP
```

Вы можете прибегнуть к команде init для изменения значения уровня выполнения на любую величину, в том числе init 0 (выключение) и init 6 (перезагрузка), однако есть специфические команды для остановки работы Linux. Преимущество таких команд, как halt, reboot, poweroff и shutdown, заключается в том, что они поддерживают параметры, позволяющие вам останавливать работу некоторых элементов перед выключением. Например:

ВНИМАНИЕ----------

Не используйте приведенные далее команды, если на самом деле не собираетесь выключать систему, особенно если речь идет об удаленной системе.

```
$  sudoreboot                 Перезагрузить компьютер
$  sudohalt -n                Не синхронизировать жесткие диски перед выключением
$  sudohalt -h                Перевести жесткие диски в режим ожидания перед
                              остановкой работы
$  sudoshutdown10             Выключить через 10 минут после предупреждения
                              пользователей
$  sudoshutdown  -r10         Перезагрузить через 10 минут после предупреждения
                              пользователей
$  sudoshutdown 10'Bye!'      Отправить пользователям 'Bye' ('До свидания')
                              перед выключением
```

Помимо команд reboot и init 6, вы также можете нажать давно используемое в сфере персональных компьютеров сочетание клавиш Ctrl+Alt+Del, чтобы перезагрузить компьютер.

Прямо в ядро

В целом при запуске ядра в системе Linux вам не придется делать с ним слишком многое. Однако есть инструменты для проверки используемого ядра, а также просмотра информации о том, как оно запустилось. Кроме того, если что-либо пойдет не так или возникнет необходимость в обеспечении дополнительной поддержки ядра, вы можете воспользоваться доступными инструментами, чтобы справиться с соответствующими задачами.

Чтобы выяснить, какое ядро запущено в системе на данный момент, введите следующее:

```
$ uname -r                    Отобразить имя выпуска ядра

3.2.0-38-generic
$ uname -a                    Отобразить всю доступную информацию о ядре
Linux ubuntutb 3.2.0-38-generic #61-Ubuntu SMP
   Tue Feb 19 12:18:21 UTC 2013 x86_64 x86_64 x86_64 GNU/Linux
```

При запуске ядра сообщения о происходящих событиях помещаются в его кольцевой буфер. Чтобы отобразить содержимое кольцевого буфера ядра, вы можете воспользоваться командой dmesg:

```
$ dmesg 11 ess
[      0.000000] Linux version 3.2.0-38-generic (buildd@akateko)
```

```
 (gcc version 4.6.3 (Ubuntu/Lina ro 4.6.3-lubuntu5) ) #61-Ubuntu SMP
  Tue Feb 19 12:18:21 UTC 2013 (Ubuntu 3.2.0-38.61-generic 3.2.37)
[    0.000000] Command line: B00T_IMAGE=/vmlinuz-3.2.0-38-generic
   root=/dev/mapper/ubuntutb-root ro
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
[    0.000000]   AMD AuthenticAMD
[    0.000000]   Centaur CentaurHauls
  ...
```

Если этот буфер окажется заполнен, он может больше не содержать начало записанной информации. В этом случае воспользуйтесь less /var/1 og/dmesg.

Вы можете найти дополнительную информацию об обработке, связанной с ядром, в файлах /var/log, в частности в файле messages. Эти файлы можно просматривать постранично:

```
$ sudo less /var/1og/messages Постраничный просмотр файла messages
May 23 21:36:05 ubuntutb kernel: [12567.769587] Inbound
 IN=eth0 0UT= MAC=40:61:86:c3:35:f8:48:5b:39:e7:95:72:08:00
 SRC=192.168.0.131 DST=192.168.0.141 LEN=60 TO
 S=0x00 PREC=0x00
 TTL=63 10=58788 DF PR0T0=TCP SPT=49304 DPT=22 WINDOW=14600
 RES=0x00 SYN URGP=0
```

В идеале все аппаратные компоненты, входящие в состав вашего компьютера, должны быть детектированы и сконфигурированы с использованием соответствующих Linux-драйверов. Однако в некоторых ситуациях может обнаружиться ненадлежащий драйвер либо необходимый драйвер может оказаться недоступен. Для таких случаев в Linux предусмотрены способы просмотра загружаемых модулей ядра, а также добавления новых модулей ядра в систему.

Команда 1 smod позволяет просматривать имена загруженных модулей, сведения об их размере, а также о том, какие другие модули их используют. Пример:

```
$ Ismod
Module                  Size Used by
 ...
ext2                   73795  :
dm_crypt               23125  0
iptable_nat            13229  0
nf_nat                 25891  2  i pt_MASQUERADE,i ptable_nat
nf_conntrack_ipv4      19716  9  iptable_nat,nf_nat
 ...
snd_hda_codec_realtek   224173  1
sndjidajntel           33773  3
snd_hda_codec         127706  2  snd_hda_codec_rea1tek,snd Jida_i ntel
snd_hwdep              17764  1  snd_hda_codec
Ip                     17799  0
parport                46562  3  parport_pc.ppdev,lp
```

Чтобы получить больше информации об определенном модуле, можете воспользоваться командой modinfo. Вот пример:

```
$ modinfo parport_pc
filename:
    /1i b/modules/3.2.0-38-generic/kernel/drivers/parport/parport_pc.ko
license:        GPL
description:    PC-style parallel port driver
author:           Phil Blundell. Tim Waugh, others
...
```

Если вы решите, что нужно **добавить или удалить загружаемый модуль,** чтобы некий аппаратный компонент системы работал должным образом, воспользуйтесь командой modprobe. Ее также можно задействовать для **просмотра всех доступных модулей и их удаления.** Примеры:

```
$ modprobe -1 | grep c-qcam          Показать c-qcam из списка модулей
kernel/dri vers/medi a/vi deo/c-qcam.ko.
$ sudo modprobe c-qcam               Загрузить модуль для Color QuickCam
$ sudo modprobe -r c-qcam            Удалить модуль для Color QuickCam
```

ПРИМЕЧАНИЕ----------

Вам, возможно, доводилось слышать о команде insmod для загрузки модулей. В то время как insmod загружает отдельный модуль, modprobe также позволяет загружать любые дополнительные модули, необходимые для функционирования нужного. В силу этого modprobe оказывается предпочтительной командой в большинстве случаев.

Вы можете **управлять параметрами ядра во время работы системы,** используя команду sysctl. Вы также можете перманентно добавлять параметры в файл /etc/ sysctl. conf, чтобы они могли запускаться как группа или при каждой перезагрузке. Взгляните на следующие примеры:

```
$ sudo sysctl -a | less                  Показать все параметры ядра
kernel.panic = O
kernel.core_uses_pid = 0
...
$ sudo sysctl kernel.hostname            Показать значение определенного параметра
$ sudo sysctl -p                         Загрузить параметры из /etc/sysctl.conf
$ sudo sysctl -w kernel.hostname=joe     Задать значение для kernel.hostname
```

Как уже отмечалось ранее, если вы захотите перманентно изменить какие-либо параметры ядра, потребуется добавить их в файл /etc/sysctl .conf. Настройки параметров в этом файле представлены в форме *«параметр = значение».* Параметры, значения для которых были заданы с использованием sysctl -w, вернутся к своим значениям по умолчанию при следующей перезагрузке системы.

Просмотр информации об аппаратном обеспечении

Если вам потребуется узнать больше об аппаратном обеспечении своего компьютера в целом, можете воспользоваться приведенными далее командами. Команда Is pci **отображает информацию о PCI-устройствах,** установленных в компьютере:

```
$ lspci                    Показать информацию об аппаратных PCI-компонентах
00:00.0 Host bridge: Intel Corporation 4 Series Chipset
    DRAM Controller (rev 03)
00:02.0 VGA compatible controller: Intel Corporation 4 Series
    Chipset Integrated Graphics Controller (rev 03)
00:02.1 Display controller: Intel Corporation 4 Series Chipset
    Integrated Graphics Controller (rev 03)
00:1b.0 Audio device: Intel Corporation N10/ICH 7 Family
    High Definition Audio Controller (rev 01)
...
$ lspci -v List Показать более подробную информацию об аппаратных
                    PCI-компонентах
$ lspci -vv List Показать еще более подробную информацию об аппаратных
                    PCI-компонентах
```

Используя команду dmi decode, вы можете **отображать информацию об аппаратных компонентах компьютера,** включая сведения о том, какие параметры поддерживаются в базовой системе ввода/вывода. Пример:

```
$ sudo dmidecode | less Показать информацию об аппаратных компонентах
SMBI0S 2.5 present.
41 structures occupying 1535 bytes.
Table at Ox0009F400Handle 0x0000, DMI type 0, 24 bytes.
BIOS Information
  Vendor: MSI
  Version: 130
  Release Date: 10/06/2013
    ...
Processor Information
  Socket Designation:
  SOCKET 775_M/B
        Type: Central Processor
        Family: Unknown
        Manufacturer: Intel
        ID: 7A 06 01 00 FF FB EB BF
        Version: Pentium(R) Dual-Core CPU £6300        @ 2.80GHz
```

Вы можете использовать команду hdparam для **просмотра и изменения информации, связанной с вашим жестким диском.**

ВНИМАНИЕ----------

Хотя просмотр информации о параметрах жестких дисков безопасен, изменение некоторых настроек этих накопителей может причинить им вред.

Вот несколько примеров отображения информации о ваших жестких дисках:

```
$ sudo hdparm /dev/sda Отобразить настройки жесткого диска
                        (с интерфейсом SATA или SCSI)
/dev/sda:
  Multcount     =16 (on)
  I0_support    = 1 (32-bit)
  Readonly      = 0 (off)
```

```
 Readahead = 256 (on)
      Geometry = 19457/255/63. sectors = 312581808. start = 0
    $ sudo hdparm /dev/hda Отобразить настройки жесткого диска
                                (с интерфейсом IDE)
$ sudo hdparm -I /dev/sda Отобразить подробную информацию о жестком диске
/dev/sda:
ATA device, with non-removable media
        Model Number:     ST3160815AS
        Serial Number:      6RA5EGGJ
        Firmware Revision: 3.AAD
...
```

Резюме

Ubuntu и другие Linux-системы позволяют с легкостью просматривать и изменять многие аспекты функционирующей системы для обеспечения ее работы с максимальной производительностью. Такие команды, как free, top, vmstat, slabtop, iostat и dstat, позволяют узнать, как система использует свою память, ресурсы центрального процессора и устройства для хранения. С помощью команд date, hwclock и cal, а также служб, например NTP, вы можете просматривать и управлять настройками даты и времени в системе.

Для управления задаваемыми параметрами и службами, запускаемыми при загрузке системы, вы можете модифицировать параметры, ассоциированные с загрузчиком GRUB и системными уровнями выполнения. Вы можете запускать, останавливать, отображать информацию, добавлять и удалять отдельные системные службы с помощью команд вроде service и update-re.d. Команды reboot, halt и shutdown дают возможность безопасно выключать или перезагружать компьютер.

Когда дело касается управления аппаратным обеспечением вашего компьютера, такие команды, как Ismod, modinfo и modprobe позволяют работать с загружаемыми модулями. Вы можете просматривать информацию об имеющемся аппаратном обеспечении с помощью команд Ispci, dmi decode и hdparm.

11 Управление сетевыми подключениями

В этой главе:

- использование команд ethtool и mi i -tool для работы с сетевыми интерфейсными картами;
- получение статистики касательно сетевых интерфейсов с помощью netstat;
- запуск сетевых устройств;
- просмотр информации об Ethernet-подключениях посредством ifconfig и ip;
- управление беспроводными картами с использованием команды iwconfig;
- проверка разрешения имен DNS с помощью dig, host и hostname;
- проверка возможности подключения посредством ping и arp;
- трассировка подключений с использованием traceroute, route и ip;
- наблюдение за сетью с помощью netstat, tcpdump и nmap.

Подсоединиться к сети из Linux чаще всего так же просто, как включить компьютер. Проводные или беспроводные интерфейсы должны запуститься и сразу предоставить вам возможность подключиться к другим компьютерам в локальной сети или Интернете. Однако если ваш сетевой интерфейс не заработает либо потребует настройки вручную, можете воспользоваться множеством команд, пригодных для конфигурирования сетевых интерфейсов, проверки сетевых подключений и настройки специальной маршрутизации.

В этой главе рассматриваются команды для конфигурирования и работы с сетевыми интерфейсными картами (Network Interface Card, NIC), например ethtool, mii-tool и ifconfig. Точнее, в ней демонстрируются способы конфигурирования проводных и беспроводных Ethernet-подключений. Здесь также приведено описание таких команд, как netstat, dig, ip и ping, наряду с использованием подключенного аппаратного обеспечения и настроенных сетевых интерфейсов.

Конфигурирование сетей с использованием GUI-интерфейса

При первой инсталляции Ubuntu установщик позволит вам сконфигурировать любые проводные Ethernet-карты, имеющиеся на компьютере, с использованием сервера DHCP, обнаруженного в вашей сети. Сервер DHCP позволит присвоить

IP-адрес сетевому интерфейсу компьютера, а также назначить шлюз по умолчанию (для доступа в Интернет или другие удаленные сети), сервер DNS (для разрешения имен в адреса) и, возможно, имя хоста.

В качестве альтернативы можно присвоить статический IP-адрес наряду с именем хоста и IP-адресами для компьютера-шлюза и серверов имен. После установки вы также получите возможность воспользоваться графическими инструментами для конфигурирования сетевых интерфейсов.

В настольных системах Ubuntu управление сетевыми интерфейсами осуществляется посредством приложения NetworkManager. Чтобы внести изменения в сетевые подключения, введите `Network Connections` (Сетевые подключения) на панели управления либо щелкните на значке сети (он похожа на сектор круговой диаграммы) на верхней панели. Затем выберите **Edit Connections** (Редактировать подключения) в меню. В появившемся окне **Network Connections** (Сетевые подключения) вы сможете сконфигурировать как проводные, так и беспроводные подключения. Выберите нужное подключение, после чего вы, если захотите, сможете изменить динамическую (DHCP) конфигурацию на статические IP-адреса.

Иногда сетевые интерфейсы могут не функционировать либо вы можете захотеть поработать с ними путями, которые не поддерживаются GUI-интерфейсом. Для таких случаев в приведенных далее разделах описывается, как работать с сетевыми интерфейсами из командной строки.

Управление сетевыми интерфейсными картами

Если сетевое аппаратное обеспечение компьютера сразу не заработает и не позволит подключиться к Интернету, необходимо выполнить описанные ниже действия, чтобы устранить проблему.

- При использовании проводной сетевой интерфейсной карты удостоверьтесь, что она установлена правильно и кабель подключен к сети (DSL-линии ISP-поставщика, коммутатору и т. д.).
- Подключив кабель, убедитесь, что у вас есть соединение без каких-либо несоответствий относительно скорости или дуплексных режимов.
- Удостоверьтесь, что кабель плотно вставлен в сетевую интерфейсную карту (вставляя его, вы должны были услышать щелчок). Что касается беспроводных сетевых интерфейсных карт, то, если нет никаких признаков наличия беспроводной карты, но вы знаете, что в вашем ноутбуке она есть, проверьте маленький переключатель сбоку ноутбука. Мне несколько раз доводилось видеть, как этот переключатель по ошибке был установлен в положение, которое отключало беспроводную карту, в результате чего она вообще не отображалась в списке доступных сетевых интерфейсных карт.

Чтобы проверить соединение из Linux, а также задать скорость и дуплексный режим, используются две команды: более старая `mii -tool` (из пакета net-tools) и более новая `ethtool` (из пакета ethtool). Используйте `ethtool`, если только у вас

нет очень старой сетевой интерфейсной карты и драйвера для нее, которые несовместимы с данной командой.

Чтобы установить пакет ethtool, а затем **просмотреть синтаксис команды ethtool,** введите следующее:

```
$ sudo apt-get remove ethtool
$ ethtool -h | less                    Просмотр параметров команды ethtool
```

Возвращаемая командой ethtool информация является встроенной в stderr помощью. Чтобы просмотреть эти сообщения с помощью less, мы перенаправили stderr в stdout.

Для **отображения настроек определенной Ethemet-карты** добавьте в команду имя соответствующего интерфейса. Например, чтобы просмотреть информацию о карте для ethO, введите следующее:

```
$ sudo ethtool ethO Показать настройки сетевой интерфейсной карты для ethO
Settings for ethO:
        Supported ports: [ TP ]
        Supported link modes: lObaseT/Half lObaseT/Full
                              lOObaseT/Half lOObaseT/Full
                              lOOOObaseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: Yes
        Advertised link modes: lObaseT/Half lObaseT/Full
                               lOObaseT/Half lOObaseT/Full
                               lOOOObaseT/Full
        Advertised auto-negotiation: Yes
        Speed: lOOMb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 1
        Transceiver:     internal
        Auto-negotiation:    on
        MDI-X: off
        Supports Wake-on: plumbg
        Wake-on: g
        Current message level: 0x00000001 (1)
                               drv
        Link detected: yes
```

Вам потребуются права доступа суперпользователя для получения информации о Ethernet-интерфейсе, в силу чего в приведенном выше примере была использована команда sudo.

Чтобы **получить сведения о драйвере, используемом для определенной сетевой карты,** задействуйте параметр -i:

```
$ sudo ethtool -i ethO Отобразить информацию о драйвере
                       сетевой интерфейсной карты
driver: eЮООе
version: 1.5,1-k
firmware-version: 0.5-7
bus-info: 0000:01:00.0
```

Для вывода на экран подробной статистики касательно сетевой интерфейсной карты используйте параметр -S:

$ sudo ethtool -S ethO *Показать статистику касательно сетевой интерфейсной карты для ethO*

```
NIC statistics:
     rx_packets: 1326384
     tx_packets: 773046
     rx_bytes: 1109944723
     tx_bytes: 432773480
     rx_errors: 5
     tx_errors: 2
     rx_dropped: 0
     tx_dropped: 0
     multicast: 0
     collisions: 0
     rx_length_errors: 0
     rx_over_errors: 0
     rx_crc_errors: 5
     rx_frame_errors: 0
     rx_fifo_errors: 0
     rx_missed_errors: 0
     tx_aborted_errors: 0
     tx carrier errors: 2
...
```

Команду ethtool можно использовать для **изменения настроек сетевой интерфейсной карты,** а также для вывода их на экран. Чтобы деактивизировать автоматическое согласование и зафиксировать скорость сетевой интерфейсной карты на уровне 100 Мбит в секунду в полнодуплексном режиме, введите следующее:

```
$ sudo ethtool -s ethO speed 100 duplex full autoneg off Изменить настройки сетевой интерфейсной карты
```

Для деактивизации автоматического согласования и фиксации скорости на уровне 10 Мбит в секунду в полудуплексном режиме введите приведенное ниже:

```
$ sudo ethtool -s ethO speed 10 duplex half autoneg off Изменить настройки сетевой интерфейсной карты
```

Изменения, только что внесенные в настройки сетевой интерфейсной карты, распространяются на текущий сеанс. Однако когда вы осуществите перезагрузку, эти настройки пропадут. Чтобы **настройки остались в силе после следующей перезагрузки и перезапуска сети,** необходимо создать новый сценарий, который будет выполняться во время загрузки. Далее приведены шаги с описанием того, как это сделать.

1. Выберите имя для нового сценария, например eth_opti ons, а затем создайте этот сценарий в каталоге /etc/init.d:

```
$ sudo vi /etc/init.d/eth_options
```

2. Вставьте следующий текст в новый сценарий:

```
#!/bin/sh
ETHTOOL="/usr/sbin/ethtool"
ETHT00L_0PTS="speed 10 duplex half autoneg off"
DEV="ethO"
case "$1" in
start)
      echo -n "Setting $DEV options to $ETHT00L_0PTS...";
      SETHTOOL -s SDEV $ETHT00L_0PTS;
      echo " done.":;
stop)
      : ;
esac
exit 0
```

3. Специфические настройки следует разместить в переменной ETHT00L_0PTS. Например:

```
ETHT00L_0PTS="speed 10 duplex half autoneg off"
```

Вы также можете изменить переменную DEV, указывающую на первый Ethernet-интерфейс, которым является ethO.

4. Сконфигурируйте сценарий как выполняемый файл:

```
$ sudo chmod +x /etc/init.d/eth_options
```

5. Создайте символьные ссылки для запуска нового сценария на разных уровнях выполнения:

```
$ sudo update-rc.d eth_options defaults
 Adding system startup for /etc/imt.d/eth_options ...
   /etc/rc0.d/K20eth_options     ->     ../init.d/eth_options
   /etc/rcl,d/K20eth_options     ->     ../init.d/eth_options
   /etc/rc6.d/K20eth_options     ->     .,/init.d/eth_options
   /etc/rc2.d/S20eth_options     ->     ../init.d/eth_options
   /etc/rc3.d/S20eth_options     ->     ../init.d/eth_options
   /etc/rc4.d/S20eth_options     ->     ../init.d/eth_options
   /etc/rc5.d/S20eth_options -> ../init.d/eth_options
```

Вы сможете запустить свой сценарий с помощью следующей команды:

```
$ sudo /etc/init.d/eth_options start
```

ПРИМЕЧАНИЕ----------

Советы, подобные этому, можно найти на сайте nixCraft по адресу www.cyberciti.biz/tips/.

Как уже отмечалось ранее, команда ethtool может не работать с некоторыми более старыми сетевыми интерфейсными картами. Поэтому, **если у вас такая карта, попробуйте воспользоваться mii-toof,** как показано далее:

```
$ sudo mii-tool          Показать согласованную скорость и статус соединения
                         при старой интерфейсной сетевой карте
ethO: negotiated 100baseTx-FD flow-control, link ok
```

Команда из этого примера выполнялась на том же компьютере, что и команды из приведенных ранее примеров, с автоматическим согласованием в случае с сетевой интерфейсной картой на скорости 1000 Мбит в секунду в полнодуплексном режиме. Команда mi i -tool неправильно истолковывает настройку скорости. Вот почему я рекомендую использовать mi i -tool только как последнее средство, если ethtool не будет работать с вашей старой сетевой интерфейсной картой.

Для **отображения более подробного вывода mii-tool** задействуйте параметр -v:

```
$ sudo mii-tool -v Показать подробный вывод с информацией о настройках
                   старой сетевой интерфейсной карты
ethO: negotiated 100baseTx-FD flow-control, link ok
  product info: Yukon-EC 88E1111 rev 0
  basic mode:   autonegotiation enabled
  basic status: autonegotiation complete, link ok
  capabilities: lOOObasel-FD lOObaseTx-FD lOObaseTx-HD
                lObaseT-FD lObaseT-HD
  advertising:  lOObaseTx-FD lOObaseTx-HD lObaseT-FD
                lObaseT-HD flow-control
  link partner: lOOObaseT-FD lOObaseTx-FD lOObaseTx-HD
                lObaseT-FD lObaseT-HD flow-control
```

В данном примере вы видите, что в каждом случае (lOObaseTx и lObaseT) поддерживается как полудуплексный (HD), так и полнодуплексный (FD) режим. Однако lOOObaseT поддерживает только полнодуплексный режим. Чтобы **деактивизировать автоматическое согласование и форсировать применение определенной настройки,** задействуйте параметр -F, как показано далее:

```
$ sudo mii-tool -F lObaseT-FD ethO Форсировать применение настройки скорости/
                                   дуплексного режима 10baseT-FD
```

Если вы передумаете и позднее захотите **снова активизировать автоматическое согласование,** воспользуйтесь параметром -г:

```
$ sudo mii-tool -г ethO Активизировать автоматическое согласование
                        для старой сетевой интерфейсной карты
restarting autonegotiation...
```

mi i -tool, в отличие от ethtool, не позволяет сохранять настройки, соответственно, вам придется выполнять эту команду после каждой перезагрузки. Для этого ее нужно добавить в конец /etc/rc. 1 ocal.

Команда netstat обеспечивает другой способ **получения статистики касательно сетевого интерфейса:**

```
$ netstat -i                    Получить статистику о сетевом интерфейсе ethO
Kernel Interface table
I face MTU Met RX-OK   RX-ERR        RX-DRP        RX-OVR TX-OK TX-ERR      TX
ethO 1500 0 1757208      6      0      0 996834      4      0      0 BMRU
```

Задействуйте параметр -c, чтобы команда netstat **обновляла статистику касательно сетевого интерфейса каждую секунду:**

```
$ netstat -ic Обновлять статистику касательно сетевого интерфейса каждую секунду
```

Вы можете **получить более четкий (экранно-ориентированный) обновляемый вывод** команды netstat путем ее комбинирования с командой watch, как показано далее:

```
$ watch netstat -i          Обновлять статистику касательно сетевого интерфейса
                            (в экранно-ориентированном режиме)
Every 2.0s: netstat -i                    Wed May 29 01:55:48 2013
Kernel Interface table
I face MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Fig
ethO 1500 0 1757208          600 996834                 4       0       0 BMRU
```

Как видно из этого вывода, статистика netstat обновляется каждые две секунды.

Управление сетевыми подключениями

Запуск и остановка сетевых интерфейсов для проводных Ethernet-подключений к LAN-сети или Интернету обычно осуществляется во время соответственно загрузки и выключения вашей системы Ubuntu. Однако вы можете использовать команды из /etc/init.d для запуска и остановки сетевых интерфейсов в любое желаемое время либо update-rc.d для конфигурирования, должна ли ваша сеть запускаться автоматически.

Команды ifconfig и ip тоже можно использовать для конфигурирования, активизации и деактивизации интерфейсов. Однако в Ubuntu и других дистрибутивах, производных от Debian, команды из каталога /etc/init .d являются более простыми инструментами для запуска и остановки сетевых интерфейсов, поэтому в большинстве случаев вам следует использовать команды ifconfig и ip только для сбора информации об Ethernet-интерфейсах и сетевых интерфейсных картах (как будет показано позднее в этом разделе).

Запуск и остановка сетевых интерфейсов для Ethernet-подключений

Ваши проводные Ethernet-интерфейсы запускаются во многих случаях при загрузке Ubuntu, поскольку служба network настроена быть активизированной при переходе системы на общие уровни выполнения загрузки. Есть набор основных конфигурационных файлов и сценариев, благодаря которым это происходит, а также несколько простых команд, позволяющих вам управлять данным процессом.

В Ubuntu управляющие сценарии и конфигурационные файлы располагаются в каталоге /etc/network/. Сетевые интерфейсные карты конфигурируются путем редактирования файла /etc/network//i nterfaces. Содержимое этого файла выглядит следующим образом:

```
auto 1o
1 face lo inet loopback
auto ethO
iface  ethO  inet  dhcp
```

```
auto ethl
iface ethl met dhcp
auto eth2
iface eth2 inet dhcp
auto athO
iface athO inet dhcp
auto wlanO
iface wlanO inet dhcp
```

Для **получения дополнительной информации об этом файле** введите показанное далее:

$ less /usr/share/doc/network-manager/README.Debian

Если вы измените файл i nterfaces и решите использовать NetworkManager для управления своими сетевыми интерфейсами, выполните следующую команду:

$ sudo service network-manager restart

Сценарием, который запускает сконфигурированные файлы network-scripts, является /etc/init. d/network. Как и в случае с другими службами Linux, вы можете останавливать и запускать службу network с помощью команды /etc/init.d/networking.

Чтобы **отключить все сетевые интерфейсные карты, а затем снова включить** их с целью позволить всем изменениям в сетевых сценариях вступить в силу, введите вот это:

```
$ sudo /etc/init.d/networking restart Перезапустить сетевые интерфейсы
 * Reconfiguring network interfaces...
...
```

Вы можете увидеть ошибки, связанные с дополнительными интерфейсами, которые были определены, однако в действительности отсутствуют в вашей системе, например, это могут быть беспроводные интерфейсы. Вы можете игнорировать любые ошибки, имеющие отношение к сетевым устройствам, которые вы не устанавливали.

Используйте параметры start и stop для соответственно **запуска и остановки сетевых интерфейсов:**

```
$ sudo /etc/init.d/networking stop Остановить сетевые интерфейсы
$ sudo /etc/init.d/networking start Запустить сетевые интерфейсы
```

Для **проверки статуса сетевых интерфейсов** введите следующее:

```
$ ifconfig                Проверить статус сетевых интерфейсов
ethO Link encap:Ethernet HWaddr 00:19:DI:5A:A9:E2
  inet addr:192.168.1.106 Beast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::219:diff:fe5a:a9e2/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:14442 errors:0 dropped:0 overruns:0 frame:0
          TX packets:13080 errors:0 dropped:0 overruns:0 carrier:0
          colli sions:434 txqueuelen:1000
          RX bytes:3732823 (3.5 MiB) TX bytes:1142020 (1.0 MiB)
          Interrupt:16 Memory:fe9e0000-fea00000
```

```
Lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: :: 1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:16436 Metric:1
          RX packets:35 errors:0 dropped:0 overruns:0 frame:0
          TX packets:35 errors:0 dropped:0 overruns:0 carrier:0
          col 1i si ons:0 txqueuelen:0
          RX bytes:2121 (2.0 KiB) TX bytes:2121 (2.0 KiB)
```

Если у вас будет несколько сетевых интерфейсов, может потребоваться **запустить или остановить один из них.** Для этого используйте команды ifup и ifdown:

```
$ sudo ifdown ethO    Остановить сетевой интерфейс ethO
$ sudo ifup ethO      Запустить сетевой интерфейс ethO
```

Когда сетевые интерфейсы запущены, вы можете использовать инструменты для просмотра информации об этих интерфейсах и ассоциированных с ними сетевых интерфейсных картах.

Просмотр информации об Ethernet-подключениях

Для просмотра адреса управления доступом к среде (MAC) сетевой интерфейсной карты и IP-адреса для TCP/IP-подключений вы можете прибегнуть к команде i fconfi g. Приведенная далее строка **отображает информацию об адресе и статусе вашего Ethernet-интерфейса ethO:**

```
$ ifconfig ethO
ethO Link encapEthernet HWaddr 00:DO:B7:79:A5:35
          inet addr:10.0.0.155 Bcast:10.0.0.255 Mask:255.255.255.0
          inet6 addr: fe80::2d0:b7ff:fe79:a535/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:1413382 errors:6 dropped:0 overruns:0 frame:6
          TX packets:834839 errors:4 dropped:0 overruns:0 carrier:4
          colli sions:0 txqueuelen:1000
          RX bytes:1141608691 (1.0 GiB) TX bytes:470961026 (449.1 MiB)
```

В этом примере интерфейс ethO представляет собой первый Ethernet-интерфейс на компьютере. MAC-адресом (HWaddr) сетевой интерфейсной карты является 00:D0:B7:79:A5:35. Вы можете видеть 1P-адрес (10.0.0.155), широковещательный адрес (10.0.0.255) и маску подсети (255.255.255.0) интерфейса ethO. К другим представленным сведениям относится количество полученных и переданных пакетов, а также информация о проблемах (количество ошибок, отброшенных пакетов и переполнений), имеющих место в случае с интерфейсом.

Для **получения информации как об активных, так и о неактивных сетевых интерфейсных картах** задействуйте параметр -а:

```
$ ifconfig -a
```

Вместо ifconfig (и нескольких других команд, описанных в этой главе) вы можете использовать более новую команду i p. Она была создана для отображения информации о сетевых интерфейсах, а также для изменения настроек сетевых устройств, маршрутизации и IP-туннелей. Вот пример использования команды ip с целью **вывода на экран сведений об интерфейсе ethO:**

```
$ ip addr show ethO
2: ethO: <BROADCAST,MULTICAST,UP,L0WER_UP>
         mtu 1500 qdisc pfifo_fast qlen 1000
    1ink/ether 00:d0:b7:79:a5:35 brd ff:ff:ff:ff:ff:ff
    inet 10.0.0.155/24 brd 10.0.0.255 scope global ethO
    inet6 fe80::2d0:b7ff:fe79:a535/64 scope link
       validjft forever preferredjft forever
```

Команда ip позволят использовать сокращенный синтаксис. Если вам знаком интерфейс командной строки Cisco IOS, знайте: команда i p работает аналогичным образом. Например, вместо ip addr show вы можете ввести следующее, чтобы **отобразить информацию обо всех интерфейсах:**

```
$ ip a
```

Команда ip может работать с множественными сетевыми компонентами, называемыми *объектами.* Одним из этих объектов является addr, который позволяет ip конфигурировать сетевые адреса. В приведенных далее примерах рассматриваются другие объекты команды i p.

Чтобы отобразить информацию касательно использования ip, задействуйте параметр help. Помимо применения help, вы можете идентифицировать объект ip для получения сведений об использовании этого объекта:

```
$ ip help                    Просмотр информации касаемо использования ip
Usage: ip [ OPTIONS ] OBJECT { COMMAND | help }
       ip [ -force ] [-batch filename
where OBJECT := { link | addr | route | rule | neigh | ntable |
                  tunnel | tuntap | maddr | mroute | mrule |
                  monitor | xfrm j netns }
       OPTIONS := { -Vfersion] | -s[tatistics] | -r[esolve] |
                  -f[amily] { inet | inet6 | ipx | dnet | link } |
                  -l[oops] { maximum-addr-flush-attempts } |
                  -o[neline] | -t[imestamp] | -b[atch] [filename] |
                  -rc[vbuf] [size]}
$ ip  addr help        Просмотр справочной информации об объекте addr
$ ip  route help       Просмотр справочной информации об объекте route
$ ip  tunnel help      Просмотр справочной информации об объекте tunnel
```

Маски подсетей могут привести в замешательство, если они непривычны вам. Возможно, вы найдете полезной команду i peal c (из пакета ipcalc) для **расчета маски подсети хост-компьютера на основе его IP-адреса CIDR:**

```
$ ipcalc -bn 192.168.1.100/27
Address:    192.168.1.100
Netmask:    255.255.255.224 = 27
Wildcard: 0.0.0.31
=>
Network:    192.168.1.96/27
HostMin:    192.168.1.97
HostMax:    192.168.1.126
Broadcast:  192.168.1.127
Hosts/Net: 30                    Class C.  Private Internet
```

В данном примере маской подсети (выступающей в качестве индикатора, какая часть IP-адреса представляет сеть, а какая хост) является 255.255.255.224. Она была рассчитана посредством значения /27 в конце IP-адреса 192.168.1.100.

Беспроводные подключения

В прошлом конфигурирование беспроводных подключений в Linux было сложным главным образом из-за отсутствия драйверов с открытым исходным кодом для многих из первых беспроводных LAN-карт. В более поздних выпусках Ubuntu ситуация заметно улучшилась.

Для обеспечения базовой беспроводной конфигурации рекомендую использовать GUI-инструменты (в частности, окно **Network Configuration** (Конфигурация сети), сведения о котором приводились ранее в этой главе, или Network Manager).

В редких случаях может потребоваться установить пакеты с инструментами для работы с беспроводными устройствами с целью обеспечения функционирования беспроводных интерфейсов, например, пакеты wireless-tools и bcm43xx-fwcutter, которые доступны в репозиториях Ubuntu. Кроме того, вам могут понадобиться прошивки, доступные в следующих пакетах: ipw2100-source, ipw2200-firmware и zdl211-firmware.

Если не удается сконфигурировать беспроводную LAN-карту посредством окна **Network Configuration** (Конфигурация сети), то, возможно, получится заставить ее работать с помощью драйверов и инструментов, доступных благодаря Atheros **(www.atheros.com),** проектам MadWifi **(www.madwifi.org)** или NDISwrapper **(ndiswrapper.sourceforge.net).** Многие пакеты программного обеспечения, разработанного в рамках этих проектов, доступны в стандартных репозиториях Ubuntu, которые были описаны в гл. 2.

Если потребуется помощь в **определении, какие именно беспроводные карты у вас имеются,** введите следующее:

```
$ dmesg | grep -i wireless                Осуществить поиск беспроводных PCI-карт
Intel(R) Wireless WiFi Link AGN driver for Linux, in-tree:
```

При условии, что беспроводная карта находится в состоянии рабочей готовности, вы сможете воспользоваться несколькими полезными командами из пакета wireless-tools для просмотра и изменения ее настроек. В частности, команда iwconfig может помочь в работе с беспроводными LAN-интерфейсами. Приведенное далее обеспечивает **сканирование ваших сетевых интерфейсов на предмет поддерживаемых беспроводных карт** и выводит на экран их текущие настройки:

```
$ iwconfig
ethO no wireless extensions,
ethl IEEE 802.11-DS ESSID:"Mylan"
     Mode:Managed Frequency:2.437 GHz Access Point: 43:5A:29:E7:95:75
          Bit Rate:54 Mb/s Tx-Power=15 dBm
          Retry long 1imit:7 RTS thr:off Fragment thr:off
          Power Management:off
```

Беспроводные интерфейсы могут иметь имя wlanX или ethX в зависимости от используемого аппаратного обеспечения и драйвера. Вам, возможно, удастся получить больше информации после установки соединения с использованием беспроводного интерфейса:

```
$ ip link set ethl up
$ iwconfig ethl
ethl IEEE 802.llabgn ESSID:"Mylar"
    Mode:Managed Frequency:2.437 GHz Access Point: 43:5A:29:E7:95:7
          Bit Rate:54 Mb/s Tx-Power=15 dBm
          Retry long 1imit:7 RTS thr:off Fragment thr:off
          Power Management:off
          Link Qua!ity=70/70 Signal level=-39 dBm
          Rx invalid nwid:0 Rx invalid crypt:0 Rx invalid frag:0
          Tx excessive retries:0 Invalid misc:0 Missed beacon:0
```

Данные настройки можно модифицировать несколькими путями. Далее приведены способы **использования iwconfig для изменения настроек беспроводного интерфейса.** В следующих примерах демонстрируются операции касательно беспроводного интерфейса с именем wlanO. Эти операции могут поддерживаться или не поддерживаться в зависимости от того, какую беспроводную карту и драйвер вы используете.

$ sudo iwconfig wlanO essid "MyWireless"	*Задеть для essid значение MyUireless*
$ sudo iwconfig wlanO channel 3	*Задать для channel значение 3*
$ sudo iwconfig wlanO mode Ad-Hoc	*Сменить Managed на режим Ad-Hoc*
$ sudo iwconfig wlanO ap any	*Использовать любую точку доступа*
$ sudo iwconfig wlanO sens -50	*Задать для sens значение -50*
$ sudo iwconfig wlanO retry 20	*Задать для retry значение 20 касательно повторных MAC-передач*
$ sudo iwconfig wlanO key 1234-5555-66	*Задать для key значение 1234-5555-66*

`essid` иногда называют сетевым именем или доменным идентификатором. Вы можете использовать его в качестве общего имени, идентифицирующего вашу беспроводную сеть. Задание значения для `channel` позволяет беспроводной LAN-сети работать на определенном канате.

В режиме `Ad-Hoc` сеть состоит только из взаимосоединенных клиентов без центральной точки доступа. Указав для `ap` значение в виде определенного MAC-адреса в режиме `Managed/Infrastructure`, можно сделать так, чтобы карта подключалась к точке доступа по этому адресу, либо вы можете задать для `ap` значение `any`, чтобы разрешить подключение к любой точке доступа.

При возникновении проблем с производительностью попробуйте изменить величину, определяющую чувствительность (`sens`), на отрицательное значение (которое представляет dBm) или положительное (которое может быть либо процентным значением, либо значением `sens`, заданным поставщиком). Если при повторной передаче происходят сбои, можете увеличить значение `retry`, чтобы карта успевала отправить больше пакетов перед сбоем.

Параметр key используется для задания ключа шифрования. Вы можете вводить шестнадцатеричные цифры (XXXX-XXXX-XXXX-XXXX или XXXXXXXX). Указав s: перед значением key, можно ввести ASCII-строку (например, s:My927pwd).

Проверка разрешений имен

IP-адреса являются наборами цифр, а люди предпочитают обращаться к вещам по имени, поэтому TCP/IP-сети (например, Интернет) полагаются на DNS-систему для разрешения имен хостов в IP-адреса. Ubuntu обеспечивает набор инструментов для поиска информации, связанной с разрешением имен DNS.

При первой установке Ubuntu вы либо обозначили серверы DNS (Domain Name System — система доменных имен) для разрешения имен, либо обеспечили их автоматическое присваивание с использованием сервера DHCP. Эта информация затем сохраняется в файле /etc/ resol v. conf, содержимое которого будет выглядеть примерно так:

```
domain exampie.com
search example.com example.net
nameserver 11.22.33.44
nameserver 22.33.44.55
```

Строка домена, если она присутствует, идентифицирует локальный домен. Это позволяет вам идентифицировать компьютер по его базовому имени, а при DNS-поиске предполагается, что вы имеете в виду локальный домен. Таким образом, если вы запросите хост с именем abc, то система будет искать abc. exampl e. com. Строка поиска позволит определить несколько доменов для поиска.

Приведенные выше цифры из файла /etc/resol v. conf заменяются реальными IP-адресами компьютеров, которые служат в качестве серверов имен DNS. При наличии возможности подключиться к работающим серверам DNS вы сможете использовать команды для выполнения запросов к этим серверам и поиска хост-компьютеров.

Команду dig (которую следует использовать вместо устаревшей nslookup) можно задействовать для поиска информации, предоставляемой сервером DNS. Команда host может применяться для поиска информации об адресе, касающейся имени хоста или доменного имени.

Для **поиска хоста с определенным именем среди ваших серверов DNS** (www.turbosphere.com в приведенных далее примерах) используйте команду dig следующим образом:

$ dig www.turbosphere.com *Осуществить поиск среди серверов DNS, указанных в /etc/resolv.conf*

Вместо использования назначенного сервера имен можете **выполнить запрос к определенному серверу имен.** В следующем примере делается запрос к серверу DNS по адресу 4.2.2.1:

$ dig www.turbosphere.com И.2.2.1

С помощью команды di g вы также можете **сделать запрос к записи определенного типа:**

$ dig turbosphere.com mx Запрос об устройстве работы с электронной почтой
$ dig turbosphere.com ns Запрос о низкоуровневых серверах имен.

Используйте параметр +trace для **трассировки рекурсивного запроса** от серверов DNS верхнего уровня до низкоуровневых серверов:

$ dig +trace www.turbosphere.com *Рекурсивное отслеживание DNS-серверов*

Если вы захотите *узнать* **IP-адрес хост-компьютера,** задействуйте параметр +short:

$ dig +short www.turbosphere.com *Отобразить только пару «имя IP-адрес»*
turbosphere.com.
66.113.99.70

Вы можете использовать dig для **обратного поиска DNS-информации на основе IP-адреса:**

$ dig -x 66.113.99.70 *Получить DNS-информацию на основе IP-адреса*

host можно также использовать для **обратного DNS-поиска:**

$ host 66.113.99.70
70.99.133.66.in-addr.arpa domain name pointer boost.turbosphere.com.

Для получения информации об имени хоста для локального компьютера задействуйте команды hostname и dnsdomainname:

$ hostname *Просмотр полного имени хоста DNS для локального компьютера*
boost.turbosphere.com

Вы также можете использовать hostname для **временного задания локального имени хоста** (до следующей перезагрузки). Вот пример:

$ sudo hostname serverl.examp1e.com *Задать локальное имя хоста*

Изменение имени хоста работающего компьютера может негативно сказаться на запущенных демонах. Вместо этого рекомендую **сделать так, чтобы локальное имя хоста задавалось при каждом запуске системы.** Для этого отредактируйте первую строку в файле /etc/hostname. Пример:

serverl.example.com

Устранение неполадок в сетях

Неполадки в сетях обычно устраняются начиная с нижнего уровня и вверх. Как уже отмечалось в начале этой главы, в первую очередь нужно убедиться, что компоненты физического сетевого уровня (кабели, сетевые интерфейсные карты и т. д.) подключены и функционируют. Затем следует проверить работоспособность соединений между физическими узлами. После этого вы можете воспользоваться

множеством инструментов для проверки возможности подключения к определенному хосту.

Проверка возможности подключения к хосту

Если вы знаете, что есть соединение и нет несоответствия дуплексных режимов, то следующим шагом будет использование ping в отношении вашего шлюза по умолчанию. Вам следует либо сконфигурировать шлюз по умолчанию (gw) в файле /etc/network/interfaces, либо позволить системе настроить его с применением такой службы, как DHCP. Чтобы **проверить шлюз по умолчанию, обратившись к фактической таблице маршрутизации,** задействуйте команду i p следующим образом:

```
$ ip route
10.0.  0.0/24 dev ethO proto kernel scope link src 10.0.0.155
169.254.0.  0/16 dev ethO scope link
default via 10.0.0.1 dev ethO
```

Шлюзом для маршрута по умолчанию в этом примере является 10.0.0.1. Чтобы **убедиться в возможности IP-подключения в случае с этим шлюзом,** используйте команду pi ng, передав ей адрес шлюза по умолчанию:

```
$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=l ttl=64 time=0.382 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.313 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.360 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=1.43 ms

--- 10.0.0.1 ping statistics ---
4  packets transmitted. 4 received. 01 packet loss, time 2999ms
rtt min/avg/max/mdev = 0.313/0.621/1.432/0.469 ms
```

По умолчанию выполнение pi ng будет продолжаться, пока вы не нажмете комбинацию клавиш Ctrl+C. К другим параметрам ping относятся следующие:

$ ping -a 10.0.0.1 *Обеспечить подачу соответствующего звукового сигнала при выполнении ping*

$ ping -c 4 10. 0.0.1 *Выполнить ping четыре раза и выйти (используется по умолчанию в Windows)*

$ ping -q -c 5 10.0.0.1 *Показать суммарную информацию о выполнении ping (лучше всего работает при использовании -с)*

$ sudo ping -f 10.0.0.1 *Выполнить ping большое количество раз подряд (необходимы привилегии* суперпользователя)

$ ping -i 3 10. 0.0.1 *Отправлять пакеты с интервалом три секунды*

$ sudo ping -I ethO 10.0.0.1 *Задать в качестве источника ethO использовать при наличии нескольких сетевых интерфейсных карт)*

```
PING 10.0.0.1 ( 10.0.0.1) from 10.0.0.155 ethO: 56(84) bytes of data.
```

$ sudo ping -I 10.0. 0.155 10.0.0.1 *Задать в качестве источника 10.0.0.155*

```
PING 10.0.0.1 ( 10.0.0.1) from 10.0.0.155 : 56(84) bytes of data.
```

$ ping -s 1500 10.0. 0.1 *Задать размер пакетов как равный 1500 байтам*

```
PING 10.0.0.1 ( 10.0.0.1) 1500(1528) bytes of data.
```

Используйте параметр -f в сочетании с ping с осторожностью. По умолчанию ping отправляет маленькие пакеты (размером 56 байт). Большие пакеты (вроде 1500-байтовых из приведенного выше примера) подходят для выявления проблемных сетевых интерфейсных карт и подключений.

Проверка протокола разрешения адресов

Если выполнение pi ng в отношении вашего шлюза не даст положительного результата, проблема может быть на MAC-уровне Ethernet. Для поиска информации на этом уровне можно задействовать ARP (Address Resolution Protocol — протокол разрешения адресов). Просмотр и конфигурирование ARP-записей осуществляется с помощью команды `arp` или ip `neighbor`. В этом примере показано, как arp используется для вывода на экран списка компьютеров по имени хоста, информация о которых содержится в ARP-кэше:

```
$ arp -v          Показать записи в ARP-кэше по имени хоста
Address        HWtype   Hwaddress           Flags  Mask   Iface
Ritchie        ether    00:10:5A:AB:F6:A7   C              eth0
Einstein       ether    00:OB:6A:02:EC:98   C              eth0
Entries: 1 Skipped: 0 Found: 1
```

В этом примере вы видите имена других компьютеров, сведения о которых содержатся в ARP-кэше локального компьютера, а также тип соответствующего аппаратного обеспечения и аппаратный адрес (MAC-адрес) сетевой интерфейсной карты каждого компьютера. Вы можете **деактивизировать разрешение имен, чтобы взамен увидеть IP-адреса этих компьютеров:**

```
$ arp -vn Показать записи в ARP-кэше по IP-адресу
Address         HWtype  Hwaddress           Flags Mask  Iface
10.0.  0.1      ether   00:10:5A:AB:F6:A7   C           eth0
10.0.  0.50     ether   00:0B:6A:02:EC:98   C           eth0
Entries: 1 Skipped: 0 Found: 1
```

Чтобы **удалить запись из ARP-кэша,** задействуйте параметр -d:

```
$ sudo arp -d 10.0.0.50 Удалить адрес 10.0.0.50 из ARP-кэша
```

Вместо разрешения ARP-кэшу динамически узнавать о других системах, вы можете **добавить в него статические ARP-записи** с помощью параметра -s:

```
$ sudo arp -s 10.0.0.51 00:0B:6A:02:EC:95      Добавить IP/MAC-адреса в ARP-кэш
```

Чтобы с помощью команды ip получить то же, что было только что сделано посредством `arp`, задействуйте объект `neighbor` (следует отметить, что объекты `neighbor`, `nei` и `n` можно использовать как взаимозаменяемые):

```
$ ip neighbor
10.0.  0.1 dev eth0 lladdr 00:10:5a:ab:f6:a7 DELAY
10.0.  0.50 dev eth0 lladdr 00:0b:6a:02:ec:98 REACHABLE
# ip nei del 10.0.0.50 dev eth0
# ip n add 10.0.0.51 lladdr 00:0B:6A:02:EC:95 dev eth0
```

Для **выполнения запроса к подсети с целью узнать, задействуется ли уже тот или иной IP-адрес,** и найти MAC-адрес устройства, которое использует его, примените команду arping. ifup использует команду arping, чтобы избегать IP-конфликтов при запуске сетевых интерфейсных Ethernet-карт. Вот примеры:

$ arping 10.0.0.50 *Выполнить запрос к подсети с целью узнать, используется ли 1P-адрес 10.0.0.50*

```
ARPING 10.0.0.50 from 10.0.0.195 ethO
Unicast reply from 10.0.0.50 [00:0B:6A:02:EC:98] 0.694ms
Unicast reply from 10.0.0.50 [00:OB:6A:02:EC:98] 0.683ms
```

$ sudo arping -I ethO 10.0.0.50 *Определить интерфейс для использования при выполнении запроса*

Команда ping, как и arping (из пакета iputils-arping), будет постоянно запрашивать адрес, пока ее выполнение не будет завершено нажатием сочетания клавиш Ctrl+C. Обычно необходимо узнать, реагирует ли целевой объект, поэтому вы можете использовать одну из следующих команд:

$ arping -f 10.0.0.50 *Запросить 10.0.0.50 и остановиться после получения первого ответа*

$ arping -c 2 **10.0.0.51** *Запросить 10.0.0.50 и остановиться после двух раз*

Трассировка маршрутов к хостам

Когда вы убедитесь, что выполнение pi ng в отношении шлюза дает положительный результат, и даже сумеете связаться с компьютерами, находящимися вне вашей сети, все равно могут возникать проблемы с подключением к определенному хосту или сети. Оказавшись в такой ситуации, **можете воспользоваться traceroute (из пакета traceroute) для выявления узкого места или точки сбоя:**

$ traceroute boost.turbosphere.com *Следовать маршруту к хосту*

```
traceroute to boost.turbosphere.com (66.113.99.70),
     30 hops max.40 byte packets
 1  10.0.0.1(10.0.0.1) 0.281ms 0.289ms 0.237ms
 2  tl-03.hbci.com (64.211.114.1) 6.213 ms 6.189 ms 6.083 ms
 3  172.17.2.153(172.17.2.153) 14.070ms 14.025ms 13.974ms
 4  so-0-3-2.ar2.MINl.gblx.net (208.48.1.117) 19 ms 19 ms 19 ms
 5  sol-0-0-2488M.ar4.SEA1,gblx.net(67.17.71.210)94.6 ms 94.6 ms 94.6ms
 6  64.215.31.114 (64.215.31.114) 99.643 ms 101.647 ms 101.577 ms
 7  dr02-vl09.tac.opticfusion.net(209.147.112.50)262.301ms 233.316ms 233.153 ms
 8 dr01-vl00.tac.opticfusion.net (66.113.96.1) 99.3 ms 99.4 ms 99.3 ms
 9 boost.turbosphere.com (66.113.99.70) 99.25 ms 96.21 ms 100.22 ms
```

Как видите, самый длинный переход имеет место между 4 (Global Crossing, вероятно, в Миннеаполисе) и 5 (Global Crossing в Сиэтле). Однако этот промежуток в действительности не является узким местом; он лишь отражает расстояние между соответствующими точками перехода. Иногда последние переходы выглядят так:

```
28 * * *
29 * * *
3Q * * *
```

Наличие строк со звездочками (*) в конце трассы может быть обусловлено брандмауэрами, блокирующими трафик к целевому объекту. Однако если вы увидите несколько звездочек перед пунктом назначения, они могут свидетельствовать о серьезной перегрузке или сбоях в работе оборудования и указывать на узкое место.

По умолчанию `traceroute` использует UDP-пакеты, которые обеспечивают более реалистичную картину производительности, чем ICMP-пакеты, так как при некоторых интернет-переходах ICMP-трафик наделяется более низким приоритетом. Однако если вы все равно захотите **осуществить трассировку с использованием ICMP-пакетов,** введите следующую команду:

$ traceroute -I boost.turbosphere.com *Использовать ICMP-пакеты для трассировки маршрута*

По умолчанию `traceroute` подключается через порт 80. Можно **задать другой порт,** воспользовавшись параметром `-p`:

$ traceroute -p 25 boost.turbosphere.com *Подключаться через порт 25 при трассировке*

Вы можете **выводить на экран IP-адреса вместо имен хостов,** деактивизировав разрешение имен в случае с переходами:

$ traceroute -n boost.turbosphere.com *Деактивизировать разрешение имен при трассировке*

Альтернативой traceroute является команда tracepath, которая тоже использует UDP-пакеты для трассировки:

$ tracepath boost.turbosphere.com *Использовать UDP-пакеты для трассировки маршрута*

Команда `route` раньше была предпочтительным инструментом для просмотра и манипулирования таблицей маршрутизации ядра, но она потихоньку заменяется командой ip `route`. Сетевые сценарии Ubuntu главным образом полагаются на ip `route`. Однако не помешает знать обе эти команды, поскольку `route` по-прежнему довольно широко используется.

Вы можете задействовать `route` для **отображения локальной таблицы маршрутизации.** Вот два примера использования команды `route` с разрешением имен DNS и без него:

$ route *Отобразить локальную таблицу маршрутизации*

```
Kernel IP routing table .
Destination     Gateway     Genmask          Flags Metric Ref Use Iface
10.0. 0.0       *           255.255.255.0    U     0      00      ethO
Default         ritchie     0.0.0.0          UG    0      00      ethO
```

$ route -n *Отобразить таблицу маршрутизации без DNS-поиска*

```
Kernel IP routing table
Destination     Gateway     Genmask          Flags Metric Ref Use Iface
10.0. 0.0       *           255.255.255.0    U     0      00      ethO
0.0.0.0         10.0.0.1    0.0.0.0          UG    0      0    0  ethO
```

Вы можете **добавить адрес шлюза по умолчанию** с помощью параметра gw:

$ sudo route add default gw 10.0.0.2 *Добавить 10.0.0.2 в качестве адреса шлюза по умолчанию*

Можно **добавить новый маршрут в вашу сеть,** указав либо интерфейс (ethO), либо IP-адрес шлюза (например, gw 10.0.0.100):

$ sudo route add -net 192.168.0.0 netmask 255.255.255.0 ethO
$ sudo route add -net 192.168.0.0 netmask 255.255.255.0 gw 10.0.0.100

Вы можете **удалить маршрут,** задействовав параметр del:

$ sudo route del -net 192.168.0.0 netmask 255.255.255.0 *Удалить маршрут*

Использование более новой команды ip дает возможность осуществлять действия, аналогичные показанным выше, выполняемым с помощью команды route. Вот три способа вывести на экран аналогичную основную информацию о маршрутизации:

$ ip route show *Отобразить основную информацию о маршрутизации*
10.0. 0.0/24 dev ethO proto kernel scope link src 10.0.0.195
169.254.0. 0/16 dev ethO scope link
default via 10.0.0.1 dev ethO
$ **ip route** *Отобразить основную информацию о маршрутизации (пример *>2)*
$ **ip r** *Отобразить основную информацию о маршрутизации (пример #3)*

Далее приведены примеры **добавления и удаления маршрутов с помощью ip:**

$ sudo ip r add 192.168.0.0/24 via 10.0.0.100 dev ethO *Добавить маршрут к ethO*
$ sudo ip r add 192.168.0.0/24 via 10.0.0.100 *Добавить маршрут без интерфейса*
$ sudo ip r del 192.168.0.0/24 *Удалить маршрут*

Чтобы **сделать новый маршрут постоянным,** отредактируйте файл /'etc/network/ i nterf aces, разместив в нем информацию о новом маршруте. Например, для добавления маршрута, который был привнесен с. использованием команды ip в предыдущем примере, вставьте следующие строки в /etc/network/interfaces:

iface ethO inet static
address 192.168.0.0
netmask 255.255.255.0
gateway 10.0.0.100

Отображение подключений и статистики netstat

Перечисленные выше средства по устранению неполадок в основном относятся к сетевому уровню (уровень 3). Для **отображения информации о пакетах, которые передаются между компьютерами, использующими протоколы транспортного уровня (TCP и UDP) и ICMP,** можно задействовать команду netstat:

$ netstat -s | less *Показать суммарную информацию о TCP-, ICMP-. UDP-активности*

Вы можете увидеть **список всех TCP-подключений,** в том числе какой именно процесс обрабатывает определенное подключение:

$ sudo netstat -tanp *Просмотр активных TCP-подключений*

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
tcp   0      0 127.0.0.1:631 0.0.0.0:*          LISTEN 2039/cupsd
tcp   0      0 127.0.0.1:25 0.0.0.0:*           LISTEN 2088/sendmail
  ...
```

Вы также можете **просмотреть активные UDP-подключения,** как показано далее:

S sudo netstat -uanp *Просмотр активных UDP-подключений*

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State PID/Program name
Udp        0      0 0.0.0.0:631     0.0.0.0:*              2039/cupsd
Udp        0      0 192.168.122.1:123 0.0.0.0:*            2067/ntpd
  ...
```

Чтобы **вывод netstat ограничивался информацией о демонах, связанных с тем или иным TCP-портом,** используйте слово `listen`:

$ sudo netstat -tanp | grep -i listen *Просмотр информации о демонах, связанных с портом*

Данная команда — отличное средство разрешения конфликтов между демонами при использовании портов.

Другие полезные инструменты для работы с сетями

Если вы захотите **просматривать заголовочную информацию о пакетах** при их отправке и получении вашей системой, используйте команду tcpdump. Она обладает множеством продвинутых функциональных возможностей, большинство из которых вращается вокруг фильтрации и «поиска иголки в стоге пакетов». Если вы запустите выполнение команды tcpdump на удаленном компьютере, то экран заполнится всем SSH-трафиком между вашим клиентом и удаленным компьютером. Чтобы приступить к ее использованию без необходимости детального изучения, как работает фильтрация tcpdump, введите следующую команду:

$ sudo tcpdump | grep -v ssh *Осуществить поиск пакетов, не ассоциированных с ssh*

Если вы пожелаете **глубже покопаться в трафике на уровне пакетов,** примените инструмент `wi reshark` (ранее известный как `ethereal`). Для этого вам потребуется установить пакет wireshark. Вы можете запускать выполнение wireshark с использованием X посредством SSH на удаленном компьютере, wireshark является отличным анализатором пакетов, соперничающим с лучшими коммерческими инструментами.

Для **исследования сетей и удаленных компьютеров с целью знать, какие службы они предлагают,** используйте `nmap`. Команда `nmap` (из пакета nmap) — самый распространенный сканер портов. Этот инструмент даже можно было увидеть в фильме «Матрица: перезагрузка»! Убедитесь, что вы явным образом авторизованы для сканирования интересующих вас систем или сетей. Команда `nmap` является частью пакета nmap и может выполняться от имени обычного пользователя, однако при некоторых типах сканирования потребуются привилегии суперпользователя.

Вот как **осуществляется базовое сканирование хоста** с помощью `nmap`:

$ sudo nmap 10.0.0.1 *Сканировать порты компьютера по адресу 10.0.0.1*

Чтобы **получить максимально подробный вывод nmap,** задействуйте параметр -vv:

$ sudo nmap -vv 10.0.0.1 *Отобразить максимально подробный вывод nmap*

Для использования `nmap` с целью **сканирования всей сети** передайте этой команде в качестве аргумента адрес соответствующей сети. В приведенном далее примере добавлен параметр `-sP`, чтобы дать `nmap` указание осуществить простое эхо-тестирование адресов:

$ sudo nmap -vv -sP 10.0.0.0/24 *Сканировать хосты во всей сети*

Вы можете очень точно определить, какую именно информацию должна собрать команда `nmap`. В примере ниже параметр `-P0` дает `nmap` указание не использовать `ping` (такой подход годится при сканировании компьютеров, не реагирующих на `ping`). Параметр `-0` обеспечивает отображение отпечатка операционной системы, установленной на сканируемом компьютере. Параметр `-p 100-200` указывает `nmap` сканировать только порты с 100-го по 200-й:

$ sudo nmap -vv -P0 -0 -p 100-200 10.0.0.1 *Не использовать ping, отобразить отпечаток операционной системы, сканировать порты 100-200*

Команда nmap поддерживает множество других параметров для продвинутого использования. Дополнительную информацию вы сможете найти на MAN-странице (man nmap), посвященной команде nmap.

Резюме

Почти каждый аспект сетевых подключений в вашей системе Ubuntu можно сконфигурировать. проверить и подвергнуть мониторингу с использованием инструментов командной строки. Вы можете просматривать и изменять настройки сетевых интерфейсных карт с помощью команд ethtool и mi i -tool. Для отображения статистики касательно сетевых интерфейсов применяется команда netstat.

Простыми в использовании командами для запуска и остановки вашей сети являются service, ifup и ifdown. Установив подключение, можно просматривать статистику относительно его с помощью команд i fconf i g и i p.

Помимо использования проводных Ethernet-карт, операционная система Linux также поддерживает применение другого сетевого аппаратного обеспечения, например беспроводных LAN-карт. Для работы с беспроводными интерфейсами используйте такую команду, как `iwconfig`.

Для проверки разрешения имен DNS задействуйте команды `dig`, `host` и `hostname`. К командам для проверки возможности подключения и маршрутов к хостам относятся `ping`, `arp`, `traceroute` и `ip`.

12 Осуществление доступа к сетевым ресурсам

В этой главе:

- поиск и просмотр информации в Интернете с использованием браузера ELinks;
- передача файлов с помощью команд wget, curl, 1 ftp и scp;
- совместное использование каталогов посредством NFS, Samba и sshfs;
- общение в чатах с применением IRC-клиента i rssi;
- управление электронной почтой с помощью mutt и mail.

За время, которое уходит на запуск графического FTP-клиента, вы могли бы успеть загрузить несколько десятков файлов с удаленного сервера с использованием инструментов командной строки. Даже при наличии GUI-интерфейса команды для передачи файлов, поиска и просмотра информации в Интернете, совместного использования каталогов и чтения электронной почты могут оказаться быстрыми и эффективными инструментами, а в случае отсутствия GUI-интерфейса — стать спасительными средствами.

В этой главе рассматриваются команды для доступа к ресурсам (файлам, электронной почте, совместно используемым каталогам и онлайн-чатам) по сети.

Выполнение команд для поиска и просмотра информации в Интернете

Браузеры, функционирующие в текстовом режиме, позволяют быстро проверять работоспособность веб-серверов, а также получать от них информацию, если отсутствует пригодный для использования GUI-интерфейс. Некогда популярный текстовый браузер Lynx был вытеснен браузерами Links и ELinks в большинстве Linux-систем.

Для использования браузера командной строки вам потребуется установить одну из этих программ с именами, соответствующими названиям пакетов, которые указаны в скобках: Lynx (из пакета lynx-cur), Links (пакет links) и ELinks (из пакета elinks). Если вам потребуется браузер командной строки, в большинстве случаев следует устанавливать пакет elinks.

Браузер ELinks работает в окне терминала. За исключением того, что он не показывает изображения в терминале, он способен обрабатывать самое простое HTML-содержимое и компоненты — таблицы, фреймы, просмотр веб-страниц с использованием вкладок, файлы cookie, историю, MIME-типы и простые каскадные таблицы стилей. Вы даже можете использовать мышь для перехода по ссылкам и выбора элементов меню.

ELinks поддерживает разные цвета, поэтому если используемый терминал тоже обеспечивает их поддержку, то вам не составит труда заменить ссылки и заголовки в тексте (цвета могут не поддерживаться во время сеанса `screen`). Вот примеры строк команды `elinks`:

$ elinks *Приглашает вести имя файла или URL-адрес*
$ elinks www.handsonhistory.cora *Открывает запрошенный вами файл или URL-адрес*

Если у вас есть мышь, щелкните рядом с верхом окна терминала, чтобы увидеть меню. Затем выберите требуемое имя или элемент, а потом — ссылку, чтобы выполнить соответствующий переход. В приведенном далее списке показаны навигационные клавиши, используемые в случае с ELinks:

- **Esc** (или **F9/F8)** — отобразить/скрыть меню (затем используйте клавиши со стрелками или мышь для навигации по меню);
- 1- перейти по следующей ссылке или к следующему редактируемому полю на странице;
- t — перейти по предыдущей ссылке или к предыдущему редактируемому полю на странице;
- -> или **Enter** — перейти вперед по выделенной ссылке, ввести текст в выделенное поле формы;
- <-----вернуться на предыдущую страницу;
- / — искать в прямом направлении;
- ? — искать в обратном направлении;
- **N** — найти далее;
- **Shift+N** — найти предыдущее;
- Page **Up** — прокрутить на одну страницу вверх;
- Page Down — прокрутить на одну страницу вниз;
- G — перейти по URL-адресу;
- **Q** или **Ctrl+C** — выйти из ELinks;
- = — просмотреть информацию о странице;
- **Ctrl+R** — перезагрузить страницу;
- **A** — добавить текущую страницу в закладки;
- OT — открыть новую вкладку в браузере;
- > — перейти на следующую вкладку;
- < — перейти на предыдущую вкладку;
- **C** — закрыть текущую вкладку;

O D — загрузить файл по текущей ссылке;

O Shift+D — просмотреть загрузки;

O A — добавить текущую ссылку в закладки;

O S — просмотреть закладки;

O V — просмотреть текущее изображение;

OH — просмотреть данные менеджера глобальной истории.

Вы можете добавлять глобальные настройки для ELinks в /etc/el inks .conf. Настройки для каждого пользователя в отдельности располагаются в его каталоге $H0ME/ .el inks. Для просмотра доступных настроек введите man el inkskeys.

Передача файлов

В Linux есть множество мощных команд для загрузки файлов с удаленных серверов (HTTP, HTTPS, FTP или SSH). Можете предпочесть одну команду другой из-за того, что она поддерживает определенные параметры, которые вам необходимы. Например, вы можете захотеть выполнить загрузку с использованием зашифрованного подключения, возобновить прерванную загрузку или произвести рекурсивные загрузки. В этом разделе описывается, как использовать wget, ftp, 1 ftp, scp и sftp.

Загрузка файлов с помощью wget

Иногда возникает необходимость загрузить файл с удаленного сервера с использованием командной строки. Например, вы нашли ссылку на программный пакет RPM, однако она несколько раз перенаправляет на разные HTTP-серверы, не позволяя загрузить этот пакет RPM прямо с исходного HTTP-сервера. Либо вам может потребоваться написать сценарий для автоматической загрузки, например, файла журнала каждую ночь.

Команда wget позволяет загружать файлы с веб-серверов (HTTP и HTTPS) и FTP-серверов. Если сервер не требует аутентификации, достаточно передать команде wget ссылку на файл, который необходимо загрузить:

```
$ wget http://design.ubuntu.com/wp-content/uploads/ubuntu-1ogol4.png
```

Однако если **FTP-сервер требует логин и пароль,** вы можете ввести эти данные в строку команды wget в следующей форме:

```
$ wget ftp://user:password@ftp.example.com/path/to/file
S wget --user=usr --password=passwd ftp://ftp.example.com/pathtofile
```

Например:

```
$ wget ftp://chri s:mykuulpwdOftp.1inuxtoys.net/home/chris/image.jpg
$ wget --user=chris --password=mykuulpwd \
ftp://ftp.li nuxtoys.net/home/chri s/i mage.jpg
```

Вы можете использовать wget для **загрузки одной веб-страницы, как показано далее:**

$ wget http://www.wiley.com *Загрузить только веб-страницу*

Если вы откроете итоговый файл i ndex. html, то увидите ссылки на изображения, собранные с сайта, среди которых будут всевозможные неработающие ссылки, если у вас отсутствует подключение к Интернету. Для загрузки всех изображений и других элементов, необходимых для правильного отображения страницы (если сайт позволяет делать это, однако не в рассматриваемом нами случае) используйте параметр `-p`:

$ wget -p http://www.wiley.com Download Web page and other elements

Однако если вы теперь откроете итоговый файл i ndex. html в браузере, то, скорее всего, будут по-прежнему показываться все неработающие ссылки, несмотря на то что все изображения были загружены. Это происходит потому, что ссылки необходимо преобразовать так, чтобы они указывали на локальные файлы. В силу этого взамен введите следующее:

$ wget -pk http://www.wiley.com *Загрузить страницы, использовать имена локальных файлов*

Если вы захотите, чтобы команда wget сохранила оригинальные файлы, а также осуществила преобразование, введите это:

$ wget -pkK http://www.wiley.com *Переименовать файлы, присвоив им локальные имена, сохранить оригинальные файлы*

Иногда загруженный HTML-файл имеет расширение . asp или . cgi, а не . html. Это может привести к тому, что браузер не будет знать, как открыть вашу локальную копию файла. Вы можете сделать так, чтобы команда wget добавляла расширение .html к таким файлам, задействовав параметр-E:

$ wget -E http://cgi-app.org/index.cgi7Examples *Добавлять расширение .html к файлам*

Используя wget, можно осуществить рекурсивное зеркалирование целого сайта. Наряду с копированием файлов и каталогов со всех уровней файловой структуры сервера использование параметра -т обеспечивает применение временных меток и сохранение перечней файлов FTP-каталогов (при условии, что сервер позволяет делать это). Используйте параметр -т осторожно, поскольку в результате может потребоваться много времени и дискового пространства.

$ wget -m http://www.linuxtoys.net

Приведенная далее строка команды, включающая некоторые из описанных выше параметров, создает наиболее пригодную для использования локальную копию сайта:

$ wget -mEkK http://www.linuxtoys.net

Если вам когда-либо доводилось сталкиваться с ситуацией, в которой загрузка файла большого размера (например, файла образа CD или DVD) прерывалась

раньше полного завершения, то, возможно, wget в сочетании с параметром -c окажется для вас спасительным средством. При использовании параметра -c команда wget возобновляет работу с места, где она прекратилась, и **продолжает прерванную загрузку файла.** Например:

```
$ wget http://examp1e.com/DVD.iso              Начать загрузку файла большого размера
...
95%[========== ] 685,251,583 55K/S Загрузка прервалась раньше
                                                полного заверешения
$ wget -c http://example.com/DVD.iso Возобновить загрузку с места.
                                                где она была прервана
...
HTTP request sent, awaiting response... 206 Partial Content
Length: 699,389,952 (667). 691,513 (66M) remaining [text/plain]
```

Благодаря параметру, позволяющему возобновить загрузку (-c), команда wget особенно полезна для пользователей с низкоскоростным подключением к Интернету, которым требуется загрузить файлы большого размера. Если у вас когда-нибудь прерывалась длившаяся несколько часов загрузка файла чуть раньше полного завершения, вы знаете, что я имею в виду (следует отметить, что если вы не задействуете -c, когда будете намереваться возобновить загрузку файла, то этот файл будет сохранен под другим именем — оригинальным, но с добавлением . 1).

Передача файлов с помощью curl

Клиент для URL-приложения (команда curl) поддерживает те же функциональные возможности, что и wget, касательно передачи файлов с использованием веб- и FTP-протоколов. Однако curl также позволяет передавать файлы с применением других популярных протоколов, включая SSH-протоколы (SCP и SFTP), LDAP, DICT, Telnet и File.

Вместо поддержки больших, рекурсивных загрузок (как это имеет место в случае с wget) команда curl предназначена для **однократных передач файлов.** Однако она поддерживает большее количество протоколов (как уже отмечалось ранее) и набор продвинутых функциональных возможностей. Для использования этой команды потребуется установить пакет curl. Вот несколько примеров **передач файлов с помощью curl:**

```
$ curl -0 ftp://kernel.org/pub/1inux/kernel/v1.0/patch[6-83.sign
$ curl -00 ftp://kernel.org/pub/linux/kernel/v2.6/ChangeLog-2.6.{1.4}
$ curl -0 ftp://chris:MyPasswd@ftp.example.com/home/chris/fileA \
    •Q '-DELE fileA'
$ curl -T install.log ftp://chris:MyPasswd0ftp.example.com/tmp/ \
    -Q "-RNFR install.log" -Q "-RNT0 Xinstall.log
$ curl ftp://ftp.kernel.org/pub//              Отобразить содержимое /pub
```

В первых двух командах показано, как использовать квадратные скобки для индикации диапазона [6-8] и фигурные для задания списка {1,4} символов или чисел, которые задействуются для выбора соответствующих файлов.

В третьей строке команды продемонстрировано, как добавить имя пользователя и пароль (chris: MyPasswd), загрузить файл (f i 1 eA) с сервера, а затем удалить этот файл с сервера, когда загрузка будет завершена (-Q ' -DELE fi 1eA').

В четвертом примере осуществляется выгрузка (-T) файла instal 1. log на FTP-сервер. Затем удаленный файл переименовывается в Xinstall .log. В последнем примере curl дается указание отобразить содержимое каталога /pub/, располагающегося по адресу ftp.kernel.org.

Передача файлов с помощью FTP-команд

Ubuntu поставляется вместе со стандартным FTP-клиентом (команда ftp), который работает так же, как и в большинстве UNIX- и Windows-систем. Рекомендую вместо этого задействовать полнофункциональную, дружественную к пользователю команду 1 ftp.

Используя такой FTP-клиент, вы можете начать сеанс работы с FTP-сервером (а не только взять тот или иной файл, как в случае применения wget или curl). Затем вы будете осуществлять навигацию по серверу подобно тому, как делали бы это в локальной файловой системе, получая и передавая документы посредством сетевого подключения.

Для использования команды 1 ftp вам потребуется ввести apt-get instal 1 1 ftp. Вот примеры, как **подключиться к FTP-серверу с помощью lftp:**

```
$ lftp rnirrors.kernel.org                    Анонимное подключение
1 ftp mirrors.kernel.org:~>
$ lftp chrisiaexaniple.com                    Подключение с аутентификацией
lftp example.com:~>
$ lftp -u chris example.com                   Подключение с аутентификацией
Password: ******
lftp example.com:->
$ lftp -u chris,Mypwd example.com             Аутентификация с помощью пароля
lftp example.com:~>
$ lftp                                        Запустить lftp без установки подключения
lftp :~> open mirrors.kernel.org              Установить подключение во время сеанса lftp
lftp mirrors.kernel,org:~>
```

ВНИМАНИЕ----------

Следует избегать использования четвертого примера в реальной жизни. Пароли, вводимые в командной строке, сохраняются как открытый текст в вашем файле bash_history. Их также могут увидеть другие пользователи в выводе ps auwx.

Установив подключение к FTP-серверу, вы сможете задействовать набор команд во время FTP-сеанса. Команды FTP похожи на ге, что применяются в командном процессоре. Как и в интерпретаторе команд bash, вы можете нажимать клавишу Tab для автоматического завершения ввода имен файлов. Команда 1 ftp также позволяет переводить множественные задания в фоновый режим (Ctrl+Z) и возвращать их обратно в приоритетный (wait или fg) во время сеанса. Это придется кстати, когда вам потребуется продолжить обход FTP-сайта, пока будет осуществляться загрузка или выгрузка файлов. Фоновые задания выполняются параллельно.

Введите jobs, чтобы увидеть список выполняющихся фоновых заданий. Для отображения списка команд 1 ftp наберите he! p.

В приведенном далее сеансе-примере 1 ftp показаны **полезные команды, используемые при осуществлении загрузки:**

```
$ lftp mirrors.kerne1.org
lftp mirrors.kernel,org:~> pwd Проверить текущий каталог
ftp://mirrors.kernel.org
lftp mirrors.kernel ,org:~> Is Показать текущий каталог
drwxr-xr-x 8 ftp ftp 4096 Mar 08 21:02 debian
..,drwxr-xr-x 6 ftp ftp 4096 Mar 09 00:11 ubuntu
. . .
lftp mirrors.kernel.org:-> cd ubuntu/dists/quantal Сменить каталог
lftp mirrors.kernel.org:...> get Contents-amd64.gz Загрузить файл
2A113137 bytes transferred in 37 seconds (646.lK/s)
lftp mirrors.kernel.org:.. > <Ctrl+Z> Перевести загрузку в фоновый режим
lftp mirrors.kernel.org:...> mget main/il8n/* Загрузить все файлы
                                                из ma1n/H8n/
lftp mirrors.kernel.org:...> !1s Запустить локальное выполнение Is
lftp mi rrors.kernel.org:...> bookmark add quantal Добавить местоположение
                                                в закладки
lftp mi rrors. kernel .org: ...> quit Закрыть lftp
```

Во время этого сеанса я вошел в систему как анонимный пользователь по адресу mi rrors .kernel .org. После перехода в каталог, который содержал нужный мне файл, я загрузил его с помощью команды get. Нажав сочетание клавиш **Ctrl+Z,** я мог бы продолжить загрузку, одновременно занимаясь другими операциями. Затем я применил команду mget (которая позволяет использовать подстановочные символы, например *), чтобы загрузить все файлы из каталога tain/il8n/.

Любая команда, перед которой указан восклицательный знак (например, ! 1 s), выполняется локальным интерпретатором команд. Команда bookmark сохраняет текущее расположение (в данном случае ftp:/ /mirrors.kernel .org/ubuntu/dists/) под именем quantal, благодаря чему в следующий раз я смогу выполнить lftp quantal, чтобы вернуться в то же место. Команда qui t завершает сеанс.

Далее приведены **полезные команды, применяемые во время сеанса выгрузки lftp с использованием аутентификации.** При этом предполагается, что вы обладаете необходимыми правами доступа к файлам на сервере:

```
$ lftp chrisiaexample.com
Password: *******
lftp example.corn:-> led /home/chris/songs      Перейти в локальный каталог
lftp example.com:-> cd pub/uploads              Перейти в каталог на сервере
lftp example.com:-> mkdir songs                 Создать каталог на сервере
lftp example.com:-> chmod 700 songs             Изменить права доступа к удаленному
                                                каталогу
lftp example.com:-> cd songs                    Перейти в новый каталог
lftp example.com:~> put song.ogg tune.ogg       Выгрузить файлы на сервер
3039267 bytes transferred
lftp example.com:~> mput /var/songs/*           Выгрузить соответствующие файлы
lftp example.com:-> quit                        Закрыть lftp
```

В случае с этим сеансом 1 ftp показано, как вы можете использовать имена команд интерпретатора команд для работы с удаленными каталогами (при условии, что у вас есть соответствующие права доступа). Команды mkdi r и chmod создают каталог и предоставляют права доступа только для вашей учетной записи пользователя. Команда put выгружает один файл или более на удаленный сервер. Команда mput позволяет использовать подстановочные символы с целью выбора соответствующих файлов для загрузки. К другим командам относятся mi ггог (для загрузки дерева каталогов) и mi ггог -R (для выгрузки дерева каталогов).

1 ftp также обеспечивает сценарий интерпретатора команд для неинтерактивных сеансов загрузки — 1 ftpget. Синтаксис 1 ftpget подобен синтаксису команды wget:

$ 1ftpget ftp://mirrors.kerne1.org/ubuntu/dists/quanta1/Re1ease

Имейте в виду, что стандартные FTP-клиенты небезопасны, поскольку вся информация при их работе передается открытым текстом, так что альтернативой для вас, особенно когда безопасность является основным вопросом, будет использование SSH-инструментов для передачи файлов.

SSH-инструменты для передачи файлов

SSH-утилиты относятся к самым важным инструментам в арсенале коммуникационных команд системного администратора, поэтому некоторые более сложные аспекты конфигурирования и использования SSH-утилит будут рассмотрены подробнее в гл. 13. Однако в самой простой форме SSH-утилиты являются инструментами, которые вам следует чаще всего использовать для базовой передачи файлов.

В частности, команда sep сделает бо́льшую часть необходимого вам для передачи файла с одного компьютера на другой, обеспечив при этом безопасность коммуникации путем применения шифрования на этапе как ввода пароля, так и передачи файла. Команда sep заменяет гер в качестве наиболее популярного инструмента для копирования файлов между хостами.

ВНИМАНИЕ----------

При использовании sep вы не получите предупреждающего сообщения перед перезаписью существующих файлов, поэтому убедитесь, что на целевом хосте нет каких-либо нужных вам файлов или каталогов, пути к которым совпадают с путями копирования ваших файлов посредством sep.

Копирование удаленных файлов с помощью sep

Чтобы использовать sep для передачи файлов, на удаленной системе должна быть запущена служба SSH (обычно это серверный демон sshd). Вот **полезные примеры использования команды sep:**

$ sep myfile chris@serverl:/tmp/ *Скопировать myfile на serverl*
Password: ******
$ sep serverl:/tmp/myfi1e . *Скопировать удаленный файл myfile в локальную папку*
Password: ******

Задействуйте параметр -p для **сохранения прав доступа и временных меток** в случае с копируемыми файлами:

```
$ sep -p myfile serverl:/tmp/
```

Если служба SSH окажется сконфигурированной на прослушивание порта, отличного от порта по умолчанию с номером 22, используйте -P в качестве **индикатора этого порта** в строке команды sep:

```
$ sep -P 12345 myfile serverl:/tmp/ Подключиться к определенному порту
```

Для **рекурсивного копирования** из определенной точки удаленной файловой системы задействуйте параметр -r:

```
$ sep -r mydir francois@serverl:/tmp/ Скопировать все содержимое mydir
                                      в удаленный каталог /tmp
```

Команда sep наиболее полезна, когда вы знаете точное местоположение файла (-ов), который необходимо скопировать, однако иногда целесообразнее просматривать и передавать файлы интерактивно.

Копирование файлов с помощью rsync

Как и sep, команда rsync позволяет копировать файлы между удаленными системами с применением инструмента ssh. Однако rsync поддерживает параметры, которые делают ее особенно полезной при осуществлении резервного копирования и синхронизации каталогов по сети.

В приведенном далее примере использования rsync замените local host на имя хоста или IP-адрес удаленной системы, куда требуется выполнить резервное копирование файлов. Благодаря тестированию команды rsync путем копирования файлов на local host вам будет проще разобраться в результатах и понять, как она работает.

Команда rsync позволяет осуществлять **рекурсивное копирование файлов из одного локального каталога на другую систему (локальную или удаленную).** Параметр -a (archive — «архив») обеспечивает рекурсивное копирование и старается сохранить временные метки и права доступа к файлам. Параметр -v используется для генерирования подробного вывода:

```
$ cd /usr/share/doc
$ sudo rsync -av anacron/ chris@localhost:/tmp/anacron Рекурсивное копирование
chris@localhost's password: ********
sending incremental file list
created directory /tmp/anacron
README.Debian
...
```

Введите Is -1 /usr/share/doc/a'nacron /tmp/anacron и обратите внимание, что дата и временные метки были сохранены в случае с файлами, скопированными посредством rsync. Однако поскольку выполнение rsync осуществлялось от имени поль-

зователя chris, он является владельцем всех файлов, скопированных на целевую систему.

Одной из замечательных особенностей команды rsync является то, что она **копирует только изменившиеся файлы,** поэтому, если снова выполните ту же команду rsync после добавления новых файлов, скопируются только новые.

```
$ cd /usr/share/doc
$ sudo touch anacron/testing                    Создать новый файл
$ sudo rsync -av anacron/ chris@localhost:/tmp/anacron/  Повторить выполнение
                                                                rsync
chris@localhost's password:
********
sending incremental file list
./
testing
```

Следует отметить, что только новый файл копируется на удаленную целевую систему. Это делает rsync очень эффективным инструментом резервного копирования. Если вместо использования rsync для резервного копирования вы захотите просто сохранить структуры каталогов синхронизированными, можете добавить параметр - -delete для удаления из целевого каталога всех файлов, которые отсутствуют в исходном. Например:

```
$ sudo rm anacron/testing
$ sudo rsync -av --delete anacron/ chrisPlocalhost:/tmp/anacron/
chris@localhost's password: ********
./
deleting testing
```

При копировании файлов посредством rsync вам также следует подумать, что делать с символически связанными файлами. Если задействовать параметр - -1 inks, скопируются символьные ссылки на файлы; в случае использования - -copy-links на удаленную систему будет скопирован файл, на который в конечном счете указывает символьная ссылка. При использовании - -links вы рискуете вообще не получить копию файла, на который указывает соответствующая ссылка, а если задействуете --copy-links, то рискуете получить слишком много копий одного файла.

Копирование удаленных файлов во время сеансов sftp и lftp

Команда sftp позволяет использовать FTP-подобный интерфейс для **поиска и копирования файлов посредством SSH-протоколов.** Вот пример, как начать сеанс sftp:

```
$ sftp chris@serverl
chris@serverl's password: *****
sftp>
```

Используйте sftp так же, как применяете обычные FTP-клиенты. Введите ? для вывода на экран списка команд. Вы можете сменять удаленные каталоги (cd)

и локальные каталоги (led), проверять текущий удаленный и локальный каталоги (pwd и 1 pwd), отображать удаленное и локальное содержимое (1 s и П s). В зависимости от прав доступа пользователя, от имени которого вы войдете в систему, вам может быть разрешено создавать и удалять каталоги (mkdl г и rmdir), изменять права доступа (chmod) и владение/группу (chown и chgrp) файлов и каталогов.

Можете также задействовать команду 1 ftp (описанную ранее в этой главе) в качестве клиента sftp. Использование 1 ftp привносит некоторые дружественные по отношению к пользователю параметры, например **завершение ввода путей** с помощью клавиши **Tab:**

```
$ 1 ftp sftp://chris@serverl
Password: ********
1 ftp chris@serverl:->
```

Windows- инструменты передачи файлов

Во многих случаях людям требуется получить файлы с Linux-серверов с использованием Windows-клиентов. Если вашей клиентской операционной системой окажется Windows, вы сможете воспользоваться одним из приведенных далее инструментов с открытым исходным кодом для получения файлов с Linux-серверов:

- **WinSCP (http://winscp.net)** — графический клиент sep, sftp и FTP для Windows с использованием протоколов SSH1 и SSH2;
- **FileZilla (http://filezilla.sourceforge.net)** — обеспечивает графические клиентские FTP- и SFTP-службы в Windows, а также предлагает функции FTP-сервера;
- **PSCP (www.chiark.greenend.org.uk/~sgtatham/putty/)** — клиент командной строки sep, являющийся частью комплекта PuTTY;
- **PSFTP (www.chiark.greenend.org.uk/~sgtatham/putty/)** — клиент командной строки sftp, являющийся частью комплекта PuTTY.

Совместное использование удаленных каталогов

И нструменты, описанные до этого момента в текущей главе, обеспечивают элементарный доступ к файлам, при котором установка подключения и передача файлов осуществляются в рамках одного сеанса. Когда же требуется более долгосрочный, постоянный доступ к удаленному каталогу с файлами, полезными могут оказаться службы для совместного использования и монтирования удаленных файловых систем. К таким службам относятся Network File System (NFS) (Сетевая файловая система), Samba и SSHFS.

Совместное использование каталогов с помощью NFS

При условии, что на сервере уже запущена служба NFS (являющаяся частью пакета nfs-kernelserver). вы сможете задействовать команды exportfs и showmount для просмотра доступных и смонтированных совместно используемых каталогов. Совместно используемый каталог монтируется посредством команды mount в сочетании со специальными параметрами. Если вы установите пакет nfs-kernelserver, то Ubuntu запустит службу NFS.

Просмотр и экспорт совместно используемых ресурсов NFS

Если выполнить команду exportfs на сервере NFS, она **отобразит все совместно используемые каталоги,** доступные на этом сервере:

```
$ sudo /usr/sbin/exportfs -v
/export/mysh client.example.com(ro,root_squash.no_subtree_check)
/mnt ,/publ ic *(rw, wdel ay, root_squash. no_subtree_check)
```

Двумя совместно используемыми каталогами в данном случае являются /export/mysh и /mnt/publ i c. Первый доступен только хост-компьютеру cl ient. exampl e. com, второй — всем. Параметры для каждого совместно используемого ресурса заключены в круглые скобки. Первый совместно используемый ресурс доступен только для чтения (no), а запросы от суперпользователя на стороне клиента соотносятся с анонимным идентификатором пользователя (root_squash). Кроме того, проводится менее тщательная проверка прав доступа к файловой системе (no_subtree_check). Второй совместно используемый ресурс допускает монтирование с возможностью чтения/записи (rw), при этом операции записи в случае с ним выполняются с задержкой для повышения производительности, когда ожидаются дополнительные операции записи (wdel ay).

Вы можете **добавлять и модифицировать совместно используемые каталоги NFS** путем внесения изменений в файл /etc/exports. Для вступления изменений в силу введите от имени суперпользователя любое из приведенного далее:

```
$ sudo /etc/init.d/nfs-kernel-server reload  Перезагрузить экспортированные каталоги
$ sudo exportfs -r                           Перезагрузить экспортированные каталоги
$ sudo exportfs -rv                          Перезагрузить совместно используемые ресурсы с генерированием подробного вывода
exporting client.example.com:/export/myshare
exporting *:/mnt/public
```

Вы можете использовать команду showmount на серверной системе Linux для **просмотра, какие совместно используемые каталоги доступны из локальной системы.** Ыапример:

```
$ sudo /usr/sbin/showmount -e
Export list for server.example.com
/export/myshare client.example.com
/mnt/public *
```

Команду showmount можно задействовать на клиентской системе Linux для **просмотра, какие совместно используемые каталоги доступны с выбранного компьютера.** Например:

```
$ sudo /usr/sbin/showmount -e server.example.com
/export/myshare client.example.com
/mnt/public *
```

ПРИМЕЧАНИЕ----------

Если не получается увидеть совместно используемые ресурсы NFS из другой системы, то, возможно, брандмауэр (iptables) блокирует доступ к службе NFS на стороне сервера.

Монтирование совместно используемых ресурсов NFS

Применяйте команду mount для монтирования удаленного совместно используемого ресурса NFS на локальном компьютере. Вот пример:

```
$ sudo mkdir /mnt/server-share
$ sudo mount server.example.com:/export/myshare /mnt/server-share
```

В этом примере показан сервер NFS (server. exampl e. com) и совместно используемый каталог, который располагается на этом сервере (/export/myshare). Локальная точка монтирования, которая должна существовать до монтирования совместно используемого ресурса, указана в конце команды (/mnt/server-share).

Передайте специфичные для NFS **параметры** команде mount, добавив их после параметра -o:

```
$ sudo mount -o rw.hard.intr server.example.com:/export/myshare /mnt/server-share
```

Параметры rw обеспечивают монтирование удаленного каталога с правами доступа с возможностью чтения/записи, при этом предполагается наличие таких прав доступа. В силу задания параметра hard любой, кто прибегнет к совместно используемому ресурсу, увидит сообщение server not responding (сервер не отвечает), когда истечет время ожидания операции чтения или записи. Если это случится, то задание параметра intr позволит вам **прервать «зависший» запрос к удаленному серверу** (для прерывания нажмите комбинацию клавиш **CtrH-C).**

В более старых системах для подключения к совместно используемым ресурсам задействовался протокол NFS версии 3 (nfs3). Для использования протокола NFS версии 4, предназначенного для работы через Интернет и брандмауэры, **обозначьте этот протокол в качестве типа файловой системы** в строке команды следующим образом:

```
$ sudo mount -t nfs4 server.example.com:/ /mnt/server-share
```

ПРИМЕЧАНИЕ----------

В зависимости от используемой версии Ubuntu реализация NFS версии 4 может быть недостаточно устойчивой для производственной системы. Возможно, будет безопаснее и/или надежнее туннелировать более ранние версии NFS посредством SSH. Вы можете найти дополнительную информацию на эту тему в Интернете, осуществив поиск по словосочетанию nfs ssh и, в частности, заглянув по адресу www.howtoforge.com/nfs_ssh_tunneling. Кроме того, посетите страницу http://tldp.org/HOWTO/NFS-HOWTO/security.html, чтобы узнать больше о безопасности NFS.

Совместное использование каталогов с помощью Samba

Samba является реализацией с открытым кодом протокола совместного использования файлов и принтеров Windows, который первоначально назывался SMB (Server Message Block — блок серверных сообщений), а теперь носит название CIFS (Common Internet File System — общая файловая система Интернета). Есть реализация Samba для Linux, а также многих других операционных систем. Для использования Samba вам потребуется установить пакеты samba и samba-doc.

К графическим инструментам для совместного использования, запроса и монтирования совместно используемых каталогов SMB из Windows относится веб-инструмент администрирования Samba SWAT. Чтобы **использовать инструмент SWAT** в I .inux, установите пакет swat. Затем прочтите инструкции по адресу **https://help.ubuntu.com/community/Swat** для получения подробной информации, как можно запустить SWAT.

Команды для работы с совместно используемыми ресурсами Samba можно задействовать для выполнения запросов к серверам SMB, монтирования и совместного использования каталогов.

Просмотр и осуществление доступа к совместно используемым ресурсам Samba

Для **сканирования вашей сети на предмет хостов SMB** введите следующее:

```
$ findsmb
                                *=DMB
                                +=LMB
IP ADDR          NETBIOS NAME WORKGROUP/OS/VERSION
------------------------------------------------------------------------
192.168.1.1      SERVER1      +[MYWORKGROUP] [Unix] [Samba 3.6.3]
```

Для **просмотра текстового представления сетевого окружения** (совместно используемых каталогов и принтеров) задействуйте smbtree:

```
$ sudo smbtree
Password: ******
MYGROUP
  WTHOMPSON            thompson server (Samba, Ubuntu)
     \\THOMPSON\hp2100 HP LaserJet 2100M Printer
     \\THOMPSON\IPC$ IPC Service (thompson server (Samba. Ubuntu))
  WEINSTEIN            Samba Server
     \\EINSTEIN\hp5550 HP DeskJet 5550 Printer
     WEINSTEIN\IPC$ IPC Service (Samba Server)
```

Чтобы **добавить существующего Linux-пользователя в качестве Samba-пользователя,** примените команду smbpasswd:

```
$ sudo smbpasswd -a chris
New SMB password: ******
Retype new SMB password: ******
Added user chris
```

ПРИМЕЧАНИЕ

Вам потребуется задать Samba-пароль для себя, чтобы выполнять любые команды, которые его запрашивают.

Чтобы **отобразить службы, предлагаемые сервером анонимному пользователю,** введите следующее (нажмите клавишу **Enter,** чтобы пропустить ввод пароля):

```
$ smbclient -L server
Enter Chris's password:
Anonymous login successful
Domain=[MYGROUP] 0S=[Unix] Server=[Samba 3.6.3]
        Server               Comment
        -----------          -------
        EINSTEIN
        THOMPSON             thompson server (Samba, Ubuntu)
```

Вот вывод smbclient касательно определенного пользователя с именем **chris:**

```
$ smbclient -L server -U chris
Enter chris's password:
Domain=[WORKGROUP] 0S=[Unix] Server=[Samba 3.6.3]
  Sharename Type Comment
  ---------    ----  -------
  Homes        Disk  Home Directories
  prints       Disk  Printer Drivers
  IPCS           IPC   IPC Service (thompson server (Samba, Ubuntu))
  chris        Disk  Home Directories
 Domain=[WORKGROUP] 0S=[Umx] Server=[Samba 3.6.3]
  Server       Comment
  ---------    -------
  EINSTEIN
  THOMPSON thompson server (Samba, Ubuntu)
  Workgroup            Master
   ---------            -------
   DATAGROUP            MYSERVER
   WORKGROUP            THOMPSON
```

Для **подключения к совместно используемому ресурсу Samba в FTP-стиле** введите следующее:

```
$ smbclient //thompson/homes -U chris
Password: ********
Domain=[WORKGROUP] 0S=[Umx] Server=[Samba 3.6.3]
smb: \>
```

Как и в случае с большинством FTP-клиентов, введите he! p или ?, чтобы увидеть список доступных команд. Более того, вы можете использовать команды, подобные командам командного процессора, например cd, 1 s, get, put и quit, для **работы с хостом SMB.**

Монтирование совместно используемых ресурсов Samba

Вы можете **монтировать удаленные совместно используемые ресурсы Samba в локальную файловую систему** подобно тому, как монтировали бы раздел локального жесткого диска или удатенную файловую систему NFS. Для монтирования совместно используемого ресурса введите следующее:

```
$ sudo mount -t cifs -o username=chris,password=MySecret \
     //192.168.1.1/homes /mnt/mymount/
```

Вы сможете **увидеть текущие подключения и блокировки файлов** на сервере, прибегнув к команде smbstatus. Она позволит узнать, смонтировал ли кто-либо ваши совместно используемые каталоги или задействует ли в настоящее время подключение посредством smbclient к вашему серверу:

```
$ sudo smbstatus
Samba version 3.6.3
PID Username Group Machine
-------------------------------------------
 5466 chris            chris 192.168.1.1 (192.168.1.1)
Service pid machine Connected at
-------------------------------------------
IPC$       30180      192.168.0.145 Sat Mar 9 12:19:28 2013
Chris      30180      192.168.0.145 Sat Mar 9 12:19:28 2013
```

Поиск хостов Samba

Для идентификации хостов в Samba используются NetBIOS-имена. Вы можете **определить IP-адрес компьютера** с помощью команды nmbl ookup, позволяющей выполнить запрос с использованием конкретного NetBIOS-имени в локальной подсети, как показано далее:

```
$ nmblookup thompson
querying thompson on 192.168.1.255
192.168.1.1  thompson<00>
```

Чтобы **найти IP-адрес сервера в определенной подсети,** используйте параметр -U:

```
$ nmblookup -U 192.168.1.255 einstein
querying einstein on 192.168.1.255
192.168.1.1  ei nsteln<00>
```

Проверка конфигурации Samba

Если не получается задействовать совместно используемый ресурс Samba или вы столкнетесь с другими проблемами при взаимодействии с вашим сервером Samba, можете проверить конфигурацию Samba на сервере. Для **проверки вашего**

главного конфигурационного файла Samba (smb.conf) задействуйте команду testparm:

```
$ testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[myshare]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press Enter to see a dump of your service definitions
```

После нажатия клавиши **Enter,** как указано в приведенном выше примере, вы увидите настройки из файла smb. conf. Вот как может выглядеть запись касательно совместно используемого каталога myshare, который задействовался ранее в примере, в файле smb. conf:

```
[myshare]
path = /home/chris
username = chris
valid users = chris
hosts allow = einstein
available = yes
```

Эта запись позволяет Samba-пользователю **chris** осуществлять доступ к каталогу /home/chris (представленному названием совместно используемого ресурса myshare) с хост-компьютера с именем **einstein.** Этот совместно используемый ресурс показан как доступный в настоящее время.

В предыдущем примере использования команды testparm были продемонстрированы записи из файла smb. conf. Однако в нем не **показано никаких записей по умолчанию, которые вы не задали.** Их можно просмотреть, прибегнув к параметру -v. Передайте его по канату команде less для постраничного просмотра соответствующих настроек:

```
$ testparm -v | less
```

Если вы захотите **проверить конфигурационный файл перед тем, как применять его,** то можете дать testparm указание использовать файл, отличный от /etc/samba/smb.conf:

```
$ testparm /etc/samba/test-smb.conf
```

Совместное использование каталогов с помощью sshfs

Еще одним интересным трюком, который можно выполнить посредством протокола SSH, является монтирование удаленных файловых систем. Используя файловую систему SSH (sshfs), вы сможете смонтировать любой каталог с SSH-сервера, к которому имеете доступ из собственной локальной системы Linux согласно

своей учетной записи пользователя, sshfs обеспечивает шифрование операции монтирования, а также всех передаваемых данных.

Другим замечательным аспектом sshfs является то, что этот инструмент не требует установки на стороне сервера (за исключением наличия запущенной службы SSH).

Далее приведена быстрая процедура **монтирования каталога с документами с удаленного сервера в локальный каталог.** Для этого потребуется только чтобы на удаленном сервере была запущена служба SSH и ваша учетная запись пользователя разрешала доступ к нужному каталогу' на сервере. В этом примере монтируется каталог с именем /var/docs с хоста по адресу 10.0.0.50 в точку монтирования с именем /mnt/docs налокадьной системе:

```
$ sudo apt-get install sshfs Установить программное обеспечение sshfs
$ rakdir /var/tmp/chris Создать точку монтирования
$ sshfs chris@10.0.0.5:/home/chris /var/tmp/chris Монтировать удаленный каталог
```

Закончив использовать удаленный каталог, вы можете **демонтировать** его посредством команды fusermount (являющейся частью пакета fuse-utils):

```
$ fusermount -u /var/tmp/chris Демонтировать удаленный каталог
```

Общение с друзьями в чатах посредством IRC

Несмотря на появление возможности мгновенного обмена сообщениями, многие по-прежнему используют IRC (Internet Relay Chat – ретранслируемый интернет-чат). Сегодня существует большое количество глобальных компаний и все больше сотрудников работает удаленно, поэтому IRC стала популярной технологией, помогающей группам людей работать вместе из разных мест.

На сайте **freenode.net** есть множество чат-комНат, посвященных поддержке крупных проектов по разработке программного обеспечения с открытым исходным кодом. Фактически многие люди находятся в них целыми днями и просто наблюдают за ходом дискуссий, которые касаются их любимых Linux-проектов. Это называется **чтением дискуссий без принятия в них участия.**

Утилита xchat. является хорошим графическим IRC-клиентом с поддержкой множества операционных систем. Вы можете установить пакет xchat или GNOME-привязки из пакета xchat-gnome. Для запуска той или иной версии xchat в Ubuntu введите команду xchat на панели управления и выберите значок **XChat IRC** или **XChat GNOME IRC.**

Однако элитный способ общения посредством IRC – это использование работающего в текстовом режиме клиента, который запускается в screen на постоянно включенном компьютере вроде старого сервера. Еще один похожий вариант заключается в использовании прокси-клиента IRC, также называемого *баунсером,,* например dircproxy (который является частью пакета dircproxy).

Оригинальным IRC-клиентом был ircll. Он позволял добавлять сценарии, в какой-то мере похожие на макросы, встречающиеся в комплектах приложений

для продуктивной работы, которые автоматизировали выполнение отдельных команд и повышали удобство пользования. Наиболее популярным был PhoEniX от Vassago. Затем появился BitchX, который сначала был сценарием ircII, а потом превратился в полнофункциональный клиент. Сегодня многие используют irssi. Для **установки и запуска irssi** в Ubuntu введите следующее:

```
$ sudo apt-get install irssi
$ irssi -n JayJoel99x    Начать чат-сеанс irssi
```

В этом примере именем пользователя (называемым *ником* человека) является JayJoel99x (вам потребуется указать свое имя пользователя). Вы должны будете увидеть синюю строку состояния внизу экрана, означающую, что вы находитесь в окне 1, которое является окном состояния. При первом запуске irssi программа отобразит справочные сообщения, в которых вас направят к документации. IRC-командам предшествует символ /. Например, чтобы **подключиться к серверу freenode** (после принятия политик по адресу **http://freenode.net),** вы введете следующее:

```
/connect chat.freenode.net
```

Если вы не добавите свое имя пользователя в строку команды, то подключитесь к **chat.freenode.net** под именем пользователя, которое ввели при входе в систему. В IRC чат-комната называется *каналом*, перед именем которого указывается знак решетки (#). Далее **попытайтесь присоединиться к IRC-каналу #ubuntu:**

```
/join #ubuntu
```

Теперь вы присоединены к каналу в окне 2, о чем свидетельствует строка состояния. Для переключения между окнами irssi используйте комбинации клавиш **Ctrl**+N и **Ctrl+P.** В качестве альтернативы вы могли бы прибегнуть к сочетаниям **Alt+1** или **Alt+2,** однако они не будут работать в окне **gnome-terminal,** поскольку **gnome-terminal** «съедает» нажатия этих клавиш.

Для **получения справочной информации** в любой момент введите /help команда, где вместо слова команда укажите имя команды, о которой нужны дополнительные сведения. Справочный текст отобразится в окне состояния, а не обязательно в текущем.

Для добавления сообщения в IRC-чат просто наберите его и нажмите клавишу **Enter,** чтобы отправить сообщение тем, кто использует соответствующий канал. Для **направления сообщения определенному пользователю в чате** введите первые несколько символов ника этого пользователя и нажмите клавишу **Tab,** чтобы автоматически завершить ввод ника. Затем наберите сообщение в остальной части строки и нажмите **Enter.** После этого ваше сообщение отобразится в окне чата и будет выделено для соответствующего пользователя.

Вы можете **сменить свой ник** в любой момент с помощью команды /nick. Например, чтобы сменить ник на **joe,** введите /nick joe. Чтобы покинуть канал, наберите /part. Для выхода из программы введите /quit.

О irssi можно рассказать еще многое. Вы можете настраивать этот инструмент и значительно улучшать взаимодействие. Дополнительные сведения о его использовании вы можете получить из документации к irssi **(www.irssi.org/documentation).**

Текстовые клиенты электронной почты

В наши дни большинство пользовательских почтовых агентов (Mail User Agent, M UA) обладает GUI-интерфейсом, поэтому если вы начали пользоваться электронной почтой десять лет назад или около того, то, когда речь идет о клиентах электронной почты, вам, вероятно, на ум приходят браузерные клиенты электронной почты (например, Gmail) или автономные графические приложения вроде Evolution, Kmail, Thunderbird или (в Windows-системах) Outlook. Однако в первых UNIX- и Linux-системах чтение электронной почты осуществлялось с использованием текстовых приложений.

Чтобы проверить электронную почту на удаленном сервере или в другой текстовой среде, можете воспользоваться старыми текстовыми почтовыми клиентами, которые все еще полезны. Более того, некоторые яростные фанаты по-прежнему пользуются исключительно текстовыми почтовыми клиентами, расхваливая их эффективность и смеясь над сообщениями на основе HTML.

Описываемые в этой главе почтовые клиенты предполагают, что ваши сообщения хранятся в стандартном формате MBOX на локальной системе. Это означает, что вы либо вошли в систему на почтовом сервере, либо уже загрузили сообщения локально (например, с использованием POP3 или схожего протокола).

В качестве альтернативы можно задействовать текстовые приложения для чтения электронной почты, например команду mutt, которая позволяет подключаться к почтовым серверам POP и IMAP (в том числе с использованием протоколов с поддержкой шифрования) для чтения электронной почты.

ПРИМЕЧАНИЕ

Текстовые почтовые клиенты можно использовать с целью чтения почты, которая уже была загружена с помощью других почтовых клиентов. Например, вы могли бы открыть свой почтовый файл Inbox, связанный с программой Evolution, введя `mail -f $HOME/.evolution/mail/loc/Inbox`.

Управление электронной почтой с помощью mail

Самой старой, а также простейшей в использовании командой, если вам требуется лишь быстро проверить расположенный на удаленном сервере почтовый ящик суперпользователя на предмет сообщений, является mai 1 (bin/mai 1), которая представляет собой часть пакета mailutils.

Прежде чем вы сможете использовать программу командной строки mail, вам потребуется сконфигурировать соответствующий пакет. Есть множество проблем, которые могут возникнуть в случае с почтовыми серверами в зависимости от вашего поставщика услуг Интернета. Процесс конфигурирования запускается как часть установки при вводе следующей команды:

```
$ sudo apt-get install mailutils
```

Команду mail можно использовать интерактивно, но часто она применяется для **отправки электронной почты на основе сценариев.** Вот примеры:

```
$ mail -s 'My Linux version' chris@localhost < /etc/lsb-release
$ ps auwx | mail -s 'My Process List' root@localhost
```

В двух приведенных выше примерах использования mai 1 продемонстрированы быстрые способы отправки текста по почте без необходимости открывать почтовое GUI-приложение. В первом примере содержимое файла /etc/l sb-release отправляется пользователю по адресу chris@localhost. Темой (-s) является 'My Linux Version'. Во втором примере суперпользователю посылается список запущенных на данный момент процессов (ps auwx), тема — 'My Process List[1 * 3 * * &].

При интерактивном использовании команда mail по умолчанию открывает почтовый ящик, определяемый значением SMAIL вашего текущего интерпретатора команд. Например:

```
$ echo $MAIL
/var/mai1/chri s
```

ПРИМЕЧАНИЕ

Вам может потребоваться задать значение для данной переменной среды. Этим значением должно быть /var/mail/имя пользователя (где вместо «имя пользователя» указывается chris или другое имя). В Ubuntu значение для переменной MAIL не задается по умолчанию, поскольку команда mail не устанавливается по умолчанию.

Чтобы прочитать почту суперпользователя, введите следующую команду:

```
$ sudo mail
Mail version 8.1 6/6/93. Type ? for help.
"/var/mail/root": 25 messages 25 new
>11  1logwatch@a.l   Sat  Jun 15 20:03 44/1667   "Logwatch for a (Linux)"
 U   2logwatch@a.l   Sun  Jun 16 04:32 87/2526   "Logwatch for a (Linux)"
     3 logwatch@a.1   Mon  Jun 17 04:32 92/2693   "Logwatch for a (Linux)"
 N   4logwatchtPa.l  Sat  Jun 22 09:28 44/1667   "Logwatch for a (Linux)"
 N 5 MAILER-DAEMON@a Sat Jun 22 09:28 93/3348 "Warning: not sent"
&
```

Рядом с текущим сообщением имеется знак «больше» (>). В начале новых сообщений стоит N, непрочитанных (но не новых) сообщений — U, а если буква отсутствует, это означает, что сообщение было прочитано. Приглашение внизу (&) выражает готовность к приему команд.

На данном этапе вы находитесь в командном режиме. Вы можете использовать простые команды для **навигации** и **выполнения базовых почтовых функций** посредством mail. Введите следующее для реализации этих возможностей:

- ? — отображает список команд;
- номер сообщения — выводит на экран соответствующее сообщение;
- v3- открывает третье сообщение в редакторе vi;
- hl8 — отображает список заголовков сообщений, который начинается с заголовка сообщения под номером 18;

- r7 — позволяет ответить на сообщение номер 7 (наберите сообщение, а затем поставьте точку в самой строке, чтобы отправить его);
- d4 — удаляет четвертое сообщение (или введите d4-9, чтобы удалить сообщения с четвертого по девятое);
- ! bash — позволяет перейти в интерпретатор команд (затем введите exit, чтобы вернуться в mail).

Перед выходом из mai 1 не забудьте, что, когда вы сделаете это, любые просмотренные вами сообщения будут скопированы из вашего файла почтового ящика в ваш файл $H0ME/mbox, если только вы не сохраните их (pre*). Чтобы все сообщения остались в вашем почтовом ящике, нажмите клавишу X для выхода. Чтобы сохранить все уже просмотренные вами сообщения в почтовый ящик, нажмите клавишу Q для выхода.

Вы можете открыть любой файл формата MBOX при использовании mail. Например, если вы войдете в систему от имени того или иного пользователя, но захотите открыть почтовый ящик пользователя chris, введите следующее:

```
$ sudo mail -f /var/mail/chris
```

Управление электронной почтой с помощью mutt

Если вы решите использовать почтовый клиент командной строки на постоянной основе, рекомендую применять mutt вместо mail. Команда mail имеет .множество ограничений, например не позволяет отправлять вложения без их предварительного кодирования (например, с помощью команды uuencode), в то время как mutt обладает множеством функциональных возможностей, позволяющих удовлетворить современные потребности касательно электронной почты. Команда mutt является частью пакета mutt, который вам потребуется установить, чтобы использовать ее. Конфигурируется mutt путем редактирования /etc/Muttrc. Вам также потребуется сконфигурировать Sendmail для обеспечения отправки почты.

Как и mai 1, команду mutt тоже можно использовать для отправки сообщений на основе сценариев. Кроме того, mutt позволяет **отправлять вложения.** Например:

```
$ mutt -s "My Linux Version" -a /etc/lsb-release \
chris0example.com < email-body.txt
$ mutt -s "My Linux Version" -a /etc/lsb-release \
chris0example.com < /dev/null
```

В первом из приведенных выше примеров файл email -body.txt выступает в качестве тела сообщения, а файл /etc/1 sb- rel ease — в роли вложения. Во втором примере отправляется вложение, однако тело сообщения является пустым (< /dev/nul 1).

Вы можете **начать почтовый сеанс mutt** (при условии, что вашим ящиком по умолчанию является SMAIL), просто введя команду mutt:

```
$ mutt
/home/chris/Mai1 does not exist. Create it? ([yes]/no): y
q:Quit d:Del u:Undel s:Save m:Mail r:Reply g:Group ?:Help
   1 0 Jun 16 logwatch0a          ( 69) Logwatch for a (Linux)
   2 0 Jun 18 logwatch0a          ( 171) Logwatch for a (Linux)
```

```
  3  C Jun 18 Mail Delivery S ( 219) Warning: could not send message
  4  0 Jun 19 logwatch@a ( 33) Logwatch for a (Linux)
--Mutt: /var/mail/root [Msgs:22 New:2 01d:20 63K]- -(date/date)--(all)--
```

Команда mutt экранно-ориентирована, поэтому ее проще применять, чем mail. Как и в случае email, для **навигации при использовании mutt задействуются соответствующие команды с клавиатуры.** Нажимайте следующие клавиши для навигации по почтовому ящику:

- O ? — как обычно, эта клавиша используется для получения справочной информации; подсказки будут отображаться на верхней панели, чтобы помочь в работе с почтой;
- O T и *I* — позволяют выделять сообщения, которые вы хотите прочитать;
- O **Enter** — обеспечивает просмотр выделенного сообщения;
- O **Page** Up и **Page Down** — позволяют осуществлять постраничный просмотр каждого сообщения;
- O I — обеспечивает возврат к заголовкам сообщений;
- O / — позволяет осуществлять поиск текста в прямом направлении;
- O **Esc-/** — позволяет выполнять поиск текста в обратном направлении;
- O N — запускает новый поиск;
- O **Tab** — обеспечивает переход к следующему новому или непрочитанному сообщению;
- O **Esc+Tab** — обеспечивает переход к предыдущему сообщению;
- O S — позволяет сохранить текущее сообщение в файл;
- O **D** — позволяет удалить сообщение;
- O U — позволяет отменить удаление сообщения.

Для написания и отправки сообщений по электронной почте вы можете использовать следующие клавиши:

- OM — отправляет новое почтовое сообщение; после того как вы укажете получателя и тему, в JOE (или любом другом редакторе согласно значению, заданному для вашей переменной SEDIT0R) откроется пустое сообщение;
- O A — после выхода из тела сообщения нажмите клавишу A, если захотите добавить вложение;
- O ? — позволяет узнать другие способы манипулирования вашими сообщениями, заголовками или вложениями;
- O Y — отправляет сообщение; если вам потребуется отменить его отправку, то нажмите клавишу Q;
- O X — позволяет выйти без сохранения изменений касательно почтового ящика, когда вы закончите работу;
- O Q — позволяет выйти и сохранить внесенные вами изменения (прочитанные, удаленные сообщения и т. д.).

По умолчанию команда mutt осуществляет чтение вашего локального файла почтового спула, однако она также позволяет подключаться к почтовым серверам

с использованием протоколов POP, POPS, ШАР и IMAPS. Если вы знаете адрес своего почтового сервера, то можете прибегнуть к приведенным далее примерам, как подключиться к серверам электронной почты POPS или IMAPS, задействуя mutt:

```
$ mutt -f imaps://chrisiaimapsserver.example.com
$ mutt -f pops://chris@popsserver.example.com
```

При использовании почтовых протоколов (SMTP), которые требуют сертификатов (например, безопасные протоколы POP и ШАР), вам будет предложен сертификат, который вы сможете принять или отвергнуть после подключения к серверу. Если вы примете сертификат и введете соответствующее имя пользователя и пароль, то сможете читать, писать и отвечать на почтовые сообщения так же, как делачи бы это при работе с почтой из своего локального файла почтового спула.

Резюме

Команды сетевого доступа представляют собой быстрые и эффективные инструменты, позволяющие получать по сети нужное вам содержимое. Браузер ELinks является популярным экранно-ориентированным приложением для поиска и просмотра информации в Интернете или быстрого ознакомления с любым HTML-файлом. В вашем распоряжении есть множество команд для загрузки файлов посредством FTP, HTTP, SSH или других протоколов, в том числе wget, curl, lftp и sep.

В этой главе было рассмотрено, как использовать командные инструменты NFS, Samba и sshfs для постоянного доступа к удаленным каталогам с файлами. Вы можете общаться в IRC-чатах, которые популярны среди участников проектов по разработке программного обеспечения с открытым исходным кодом, используя команду i rssi. Что касается текстовых клиентов электронной почты, то здесь вам на выбор предоставляются такие команды, как mai 1 и mutt.

13 Удаленное системное администрирование

В этой главе:

- конфигурирование SSH;
- использование SSH для удаленного входа в систему;
- применение SSH для туннелирования;
- использование SSH для обеспечения прокси-службы;
- применение SSH в сочетании с закрытыми ключами;
- использование команд byobu и screen для управления удаленными интерпретаторами команд;
- осуществление доступа к удаленному Рабочему столу Windows с помощью различных команд;
- совместное использование удаленного рабочего стола Linux посредством VNC.

Большинство профессиональных Linux-администраторов не используют графический интерфейс на интернет-серверах. В силу этого, если вам потребуется доступ к другим компьютерам для удаленного администрирования, то в какой-то момент почти наверняка придется работать из командной строки. К счастью, есть множество функционально богатых Linux-команд, которые помогут вам в этом.

Инструменты, ассоциированные со службой SSH, не только позволяют осуществлять удаленный вход в систему и передачу файлов, но и предоставляют зашифрованные коммуникации для обеспечения защиты при выполнении вами работы, связанной с удаленным администрированием. Используя такие средства, как VNC (Virtual Network Computing — вычисления посредством виртуальной сети), вы сможете сделать так, чтобы удаленный рабочий стол сервера оказался доступен на вашем локальном клиентском компьютере.

Описание этой и другой функциональности, связанной с удаленным системным администрированием, вы найдете в текущей главе.

Удаленный вход в систему и туннелирование посредством SSH

Система UNIX, являющаяся старшей сестрой Linux, взрослела в университетских сетях. Во времена, когда этими сетями пользовались только студенты и профессора, а сети были по большей части изолированы друг от друга, необходимость в обеспечении их безопасности отсутствовала.

Приложения и протоколы, созданные в те времена (в 1970-1980-е годы), отражают недостаточную заботу о шифровании и аутентификации. Отличным тому примером является SMTP. То же можно сказать и о первом поколении UNIX-инструментов для удаленной работы: telnet, ftp (File Transfer Protocol — протокол передачи файлов), rsh (Remote Shell — удаленный интерпретатор команд), rep (Remote Copy — удаленное копирование), rexec (Remote Execution — удаленное выполнение) и rlogin (Remote Login — удаленный вход в систему). В случае применения этих инструментов учетные данные пользователей и трафик передаются открытым текстом. По этой причине их очень опасно использовать в такой общедоступной и ненадежной сети, как Интернет, и они главным образом были признаны устаревшими и заменены командами SSH (командами ssh, sep, sftp и соответствующими службами).

Унаследованным командам для удаленной работы все еще находится некоторое применение (см. раздел «Унаследованные коммуникационные инструменты» данной главы, большая часть этого раздела посвящена описанию использования SSH-команд для удовлетворения большинства ваших потребностей касательно удаленных коммуникаций).

Унаследованные коммуникационные инструменты

Несмотря на то что SSH обеспечивает лучшие инструменты для удаленных коммуникаций, унаследованные коммуникационные команды, иногда называемые r-командами, по-прежнему включаются в большинство крупных дистрибутивов Linux. Некоторые эти средства работают быстрее эквивалентных SSH-команд, поскольку им не нужно осуществлять шифрование, поэтому UNIX-администраторы старой закалки могут время от времени использовать их в частных сетях или все еще включать в старые сценарии. По большей части вам не следует использовать эти унаследованные команды для удаленной работы, однако одна из них может оказаться полезной в некоторых случаях — telnet.

Команда telnet по-прежнему применяется для взаимодействия с сетевыми устройствами (маршрутизаторами, коммутаторами, UPS-источниками и т. д.), у которых недостаточно мощности для запуска демона ssh. Несмотря на то что она ставит под угрозу безопасность, некоторые производители устройств все равно закладывают в них поддержку tel net.

Один из хороших способов применения команды tel net заключается в ее использовании для устранения неполадок со многими интернет-протоколами, например

POPS, SMTP, HTTP и другими. По сути, эти текстовые протоколы используются просто для обеспечения автоматических сеансов telnet, во время которых клиент (например, браузер или пользовательский почтовый агент) обменивается текстовыми данными с сервером. Единственное отличие состоит в задействуемом TCP-порте. Вот пример, как вы могли бы использовать telnet через HTTP-порт (80) веб-сервера:

```
$ telnet www.example.com 80
Trying 208.77.188.166...
Connected to www.example.com.
Escape character is '^A]'.
GET / HTTP/1.0
        Введите здесь второй символ возврата каретки
НИР/1.1 200 ОК
```

Аналогичным образом вы можете использовать telnet в случае с почтовым сервером через порт 25 (SMTP) и 110 (POP3) и вводить соответствующие команды для устранения неполадок с электронной почтой.

ПРИМЕЧАНИЕ

Более полное описание использования команды telnet для устранения неполадок с сетевыми протоколами вы сможете найти в книге «Библия пользователя Linux по устранению неполадок» (Linux Troubleshooting Bible) (под авторством Кристофера Негуса и Томаса Викса, Wiley Publishing, 2004), на страницах 505-508.

Если вам потребуется принудительно завершить сеанс telnet, воспользуйтесь управляющей последовательностью (по умолчанию ею является комбинация клавиш **Ctrl+]).** В результате клавиатурный ввод перестанет отправляться на удаленную сторону, а также отобразится приглашение командной строки telnet, где вы сможете ввести quit или ?, чтобы увидеть дополнительные параметры.

Конфигурирование SSH

В настоящее время своего рода швейцарским армейским ножом для удаленного системного администрирования является SSH. SSH-команды и службы заменяют все старые инструменты для удаленной работы и привносят стойкое шифрование, открытые ключи и множество других возможностей. Наиболее распространенной в мире Linux реализацией SSH является OpenSSLI **(www.openssh.com),** сопровождением которой занимаются участники проекта OpenBSD. OpenSSH обеспечивает как клиентский, так и серверный компоненты.

Если он еще не установлен, то инсталлируйте OpenSSH-сервер, введя следующую команду:

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

Вот несколько фактов о SSH.

- В Windows вы можете применять SSH-инструменты Linux в рамках Cygwin **(www.cygwin.com).** Однако, если вы еще не используете Cygwin (Linux-подобная среда для Windows), рекомендую задействовать PuTTY **(www.chiark.greenend.org**

/uk/sgatatham/putty). PuTTY представляет собой мощный Telnet/SSH-клиент с открытым исходным кодом.

- По возможности используйте SSH версии 2, поскольку она наиболее безопасна. Некоторые сетевые устройства, работающие с SSH, могут поддерживать только более старые, менее безопасные версии. OpenSSH поддерживает все версии. Некоторые ранние версии Ubuntu принимали подключения SSH версии 1 и 2. Более новые выпуски принимают подключения SSH версии 2 по умолчанию.
- Чтобы **запустить службу SSH** (демон sshd) в Ubuntu, введите service ssh start. **Конфигурирование этой службы** осуществляется путем редактирования файла /etc/ssh/ssh_config.
- Для **конфигурирования SSH-клиента** необходимо отредактировать файл /etc/ssh/ssh_config.

Если вы предпочитаете использовать графические инструменты для администрирования своей удаленной системы Linux, можете активизировать *туннелирование X11* (также называемое *переадресацией портов X11).* Активизировав туннелирование XII (как на SSH-клиенте, так и на сервере), вы сможете запустить приложение X на сервере, которое при этом будет отображаться на клиенте. Все коммуникации с использованием этого подключения будут шифроваться.

По умолчанию в Ubuntu переадресация XI1 активизирована (XIIForwarding yes) в случае с сервером (демон sshd). Однако вам придется самостоятельно активизировать ее на стороне клиента. Для **активизации переадресации XII на клиенте на один сеанс** подключитесь с использованием следующей команды:

```
$ ssh -X chris@myserver
```

Чтобы **перманентно активизировать переадресацию XII для всех пользователей,** добавьте строку ForwardXll yes в /etc/ssh/ssh_config. Для ее постоянной активизации только для определенного пользователя добавьте аналогичную строку в ~. ssh/config этого пользователя. Чтобы проверить работоспособность туннелирования, введите xclock, подключившись к удаленному компьютеру с использованием ssh, после чего вы должны увидеть соответствующий результат на своем клиентском рабочем столе. Для использования этих команд вам потребуется установить пакет xl 1-apps.

SSH-туннелирование является отличным способом безопасного использования графических инструментов для удаленной работы!

Удаленный вход в систему посредством SSH

Для **безопасного входа в систему на удаленном хосте** вы можете использовать любой из двух разных синтаксисов, позволяющих указать имя пользователя:

```
$ ssh -1 chris myserver
$ ssh chris@myserver
```

Однако команды sep и sftp (рассмотренные в гл. 12) поддерживают только синтаксис имя пользователя@имя сервера, поэтому рекомендую привыкнуть к нему. Если вы не укажете имя пользователя, то ssh попытается обеспечить удаленный

вход в систему от имени того же пользователя, от имени которого вы вошли в систему локально. Если после подключения потребуется **принудительно завершить сеанс ssh,** воспользуйтесь управляющей последовательностью в виде тильды с точкой (~.). Если она не сработает, нажмите комбинацию клавиш **Ctrl+C.**

Подключение к SSH через другой порт

Из соображений безопасности на удаленном хосте **служба SSH может вести прослушивание другого порта,** а не того, который является портом по умолчанию и имеет номер 22. Если это случится, используйте команду ssh в сочетании с параметром -p для установки контакта с этой службой:

```
$ ssh -p 12345 chris@turbosphere.com Подключиться к SSH через порт 12345
```

Использование SSH для осуществления туннелирования (переадресации портов XII)

Если сконфигурировать SSI 1-туннелирование так, как было описано ранее, то служба SSII будет перенаправлять клиентов X Window System на ваш локальный экран. Однако туннелирование также можно использовать наряду с другими протоколами на основе TCP.

Туннелирование для клиентов XII

Приведенная далее последовательность команд предназначена для **начала сеанса ssh** и последующего **запуска нескольких приложений X, чтобы они отобразились на локальном рабочем столе.**

```
$ ssh -X chrisftnyserver                Установить SSH-подключение к myserver
chris@myserver's password: *******
[chris@myserver ~}$ echo SDISPLAY       Показать текущее значение $DISPLAY для X
local host:10.0                         SSH задает для $D1 SPLAY
                                        значение local host:10.0
[chris@myserver ~}$ xeyesS              Показать «глаза», следящие
                                        за указателем мыши на рабочем столе
[chris@myserver ~}$ gnome-calculators   Использовать gnome-calculator
[chris@myserver ~}$ gksu system-config - printers Сконфигурировать принтеры
```

Туннелирование для удаленного администрирования системы печати CUPS

XII — не единственный протокол, который можно туннелировать посредством SSH. Вы можете осуществлять **переадресацию любого TCP-порта** с использованием SSH. Это отличный способ быстрого и легкого конфигурирования безопасных туннелей. При этом никакого конфигурирования на стороне сервера не потребуется.

Например, myserver является принт-сервером с активизированным веб-интерфейсом пользователя службы печати CUPS (работает через порт 631). Этот GUI-интерфейс доступен только с локального компьютера. В приведенном далее при-

мере на персональном компьютере осуществляется туннелирование к этой службе с использованием ssh в сочетании со следующими параметрами:

```
$ ssh -L 1234: local host:631 myserver
```

В этом примере выполняется переадресация порта 1234 на клиентском компьютере на **localhost** с номером 631 на сервере. Теперь вы можете открыть **http://localhost:** 1234 на клиентском компьютере. В результате произойдет перенаправление к приложению cupsd, которое прослушивает порт 631 на сервере.

Туннелирование к интернет-службе

Если окажется, что ваш локальный компьютер не позволяет подключиться к Интернету, однако у вас будет возможность связаться с другим компьютером (myserver), который к нему подключен, то SHH-туннелирование сможет **обеспечить интернет-доступ в случае с локальным компьютером.** В приведенном далее примере показано, как вы можете посетить сайт по адресу **google.com** (HTTP, TCP-порт 80) посредством SSH-подключения к компьютеру с именем myserver, который обеспечивает доступ в Интернет:

```
$ ssh -L 12345:google.com:80 chris@myserver
```

При использовании команды из этого примера любое подключение через локальный порт 12345 будет направляться по SSH-туннелю на myserver, который, в свою очередь, установит подключение через порт 80, чтобы зайти на сайт по адресу **google.com.** Эта команда попытается обеспечить вход в систему от имени пользователя **chris** (если не указать имя пользователя, то на удаленной системе будет задействовано имя текущего).

Теперь вы можете открыть **http://localhost: 12345** и использовать myserver как ретранслятор к сайту по адресу **google.com.** Вы задействуете ssh только для перенаправления порта, а не для того, чтобы добраться до интерпретатора команд на сервере, поэтому можете добавить параметр -N, чтобы **предотвратить выполнение команд для удаленной работы:**

```
$ ssh -L 12345:google.com:80 -N myserver
```

Использование SSH в качестве SOCKS-прокси

В приведенном выше примере было показано, что вы можете осуществлять переадресацию порта с клиента на компьютер, отличный от сервера. В реальном мире наилучшим способом **вывести трафик браузера за пределы вашей локальной сети** по зашифрованному туннелю является использование встроенного в SSH компонента SOCKS-прокси. Например:

```
$ ssh -D 12345 myserver
```

Динамический параметр (-D) команды ssh позволяет вам войти в систему на myserver (как обычно). При установленном подключении все запросы, направляемые через порт 1235, затем будут перенаправляться на myserver. Далее настройте свой излюбленный браузер на использование порта localhost с номером 1235

в качестве SOCKS-прокси версии 5, и все готово. Ничего не вводите в поля, касающиеся HTTP и других протоколов. Они все будут работать посредством SOCKS. На рис. 13.1 изображено окно **Connections Settings** (Параметры соединения) в браузере Firefox.

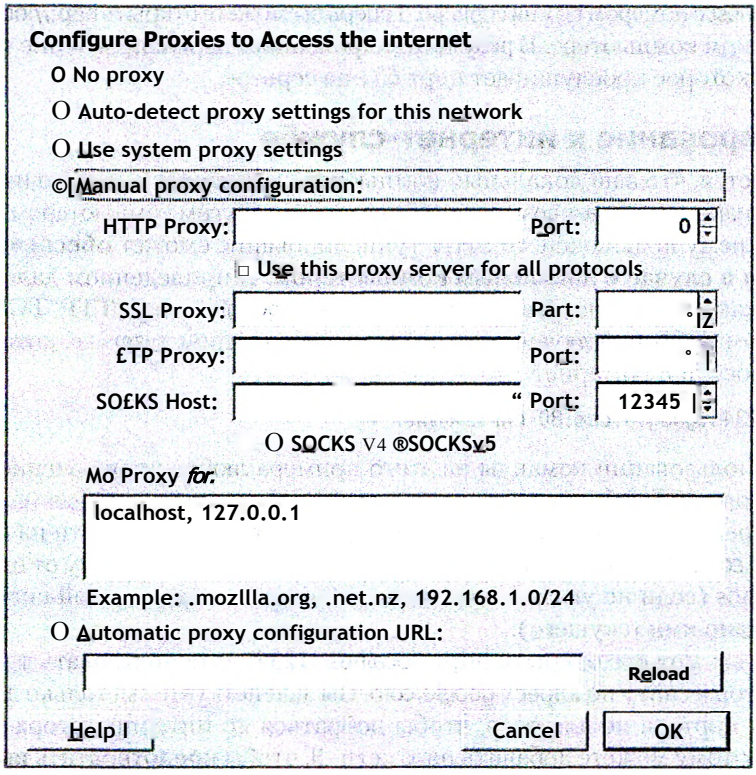

Рис. 13.1. Использование окна Connections Settings (Параметры соединения) в браузере Firefox для конфигурирования прокси

Чтобы протестировать собственную конфигурацию, попробуйте завершить сеанс ssh и зайти на любой сайт. Ваш браузер должен вывести сообщение об ошибке прокси.

При использовании Windows-клиента аналогичную переадресацию портов можно обеспечить в PuTTY, выполнив команду **Connection** (Подключение) ► **SSH Tunnels** (Туннели).

Применение ssh в сочетании с аутентификацией с использованием открытого ключа

До сих пор мы задействовали ssh только наряду с аутентификацией на основе пароля по умолчанию. Однако команда ssh также поддерживает аутентификацию с использованием открытого ключа. Такой подход обеспечивает некоторые преимущества.

- **Автоматический вход в систему для сценариев и заданий cron** — задав пустую фразу-пароль, вы сможете использовать ssh в сценарии для автоматического входа в систему. Это удобно, но в то же время небезопасно, поскольку любой, кто получит ваш файл ключа, сможет подключиться к любому компьютеру, к которому способны подсоединиться вы сами. Конфигурирование автоматического входа в систему также можно осуществить с использованием фразы пароля и агента для управления ключами. Это компромисс между удобством и безопасностью.
- **Двухфакторная аутентификация** — при использовании ключа с защитой фразой-паролем для интерактивного входа в систему аутентификация осуществляется на основе двух факторов (ключа и фразы-пароля) вместо одного.

Вход в систему с использованием общего ключа

Рассмотрим процесс **настройки коммуникаций на основе ключей** между двумя Linux-системами. В приведенном далее примере используются пустые фразы-пароли для беспарольного входа в систему. Если вы предпочитаете защитить свой ключ фразой-паролем, то просто введите ее, когда получите соответствующее приглашение во время первого этапа (создание пары ключей).

Выполните команду ssh-keygen на стороне клиента, чтобы **сгенерировать пару ключей,** после входа в систему от имени пользователя, которому необходимо инициировать коммуникации:

```
$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/chris/.ssh/id_rsa): «нажмите Enter>
Enter passphrase (empty for no passphrase): <нажмите Enter>
Enter same passphrase again: «нажмите Enter>
Your identification has been saved in /home/chris/.ssh/id_rsa.
Your public key has been saved in /home/chris/,ssh/id_rsa.pub.
The key fingerprint is:
ac: db: a4: 8e: 3f: 2a: 90: 4f: 05: 9f: b4:44:74: Oe: d3: db chri s@host. exampl e. com
```

Следует отметить, что при каждом приглашении было необходимо нажать клавишу **Enter,** чтобы создать файловое имя ключа по умолчанию и ввести (и проверить) пустую фразу-пароль. Теперь у вас имеется закрытый ключ, который вам необходимо беречь, поскольку во время этой процедуры вы не защитили его фразой-паролем.

Кроме того, у вас есть и открытый ключ (id_rsa.pub), который был создан посредством предыдущей команды. Его необходимо установить на хостах, к которым вы желаете подключаться. Содержимое ~/. ssh/i d_rsa. pub потребуется скопировать (безопасно) в ~/. ssh/authori zed_keys для пользователя, с кем вы хотите связываться на удаленном сервере посредством ssh. Файл authori zed_keys может содержать более одного открытого ключа, если сразу несколько пользователей будут задействовать ssh для подключения к этой учетной записи.

Раньше копирование собственного открытого ключа в учетную запись пользователя на другой системе осуществлялось вручную, однако теперь вы можете легко сделать это с помощью команды ssh-copy-id. Войдя в систему под учетной

записью пользователя, под которой вы создавали ключ, введите приведенную далее команду, чтобы скопировать свой открытый ключ в файл authorized_keys пользователя на другой системе:

```
$ ssh-copy-id chris@hostl.example.com
chris@hostl's password:
*********

Now try logging into the machine, with "ssh 'chris@host1'".
and check in:
  ~/.ssh/authorized_keys
to make sure we haven't added extra keys that you weren't expecting.
```

В данном примере вы получаете приглашение ввести пароль удаленного пользователя **chris.** После того как вы введете его, это должен быть последний раз, когда вам приходится вводить этот пароль для входа в систему сначала под своей учетной записью локального пользователя, а затем — учетной записью удаленного пользователя.

Войдите в систему под учетной записью удаленного пользователя и загляните в файл authorized_keys, как сказано в сообщении из приведенного выше примера. Вы сможете автоматически войти в систему без получения приглашения ввести пароль. Вот пример:

```
$ ssh chris@hostl
Last login: Wed Mar 20 11:48:58 2013
$ cat .ssh/authorizedjceys
ssh-rsa AAAAB3NzaClyc2EAAAADAQABABABAQCdinPlzlSjW5IfsbfGBk05VDLb+5hNl
PaJhTJLv+8478yox4NlUb0KhEhjVZqcEd8YFpMgUjJqkEQ0IV0+TneGyFPDW5666k5nKo
cgAlw21VELGfHZtTMSlAaq0vwK/bUE2+rrDgwo+Y1KhaA+/2DPaU4XRcnI86yS4NEdENi
PAfoYIZj8DTMI+G0KnHqunIfkiP9g8wnvKIrJsF0JSi62ZeQ2CbXp3PQCAASNSt2r/o0S
tXI9rpdIwLPuwKkSj+FDJaFX5cbbecG5m6uH5IcGbTjeZgfl(JpWNZEK/AkhA2E8HEta/D
4HqtbqfeqNq48x5zGEgyQphwqn8cDlnEYc7 chri s@mydesktop
```

На данном этапе вы также сможете выполнить любую из SSH-команд (sep, sftp, rsync и т. д.) без получения приглашения ввести пароль из локальной системы на удаленной системе. Однако имейте в виду, что если вы зададите фразу-пароль для своего ключа, то вас попросят ввести ее перед тем, как ключ можно будет использовать для подключения к удаленной системе.

Сохранение закрытых ключей для использования с флеш-диска USB

Если вы захотите **сохранить свой закрытый ключ** в более надежном месте, чем жесткий диск, можете воспользоваться *флеш-диском USB* (называемым также флеш-накопителем или флешкой):

```
$ mv -/.ssh/id rsa /media/THUMBDRIVEl/myprivatekey
```

Когда вам потребуется задействовать этот ключ, вставьте USB-диск и введите следующее:

```
$ ssh -i /media/THUMBDRIVEl/myprivatekey chris@myserver
```

Использование ключей с фразами-паролями обеспечивает большую безопасность, чем применение простых паролей, но в то же время является более обременительным. Чтобы облегчить себе жизнь, можете использовать ssh-agent для **хранения разблокированных ключей на протяжении своих сеансов.** Добавив разблокированный ключ в запущенное приложение ssh-agent, вы сможете выполнять ssh с использованием этого ключа без получения каждый раз приглашения ввести фразу-пароль.

Чтобы увидеть, что делает команда ssh-agent, введите ее без параметров. Отобразится трехстрочный сценарий bash:

```
$ ssh-agent
SSH_AUTH_S0CK=/tmp/ssh-SkEQZ18329/agent.18329: export SSH_AUTH_SOCK:
SSH_AGENT_PID=18330: export SSH_AGENT_PID:
echo Agent pid 18330:
```

Первые две строки приведенного выше вывода должны быть выполнены вашим интерпретатором команд. Скопируйте и вставьте их в свой интерпретатор команд прямо сейчас. Вы можете избежать этого дополнительного шага, если запустите выполнение команды ssh-agent и дадите интерпретатору команд bash указание оценить ее вывод, введя следующее:

```
$ eval 'ssh-agent'
Agent pid 18408
```

Теперь вы можете разблокировать ключи и добавить их в свой запущенный агент. При условии, что вы уже выполнили команду ssh-keygen для создания ключа по умолчанию, давайте **добавим этот ключ по умолчанию** с помощью команды ssh-add:

```
$ ssh-add
Enter passphrase for /home/chris/.ssh/id_rsa: *******
Identity added: /home/chris/.ssh/id_rsa (/home/chris/.ssh/id_rsa)
```

Далее вы можете добавить ключ, сохраненный на флешке:

```
$ ssh-add /media/THUMBDRIVEl/myprivatekey
```

Задействуйте ssh-add в сочетании с параметром -1 для **вывода на экран ключей, хранящихся в агенте:**

```
$ ssh-add -1
2048 f7:Ь0:7a:5a:65:3c:cd:45:Ь5:1c:de:f8:26:ee:8d:78
/home/chris/.ssh/i d_rsa
(RSA)
2048 f7:b0:7a:5a:65:3c:cd:45:b5:lc:de:f8:26:ee:8d:78
/mdU/THUMBDRIVE1 /mypri vatekey (RSA)
```

Чтобы **удалить один ключ из агента,** например взятый с флешки, выполните ssh-add с параметром -d, как показано далее:

```
$ ssh-add -d /media/THUMBDRIVEl/myprivatekey
```

Для **удаления всех ключей, хранящихся в агенте,** задействуйте параметр -D:

```
$ ssh-add -D
```

Команды byobu и screen для управления удаленными интерпретаторами команд

Команда ssh открывает только один экран (интерпретатор команд). Если вы закроете этот экран, то пропадет все, что вы делали на удаленном компьютере. Это может иметь серьезные негативные последствия, если случится в разгар чего-то важного, например 12-часового компилирования. Если вы захотите выполнить три команды за один раз, в частности vi httpd.conf, tail -f error_log и service httpd reload, то потребуется начать три отдельных сеанса ssh.

По традиции команда screen используется для обеспечения множественных активных сеансов интерпретатора команд в рамках одного сеанса входа в систему. Используя screen, вы также сможете отсоединиться от сеанса screen, а затем снова присоединиться к нему, чтобы продолжить с места, где вы остановились, когда следующий раз войдете в систему.

Вместо использования screen напрямую вы можете прибегнуть к команде byobu, которая предлагает упрощенный интерфейс и более широкие возможности, чем screen. В следующих разделах рассматривается, как использовать команду screen напрямую, а затем следует описание команды byobu.

Управление удаленными интерпретаторами команд с помощью screen

По сути, screen представляет собой мультиплексор терминалов. Если вы являетесь системным администратором, работающим на удаленных серверах, то screen станет отличным инструментом для управления тем или иным удаленным компьютером, где доступен только интерфейс командной строки. Помимо обеспечения множественных сеансов интерпретатора команд screen также позволяет отсоединиться от сеанса, а затем снова присоединиться к нему же позднее.

Программный пакет screen устанавливается вместе с Ubuntu по умолчанию.

Для использования screen запустите выполнение команды ssh с клиентской системы, чтобы подключиться к Linux-серверу, на котором была произведена установка screen. Затем введите следующую команду:

```
$ screen
```

Если вы запустите выполнение screen из окна терминала, то сначала увидите сообщение-приветствие с предложением угостить разработчиков пиццей и пивом, а затем — обычное приглашение bash в окне. В следующем сообщении вам предложат взамен задействовать byobu (как описывается в подразделе «Команда byobu для управления удаленными интерпретаторами команд» данной главы).

Для управления screen нажмите сочетание клавиш Ctrl+A, а затем еще одну клавишу. Например, нажатие Ctrl+A и следом ? (упоминается как Ctrl+A, ?) отображает справочный экран. Далее приведены команды и клавиши управления, которые вы сможете использовать для манипулирования screen во время ее выполнения.

```
$ screen -Is                                  Показать активные сеансы screen
There is a screen on:
        7089.pts-2.myserver (Attached)        Свидетельствует о том,
                                              что сеанс screen присоединен
1 Socket in /var/run/screen/S-chris.
$ Ctrl+A, Shift+A                             Изменить заголовок окна
Set window's title to: My Server              Задать новый заголовок
$ Ctrl+A, C                                   Создать новое окно
$ Ctrl+A, "                                   Показать заголовки активных окон
Num Name                Flags
  0 My Server                                 Клавиши t и I позволяют сменять окна
  1 bash
$ Ctrl+A, D                                   Отсоединить сеанс screen от терминала
$ screen -Is                                  Показать активные сеансы screen
There is a screen on:
7089.pts-2.myserver (Detached)                Свидетельствует о том,
                                              что сеанс screen отсоединен
1 Socket in /var/run/screen/S-chris.
```

Результатом сеанса screen из приведенного выше примера является создание двух окон (в каждом из которых запущен интерпретатор команд bash). Вы можете создать любое количество окон и присвоить им выбранные вами имена. Кроме того, вместо отсоединения от сеанса screen можно просто завершить его, выйдя из интерпретатора команд в каждом из открытых окон (путем ввода exit или нажатия комбинации клавиш **Ctrl+D).**

Отсоединившись от сеанса screen, вы вернетесь в интерпретатор команд, который был открыт, когда вы в первый раз вошли в систему на сервере. Вы сможете снова присоединиться к этому сеансу screen так, как описано в подразделе «Повторное присоединение к сеансу screen» данной главы.

В приведенном далее списке перечислены другие последовательности нажатия клавиш управления, к которым можно прибегнуть в случае с инструментом screen:

- **Ctrl+A,** ? — показать справочный экран;
- **Ctrl+A, C** — создать новое окно;
- **Ctrl+A, D** — отсоединить сеанс screen от терминала; этот сеанс screen и соответствующие окна продолжат работать;
- **Ctrl+A,"** — просмотр списка окон;
- **Ctrl+A,'** — отобразить приглашение ввести номер или имя окна, на которое необходимо переключиться;
- **Ctrl+A,** N — просмотр следующего окна;
- **Ctrl+A, P** — просмотр предыдущего окна;
- **Ctrl+A,** [— вертикальная прокрутка терминала деактивизирована в screen; эти клавиши активизируют режим обратной прокрутки в screen; для выхода дважды нажмите **Enter;**
- **Ctrl+ A, Shift+A** — переименовать текущее окно;
- **Ctrl+A, W** — показать список имен окон в строке заголовка.

Повторное присоединение к сеансу screen

Отсоединившись от сеанса screen, вы сможете снова присоединиться к нему позднее (даже после того, как выйдете из системы и отключитесь от сервера). Чтобы снова **присоединиться к одному начатому сеансу screen,** введите следующее:

```
$ screen -г
```

Если же начато несколько сеансов, то screen -г не сработает. Например, далее показано, что произойдет, если имеют место два отсоединенных сеанса screen:

```
$ screen -г
There are several suitable screens on:
        7089.pts-2.myserver (Detached)
        7263.pts-2.myserver (Detached)
Type "screen [-d] -r [pid.]tty.host" to resume one oT them.
```

Как следует из вывода, вы можете обозначить нужный сеанс screen посредством его имени (которым по умолчанию является комбинация из идентификатора процесса сеанса, имени tty и имени хоста). Например:

```
$ screen -г 7089.pts-2.myserver
```

Присваивание имен сеансам screen

Вместо использования имен по умолчанию вы можете **создавать более описательные имена для своих сеансов screen** при запуске выполнения команды screen. Например:

```
$ screen -S mysession
$ screen -Is
There is a screen on:
        26523.mysession (Attached)
```

Совместное использование сеансов screen

Команда screen также позволяет **совместно использовать сеансы screen.** Эта функциональная возможность отлично подходит для технической поддержки, поскольку каждый человек, присоединенный к этому сеансу, сможет осуществлять как ввод, так и наблюдение за текущим сеансом. Присваивание имени сеансу screen, как было показано в предыдущем разделе, облегчает процесс. Таким образом, кто-то еще, работающий на другом компьютере, сможет подключиться к серверу посредством ssh (задействовав аналогичное имя пользователя) и ввести следующее:

```
$ screen -x mysession
```

Как и при использовании screen -г, если начат только один сеанс screen, не потребуется указывать, к какому именно сеансу screen вы хотите присоединиться:

```
$ screen -x
```

Команда byobu для управления удаленными интерпретаторами команд

Команда byobu была создана сообществом Ubuntu для обеспечения уровня конфигурации для screen. Ее возможности шире, а интерфейс проще, к тому же последний больше полагается на использование функциональных клавиш, чем **Ctrl.** У нее также имеется набор замечательных дополнительных качеств, например способность показывать информацию о состоянии компьютера, на котором вы вошли в систему.

Для запуска выполнения byobu во время сеанса удаленного интерпретатора команд войдите в удаленную систему с использованием ssh, а затем введите либо byobu, чтобы **немедленно начать сеанс,** либо byobu -enabl e, чтобы сделать сеанс byobu сеансом своего удаленного интерпретатора команд по умолчанию, когда вы начнете любой сеанс входа в систему в текстовом режиме из другой системы (введите byobu-di sable, чтобы деактивизировать это поведение).

При первом запуске выполнения byobu у вас будет возможность выбрать работу в режиме screen или Emacs (см. man 1 tmux). Для наших целей я выбрал режим byobu:

```
Configure Byobu's ctrl-a behavior...
When you press ctrl-a in Byobu, do you want it to operate in:
    (1) Screen mode (GNU Screen's default escape sequence)
    (2) Emacs mode (go to beginning of line)
Note that:
  - F12 also operates as an escape in Byobu
  - You can press F9 and choose your escape character
  - You can run 'byobu-ctrl-a' at any time to change your selection
Select [1 or 2]: 1
```

После запуска выполнения byobu внизу страницы появится строка состояния. Вы можете использовать функциональные клавиши (с **FI** по **F12),** расположенные вверху клавиатуры, чтобы **открывать новые окна интерпретатора команд и управлять ими.** Вот примеры задействования функциональных клавиш для открытия и управления новыми окнами интерпретатора команд при выполнении byobu:

F2 Открывает новое окно интерпретатора команд (попробуйте открыть несколько таких окон)
F3 Обеспечивает переход в предыдущее окно
F4 Обеспечивает переход в следующее окно

Для обновления уведомлений о состоянии используйте функциональную клавишу **F5:**

F5 Обновить уведомления о состоянии

Закончив работу в рамках сеанса byobu, вы можете **отсоединиться от этого сеанса,** оставив его активным, чтобы снова присоединиться к нему позднее. Нажмите комбинацию клавиш **Shift+F6,** чтобы сеанс входа в систему не завершился при отсоединении:

F6 Отсоединиться (отключиться) от сеанса byobu и завершить его
Shift+F6 Отсоединиться (отключиться) от сеанса byobu, но не завершать его

Вы можете **вернуться назад даже на 10 000 строк текста** на экране во время сеанса byobu. Когда на экране отображается много текста, нажмите F7, чтобы переключиться в режим обратной прокрутки. Затем используйте клавиши **Page Up** и **Page Down** для перехода к нужному тексту на экране. Чтобы выйти из этого режима, нажмите **Enter:**

```
$ find /          Выполнить какую-либо команду для заполнения экрана текстом
F7                Переключиться в режим обратной прокрутки/поиска
Page Up           Перейти на одну страницу текста назад на экране
Page Down         Перейти на одну страницу текста вперед на экране
```

Вы можете изменять конфигурацию byobu разными путями. Нажмите F8 для изменения имени текущего окна. Чтобы открыть меню конфигурации byobu, позволяющее изменять многие настройки, нажмите клавишу F9.

```
F8        Изменить имя текущего окна
F9        Открыть меню конфигурации byobu
```

Открыв меню конфигурации byobu, вы сможете изменить некоторые настройки. Вот как оно выглядит:

```
          Byobu Configuration Menu
Help -- Quick Start Guide
Toggle status notifications
Change escape sequence
Byobu currently does not launch at login (toggle on)
   <Exit>
```

Используйте клавиши со стрелками, **Tab, Page Up** и **Page Down** для **навигации по меню.** Чтобы вернуться к предыдущему меню, нажмите **Esc.** Выбрав **Help** (Справка), вы сможете увидеть описание нажатий клавиш, к которым можно прибегнуть при использовании byobu. Выберите **Toggle status notifications** (Переключение уведомлений о состоянии), чтобы изменить состояние, отображаемое в нижней строке состояния. Укажите другие варианты для изменения управляющих последовательностей или того, следует ли задействовать byobu по умолчанию при начале сеанса входа в систему в текстовом режиме.

Чтобы закрыть текущее окно byobu, просто выйдите из интерпретатора команд (введите exit или нажмите комбинацию клавиш **Ctrl+D).** Закрытие последнего интерпретатора команд, ассоциированного с сеансом byobu, приведет к завершению этого сеанса.

Для получения дополнительной информации о byobu введите man byobu или посетите страницу с документацией к byobu на сайте сообщества Ubuntu: **https://help.ubuntu.com/community/Byobu.**

Удаленный Рабочий стол Windows

Многие системные администраторы, кому удобно пользоваться рабочим столом Linux, предпочитают заниматься администрированием своих Windows-систем из

Linux во всех возможных случаях. Linux предоставляет такие инструменты, как `rdesktop` и `tscl ient`, которые позволяют подключиться к системе Windows, где запущены терминальные службы Windows.

- Чтобы иметь возможность **подключаться к Рабочему столу системы Windows** из Linux, вам потребуется активизировать удаленный **Рабочий стол** в своей системе Windows. Эта процедура различается в зависимости от версии Windows: чтобы активизировать доступ к удаленному **Рабочему столу** в Windows XP (и других), щелкните правой кнопкой мыши на **My Computer** (Мой компьютер) и выберите **Properties** (Свойства). Затем перейдите на вкладку **Remote** (Удаленные сеансы) в окне **System Properties** (Свойства системы) и установите флажок **Allow users to connect remotely to this computer** (Разрешить удаленный доступ к этому компьютеру). 11осле этого выберите в соответствующем окне пользователей, которым хотите разрешить подключаться к этому Windows-компьютеру, и щелкните ОК.
- Для активизации доступа из Linux к удаленному **Рабочему столу** в операционной системе Windows **7** выберите **System and Security** (Система и безопасность) на **Панели управления.** Щелкните на **Remote Settings** (Настройка удаленного доступа) на панели слева. Затем в диалоговом окне **System Properties** (Свойства системы) перейдите на вкладку **Remote** (Удаленный доступ) и выберите **Allow Connections from Computers Running Any Version of Remote Desktop** (Разрешать подключения от компьютеров с любой версией удаленного рабочего стола). После этого щелкните на **Select Users** (Выбрать пользователей), чтобы указать, каким учетным записям пользователей будет разрешен доступ.
- Для получения доступа к удаленному **Рабочему столу** Ubuntu в Windows 8 щелкните правой кнопкой мыши на соответствующем меню в стиле Windows 8 и выберите **All Apps** (Все приложения), а затем правой кнопкой мыши — на элементе меню **Computer** (Компьютер). После этого выберите **Properties** (Свойства). В диалоговом окне **Properties** (Свойства) щелкните на ссылке **Remote Settings** (Настройка удаленного доступа).

Теперь вы можете использовать либо `rdesktop`, либо `remmina` для подключения к системе Windows из Linux посредством RDP (Remote Desktop Protocol — протокол удаленного рабочего стола). Оба этих приложения по умолчанию устанавливаются в Ubuntu.

Подключение к удаленному Рабочему столу Windows с помощью remmina

Если вы привыкли использовать Windows-инструмент **Remote Desktop Connection** (Подключение к удаленному рабочему столу) (ранее называвшийся *Terminal Services Client (Клиент терминальных служб))* для подключения одного Windows-компьютера к другому, то, вероятно, инструмент remmi `na` (Remmina Remote Desktop — удаленный рабочий стол Remmina) покажется вам хорошим средством для подключения к **Рабочему столу** Windows из Linux. После ввода команды `remmina`

появится окно **Remmina Remote Desktop Client** (Клиент удаленного рабочего стола Remmina), которое обеспечивает доступ к интерфейсу пользователя клиента удаленного **Рабочего стола** Windows.

Располагая установленным пакетом remmina (apt-get install remmina), **запустите выполнение remmina,** набрав Remmina Remote Desktop (Удаленный рабочий стол Remmina) на панели управления Ubuntu либо введя в интерпретаторе команд следующее:

```
$ remmina &
```

Когда появится соответствующее окно, выполните команду **Connection** (Подключение) ► **New** (Новое). На рис. 13.2 изображены окна **Remmina Remote Desktop Client** (Клиент удаленного рабочего стола Remmina) и **Remote Desktop Preferences** (Параметры удаленного рабочего стола).

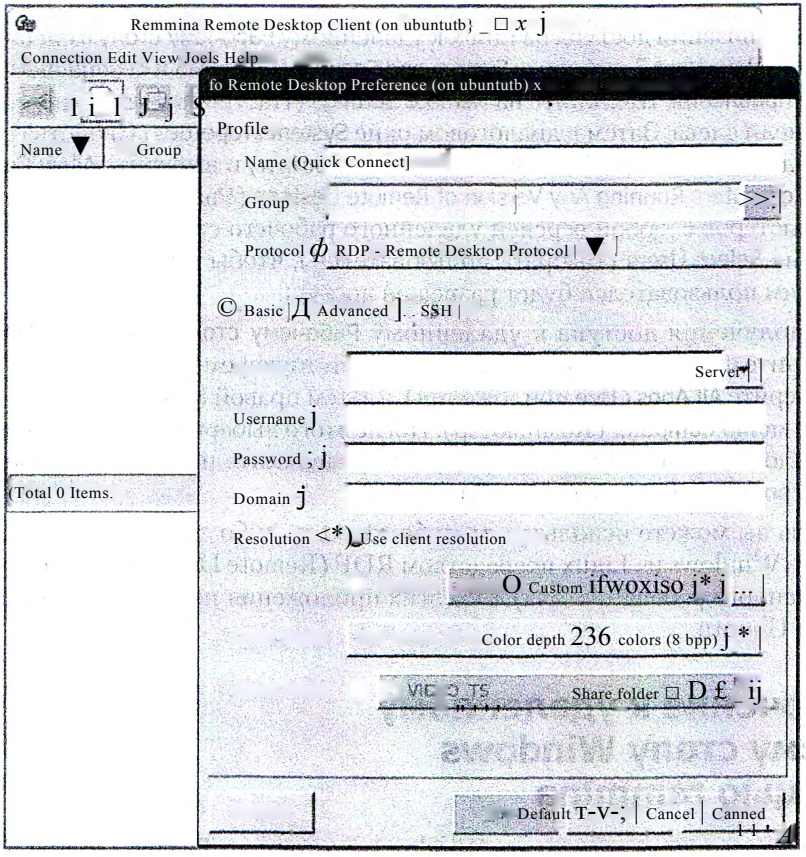

Рис. 13.2. Remmina Remote Desktop Client (remmina) (Клиент удаленного рабочего стола Remmina) позволяет подключиться к Рабочему столу Windows

Возможно, в этом окне вам потребуется ввести только имя хоста или IP-адрес Windows-компьютера. Далее вам также может быть предложено ввести имя ноль-

зователя и пароль. Зайдите па разные вкладки для дополнительной настройки своего подключения к удаленному Рабочему столу Windows.

Следует отметить, что remmina можно использовать и в качестве клиента в случае со службами VNC, SSH и SFTP.

Подключение к удаленному Рабочему столу Windows с помощью rdesktop

Если вы предпочитаете не использовать команду remmina, описанную выше, можете **войти в систему и подключиться к удаленному Рабочему столу Windows** с помощью команды rdesktop. Чтобы установить rdesktop, введите следующее:

```
$ sudo apt-get install rdesktop
```

Команда rdesktop запросит вход в систему на Windows-компьютере, а затем откроет Рабочий стол Windows для пользователя после того, как он войдет в систему. Вот примеры использования команды rdesktop:

```
$ rdesktop 172.16.18.66                  Войти в систему и подключиться
                                         к Рабочему столу по IP-адресу
$ rdesktop -u chris -p МбруХХ winl       Определить пользователя/пароль для хоста
                                         winl
$ rdesktop -f winl                       Запустить выполнение rdesktop
                                         в полноэкранном режиме
$ rdesktop -0 -r sound:local winl        Направлять звук с сервера на клиент
$ rdesktop -E winl                       Деактивизировать клиент-серверное
                                         шифрование
```

Если вы деактивизируете клиент-серверное шифрование, то пакет, касающийся входа в систему, будет зашифрован, в отличие от всего остального, что последует за ним. Это может значительно улучшить производительность, однако любой, кто будет анализировать трафик вашей LAN-сети, сможет увидеть данные, передаваемые открытым текстом во время ваших коммуникаций (включая информацию относительно всех интерактивных входов в систему, которая будет передана после первоначального пакета, касающегося входа в систему). К другим параметрам rdesktop, которые могут улучшить производительность или работу вашего Рабочего стола Windows, относятся -т (не отправлять события, инициируемые при перемещении мыши), -D (скрыть оформление оконного менеджера) и -К (не переопределять привязки клавиш оконного менеджера).

Удаленный рабочий стол Linux и приложений

Систему X Window System (X) не следует запускать на типичных производственных серверах из соображений безопасности и производительности. Однако в силу клиент-серверной природы X вы можете запустить программу с поддержкой X на удаленном компьютере, чтобы при этом ее графический вывод направлялся на ваш

локальный рабочий стол. При таком взаимодействии приложение, выполняющееся на удаленном компьютере, будет называться *клиентом X,* а ваш рабочий стол — *сервером X.* Запуская приложения X в ненадежных сетях или Интернете, используйте SSH-переадресацию, как было описано ранее. В надежных LAN-сетях делайте это без использования SS H, как рассматривается далее.

По умолчанию ваш рабочий стол X не позволит удаленным приложениям X подключаться к нему (отображаться на нем). Вы можете **разрешить удаленным приложениям отображаться на вашем рабочем столе** с помощью команды xhost. Используйте ее на своем локальном экране Linux для контроля над тем, какие компьютеры смогут подключаться к X и отображать приложения на вашем рабочем столе. Вот примеры использования xhost:

```
$ xhost                          Показать хосты, которым разрешен доступ
access control enabled, only authorized clients can connect
SI: 1 oca 1 user:chris
$ xhost +                        реактивизировать контроль доступа (опасно)
access control disabled, clients can connect from any host
$ xhost -                        Снова активизировать контроль доступа
access control enabled, only authorized clients can connect
$ xhost remotemachine            Добавить хост, которому разрешен доступ.
remotemachine being added to access control list
```

Контроль доступа следует полностью деактивизировать только с целью устранения неполадок. Однако если определенному хост-компьютеру будет разрешен доступ (remotemachine в данном случае), то вы сможете выполнить приведенные далее команды из интерпретатора команд на удаленном компьютере, чтобы приложения X, запущенные на этом компьютере, отображались на локальном рабочем столе (в данном случае его имя — local machine):

```
$ export DISPLAYS ocal machine: O   Задать для DISPLAY значение localmachine:0
$ xterm &                           Открыть удаленный терминал на локальном компьютере
$ xclock &                          Открыть удаленные часы на локальном компьютере
$ gnome-calculator &                Открыть удаленный калькулятор на локальном
                                    компьютере
```

После задания для переменной DISPLAY в случае с remotemachine значения, при котором она будет указывать на local machine, любое приложение, запускаемое из интерпретатора команд на remotemachine, должно отображаться на рабочем столе O на 1 ocalmachi пе. В данном случае мы запустили окно терминала, а также такие приложения, как часы и калькулятор.

ПРИМЕЧАНИЕ

В нескольких последних версиях Ubuntu сервер X по умолчанию не осуществляет прослушивание на предмет TCP-подключений. Чтобы разрешить удаленные подключения к X, отредактируйте файл /etc/lightdm/lightdm.conf на сервере X, добавив в него строку xserver-allow-tcp=true, чтобы содержимое этого файла стало выглядеть следующим образом:

```
[SeatDefaults]
   greeter-session=unitygreeter
   user-session=ubuntu
  xsserver-allow-tcp=true
```

Затем перезапустите сервер X Window или оконный менеджер (sudo restart lightdm).

Совместное использование приложений X хостами Linux и UNIX подобным образом обеспечивается довольно легко. Однако запуск приложений X таким путем, чтобы они отображались на других платформах для настольных компьютеров, потребует немного дополнительного конфигурирования. Если ваш настольный компьютер работает под управлением операционной системы Windows, придется запустить сервер X. Бесплатным решением в данном случае является среда Cygwin, включающая сервер X. Существуют также функционально богатые коммерческие серверы X, однако они могут оказаться очень дорогими. Для совместного использования удаленных рабочих столов в разных операционных системах рекомендую задействовать VNC (Virtual Network Computing — вычисления посредством виртуальной сети).

Совместное использование рабочих столов с помощью VNC

Система VNC включает серверное и клиентское программное обеспечение, которое позволяет вам взять на себя удаленное управление полным отображением рабочего стола с одного компьютера на другом. В Ubuntu предустановлена команда vncviewer, с помощью которой вы можете осуществлять доступ к удаленному рабочему столу на вашем экране (клиент), однако вам потребуется пакет vncserver для совместного использования удаленного рабочего стола со своего компьютера (сервер). Чтобы установить пакет vncserver, введите следующее:

```
$ sudo apt-get install vncserver
```

VNC-клиенты и серверы доступны для множества операционных систем, с которыми могут взаимодействовать. VNC-серверы доступны для систем Linux, Windows (32-битных версий), Mac OS X и UNIX. Предлагаются VNC-клиенты как для этих систем, так и для систем многих других типов (включая OS/2, PalmOS, кроме того, VNC-клиенты даже могут использоваться как Java-приложения, выполняющиеся в браузере).

Конфигурирование VNC-сервера

Предположим, что в настольной системе Linux вы используете экран по умолчанию (DISPLAY=:0) в качестве локального рабочего стола, поэтому мы создадим независимые экраны, доступные посредством VNC.

Затем вам нужно выполнить команду vnepasswd от имени каждого пользователя для создания пароля, который понадобится каждому из них для подключения к их рабочим столам на VNC-сервере. В данном примере мы выполняем от имени пользователя chris следующее:

```
$ vnepasswd
Password: *******
Verify *******
```

Наконец, вы можете запустить VNC-сервер (vncserver). Введите следующее от имени обычного пользователя (в нашем примере это делается от имени chris):

```
$ vncserver
New 'myserver: 1 (chris)' desktop is myservenl
Starting applications specified in /home/chris/.vnc/xstartup
Log file is /home/chris/.vnc/myserver:l.logs
```

Если вы используете брандмауэр iptables, встроенный в вашу систему, то убедитесь, что вы открыли порт (-ы) для VNC. Каждый экран работает через свой порт. Доступ к экрану с номером N осуществляется через TCP-порт 5900 + N. Например, экран 1 доступен через порт 5901. Подробнее о брандмауэре iptables вы сможете узнать из гл. 14.

Для изменения конфигурации имеющихся у вас VNC-серверов отредактируйте файл xstartup в своем каталоге $HOME/. vnc. В этом каталоге также располагаются файл passwd (для хранения пароля пользователя, применяемого для защиты VNC-сеанса) и файлы журналов для каждого начатого VNC-сеанса.

Закончив работу в рамках VNC-сеанса, используйте команду vncserver для его завершения. Например, чтобы завершить сеанс : 1, введите следующее:

```
$ vncserver -kill :1
Killing Xvnc4 process ID 11651
```

Запуск VNC-клиента

Запустив VNC-сервер, вы сможете подключиться к рабочему столу на этом сервере из любой клиентской системы, упомянутой ранее (Windows, Linux, Mac OS X, UNIX и т. д.). Например, если ваш VNC-сервер будет располагаться на компьютере с именем myserver, то вы могли бы ввести следующую команду для запуска удаленного рабочего стола из другой Linux-системы:

```
$ apt-get install vncviewer
$ vncviewer myserver:1 Подключиться от имени chris к экрану 1
Connected to RFB server, using protocol version 3.8
Performing standard VNC authentication
Password: *********
$ vncviewer myserver:2 Подключиться от имени joe к экрану 2
```

Если только вы не определите команды для запуска, то увидите лишь фоновую область экрана X Window System с поперечной штриховкой. Чтобы пойти далее, вам потребуется запустить программы X Window на серверной системе или со своей клиентской системы, указывающей на экран X VNC. Например:

```
$ xterm -display myserver:1 &
$ metacity --display myserver:1 &
```

ПРИМЕЧАНИЕ

Какой именно сервер X следует использовать (VNC-сервер в этом примере), в случае с большинством программ X Window указывается посредством параметра -display. При этом оконный менеджер metacity предполагает размещение двух тире перед этим параметром, чтобы он выглядел как --display.

Вы также можете использовать remmina для подключения; в этом примере вы указали бы myserver: 1 в качестве компьютера, а VNC — в роли протокола. Вы также можете ввести имя пользователя и пароль либо дождаться приглашения сделать это.

Использование VNC в ненадежных сетях наряду с SSH

VNC считается небезопасным протоколом. При передаче пароля используется довольно слабое шифрование, а остальная часть сеанса вообще не шифруется. В силу этого при использовании VNC в ненадежных сетях или Интернете рекомендую туннелировать его посредством SSH.

Общее описание работы службы SSH можно найти в подразделе «Удаленный вход в систему посредством SSH» ранее в этой главе. Для переадресации VNC-экрана 2 (порт 5902) на компьютере с именем myserver на тот же локальный порт введите следующее;

```
$ ssh -L 5902:local host:5902 myserver
```

ПРИМЕЧАНИЕ----------

Если вы начнете регулярно использовать VNC, то, возможно, захотите взглянуть на tightvnc (из пакета с аналогичным именем). Этот инструмент не включен в Ubuntu, однако является еще одной реализацией протокола VNC с открытым исходным кодом, которая находится на стадии активной разработки, обладает более новой функциональностью и оптимизациями. К этой функциональности относится встроенное SSH-туннелирование.

Резюме

Если вы когда-нибудь окажетесь в ситуации, в которой придется заниматься администрированием сразу нескольких Linux-систем, можете воспользоваться богатым набором команд в Linux для удаленного системного администрирования. Инструмент SSLI (Secure Shell — безопасный интерпретатор команд) обеспечивает зашифрованные коммуникации между клиентами и серверами при удаленном входе в систему, туннелировании и передаче файлов.

Благодаря VNC (Virtual Network Computing — вычисления посредством виртуальной сети) система Linux может совместно использовать свой рабочий стол с клиентской системой таким образом, чтобы удаленный рабочий стол отображался прямо на рабочем столе клиента. Для совместного использования удаленных рабочих столов в Linux можно задействовать как графические инструменты (например, Remmina Remote Desktop (Удаленный рабочий стол Remmina)), так и команды (к примеру, vncserver и vncviewer).

14 Повышение безопасности

В этой главе:

- добавление учетных записей пользователей и изменение пользовательских настроек с помощью команды `useradd`;
- изменение учетных записей пользователей посредством `usermod`;
- удаление учетных записей пользователей с помощью userdel;
- добавление, изменение и управление паролями с использованием команды `passwd`;
- управление группами посредством `groupadd`, `groupmod` и `groupdel`;
- просмотр информации о вошедших в систему пользователях с помощью `last`, `lastb` и `who`;
- конфигурирование брандмауэров с использованием команды `iptables`;
- обеспечение повышенной безопасности с помощью инструментов SELinux, Tripwire и APT.

Обеспечение безопасности вашей операционной системы Linux означает, во-первых, ограничение доступа к учетным записям пользователей и службам в ней, во-вторых, проверку, что никому не удалось обойти защиту, которую вы установили.

Ubuntu, Debian и другие системы, основанные на этих дистрибутивах Linux, задумывались безопасными по умолчанию. Это означает отсутствие учетных записей пользователей с пустыми паролями, а также то, что большинство сетевых служб (веб-службы, FTP и т. д.) деактивизировано по умолчанию (даже если установлено соответствующее программное обеспечение).

Многие команды, рассматриваемые в этой книге, можно использовать для проверки и повышения безопасности вашей системы Linux, однако определенные базовые функции в Linux сосредоточены именно на обеспечении безопасности. Например, безопасные учетные записи пользователей с хорошей парольной защитой, надежный брандмауэр и последовательное протоколирование (и мониторинг файлов журналов) имеют критически важное значение для того, чтобы у вас была безопасная система Linux. В этой главе описываются команды, связанные с этими функциями, а также продвинутые инструменты, например SELinux и Tripwire.

Работа с пользователями и группами

Во время большинства процедур установки] Jnux вас будут просить задать пароль для суперпользователя (для системного администрирования). Затем вам могут предложить ввести имя пользователя и задать для него пароль (для повседневного пользования компьютером). Призываю вас всегда входить в систему от имени обычного пользователя и прибегать к su или sudo для получения прав доступа, обеспечиваемых учетной записью суперпользователя, только когда это действительно необходимо. Установив Linux, вы сможете задействовать команды или графические инструменты для добавления пользователей, модификации учетных записей, а также задания и изменения паролей.

Ubuntu повышает безопасность путем запрета входа в систему от имени суперпользователя по умолчанию. Вместо этого во время установки вы указываете пользователя с паролем, который сможет выполнять административные функции. Используйте sudo в Ubuntu для выполнения отдельных команд от имени суперпользователя. Команда sudo генерирует приглашение ввести пароль администратора, которым обычно является ваш пароль. Это позволяет избежать большинства проблем из-за выполнения некоторых команд от имени суперпользователя, которые вам не требовались.

Управление пользователями с помощью GUI-интерфейса

В настольных системах Ubuntu с X вы можете управлять пользователями и группами посредством окна User Manager (Менеджер пользователей) (команда System (Система) ► Administration (Администрирование) ► Users and Groups (Пользователи и группы)).

При управлении учетными записями пользователей для серверов один из вариантов, к которым можно прибегнуть, заключается в применении веб-интерфейсов GUI. Самый распространенный инструмент общего назначения — Webmin (www.webmin.com). Вам потребуется убедиться, что Webmin не работает через свой порт (10 000) по умолчанию из соображений безопасности. Вы также можете использовать веб-интерфейсы специального назначения. Например, существует множество GUI-интерфейсов для автоматизации веб-хостинга вроде cPanel (www.cpanel.com), Plesk (www.swsoft.com/plesk) и Ensim (www.ensim.com).

Добавление учетных записей пользователей

Для добавления новых пользователей вы можете прибегнуть к команде useradd. Единственным параметром, который ей нужно передать, будет имя добавляемого вами пользователя. Вы сможете увидеть некоторые настройки по умолчанию для добавления новых пользователей, задействовав параметр -0:

```
$ useradd -D          Показать значения useradd по умолчанию
GROUP=100             Задать для группы идентификатор в виде значения 100
                      (пользователи)
```

```
H0ME=/home          Задать /home в качестве базового домашнего каталога
INACTIVE=-1         Параметр окончания срока действия пароля
                    деактивизирован (-1)
EXPIRE-             Не устанавливать дату деактивизации учетной записи
                    пользователя
SHELL=/bi n/sh      Задать /Ыn/bash в качестве интерпретатора команд
                    по умолчанию
SKEL=/etc/skel      Скопировать конфигурационные файлы из /etc/skel в $HOME
CREATE_MAIL_SPOOL=no Создать каталог почтового спула
```

Ubuntu и другие Debian-системы переопределяют группу по умолчанию (100) и создают новую группу для каждого пользователя. По умолчанию идентификатором, присваиваемым первому созданному пользователю, а также идентификатором группы является значение 1000. Название группы соответствует имени пользователя. Названием домашнего каталога является имя пользователя, добавленное к /home. Так, например, вы могли бы создать первую учетную запись обычного пользователя в системе:

```
$ sudo useradd -m willz
```

Результатом была бы новая учетная запись пользователя с именем пользователя willz (идентификаторпользователя — 1001) и именем группы willz (идентификатор группы — 1001). Параметр -т дает useradd указание создать домашний каталог /home/willz и скопировать набор конфигурационных файлов (имя каждого из которых начинается с точки) в домашний каталог из /etc/skel. Учетная запись оставалась бы активной неопределенно долго (дата окончания срока ее действия отсутствовала бы). **Добавьте пароль,** как показано далее, причем в большинстве случаев это будет все, что вам потребуется сделать для получения работоспособной учетной записи пользователя.

```
$ sudo passwd horatio
Changing password for user horatio
New UNIX password: ********
Retype new UNIX password: ********
passwd: all authentication tokens updated successfully.
```

ПРИМЕЧАНИЕ----------

Не забывайте, что нужно использовать стойкие пароли, которые нельзя подобрать с помощью словаря.

Есть множество параметров, которые вы можете задействовать для **переопределения значений по умолчанию** при добавлении пользователя. Комбинируйте разные параметры. Вот несколько примеров:

```
$ sudo useradd -u 1101 -g 60 -m skolmes   Задействовать определенные
                                          идентификаторы пользователя и группы
$ sudo useradd -m -d /home/jj jones       Создать домашний каталог /home/jj
$ sudo useradd -m -G games,man timd       Добавить пользователя в группы games и man
$ sudo useradd -c "Tom G. Lotto" tlot     Добавить полное имя в поле комментария
$ sudo useradd -s /bin/sh joeq            Задать новый интерпретатор команд
                                          по умолчанию (tcsh)
```

```
$ sudo useradd -e 2014-04-01 jerry   Срок действия новой учетной записи
                                     заканчивается 1 апреля 2014 года
$ sudo useradd -f 0 jdoe             Создать деактивизированную учетную запись
$ sudo useradd -s /sbin/nologin bint Не позволять пользователю входить
                                     в систему
$ sudo useradd billyq                Предотвратить создание домашнего каталога
```

Прежде чем вы сможете добавить пользователя в группу, ее необходимо создать (см. о команде groupadd в подразделе «Добавление групп» данной главы). Исключение состоит в том, что если не указать группу, то будет создана группа с названием, идентичным имени нового пользователя, который окажется к ней приписан. Пользователь должен принадлежать к одной начальной группе, которая может быть переопределена с помощью -д, а Также может принадлежать к дополнительным группам, определяемым посредством -G.

Для **отображения групп (-ы), к которой принадлежит пользователь,** используйте команду groups:

```
$ groups timd  Показать группы, к которым принадлежит пользователь
timd : timd man games
```

В примере с применением -е **устанавливается дата окончания срока действия** учетной записи пользователя, которая, как вы знаете, временная. Смените интерпретатор команд по умолчанию на /usr/sbin/nologin, если нужно, чтобы пользователь смог осуществлять доступ к компьютеру (посредством FTP, POP3 и т. д.), однако при этом вы не хотите, чтобы у него был доступ к обычному интерпретатору команд Linux, назначаемому при входе в систему. Аналогичным образом в последнем примере, в котором не задействуется -ш для создания домашнего каталога, пользователю может быть предоставлен доступ к компьютеру, однако домашнего каталога у этого пользователя не будет. Следует отметить, что во всех примерах, если только вы не укажете параметр -т, команда useradd не будет создавать домашний каталог для пользователя.

Изменение значений useradd по умолчанию

Значения по умолчанию, применяемые при создании новой учетной записи пользователя с помощью useradd (интерпретатор команд по умолчанию, идентификатор группы, даты окончания срока действия и т. д.), задаются в соответствии со значениями в файлах /etc/login.defs и /etc/default/useradd. Вы можете отредактировать эти файлы с целью изменить значения по умолчанию либо ввести команду useradd с параметром -0, чтобы отобразить на экране или выборочно изменить значения:

```
$ useradd -D                         Показать настройки useradd по умолчанию
$ sudo useradd -D -b /home2 -s /bin/csh Задать базовый каталог и интерпретатор
                                     команд по умолчанию
$ sudo useradd -D -e 2014-01-01      Задать первый день 2014 года в качестве
                                     даты окончания срока действия всех
                                     новых учетных записей пользователей
```

Как уже отмечалось ранее, если передать параметр -ш при создании учетной записи, то файлы и каталоги, содержащиеся в каталоге /etc/skel, будут скопированы в домашний каталог нового пользователя. К этим файлам относятся файлы интерпретатора команд bash и ссылка на папку-образец. Вы можете добавить другие файлы и каталоги в /etc/skel, чтобы их получал каждый новый пользователь. Например, если вы будете конфигурировать веб-сервер, то, возможно, создадите каталоги publ i c_html и publ i c_ftp, чтобы пользователи смогли добавлять в них веб-страницы и файлы, которыми они захотят поделиться с другими.

Модификация учетных записей пользователей

Создав учетную запись пользователя, можно **изменить значения для нее с помощью команды usermod.** Большинство ее параметров соответствует используемым с useradd. Например:

```
$ sudo usermod -c "Thomas Lotto" tlot   Изменить имя пользователя в поле
                                        комментария
$ sudo usermod -s /bin/sh joeq          Сменить интерпретатор команд
                                        по умолчанию на sh
$ sudo usermod -L swanson               Заблокировать учетную запись пользователя
                                        с именем swanson
$ sudo usermod -U travis                Разблокировать учетную запись пользователя
                                        с именем travis
```

Следует отметить, что в последних двух примерах осуществляется соответственно блокировка и разблокировка учетной записи пользователя. Блокировка учетной записи не приводит к ее удалению из системы или стиранию каких-либо файлов или каталогов пользователя. Однако она не позволит этому пользователю войти в систему. После снятия блокировки пароль пользователя тоже будет восстановлен, благодаря чему не придется задавать новый, если тот же самый пользователь возобновит применение учетной записи.

Блокировка учетной записи может быть полезна, когда работник покидает компанию, однако информацию, которая находится в файлах этого работника, необходимо передать другому человеку. При таких обстоятельствах блокировка учетной записи вместо ее удаления позволит избежать того, что файлы, которыми владел пользователь, окажутся принадлежащими идентификатору пользователя, который никому не присвоен.

Обычный пользователь не может задействовать команды useradd и usermod, поэтому существуют специальные команды для **изменения персональных данных в учетных записях.** Взгляните на следующие примеры:

```
$ chsh -s /bin/sh                   Сменить текущий интерпретатор команд
                                    на /bin/sh
$ sudo chsh -s /bin/sh chris        Сменить интерпретатор команд пользователя
                                    на /bin/sh

$ sudo chfn \
      -h "212-555-1212"     \       Изменить номер домашнего телефона
```

```
    -w "212-555-1957"          Изменить номер рабочего телефона
$ finger chris
Login: chris                    Name: Chris Negus
Office Phone: 919-555-1957 Home Phone: 919-555-1212
Directory: /home/chris          Shell: /bin/bash
On since Sat Mar 9 14:43 (EST) on pts/0 from :0
    10 seconds idle
      (messages off)
Mail last read Sat Mar 9 16:26 2013 (EST)
No Plan.
```

Информация, измененная посредством приведенной выше команды chfn (номера домашнего и рабочего телефонов) и отображенная с помощью finger, располагается в пятом поле файла /etc/password для выбранного пользователя (непосредственное редактирование файла /etc/password может осуществлять только суперпользователь, причем редактировать этот файл следует с помощью команды vi pw и действуя предельно осторожно).

В других версиях Linux вы можете использовать chfn в сочетании с параметром -f для изменения реального или полного имени. В Ubuntu права доступа, позволяющие выполнить данную операцию, деактивизированы по умолчанию. Вы можете изменить это, отредактировав файл /etc/login.defs. Найдите в нем строку:

CHFN_RESTRICT rwh

и модифицируйте ее следующим образом:

CHFN_RESTRICT frwh

Удаление учетных записей пользователей

Задействовав команду userdel, из системы вы сможете удалить учетные записи пользователей, а также другие файлы (домашние каталоги, файлы почтовых спулов и т. д.). Взгляните на следующие примеры:

```
$ sudo userdel jimbo          Удалить учетную запись пользователя, но не его
                              домашний каталог
$ sudo userdel -r lily Удалить учетную запись пользователя, домашний каталог
                              и файл почтового с пула
```

Имейте в виду, что удаление домашнего каталога пользователя с помощью userdel не приведет к удалению остальных файлов, владельцем которых является этот пользователь, в других местах системы. Помните следующее:

- при выводе на экран длинного списка файлов, которыми владел удаленный пользователь (Is -1), идентификатор пользователя и идентификатор группы бывшего пользователя будут представлены как владельцы файла;
- если вам потребуется добавить нового пользователя, для которого были заданы эти идентификаторы пользователя и группы, новый пользователь станет владельцем соответствующих файлов.

Вы можете выполнить команду find / -nouser -Is, чтобы выполнить повсеместный поиск файлов в системе, которыми не владеет ни один пользователь.

Управление паролями

Добавить или изменить пароль обычно очень просто — с использованием команды passwd. Для passwd есть параметры, которые позволяют администратору управлять блокировкой учетных записей, окончанием срока действия паролей и предупреждающими сообщениями о необходимости сменить пароль. Помимо passwd, есть и другие команды, например chage, chfn и vipw, для работы с паролями пользователей.

Обычные пользователи имеют возможность изменять только свои пароли, в то время как суперпользователь может сменить пароль любого пользователя. Например:

```
$ passwd                    Изменить пароль обычного пользователя
Changing password for user chris.
Changing password for chris.
(current) UNIX password: ********
New UNIX password: *
BAD PASSWORD: it's WAY too short
New UNIX password: *********
Retype new UNIX password: *********
passwd: password updated successfully
$ sudo passwd joseph        Суперпользователь может изменить пароль любого
                            пользователя
Changing password for user joseph.
New UNIX password: *
Retype new UNIX password: *
passwd: password updated successfully
```

В первом примере обычный пользователь (chris) изменяет собственный пароль. Даже войдя в систему, пользователю все равно придется ввести текущий пароль перед вводом нового. Кроме того, passwd не позволяет пользователю установить пароль, который слишком короткий, представляет собой слово из словаря, включает недостаточное количество разных символов или обладает иными недостатками, из-за которых его легко угадать. Во втором примере суперпользователь может изменить пароль любого пользователя без необходимости вводить старый.

Пароли должны состоять минимум из восьми символов, быть комбинацией из букв и других символов (цифр, знаков препинания и т. д.) и не должны включать реальных слов. Придумывайте пароли, которые легко запомнить, но сложно угадать.

Системный администратор может задействовать passwd для блокировки и разблокировки учетных записей пользователей. Например:

```
$ sudo passwd -1 carl       Блокировать учетную запись пользователя (carl)
Locking password for user carl.
passwd: Success
```

```
$ sudo passwd -u carl          Разблокировать заблокированную учетную запись пользователя (carl)
Unlocking password for user carl,
passwd: Success
$ sudo passwd -u Jordan Разблокировка терпит неудачу из-за пустого пароля
Unlocking password for user Jordan.
passwd: Warning: unlocked password would be empty.
passwd: Unsafe operation (use -f to force)
```

Блокировка учетной записи пользователя с помощью passwd приводит к установке восклицательного знака (!) перед полем пароля в файле /etc/shadow (в котором хранятся пароли пользователей). При разблокировке учетной записи восклицательный знак удаляется, а предыдущий пароль пользователя восстанавливается.

Администратор может задействовать команду passwd, чтобы требовать от пользователей регулярно менять их пароли, а также предупреждать пользователей, когда срок действия их паролей заканчивается. Для использования возможности окончания срока действия пароля необходимо, чтобы она была активизирована в случае с учетной записью пользователя. В приведенных далее примерах показано использование passwd для внесения изменений касательно окончания сроков действия паролей:

```
$ sudo passwd -n 2 vern        Задать минимальный срок действия пароля как равный двум дням
$ sudo passwd -x 300 vern      Задать максимальный срок действия пароля как равный 300 дням
$ sudo passwd -w 10 vern       Предупредить об окончании срока действия пароля за десять дней
$ sudo passwd -i 14 vern       Количество дней после окончания срока действия пароля, через которое будет деактивизирована учетная запись
```

В первом примере пользователю придется подождать минимум два дня (-n 2), прежде чем устанавливать новый пароль. Во втором случае пользователь должен изменить пароль в течение 300 дней (-x 300). В следующем примере пользователь предупреждается за десять дней до окончания срока действия пароля (-w 10). В последнем случае учетная запись пользователя деактивизируется через 14 дней после окончания срока действия пароля (-1 14).

Для просмотра информации об окончании сроков действия паролей вы можете использовать команду chage:

```
$ sudo chage ■) vern           Просмотр информации об окончании сроков действия паролей
Last password change                                    : Mar 04,   2013
Password expires                                        : May 31.   2014
Password inactive                                       : Jun 14.   2014
Account expires                                         : never
Minimum number of days between password change          : 2
Maximum number of days between password change          : 300
Number of days of warning before password expires       : 10
```

Будучи системным администратором, вы также сможете использовать команду chage для управления окончанием сроков действия паролей. Помимо задания минимального (-гг) и максимального (-М) сроков действия, а также количества дней до окончания срока действия пароля, за которое необходимо предупредить (-W) пользователя, chage позволяет задать день, в который пользователь должен будет установить новый пароль, или определенную дату, когда учетная запись станет неактивной:

```
$ sudo chage -I 40 frank          Сделать учетную запись неактивной через 40 дней
$ sudo chage -d 5 perry           Форсировать окончание срока действия пароля пользователя через пять дней
```

Вместо пяти дней (-d 5) вы можете задать для соответствующего параметра значение 0, в результате чего пользователю придется установить новый пароль, когда он в следующий раз войдет в систему. Например, если задать -d 0, то при следующем входе пользователя perry в систему он получит приглашение установить новый пароль, как показано далее:

```
login: perry
Password: ********
You are required to change your password immediately (root enforced)
Changing password for perry.
(current) UNIX password:
New UNIX password: *********
Retype new UNIX password: *********
```

Добавление групп

Каждый новый пользователь приписывается к одной или нескольким группам. Вы можете создавать группы в любое время и добавлять в них пользователей. Права доступа к файлам и каталогам, которые получает каждая группа в Linux, зависят от того, какие биты прав доступа заданы для каждого элемента. Приписка пользователей к группе позволяет вам назначить владение файлами, каталогами и приложениями, чтобы эти пользователи смогли работать вместе над проектом или имели общий доступ к ресурсам.

Для управления вашими группами доступны команды, подобные применяемым для работы с пользователями. Вы можете добавлять группы (groupadd), изменять настройки групп (groupmod), удалять группы (groupdel), а также добавлять и удалять членов этих групп (groupmems). Вот несколько **примеров добавления новых групп** с помощью команды groupadd:

```
$ sudo groupadd marketing         Создать новую группу со следующим идентификатором группы
$ sudo groupadd -g 1701 sales     Создать новую группу с идентификатором группы 1701
$ sudo groupadd -o -g 74 mysshd   Создать группу с существующим идентификатором группы
```

Используя команду groupmod, вы сможете **изменить имя или идентификатор** существующей группы. Взгляните на следующие примеры:

```
$ sudo groupmod -g 491 myadmin          Изменить идентификатор группы myadmin
                                        на 491
$ sudo groupmod -n myad myadmin         Изменить имя группы myadmin на my ad
```

Чтобы удалить существующую группу, задействуйте команду groupdel. Вот пример:

```
$ sudo groupdel myad                    Удалить существующую группу myad
```

Имейте в виду, что удаление группы или пользователя не приводит к удалению файлов, каталогов, устройств или других элементов, которыми владеет эта группа или пользователь. При выводе на экран длинного списка (Is -1) файлов или каталогов, которыми владели пользователь или группа, подвергнутые удалению, отобразится идентификатор удаленных пользователя или группы.

Наблюдение за пользователями

Создав учетные записи пользователей и предоставив этим пользователям доступ к своему компьютеру, вы можете задействовать несколько разных команд для отслеживания, как они используют ваш компьютер. Приведенные далее команды для наблюдения за активностью пользователей в вашей системе Linux были рассмотрены ранее:

- задействуйте команду find (см. гл. 4) для повсеместного поиска в системе файлов, которыми владеют выбранные пользователи;
- используйте команду du (см. гл. 7), чтобы узнать, сколько дискового пространства занимают домашние каталоги выбранных пользователей;
- задействуйте такие команды, как fuser, ps и top (см. гл. 9), для выяснения, какие процессы запущены пользователями.

Помимо вышеупомянутых, также существуют команды для проверки, например, того, кто вошел в вашу систему, и отображения общей информации о пользователях с учетными записями в вашей системе. Далее приведены примеры команд для **получения сведений о людях, вошедших в вашу систему:**

```
$ last                  Показать информацию о самых последних удачных попытках
                        входа в систему
greek    tty3                   Tue Mar  5 18:05   still   logged in
chris    ttyl                   Mon Mar  4 13:39   still   logged in
root     pts/4    thompson      Tue Mar  5 14:02   still   logged in
chris    pts/1     : 0.0        Mon Mar  4 15:47   still   logged in
Jim      pts/0    10.0.0.50     Sun Man  3 13:46 -  15:40   (01:53)
James    pts/2                  Sat Mar  2 11:14 -  13:38   (2+02:24)
$ last -a Делает имя хоста удаленного клиента более удобочитаемым
$ sudo lastb Показать информацию о самых последних неудачных попытках
                        входа в систему
```

```
Julian ssh:notty , ritchie                    Wed Mar6 12:28 - 12:28    (00:00)
morris ssh:notty thompson                     Thu Feb 28 13:08 - 13:08  (00:00)
baboon ssh:notty 10.0.0.50                    Thu Feb 8 09:40 - 09:40   (00:00)
James ssh:notty 000db9034dce.cli Fri Jun 22 17:23 - 17:23               (00:00)
```

$ who -u *Показать информацию о пользователях, находящихся в системе на данный момент (подробный вывод)*

```
greek    tty3    2013-08-05 18:05   17:24     18121
Jim      pts/0   2013-08-06 12:29              20959 (serverl.example.com)
root     pts/3   2013-08-04 18:18   13:46     17982 (server2.example.com)
james    pts/2   2013-07-31 23:05   old       4700 (0a0d9b34x.example.com)
chris    pts/1   2013-08-04 15:47   old       17502 (:0.0)
```

$ users *Показать информацию о пользователях, находящихся в системе на данный момент (краткий вывод)*

```
chris James greek Jim root
```

С помощью команды 1 ast вы можете увидеть информацию о каждом пользователе, который вошел в систему (или открыл новый интерпретатор команд), а также узнать, как долго он находился в системе или все еще находится в ней (still 1 ogded i n). В строках терминалов ttyl и tty3 показаны пользователи, работающие из виртуальных терминалов в консоли. В строках pts указаны имена пользователей, открывших интерпретатор команд с удаленного компьютера (thompson) или локального экрана X (:0.0). Рекомендую использовать параметр -a для обеспечения удобочитаемости. Команда lastb показывает неудачные попытки входа в систему, а также откуда они исходили. Команды who -и и users отображают информацию о пользователях, находящихся в системе на данный момент.

Вот команды для **получения дополнительной информации об отдельных пользователях** в вашей системе:

$ id *Ваши идентификационные данные (идентификатор пользователя.идентификатор группы и группа для текущего интерпретатора команд)*

```
uid=1000(chris) gid=1000(chris)
groups=4(adm).20(di alout),24(cdrom),25(floppy).
29(audio),
30(dip),44(video),46(piugdev),104(scanner),112(netdev),
113(1padmi n),
115(powerdev).117(admi n).1000(chri s)
```

$ who am I *Ваши идентификационные данные (пользователь, tty. дата входа в систему, местоположение)*

```
chris pts/0          Mar 3 21:40 (:0.0)
```

$ finger -s chris *Информация о пользователе (краткий вывод)*

```
Login Name           Tty    Idle   Login Time      Office     Office Phone
chris Chris Negus    ttyl   Id     Mar 4 13:39 A-lll          555-1212
```

$ finger -1 chris *Информация о пользователе (подробный вывод)*

```
Login: chris                        Name: Chris Negus
Directory: /home/chris              Shell: /bin/bash
Office: A-lll, 555-1212             Home Phone: 555-2323
On since Mon Mar 4 13:39 (CDT) on ttyl 2 days idle
New mail received Wed Mar 6 13:46 2013 (CDT)
     Unread since Mon Mar 4 09:32 2013 (CDT)
No Plan.
```

Помимо отображения основной информации о пользователе (логин, имя, домашний каталог, интерпретатор команд и т. д.), команда finger также выведет на экран любую информацию, хранящуюся в специальных файлах в домашнем каталоге этого пользователя. Например, содержимое файлов -7. pi an и ~/. project, если они существуют, отображается в конце вывода fi nger. Вывод в случае с однострочным файлом . project и многострочным файлом . pi an может выглядеть следующим образом:

$ finger -1 chris *Информация о пользователе (подробный вывод, файлы .project и .plan)*

...
Project:
My project is to take over the world.
Plan:
My grand plan is
to take over the world
by installing Linux on every computer

Конфигурирование встроенного брандмауэра

Брандмауэр — это критически важный инструмент для защиты компьютера от вторжений через Интернет или другую сеть. Он способен защитить ваш компьютер путем проверки каждого пакета данных, входящего, исходящего или проходящего через сетевые интерфейсы компьютера, с последующим принятием решения, что делать с этим пакетом, на основе заданных вами параметров.

Брандмауэром, встроенным в текущее ядро Linux, является Netfilter, хотя его чаще называют именем команды и службы, которая представляет этот инструмент, — i ptabl es. Команда i ptabl es устанавливается во время инсталляции Ubuntu, а ее функции встроены в ядро, поэтому вы будете готовы сразу приступить к созданию правил брандмауэра для своей системы.

Инструмент iptables (www.netfilter.org) — очень мощный, хотя и сложный в использовании из командной строки. По этой причине многие устанавливают основные правила брандмауэра посредством графического интерфейса. Чтобы получить в распоряжение графический интерфейс, установите пакет Firestarter (sudo apt-get install firestarter).

В Firestarter предусмотрен мастер для конфигурирования вашего брандмауэра. Чтобы запустить Firestarter, введите в окне терминала на своем рабочем столе следующее:

 # sudo firestarter

Вы также можете опробовать такие инструменты, обеспечиваемые дополнительными пакетами, как FWBuilder (из пакета fwbuilder) и Shorewall (из пакета shorewall), для конфигурирования брандмауэров графическим путем.

ПРИМЕЧАНИЕ

Прежде чем двигаться вперед, прочтите документ IpTablesHowTo для Ubuntu, расположенный по адресу https://help.ubuntu.com/community/IptablesHowTo. В нем содержится множество информации об использовании iptables в Ubuntu, поскольку подход к применению этого инструмента в Ubuntu значительно отличается от имеющего место в других версиях Linux, например Fedora.

Понятие брандмауэра iptables

Инструмент iptables работает, исследуя все пакеты, проходящие через сетевые интерфейсы системы Linux, и выполняя дальнейшие действия касательно этих пакетов согласно установленным вами правилам. Правила добавляются в одну из *таблиц*, ассоциированных с iptables. Доступными таблицами являются fi Iter, nat и mangle.

Большую часть времени вы будете работать с таблицей filter. Действие будет выполняться в зависимости от выбранных вами атрибутов (например, запрашиваемого порта, хоста, отправившего пакет, или сетевого интерфейса, получающего пакет). Этим действием в отношении запроса от каждого пакета может быть «принять» (ACCEPT), «отклонить» (REJECT или DROP) или «занести в журнал и продолжить» (LOG).

Каждое правило в любой таблице также относится к определенной *цепочке*. В таблице filter имеется три цепочки:

- INPUT — для пакетов, поступающих в систему;
- OUTPUT — для пакетов, покидающих систему;
- FORWARD — для пакетов, пересылаемых через систему.

Вероятно, большинство ваших правил фильтрации пакетов будет создано для цепочки `INPUT`. Эта цепочка фильтрует поступающие пакеты, по сути блокируя или разрешая доступ к службам в вашей системе. Распространенной практикой является разрешение доступа для пакетов, которые поступают из вашей локальной системы (интерфейс 1 o), ассоциированы с установленными подключениями и запрашивают одну из нескольких служб (например, ssh или http). Затем система может отбросить все остальные предназначенные для нее пакеты.

Помимо таблицы fi 1 ter, в вашем распоряжении также будет таблица nat, которая используется для трансляции сетевых адресов и пересылки пакетов. Кроме того, есть таблица mangle, которую можно использовать для изменения информации о пакетах (например, чтобы все выглядело, как будто они поступают с другого адреса, а не с того, с которого это происходит на самом деле).

Этапы конфигурирования брандмауэра следующие.

1. Создание правил, которые можно будет добавить в запущенное ядро с использованием команды iptables.
2. Сохранение этих правил в файл.
3. Загрузка этих правил до активизации ваших сетевых интерфейсов.

В примерах использования iptables, приведенных в этом подразделе, показан хороший набор образцов правил фильтрации брандмауэра, с которых вы можете

начать, разрешающей доступ к нескольким службам в локальной системе, но блокирующей доступ ко всему остальному.

ПРИМЕЧАНИЕ----------

Последовательность правил имеет важное значение. Правила читаются по порядку. Если пакет соответствует правилу, то он покидает цепочку (будучи принятым, отброшенным или отклоненным). Использование -A позволяет добавить правило в конец цепочки. С помощью -I# вы можете вставить правило в строку с определенным номером (#) в цепочке.

Сначала вам потребуется удалить все текущие правила iptables из ядра:

```
$ sudo iptables -F
```

Приведенное далее правило разрешает доступ для любых поступающих в систему пакетов, которые ассоциированы с установленными подключениями (например, для возврата данных с веб-страницы, запрошенных из локального браузера):

```
$ sudo iptables -A INPUT -m state \
    --state ESTABLISHED,RELATED -j ACCEPT
```

Это разрешает ICMP-запросы (например, запросы ping):

```
$ sudo iptables -A INPUT -p icmp -j ACCEPT
```

Такое правило разрешает любые запросы из локальной системы:

```
$ sudo iptables -A INPUT -i lo -j ACCEPT
```

Это разрешает внешние запросы через TCP-порт 22 (служба SSIT) для обеспечения удаленного входа в систему и служб удаленного копирования (повторно введите эту команду в случае с портами, имеющими другие номера, чтобы разрешить доступ к иным службам):

```
$ sudo iptables -A INPUT -rn state --state NEW \
    -m tcp -p tcp --dport 22 -j ACCEPT
```

Эти два правила разрешают запросы служб печати (порт 631) как посредством протокола TCP, так и UDP:

```
$ sudo iptables -A INPUT -m state --state NEW \
    -nt udp -p udp --dport 631 -j ACCEPT
$ sudo iptables -A INPUT -m state --state NEW
    -m tcp -p tcp --dport 631 -j ACCEPT
```

Данное правило отбрасывает любые пакеты, которые еще не были приняты явным образом:

```
$ sudo iptables -A INPUT -j REJECT \
    --reject-with icmp-host-prohibited
```

Эта команда сохраняет текущие правила из ядра в файл:

```
$ sudo iptables-save > /root/myiptables
```

Эта команда восстанавливает правила из файла в ядро:

```
$ sudo iptables-restore /root/myiptables
```

Вы можете добавить данную команду в сценарий запуска, как описано в подразделе «Сохранение и перезагрузка правил брандмауэра» данной главы, с целью загрузки правил брандмауэра iptables до активизации сетевых интерфейсов.

Отображение правил iptables

Чтобы проверить конфигурацию брандмауэра в своей системе, используйте iptables в сочетании с параметром -L. Вот как можно **вывести на экран набор текущих правил** брандмауэра в Linux (для таблицы по умолчанию с именем fi Iter):

```
$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target prot opt source destination
ACCEPT    all    --    anywhere anywhere      state RELATED.ESTABLISHED
ACCEPT    icmp   --    anywhere anywhere
ACCEPT    all    --    anywhere anywhere
ACCEPT    tcp    --    anywhere anywhere      state   NEW   tcp   dpt:ssh
ACCEPT    udp    --    anywhere anywhere      state   NEW   udp   dpt:ipp
ACCEPT    tcp    --    anywhere anywhere      state NEW tcp dpt:i pp
REJECT    all    --    anywhere               anywhere reject-with icmp-host-prohibited
Chain FORWARD (policy ACCEPT)
target          prot opt source                destination
Chain OUTPUT (policy ACCEPT)
target          prot opt source                destination
```

Добавьте параметр -v, чтобы увидеть более подробный вывод, содержащий правила. Следует отметить, что, как показывает -v, правило «все любой любой» фактически обеспечивает принятие всех пакетов из локальной системы (1о). Вы также можете увидеть номера портов вместо имен и количество пакетов и байтов, соответствующее каждому правилу:

```
$ sudo iptables -vnL          Подробный вывод, службы цифровых сообщений,
                              количество пакетов
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target prot opt in     out  source          destination
   85  6514 ACCEPT all -- *      *    0.0.0.0/0       0.0.0.0/0
state RELATED.ESTABLISHED
    0     0 ACCEPT icmp -- *      *    0.0.0.0/0       0.0.0.0/0
    0     0 ACCEPT all -- lo     *    0.0.0.0/0       0.0.0.0/0
    0     0 ACCEPT tcp -- *      *    0.0.0.0/0       0.0.0.0/0
          state NEW tcp dpt:22
   24  5144 ACCEPT udp - - *     *    0.0.0.0/0       0.0.0.0/0
          state NEW udp dpt:631
    0     0 ACCEPT tcp -- *      *    0.0.0.0/0       0.0.0.0/0
          state NEW tcp dpt:631
```

В этом примере показана таблица брандмауэра iptables с именем filter. Здесь видно, что для пакетов, поступающих в сетевые интерфейсы компьютера, и пакетов для IPP разрешены протоколы UDP и TCP. Кроме того, принимаются TCP-паке-

ты, соответствующие SSH. Пакеты также принимаются, если они ассоциированы с установленным подключением. Далее вы можете **взглянуть на таблицу nat:**

```
$ sudo iptables -t nat -L Отобразить текущую таблицу iptables с именем nat
Chain PREROUTING (policy ACCEPT)
target prot opt source destination
DNAT     tcp --     0.0.0.0/0          11.22.33.44       tcp dpt:8785 to:10.0.0.155:22
DROP     tcp --     0.0.0.0/0          0.0.0.0/0         tcp dpt:135
DROP     udp --     0.0.0.0/0          0.0.0.0/0         udp dpt:135
Chain POSTROUTING (policy ACCEPT)
target          prot    opt source                destination
MASQUERADE      all       -- 0.0.0.0/0             0.0.0.0/0
Chain OUTPUT (policy      ACCEPT)
target          prot    opt source                destination
```

Данная таблица nat касается такой функции, как трансляция сетевых адресов. Таблица nat позволяет вам, к примеру, использовать частные IP-адреса за пределами вашего брандмауэра. Когда пакеты с компьютеров, подключенных к внутренней LAN-сети, выходят из брандмауэра, исходный частный адрес перезаписывается IP-адресом внешнего интерфейса брандмауэра. Брандмауэр отслеживает эти сеансы, чтобы пропускать обратный трафик к компьютерам, подключенным к LAN-сети. Все это конфигурируется в строке MASQUERADE в цепочке POSTROUTING.

В показанном выше примере строка DNAT в цепочке PREROUTING приводит к тому, что все запросы через порт 8785 компьютера с IP-адресом 11.22.33.44 перенаправляются на IP-адрес подключенного к внутренней LAN-сети компьютера 10.0.0.155 через порт 22 (это трюк, позволяющий кому-либо подключиться посредством ssh к компьютеру за пределами брандмауэра через нестандартный порт).

Далее приведены другие примеры, как можно **вывести на экран информацию о вашем брандмауэре:**

```
$ sudo iptables -L --line-numbers          Показать номер строки
                                           для каждого правила
$ sudo iptables -nvL --line-numbers Добавить нумерацию (-n) и обеспечить
                                           генерирование подробного вывода (-v)
```

Вы можете **остановить применение или удалить все правила iptables,** имеющиеся в системе Ubuntu, как показано далее:

```
$ sudo iptables -F                         Удалить все правила iptables
```

Эта команда удаляет все правила, поэтому будьте осторожны, используя ее. Вам потребуется сразу же добавить новые правила.

Установка других правил брандмауэра

Вот дополнительные примеры, как можно использовать команду iptables для **изменения правил активного брандмауэра:**

```
$ sudo iptables -A INPUT -p TCP \ Добавить правило фильтрации
                                           INPUT для TCP
    -i ethO                    |           Использовать первый Ethernet-интерфейс.
```

```
    -■destination-port 25 \      Поступление через порт электронной почты (25)
    -j ACCEPT                    Принимать получаемые пакеты
$ sudo iptables -t nat \         Добавить nat-правило
    -A POSTROUTING \             Цепочка POSTROUTING
    -o ethl \                    Пакеты, получаемые посредством интерфейса ethl
    -j SNAT \                    Перейти к трансляции сетевых адресов
    --to-source 11.22.33.1       Использовать исходящий адрес 11.22.33.1
```

В нервом из двух приведенных выше примеров создается правило, которое разрешает поступление новых входящих запросов в вашу систему через порт 25. Возможной причиной этого может быть следующая: вы сконфигурировали свой компьютер как почтовый сервер (с Sendmail, Postfix или другой SMTP-службой). Во втором примере создается правило, которое добавляется в таблицу nat и позволяет брандмауэру осуществлять трансляцию исходных сетевых адресов (Source Network Address Translation, SNAT). Функция SNAT дает вам возможность обладать частными IP-адресами за пределами брандмауэра, которые можно задействовать при подключении к такой общедоступной сети, как Интернет, также используя при этом внешний IP-адрес брандмауэра.

Для использования SNAT или любой другой формы трансляции сетевых адресов вам потребуется **активизировать также IP-перенаправление** на компьютере. Это можно сделать, отредактировав файл /etc/sysctl .conf и убрав комментарий в случае с приведенной далее переменной:

```
net.ipv4.i p_forward=l
```

Если есть работающая через Интернет служба, предлагаемая на компьютере за пределами вашего брандмауэра, вы можете дать брандмауэру указание **перенаправлять запросы касательно этой службы** на данный компьютер. В приведенном далее примере используется функция под названием *«переадресация портов»* для передачи запросов касательно службы через брандмауэр на целевой компьютер за пределами брандмауэра:

```
$ sudo iptables -t nat -A PREROUTING \   Добавить nat-правило PREROUTING
    -p tcp -d 11.22.33.1          \      TCP-запросы по адресу 11.22.33.1
    --dport 80                    \      Через порт 80 (веб-служба)
    -j DNAT                       \      Перейти к DNAT-цели
    --to-destination 10.0.0.2            Перенаправлять пакеты по адресу 10.0.0.2
```

Есть множество других типов правил, которые вы можете создавать с целью изменения поведения своего брандмауэра. Загляните на MAN-страницу, посвященную i ptabl es, или на сайт Netfilter **(www.netfilter.org),** чтобы получить дополнительную информацию об использовании инструмента iptables.

После внесения показанных выше изменений вы увидите, что определены следующие правила:

```
$ sudo iptables -t nat -L
Chain PREROUTING (policy ACCEPT)
target     prot  opt source          destination
DNAT       tcp   --  anywhere        11.22.33.1  tcp dpt:www to:10.0.0.2
Chain POSTROUTING (policy ACCEPT)
target     prot  opt source          destination
```

```
SNAT       0      -- anywhere anywhere              to:11.22.33.1
Chain OUTPUT (policy ACCEPT)
target     prot opt source           destination
```

Сохранение и перезагрузка правил брандмауэра

Все изменения в правилах iptables, которые вы просто вводите в интерпретаторе команд, сохраняются только на протяжении текущего сеанса ядра. Если как-то не сохранить эти правила, то после следующей перезагрузки системы Ubuntu все они исчезнут, что, вероятно, окажется не тем, что вам нужно. Для сохранения своих правил iptables выполните iptables-save:

```
$ sudo iptables-save > iptables.rules      Сохранить правила в файл в текущем каталоге
$ sudo cp iptables.rules /etc              Скопировать сохраненные правила в /etc
```

Этот двухэтапный процесс необходим из-за ограничений в случае с каталогом /etc (вы можете изменить эти ограничения, но, вероятно, это нелучшая идея). В результате правила будут сохранены для использования позднее.

Далее вы можете сконфигурировать систему Ubuntu на загрузку сохраненных правил каждый раз, когда она будет активизировать Ethernet-интерфейс (это характерно для каждой сетевой карты или интерфейса в вашей системе). Отредактируйте файл /etc/network/interfaces. После конфигурации iface касательно Ethernet-интерфейса, например, ethO, вызовите iptables-restore, как показано в приведенном далее фрагменте:

```
auto ethO
iface ethO inet dhcp
  pre-up iptables-restore < /etc/iptables.rules
```

Это дополнение, внесенное в файл i nterfaces, обеспечит вызов i ptabl es - restore для восстановления правил, сохраненных ранее в файл /etc/iptables. rules.

Инструменты обеспечения повышенной безопасности

Дюжины страниц с описанием связанных с безопасностью команд далеко не достаточно, чтобы глубоко рассмотреть инструменты обеспечения безопасности, доступные вам как системному Linux-администратору. Помимо команд, рассмотренных в этой главе, далее приведено описание некоторых инструментов, которые вы можете захотеть изучить, чтобы затем использовать с целью повышения безопасности своей системы Linux.

- **Security Enhanced Linux (SELinux) (Linux с улучшенной безопасностью)** — обеспечивает средства защиты файлов, каталогов и приложений в вашей системе Linux так, чтобы эксплуатацию одной из областей вашей системы нельзя

было использовать для несанкционированного доступа к другим областям. Например, если бы злоумышленники скомпрометировали ваш веб-демон, то вовсе не обязательно, что им удалось бы сделать то же с остальной системой.

SELinux был разработан Агентством национальной безопасности США (National Security Agency, NSA), на сайте которого есть соответствующий раздел с часто задаваемыми вопросами: **www.nsa.gov/selinux/info/faq.cfm.** Вам потребуется установить SELinux в виде отдельных пакетов. Подробности можно узнать по адресу **https://wiki.ubuntu.com/SELinux.**

- **Протоколирование на центральном syslog-сервере** — если вы будете управлять более чем одной парой Linux-серверов, то предпочтительно размещать файлы журналов всех систем на центральном syslog-сервере. При реализации своего syslog-сервера вы, возможно, захотите использовать syslog-ng. Кроме того, если 1 ogwatch покажется вам больше не соответствующим вашему уровню, следует рассмотреть возможность использования такого анализатора файлов журналов, как, например, Splunk.
- **Tripwire** — задействовав пакет tripwire, вы сможете сделать моментальный снимок всех файлов в системе и использовать его позднее, чтобы узнать, изменился ли какой-либо из этих файлов. Этот инструмент особенно полезен для выяснения, подвергались ли какие-либо приложения изменениям, которые не следовало в них вносить. Сначала вы зафиксируете базовое состояние системного файла. Затем через равные интервалы будете запускать проверку целостности посредством Tripwire с целью узнать, подвергались ли изменениям какие-либо из ваших приложений или конфигурационных файлов.
- **База данных APT** — еще один способ проверить ваши приложения на предмет изменений заключается в использовании APT-команд, позволяющих проводить валидацию приложений и конфигурационных файлов, установленных вами в системе. Информацию, касающуюся использования APT-команд и dpkg для проверки содержимого установленных пакетов, можно найти в гл. 2.
- chkrootki t — если у вас возникнут подозрения, что система была скомпрометирована, установите chkrootkit, загрузив его с сайта **www.chkrootkit.org.** Этот инструмент поможет вам обнаружить наборы утилит, которые могли быть использованы для получения контроля над вашим компьютером. Рекомендую запускать chkrootki t с Live CD либо после монтирования подозрительного диска в чистую систему.

Резюме

Существует множество инструментов, доступных для защиты вашей системы Linux, однако первая линия обеспечения безопасности начинается с защиты учетных записей пользователей в вашей системе, а также служб, которые в ней запущены. Такие команды, как useradd, groupadd и password, являются стандарт-

ными инструментами для конфигурирования учетных записей пользователей и групп.

Большинство серьезных нарушений безопасности извне может иметь место из-за злоумышленников, осуществляющих доступ к вашим системам посредством общедоступных сетей, поэтому конфигурирование надежных брандмауэров имеет важное значение для любой системы, подключенной к Интернету. Инструмент iptables обеспечивает функции брандмауэра, встроенные в ядро Linux.

15 Конфигурирование хоста виртуализации и виртуальных машин

В этой главе:

- проверка компьютера на предмет поддержки виртуализации;
- установка программного обеспечения для виртуализации;
- создание и управление виртуальными машинами с помощью vi rt-manager;
- работа с виртуальными машинами из командной строки.

Используя свою систему Ubuntu в качестве хоста виртуализации, вы сможете запускать несколько операционных систем на одном компьютере. Системы, создаваемые вами на хосте, называются виртуальными машинами. Па виртуальной машине может функционировать Microsoft Windows, а также Fedora или другая Linux-системалибо почти любая ОС, способная работать непосредственно на компьютерной архитектуре хоста.

Установив виртуальную машину, вы сможете работать с ней во многом так же, как с операционными системами, установленными непосредственно на компьютерном аппаратном обеспечении. Однако виртуальные машины легче дублировать, переносить на другие виртуальные хосты с целью повышения производительности и конфигурировать на переключение на другой хост, если текущий окажется неработоспособным. Используя виртуальные машины, вы сможете эффективнее задействовать свою компьютерную инфраструктуру.

Есть множество причин для конфигурирования виртуального хоста. Например, вам может потребоваться сделать следующее:

- опробовать другую операционную систему, не расходуя на нее все ресурсы компьютера;
- запустить приложение, для работы которого необходима определенная версия ОС и конфигурация, отличные от используемых вами обычно;
- сконфигурировать систему для тестирования нового приложения без прерывания другой работы на вашем компьютере;

- создать копию установленной операционной системы и быстро развернуть новую версию с помощью этой копии;
- поэкспериментировать с ОС путем, который не приведет к нарушению хост-системы.

Задействовав свою систему Ubuntu в качестве хоста виртуализации, вы сможете приступить к созданию компьютерной инфраструктуры со способностью к масштабированию, если вам потребуется больше вычислительной мощности. Располагая несколькими хостами, можно переносить виртуальные машины с целью повышения производительности или завершать работу хостов с низким коэффициентом использования.

В этой главе описывается, как сконфигурировать Ubuntu в качестве хоста виртуализации с использованием KVM (Kernel-based Virtual Machine — виртуальная машина на базе ядра) — возможности, встроенной в ядро Linux. Затем следует описание команд, которые можно применять для создания и управления виртуальными машинами.

Поддерживает ли ваш компьютер виртуализацию

Критически важным требованием для использования системы Ubuntu в качестве хоста виртуализации KVM является наличие центрального процессора с поддержкой виртуализации. Кроме того, в большинстве случаев вам будет необходимо удостовериться, что компьютер обладает достаточными ресурсами для обеспечения эффективной работы виртуальных машин.

Помимо надлежащего центрального процессора, хосту KVM потребуется достаточный объем (оперативной) памяти и доступного дискового пространства для обеспечения соответствующего уровня производительности, необходимого для функционирования ваших виртуальных машин. Вам также понадобится учесть, какая пропускная способность сети нужна для обеспечения требуемого уровня обслуживания.

В нижеследующих разделах описывается, как проверить компьютер с целью убедиться в наличии ресурсов, необходимых для использования Ubuntu в качестве хоста виртуализации.

Проверка центрального процессора на предмет поддержки виртуализации

Чтобы использовать KVM, процессор, установленный в вашей системе, должен поддерживать либо технологию виртуализации Intel VT, либо AMD-V. В Ubuntu есть команда kvm-ok, доступная благодаря пакету cpu-checker, которуювы можете задействовать для проверки центрального процессора на предмет поддержки виртуализации. Вот как можно **установить и использовать команду kvm-ok:**

```
$ sudo apt-get install cpu-checker
$ sudo kvm-ok
INFO: /dev/kvm does not exist
HINT:   sudo modprobe kvm_intel
INFO: Your CPU supports KVM extensions
KVM acceleration can be used
```

Из вывода kvm-ok видно, что центральный процессор поддерживает соответствующие расширения, требуемые KVM. Однако программное обеспечение, необходимое для использования KVM, еще не установлено, а также пока не загружены нужные модули. Мы сделаем это позднее.

Между тем есть ручной способ **провести проверку на предмет под держки виртуализации.** Вы можете проверить флаги, заданные для центрального процессора в /proc/cpuinfo. Задействовав команду egrep, можно осуществить поиск в этом файле информации касательно поддержки Intel VT (vmx) или AMD-V (svm), как показано далее:

```
$ egrep "(svm|vmx)" /proc/cpuinfo
flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 cl flush dts acpi mmx fxsr sse sse2 ss ht tm
pbe syscall nx lm constant_tsc arch_perfmon-pebs bts rep_good nopl
aperfmperf pni dtes64 monitor ds_cpl vmx est tm2 ssse3 cxl6 xtpr
pdcm xsave lahfjm dtherm tpr_shadow vnmi flexpriority
```

Вы должны будете увидеть либо флаг vmx, либо svm (но не оба сразу). Если команда egrep не сгенерирует какого-либо вывода, это означает, что центральный процессор (-ы) вашего компьютера не поддерживает виртуализацию KVM. Следующим этапом будет проверка базовой системы ввода/вывода.

Активизация поддержки виртуализации в базовой системе ввода/вывода

Если вы полагаете, что компьютер способен поддерживать виртуализацию, однако соответствующий флаг не задан, вам, возможно, потребуется **активизировать поддержку виртуализации в базовой системе ввода/вывода.** Для этого перезагрузите компьютер и прервите процесс загрузки, когда появится первый экран базовой системы ввода/вывода. На этом экране вы, вероятно, увидите что-то вроде **Press F12 to go into Setup** (Нажмите клавишу **F12,** чтобы войти в программу настройки). Нажмите **Delete,** функциональную клавишу либо другую указанную.

На появившемся экране базовой системы ввода/вывода найдите что-то вроде заголовка **CPU** (Центральный процессор) или **Performance** (Производительность) и выберите его. Затем отыщите функцию виртуализации, например **Intel Virtualization Tech** (Технология виртуализации Intel), и активизируйте ее. На рис. 15.1 приведен пример такого экрана базовой системы ввода/вывода с активизированной поддержкой виртуализации.

После того как вы модифицируете настройку виртуализации в базовой системе ввода/вывода и сохраните ее, потребуется выключить компьютер, чтобы соответствующее изменение в базовой системе ввода/вывода вступило в силу. Затем вы сможете проверить другие параметры своего компьютера.

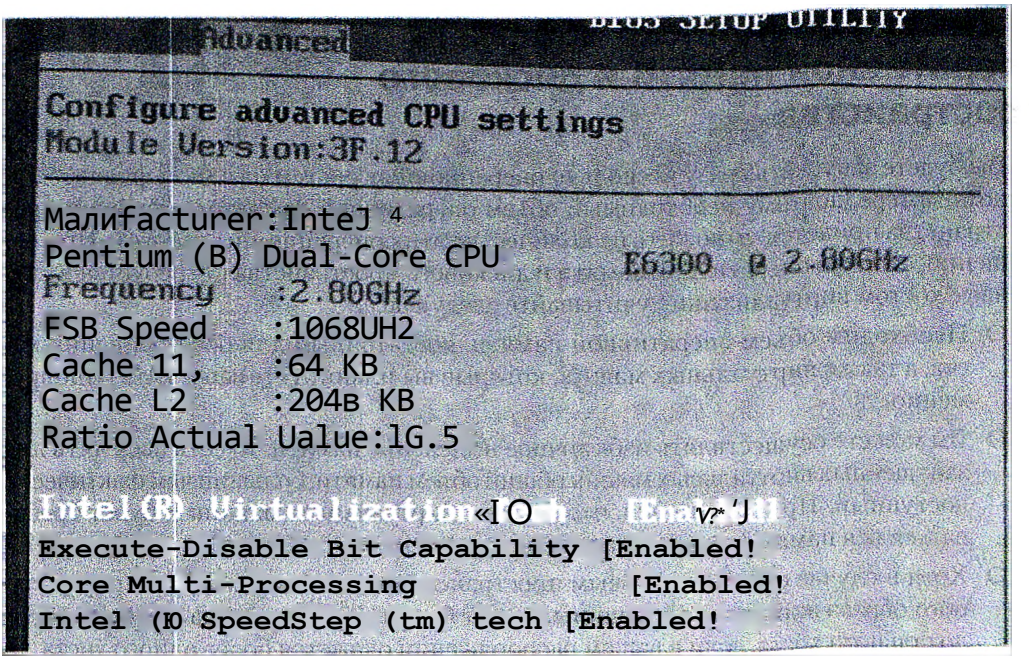

Рис. 15.1. Активизация поддержки виртуализации в базовой системе ввода/вывода

Является ли центральный процессор хост-компьютера 32- или 64-битным

Если это представляется возможным, то используйте компьютер с 64-битным центральным процессором в качестве хоста виртуализации. Некоторые операционные системы, например Red Hat Enterprise Linux, даже не поддерживают KVM на 32-битных системах. Одним из основных недостатков хоста KVM с 32-битным центральным процессором является то, что каждая виртуальная машина ограничивается памятью 2 Гбайт.

Чтобы **узнать, является ли центральный процессор вашего компьютера 32- или 64-битным,** вы можете исследовать флаги центрального процессора (подобно тому как делали это при проведении проверки на предмет поддержки виртуализации). Введите следующее:

```
$ sudo egrep 1m /proc/cpuinfo
Flags : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
pge mca cmov pat pse36 cl flush dts acpi mmx fxsr sse sse2 ss ht tm
pbe syscall nx lm constant_tsc arch_perfmon pebs bts rep_good nopl
aperfmperf pni dtes64 monitor ds_cpl vmx est tm2 ssse3 cxl6 xtpr
pdem xsave lahfjm dtherm tpr_shadow vnmi flexpriority
```

Флаг lm — это сокращение от long mode («длинный режим»). Если такой флаг присутствует, это означает, что центральный процессор вашего компьютера является 64-битным.

Проверка доступного объема оперативной памяти и дискового пространства

Вы будете запускать сразу несколько операционных систем на одном физическом компьютере, поэтому необходимый объем оперативной памяти и дискового пространства придется умножить на количество этих операционных систем. При решении, сколько оперативной памяти и дискового пространства потребуется в случае с хостом виртуализации, учитывайте следующее.

- Необходим объем оперативной памяти, достаточный для обслуживания хоста, а также виртуальных машин, которые вы планируете запускать одновременно.
- Вы можете осуществлять избыточное выделение памяти, то есть выделять для запущенных виртуальных машин общий объем памяти, больший чем фактически доступный. Предполагается, что не всем виртуальным машинам будет требоваться вся память, выделяемая им одновременно.
- Хотя в случае с KVM дисковым пространством, используемым для виртуального образа виртуальной машины, может быть пространство на жестком диске локального хоста, если в вашем распоряжении окажется более одного хоста, то рассмотрите возможность использования сетевого запоминающего устройства (например, с поддержкой iSCSI). Наличие совместно используемого запоминающего устройства Г юз вол ит динам ически переносить ваши виртуальные машины на другие хосты.

Чтобы узнать объем доступной памяти, используйте команду free:

```
$ free -m
             total       used       free     shared    buffers     cached
Mem:          3920       3442        478          0        165       2616
-/+ buffers/cache:        659       3261
Swap:         4059          0       4059
```

Общий объем оперативной памяти в случае с этой системой равен 4 Гбайт (3920 Мбайт). Это не слишком большой размер оперативной памяти для хоста виртуализации. При этом можно запустить одну или две виртуальные машины на этой системе одновременно, просто чтобы удостовериться, что виртуализация работает.

Не забывайте, что при оценке, сколько памяти нужно, вы можете использовать такую команду, как top, которая позволит выяснить размер требуемой памяти. При подсчете посредством top фактически используемого объема памяти суммируйте показатели в столбце резидентной, а не виртуальной памяти (в столбце которой показывается объем памяти, выделяемый каждому процессу, однако обычно эти показатели оказываются больше фактически используемого объема).

Что касается дискового пространства, потребуется убедиться, что его доступно достаточно в случае с каталогом, в котором будут располагаться образы дисков, используемые виртуальными машинами. По умолчанию путь к нему выглядит так:

/var/lib/libvi rt/images. Воспользуйтесь командой df для **проверки, какой объем дискового пространства доступен** в случае с этим каталогом.

ПРИМЕЧАНИЕ---------- ----------

Каталог /var/lib/libvirt/images не будет создан, пока вы не установите пакет libvirt-bin, о чем мы поговорим в следующем разделе, поэтому до его инсталляции можете использовать каталог /var/lib:

```
$ df -h /var/lib/libvirt/images
Filesystem            Size Used Avail Use% Mounted on
/dev/mapper/ubuntutb-root 88G 12G 72G 14% /
```

В данном случае выбранный каталог является частью корневой (/) файловой системы, для которой доступно 72 Гбайт дискового пространства. Некоторые монтируют файловую систему с сетевого запоминающего устройства либо создают логический том LVM в локальной системе, который можно увеличивать по мере необходимости. Так или иначе, 72 Гбайт достаточно, чтобы опробовать пару виртуальных машин.

Добавление программного обеспечения для виртуализации

К программным компонентам, которые вам потребуется добавить для виртуализации KVM, относятся следующие:

- libvirt — обеспечивает интерфейс в случае с аппаратным обеспечением для виртуализации;
- qemu — эмулирует аппаратное обеспечение персонального компьютера для виртуальных машин;
- bridge-utils — позволяет создать сетевой мост от виртуальных машин через хост.

Выполните приведенную далее команду apt-get, чтобы **установить базовое программное обеспечение, необходимое для осуществления виртуализации KVM:**

```
$ sudo apt-get install libvirt-bin kvm bridge-utils qemu-common \
    qemu-kvm qemu-utils
```

Помимо базового программного обеспечения для виртуализации KVM, вы также можете добавить графический интерфейс для управления своими виртуальными машинами. Вот как можно установить графическое программное обеспечение virt-manager. позволяющее управлять виртуальными машинами:

```
$ sudo apt-get install virt-manager
```

После установки virt-manager вы сможете управлять своими виртуальными машинами посредством графического интерфейса либо из командной строки. Далее вам потребуется убедиться, что учетная запись пользователя, с применением которой, как вам необходимо, должно осуществляться управление виртуализацией, сконфигурирована соответствующим образом.

Добавление пользователя с нужной учетной записью в группу iibvirtd

Если пользователь, чью учетную запись вы собираетесь задействовать для управления KVM, не является членом группы 1 i bvi rtd, потребуется добавить его в группу. Например, если вы захотите, чтобы управлять виртуализацией смог тот, кто обладает учетной записью пользователя **chris,** вот как можно **добавить соответствующего пользователя в группу Iibvirtd:**

```
$ sudo adduser chris Iibvirtd
Adding user 'chris' to group 'Iibvirtd' ...
Adding user chris to group Iibvirtd
Done.
```

На данном этапе вы должны быть готовы приступить к использованию виртуализации. Однако может потребоваться перезагрузить систему, чтобы убедиться, что все необходимые службы функционируют, а пользователь, только что добавленный вами в группу Iibvirtd, вошел в систему и готов к работе.

Управление виртуальными машинами с использованием virt-manager

Графический интерфейс Virtual Machine Manager (vi rt-manager) (Менеджер виртуальных машин) является популярным инструментом управления виртуальными машинами KVM. Прежде чем использовать такие команды, как virt-instal 1 (для установки новой виртуальной машины) или virsh (для управления виртуальными машинами), вы, возможно, захотите опробовать virt-manager, который проведет вас через процесс создания ваших первых виртуальных машин интуитивно понятным образом.

Для начала работы с vi rt-manager вам необходимо сделать следующее.

- **Загрузить ISO-образы.** Загрузите ISO-образы операционных систем, которые вы хотите либо установить, либо запустить прямо с виртуальной машины. Например, в этой главе я использую Live CD с Fedora в качестве установочного CD, а также непосредственно как Live CD.
- **Запустить virt-manager.** Выберите на панели управления Ubuntu значок **Virtual Machine Manager** (Менеджер виртуальных машин) (либо введите virt-manager в командной строке) от имени пользователя, добавленного вами в группу kvm. Появится окно **Virtual Machine Manager** (Менеджер виртуальных машин).

Прежде чем вы приступите к созданию первой виртуальной машины, исследуйте окно **Virtual Machine Manager** (Менеджер виртуальных машин) на предмет информации о своих настройках виртуализации. Вот примеры.

- **Virtual Networks** (Виртуальные сети) — выберите **Edit** (Редактировать), а затем **Connection Details** (Детали подключения). В появившемся после этого окне **Connection Details** (Детали подключения) перейдите на вкладку **Virtual Networks** (Виртуальные сети). Вы увидите, что сеть по умолчанию, управляемая вашим виртуальным хостом, обладает адресом 192.168.122.0/24. Виртуальный хост

будет передавать IP-адреса виртуальным машинам в диапазоне от 192.168.122.2 до 192.168.122.254. Чтобы все при этом работало, хост обеспечит трансляцию сетевых адресов (Network Address Translation, NAT) на устройстве vibrO. Пример соответствующего окна приведен на рис. 15.2.

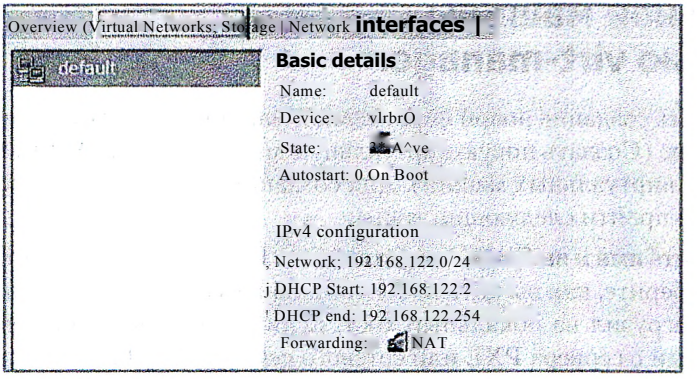

Рис. 15.2. **Виртуальная сеть по умолчанию использует трансляцию сетевых адресов и частный пул сетевых адресов**

О **Storage** (Хранение) — выберите **Edit** (Редактировать), а затем **Connection Details** (Детали подключения). В появившемся окне **Connection Details** (Детали подключения) перейдите на вкладку **Storage** (Хранение), /var/1 ib/1 ibvi rt/images будет каталогом по умолчанию, используемым для хранения томов. В примере, который приведен на рис. 15.3, показан объем свободного дискового пространства и список всех существующих томов. Вы можете создать здесь новые тома или сделать это, когда создадите свои виртуальные машины.

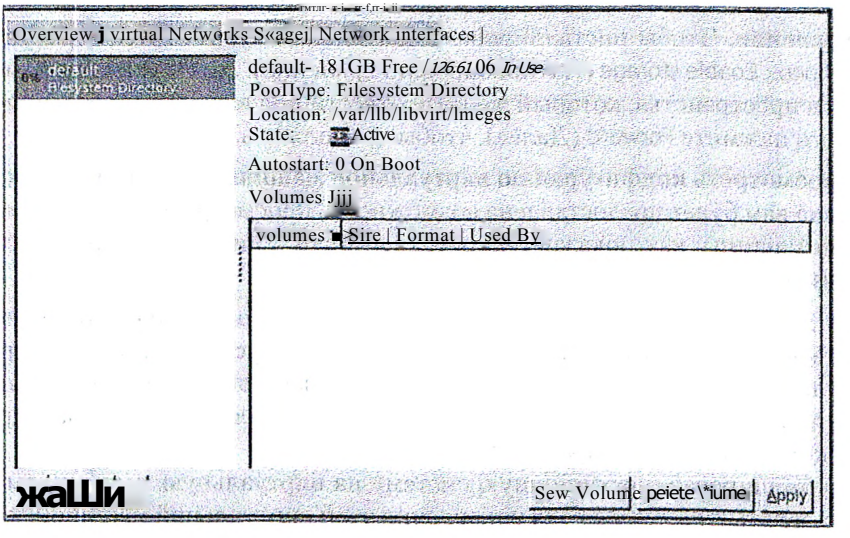

Рис. 15.3. **Проверка объема свободного дискового пространства в случае с томами**

Когда все будет на месте, вы сможете приступить к конфигурированию и установке своей первой виртуальной машины.

Создание виртуальной машины с помощью virt-manager

Чтобы начать создание новой виртуальной машины, выберите значок **Create a new virtual machine** (Создать новую виртуальную машину) в окне **Virtual Machine Manager** (Менеджер виртуальных машин). Для создания новой виртуальной машины вам необходимо пройти следующие этапы.

1. **Присвоить имя и выбрать тин установки.** Введите имя для виртуальной машины и выберите, как вы хотели бы установить ее. Я использовал ISO-образ, который загрузил на локальный хост. Если у вас есть установочный носитель, доступный с сервера PXE или сетевого сервера (HTTP, FTP или NFS), то выберите любой из соответствующих вариантов. Затем нажмите кнопку **Forward** (Далее).
2. **Указать местоположение установочного носителя и тип операционной системы.** Определите точное местоположение своего установочного носителя. Я выбрал **Use** ISO **image** (Использовать ISO-образ) и щелкнул на кнопке **Browse** (Обзор), чтобы указать путь к своему ISO-образу в локальной системе. Тип операционной системы я оставил как **Generic** (Общий). В конце нажмите **Forward** (Далее).
3. **Оперативная память и центральные процессоры.** Укажите объем оперативной памяти, который будет использоваться виртуальной машиной, а также количество центральных процессоров. Затем щелкните **Forward** (Далее).
4. **Хранение.** Чтобы инсталлировать новую систему, убедитесь, что установлен флажок **Enable storage** (Активизировать хранение). Затем укажите объем дискового пространства, который желаете отвести под виртуальную машину. После этого нажмите **Forward** (Далее), чтобы продолжить.
5. **Просмотреть конфигурацию виртуальной машины.** Перед запуском установщика вам будет предоставлена возможность просмотреть настройки виртуальной машины, как показано на рис. 15.4. Если с ними все в порядке, нажмите **Finish** (Завершить).

На данном этапе ISO-образ загрузится в новом окне. Происходящее дальше будет зависеть от того, используете ли вы Live CD, установщик или загрузочный образ другого типа. При условии, что это будет загрузчик, пройдите процесс установки так же, как сделали бы это, если бы осуществляли установку непосредственно на компьютерное аппаратное обеспечение.

Инсталлировав операционную систему на виртуальную машину, вы сможете использовать vi rt-manager для управления этой виртуальной машиной, как описано в следующем разделе.

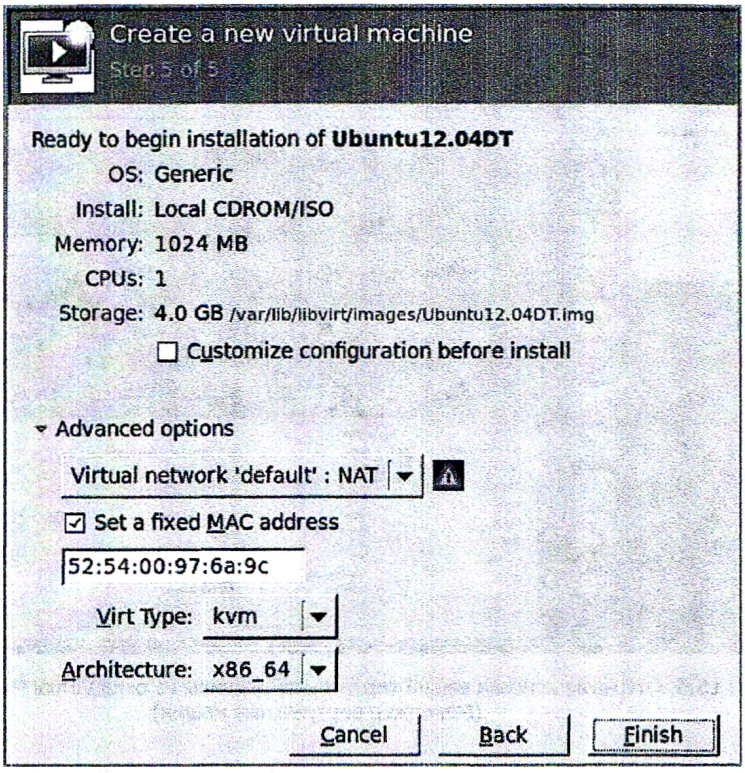

Рис. 15.4. Просмотр настроек вашей виртуальной машины перед запуском установщика

Запуск и остановка виртуальных машин с помощью virt-manager

После того как вы установите виртуальную машину с использованием virt-manager, в окне Virtual Machine Manager (Менеджер виртуальных машин) появится связанный с ней элемент. Чтобы открыть консольное окно для использования виртуальной машины, дважды щелкните кнопкой мыши на элементе, который с ней связан. На рис. 15.5 показана виртуальная машина, только что запущенная из окна Virtual Machine Manager (Менеджер виртуальных машин).

Закончив использовать виртуальную машину на данный момент, просто закройте окно. Виртуальная машина продолжит работать и останется доступной посредством любого сетевого интерфейса, сконфигурированного вами для нее. Вот некоторые другие действия, которые вы сможете предпринять в отношении виртуальной машины, щелкнув правой кнопкой мыши на связанном с ней элементе.

- Run (Запустить) — если виртуальная машина не запущена в настоящее время, то выбор этого варианта приведет к ее загрузке.
- Pause (Поставить на паузу) — ставит виртуальную машину на паузу.

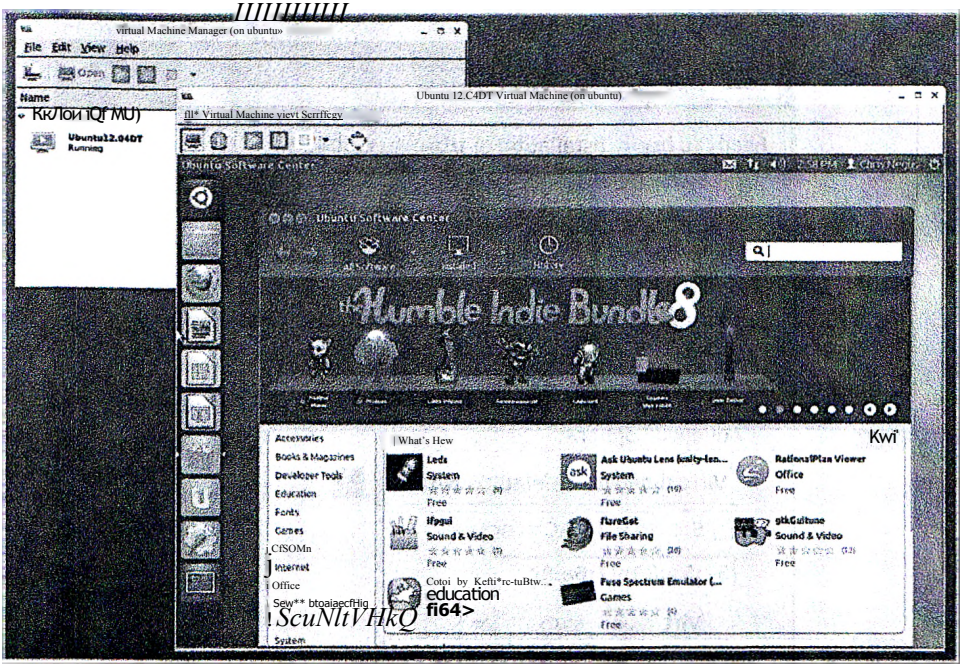

Рис. 15.5. Открытие консоли вашей виртуальной машины из окна Virtual Machine Manager (Менеджер виртуальных машин)

- **Shutdown** (Завершить работу) — указав этот вариант, вы затем также сможете выбрать **Reboot** (Перезагрузить), **Shut Down** (Выключить), **Force Off** (Принудительно выключить) или **Save** (Сохранить). Выбор **Force Off** (Принудительно выключить) аналогичен извлечению вилки из розетки в случае с физической машиной. **Save** (Сохранить) обеспечивает сохранение всего находящегося в памяти виртуальной машины в файл, который вы сможете исследовать позднее.
- **Clone** (Клонировать) — позволит клонировать текущую виртуальную машину, благодаря чему вы сможете получить в распоряжение несколько экземпляров этой виртуальной машины.
- **Migrate** (Перенести) — если у вас будет еще один совместимый хост KVM, вы сможете перенести на него виртуальную машину.

Теперь, когда вы видели, что можно делать с виртуальными машинами KVM посредством графических инструментов, следующий раздел поможет вам в использовании команд для выполнения подобных действий.

Управление виртуальными машинами с помощью команд

Если можно устанавливать и управлять виртуальными машинами посредством изящных графических инструментов, таких, например, как virt-manager, то за-

чем утруждать себя управлением виртуальными машинами с помощью команд? На то есть множество причин. Например, может оказаться, что на вашем хосте KVM не установлен рабочий стол, либо вам может потребоваться выполнить команды для работы с виртуальными машинами из сценария интерпретатора команд.

Для начала можете прибегнуть к команде virt-install, чтобы установить виртуальную машину. Задействовав virt-clone, можно клонировать существующий виртуальный образ. Для управления виртуальными машинами вы можете использовать команду virsh, которая позволяет отображать информацию о виртуальных машинах, а также запускать, останавливать и перезагружать их.

Создание виртуальной машины посредством virt-install

Вместо того чтобы щелкать кнопкой мыши в окне Virtual Machine Manager (Менеджер виртуальных машин) с целью создания виртуальной машины, вы можете передать параметры, необходимые для ее создания с использованием команды virt-install. Прежде чем задействовать virt-install, потребуется создать образ накопителя информации. Один из способов сделать это заключается в применении команды qemu-img.

Создание образа накопителя информации для виртуальной машины с помощью qemu-img

Команда qemu-img задействуется для создания файлов образов, которые виртуальные машины смогут использовать в качестве своих накопителей информации. Для установщика образы будут выглядеть как обычные жесткие диски или другие блочные устройства для хранения. Вот описание поддерживаемых форматов образов.

- **raw** — это тип исходных образов по умолчанию для qemu-img. Этот тип образов прост. Он является форматом, который вам следует использовать, если вы рассчитываете на экспорт образов в другие виртуальные среды.
- **qcow2** — если вы ожидаете, что qemu продолжит использовать определенный образ, то задействуйте данный формат. Он не расходует сразу все выделенное дисковое пространство, а потребляет его по мере увеличения размера образа. Он также поддерживает шифрование.
- **Другие форматы** — большинство других форматов образов, сочетаемых с qemu-img, поддерживается главным образом ради обеспечения совместимости с более старыми версиями. К таковым относятся qcow, cow, vdi, vmdk, vpc и cloop. Введите man qemu-img, чтобы увидеть информацию об этих типах образов.

В приведенном далее примере использования qemu-img создается qcow2-образ, путь к которому выглядит как /var/stuff/mine.qcow2. Для этого образа выделяется 8 Гбайт дискового пространства (хотя в действительности он занимает на диске

только объем пространства, который использует виртуальная машина). Параметр preallocation=metadata позволяет улучшить производительность образов по мере увеличения их размеров.

```
$ sudo qemu-img create -f qcow2 -o preallocation=metadata \
    /var/stuff/mine.qcow2 8G
```

После создания qcow2-обра3а вы можете провести в отношении его проверку на согласованность посредством qemu-img в сочетании с параметром check:

```
$ sudo qemu-img check /var/stuff/mine.qcow2 Проверить, все ли в порядке
                                            с образом
```

На любом этапе процесса вам может потребоваться проверить, сколько дискового пространства занимает виртуальная машина. Воспользуйтесь qemu-img вместе с параметром info:

```
$ sudo qemu-img info mine.qcow2
  See allocated/actual image sizes
image: mine.qcow2
file format: qcow2
virtual size: 8.0G (8589934592 bytes)
disk size: 2.6G
cluster_size: 65536
```

Команда qemu-img info из приведенного выше примера была выполнена после установки виртуальной машины. Вы видите, что, хотя выделено 8 Гбайт дискового пространства, на данный момент занято только 2,6 Гбайт.

Установка виртуальной машины с помощью virt-install

Команда virt-install является универсальным инструментом, позволяющим создавать новые виртуальные машины. Вы можете определить атрибуты среды виртуальной машины в командной строке.

Далее приведен пример строки команды virt-install, создающей виртуальную машину Ubuntu. Эта команда включает большое количество параметров, вместо которых вам пришлось бы щелкать кнопкой мыши или указывать соответствующие варианты в окне virt-manager. Следует отметить, что эта команда включает образ, который я создал ранее в текущей главе посредством команды qemu-img.

```
$ virt-install --connect qemu:///system --name cn_ubuntul2.04 \
    --ram 1024 --disk path=/var/stuff/mine.qcow2,format=qcow2 \
    --network=bridge:virbrO,model=virtio --vnc --os-type=linux \
    --cdrom /var/stuff/ubuntu-12.04.2-desktop-amd64.iso                \
    --noautoconsole --keymap=en-us
```

Вот что означают параметры команды virt-install.

- -connect определяет местоположение службы виртуализации в случае с гипервизором. Определенный здесь аргумент qemu: ///system является местоположением по умолчанию, используемым KVM.

- -паше указывает имя для представления виртуальной машины. Вы можете присвоить ей любое имя, однако обычно выбирается такое, которое будет обозначать тип операционной системы и, возможно, номер выпуска.
- - - ram позволяет задать объем оперативной памяти, который сможет использовать виртуальная машина.
- --disk_path сообщает местоположение образа диска и его формат (в данном случае qcow2).
- network-bri dge: vi rbrO определяет сетевой интерфейс, который виртуальная машина должна использовать на хосте для коммуникаций с другими хостами.
- --vnc дает указание использовать VNC для обеспечения доступа к консоли виртуальной машины.
- - - os-type определяет виртуальную машину как такую, на которой установлена операционная система Linux.
- --cdrom указывает местоположение ISO-образа, откуда будет запущена установка.
- - - noautoconsol e предотвращает автоматическое открытие консоли виртуальной машины. Благодаря этому вы можете использовать другие средства просмотра для наблюдения за виртуальной машиной, например virt-manager или virt-viewer.
- --keymap=en-us переключает раскладку клавиатуры на американский вариант английского языка.

Загляните на MAN-страницу, посвященную virt-install (для этого введите man vi rt-i nstal 1), чтобы увидеть другие параметры, которые можно использовать в сочетании с командой vi rt-i nstal 1.

Как только будет запущено выполнение команды vi rt-i nstal 1, вы сможете открыть приложение на рабочем столе, чтобы наблюдать за ходом установки. Команды virt-manager или vi rt-viewer относятся к тем, которые вы можете задействовать для вывода на экран консоли своей виртуальной машины. При использовании Ubuntu Live CD, как в этом примере, вам пришлось бы задействовать мышь в процессе установки, а также ввести некоторую информацию для завершения инсталляции.

Когда виртуальная машина будет установлена, вы сможете управлять ею с помощью команды virsh.

Запуск и остановка виртуальных машин с помощью virsh

Команда virsh является хорошим инструментом управления созданными виртуальными машинами. Ее можно использовать, чтобы узнать, какие виртуальные машины запущены; кроме того, вы сможете запускать, останавливать, ставить на паузу и иным образом управлять ими. Вот несколько примеров применения команды virsh.

```
$ virsh help                                  Просмотр списка подкоманд virsh
$ virsh list                                  Показать запущенные на данный момент
                                                виртуальные машины
 Id Name                 State
----------------------------------
  5 Ubuntul2.04DT        running
  6 Fedoral7DT           running
$ virsh shutdown Fedoral7DT                   Завершить работу виртуальной машины
                                                Fedora
Domain Fedoral7DT is being shutdown
$ virsh destroy Ubuntul2.04DT                 Немедленно остановить виртуальную машину
                                                Ubuntu
Domain Fedoral7DT destroyed
$ virsh undefine Fedoral7DT                   Полностью удалить определение виртуальной
                                                машины
Domain Fedoral7DT has been undefined
$ virsh version                               Показать информацию о текущей версии
Compiled against library: libvir 0.9.8
Using library: libvir 0.9.8
Using API: QEMU 0.9.8
Running hypervisor: QEMU 1.0.0
$ virsh hostname                              Показать имя хоста гипервизора
$ virsh autostart Ubuntul2.04DT               Настроить виртуальную машину на запуск
                                                во время загрузки
```

Команда virsh поддерживает множество других параметров, которые вы можете использовать для управления своими виртуальными машинами. Подробности вы найдете на MAN-странице, посвященной virsh (введитеman virsh).

Резюме

Все больше и больше компьютеров задействуется в качестве хостов виртуализации (иногда также называемых гипервизорами) для эффективного использования компьютерных инфраструктур. Задействовав компьютер как хост виртуализации, вы сможете одновременно запускать сразу несколько разных операционных систем на одном и том же физическом компьютере.

KVM-параметры Linux позволяют использовать Ubuntu и другие Linux-системы в качестве виртуальных хостов. Если вы пожелаете задействовать компьютер как виртуальный хост, эта глава поможет вам предпринять шаги, необходимые, чтобы убедиться в поддержке виртуализации вашим компьютерным аппаратным обеспечением.

Окно Virtual Machine Manager (Менеджер виртуальных машин) (команда vi rt-manager) позволяет устанавливать и управлять виртуальными машинами графическим путем. Благодаря тому что использование этого окна интуитивно понятно, оно является хорошим средством, чтобы начать применять виртуализацию в Ubuntu.

Если вы предпочитаете инструменты командной строки для виртуализации, можете использовать команду qemu-img, позволяющую создавать файлы образов, которые необходимы виртуальным машинам в качестве накопителей информации. Затем вы можете задействовать команду virt-install для фактического создания виртуальных машин.

Создав виртуальные машины, вы сможете управлять ими с помощью команды vi rsh. Задействуя vi rsh, можно запускать, останавливать, ставить на паузу и предпринимать иные действия в отношении ваших виртуальных машин.

Приложение А
Редакторы vi и vim

В этом приложении:

О использование редактора vi;

О запуск/выход из редактора vi;

О навигация в vi;

О изменение и удаление текста;

О использование команд ex;

О работа в визуальном режиме.

Хотя простые в использовании графические текстовые редакторы (например, gedit и kedit) без труда доступны в Linux, большинство продвинутых пользователей по-прежнему предпочитает vi, nano или Emacs для редактирования текстовых файлов. Эти текстовые редакторы работают из любого интерпретатора команд (GUI-интерфейс не требуется) и обладают другими преимуществами — например, вам не придется отрываться от клавиатуры, а также они обеспечивают интеграцию с полезными утилитами. Кроме того, в отличие о GUI-редакторов, текстовые редакторы могут работать при отсутствии установленного графического интерфейса (как и многие Linux-серверы и специализированные устройства).

В этом приложении внимание сосредоточено на функциональных возможностях редактора vi, который поможет как в базовом редактировании, так и в продвинутых манипуляциях с текстом. Я решил рассмотреть vi, а не Emacs потому, что vi более универсален и компактен, а также в силу того, что использование клавиатурных сокращений в vi требует меньшей сноровки.

Во многих Linux-системах вместо старого редактора vi по умолчанию используется редактор vim (Vi IMproved), поэтому описания, приведенные в этом разделе, расширены, чтобы охватить и vim. К функциональным возможностям vim, отсутствующим в vi, относятся множественные уровни отмены, подсветка синтаксиса и интерактивная справка.

ПРИМЕЧАНИЕ----------

Если вам никогда раньше не доводилось применять текстовые редакторы vi или vim, то воспользуйтесь обучающей программой, которая входит в пакет vim-enhanced. Введите команду vimtutor и следуйте инструкциям для ознакомления с ключевыми функциональными возможностями редакторов vi и vim.

Запуск и выход из редактора vi

Если вы захотите поэкспериментировать с использованием vi, сделайте копию текстового файла, на котором будете практиковаться. Например, введите следующее:

$ cp /etc/services /tmp

Затем откройте этот файл с помощью команды vi, как показано далее:

$ vi /tmp/services

Чтобы извлечь пользу из всех улучшений vim, убедитесь, что у вас установлен пакет vim-athena (который по умолчанию инсталлируется в Ubuntu). Во многих системах vi является символьной ссылкой на команду vim. В Ubuntu обе эти команды запускают vim.

Вот другие способы **запустить vi:**

$ vi +25 /tmp/services	*Начать со строки номер 25*
$ vi + /tmp/services	*Начать редактирование файла с последней строки*
$ vi +/tty /tmp/services	*Начать с первой строки, со слова tty*
$ vi -r /tmp/services	*Восстановить файл после краха сеанса редактирования*
$ view /tmp/services	*Редактировать файл в режиме «только для чтения»*

Закончив работать в рамках сеанса vi, вы можете прибегнуть к одному из нескольких способов, чтобы сохранить данные и выйти. Итак, чтобы:

О сохранить файл до того, как вы будете готовы выйти, введите : w;

О выйти и сохранить изменения, наберите либо ZZ, либо :wq;

О выйти без сохранения изменений, введите: q!.

Если вы обнаружите, что не получается осуществить запись в файл, который редактируете, то, возможно, он открыт в режиме «только для чтения». Если причина именно в этом, можете попытаться форсировать запись, набрав : w!, либо **сохранить содержимое файла под другим именем.** Например, введите следующее для сохранения содержимого текущего файла в файл с именем myfi 1 e. txt:

:w /tmp/myfile.txt

Редактор vi также позволяет **выстроить в ряд сразу несколько файлов для редактирования.** Например, наберите следующее:

```
$ cd /tmp
$ touch a.txt b.txt c.txt
$ vi a.txt b.txt c.txt
```

В этом примере в редакторе vi первым откроется файл a. txt. Вы можете **перейти к следующему файлу,** введя **:** п. Вам, возможно, потребуется **сохранить изменения перед переходом к следующему файлу** (:w) либо **сохранить изменения при переходе к следующему файлу** (:wn). Чтобы **отменить изменения при переходе к следующему файлу,** введите : п!.

Вы, вероятно, обнаружите, что открыт!) несколько файлов легче, если разделить экран vi. Находясь в редакторе vi, в котором открыт файл, вы можете **разделить экран несколько раз** горизонтально либо вертикально:

```
:split /etc/motd.tail
:vsplit /etc/motd.tail
```

Используйте клавишу **Tab** для завершения ввода путей к файлам так же, как вы делали бы в интерпретаторе команд bash. Для **навигации между разделенными окнами** нажимайте комбинацию **Ctrl+W,** а затем — клавишу **W.** Чтобы закрыть текущее окно, воспользуйтесь обычной командой выхода из vi (:q).

Навигация в vi

Первое, к чему следует привыкнуть в случае с vi, — что нельзя просто начать печатать. vi поддерживает множество режимов для выполнения различных наборов задач. Вы можете начать сеанс vi в режиме **Normal** (Обычный), в котором vi будет ожидать ввода вами команды, чтобы приступить к работе.

Находясь в режиме **Normal** (Обычный), вы сможете осуществлять навигацию по содержимому файла для поиска в нем нужного вам места. Для ввода или изменения текста вам потребуется перейти в режим **Insert** (Вставка) или **Replace** (Замена).

При условии, что в vi будет открыт файл, содержащий несколько страниц текста, вы сможете использовать клавиши и их сочетания из табл. АЛ для **навигации по содержимому файла в режиме Normal (Обычный).**

Таблица А.1. Горячие клавиши для навигации

Page Down или Ctrl+F — переместиться на одну страницу вниз	Page Up или Ctrl+B — переместиться на одну страницу вверх
Ctrl+D — переместиться на полстраницы вниз	Ctrl+U — переместиться на полстраницы вверх
Shift+G — перейти к последней строке файла	:1 — перейти к первой строке файла (используйте клавишу с любой цифрой для перехода к соответствующей строке)
Shift+H — переместить курсор в верхнюю часть экрана	Shift+L — переместить курсор в нижнюю часть экрана
Shift+M — переместить курсор в среднюю часть экрана	Ctrl+L — обновить изображение на экране (если наблюдаются искажения)
Enter — переместить курсор в начало следующей строки	— переместить курсор в начало предыдущей строки
Home или $ — переместить курсор в конец строки	End, или ^, или 0 — переместить курсор в начало строки
(— переместить курсор в начало предыдущего предложения	) — переместить курсор в начало следующего предложения
{ — переместить курсор в начало предыдущего абзаца	} — переместить курсор в начало следующего абзаца

W – переместить курсор к следующему слову (пробел, новая строка или знак препинания)	Shift+W – переместить курсор к следующему слову (пробел или новая строка)
B – переместить курсор к предыдущему слову (пробел, новая строка или знак препинания)	Shift+B – переместить курсор к предыдущему слову (пробел или новая строка)
E – переместить курсор в конец следующего слова (пробел, новая строка или знак препинания)	Shift+E – переместить курсор в конец следующего слова (пробел или новая строка)
<- или Backspace – переместить курсор на одну букву влево	-> или 1 – переместить курсор на одну букву вправо
K или t – переместить курсор на одну строку вверх	j или *1* - переместить курсор на одну строку вниз
/строка – найти следующий случай употребления строки	?строка – найти предыдущий случай употребления строки
N – новый поиск аналогичной строки (в прямом направлении)	Shift+N – новый поиск аналогичной строки (в обратном направлении)

Изменение и удаление текста в vi

Чтобы приступить к изменению или добавлению текста в vi, вам потребуется перейти в режим **Insert** (Вставка) или **Replace** (Замена), как показано в табл. А.2. Когда вы перейдете в один из этих режимов, набираемые вами символы будут отображаться в текстовом документе (а не интерпретироваться как команды).

Нажмите клавишу **Esc** для возврата в режим **Normal** (Обычный), после того как вы закончите осуществлять вставку или замену текста.

Таблица А.2. Клавиши для изменения текста

I – набираемый текст будет отображаться перед текущим символом	Shift+I – набираемый текст будет отображаться в начале текущей строки
A – набираемый текст будет отображаться после текущего символа	Shift+A – набираемый текст будет отображаться в конце текущей строки
O – начать новую строку под текущей строкой, чтобы приступить к набору текста	Shift+O – начать новую строку над текущей строкой, чтобы приступить к набору текста
S – удалить текущий символ и заменить новым текстом	Shift+S – удалить текущую строку и набрать новый текст
C? – нажмите вместо ? клавишу L, W, $ или C, чтобы изменить соответственно текущую букву, слово, конец строки или строку	Shift+C – удалить все, начиная с позиции курсора и до конца строки, и набрать новый текст
R – заменить текущий символ следующим набранным вами символом	Shift+R – заменять все по мере набора вами текста, начиная с текущего символа и далее

В приведенном ниже списке указаны клавиши, которые вы можете использовать для удаления или вставки текста:

- X — удалить текст под курсором;
- C? — нажмите вместо ? клавишу L, W, $ или D, чтобы вырезать соответственно текущую букву, слово, конец строки, начиная с позиции курсора, или всю строку;
- Y? — нажмите вместо ? клавишу L, W или $, чтобы скопировать (поместить в буфер) соответственно текущую букву, слово или конец строки, начиная с позиции курсора;
- P — вставить вырезанный или помещенный в буфер текст после позиции курсора;
- Shift+X — удалить текст слева от позиции курсора;
- Shift+D — вырезать текст, начиная от позиции курсора и до конца строки;
- Shift+Y — поместить в буфер текущую строку;
- Shift+P — вставить вырезанный или помещенный в буфер текст перед позицией курсора.

Вспомогательные команды

В приведенном далее списке представлено несколько клавиш для ввода вспомогательных, но важных команд, которые вам следует знать.

- U — нажмите эту клавишу, чтобы отменить предыдущее изменение. Если нажать ее несколько раз подряд, это приведет к отмене соответствующего количества изменений.
- . — нажатие клавиши с точкой обеспечивает повторное выполнение предыдущей команды. Таким образом, если, к примеру, вы удалите строку, замените слово, измените четыре буквы и т. д., то аналогичная команда будет выполнена снова, начиная с текущей позиции курсора (повторный переход в режим ввода приведет к ее сбросу).
- Shift+J — нажатие этой комбинации объединяет текущую строку со следующей.
- Esc — если вы еще сами не догадались, то знайте, что ее нажатие позволяет вернуться из режима ввода в командный режим. Эта клавиша является одной из тех, которыми вы будете пользоваться чаще всего.

Модификация команд с использованием чисел

Почти каждую из команд, описанных до сих пор, можно модифицировать с использованием чисел. Другими словами, вместо удаления слова, замены буквы или изменения строки вы можете удалить шесть слов, заменить двенадцать букв и изменить девять строк. В следующем списке приведены некоторые примеры;

- 7CW — удалить следующие семь слов и заменить их набранным вами текстом;
- 5 Shift+D — вырезать следующие пять строк (включая текущую строку);

- 3P — трижды вставить удаленный до этого текст после текущей позиции курсора;
- 9DB — вырезать девять слов перед текущей позицией курсора;
- 103 — переместить курсор на десять строк вниз;
- Y2) — скопировать (поместить в буфер) текст, начиная с позиции курсора и до конца следующих двух предложений;
- 5 Ctrl+F — переместиться на пять страниц вперед;
- 6 Shift+J — объединить следующие шесть строк.

Как видно из этих примеров, большинство команд, вводимых в vi посредством нажатия соответствующих клавиш, для изменения текста, его удаления или навигации по содержимому файла, можно модифицировать с использованием чисел.

Команды ex

Редактор vi изначально был создан на основе редактора под названием ex. Некоторые команды vi, которые вы видели до сих пор, начинаются с точки с запятой и называются командами *ex*. Для ввода команд ex начните работу в режиме Normal (Обычный) и наберите двоеточие (:). В результате этого вы перейдете в режим командной строки.

Находясь в режиме командной строки, вы можете использовать клавишу Tab для завершения ввода команд или имен файлов, а также клавиши со стрелками для навигации по истории команд, как вы делали бы это в интерпретаторе команд bash. Нажав клавишу Enter в конце своей команды, вы вернетесь в режим Normal (Обычный).

В табл. Л.3 показаны примеры команд ex.

Таблица А.3. Примеры ex-команд

`: Ibash` — перейти в интерпретатор команд bash. Закончив работу, введите `exit`, чтобы вернуться в vi	`:s/RH/Red Hat` — заменить на Red Hat первый случай употребления RH в текущей строке
`:!!` — повторно выполнить предыдущую команду	`: Idate` — выполнить `date` (или любую команду на ваш выбор). Для возврата нажмите клавишу Enter
`:20` — перейти к строке номер 20	`:s/RH/Red Hat/g` — заменить на Red Hat все случаи употребления RH в текущей строке
`:5,10w abc.txt` — записать строки с 5-й по 10-ю в файл `abc.txt`	`:%s/RH/Red Hat/g` — заменить на Red Hat все случаи употребления RH во всем файле
`:e abc.txt` — выйти из текущего файла и начать редактирование файла `abc.txt`	`:g/Red Hat/p` — показать каждую строку в файле, содержащую строку Red Hat
`:.r def.txt` — добавить содержимое файла `def.txt` в файл под текущей строкой	`:g/gaim/s//pidgin/gp` — найти все экземпляры gaim и заменить их на pidgin

В приглашении ex вы также сможете увидеть и изменить настройки, связанные с вашим сеансом vi, с помощью команды set. Вот несколько примеров:

- :set all — показать все настройки;
- : set — отобразить только настройки, которые изменились и отличаются от тех, которыми были по умолчанию;
- : set number — показывать номера строк слева от каждой строки (для сброса используйте set nonu);
- :set ai — задать автоматический отступ, чтобы новая строка начиналась с такого же отступа, что и предыдущая;
- :set ic — игнорировать регистр, чтобы при выявлении соответствий во время поиска в тексте не учитывался их регистр;
- :set list — отображать $ в конце строк и Ч там, где используется табуляция;
- : set wm — сделать так, чтобы редактор vi добавлял разрывы строки между словами рядом с ее концом.

Работа в визуальном режиме

Редактор vim обеспечивает более интуитивно понятный механизм выделения текста, называемый *визуальным режимом.* Чтобы перейти в этот режим, установите курсор рядом с первым символом текста, который хотите выделить, и нажмите клавишу **V** или комбинацию **Shift+V.** Нажав **V,** вы сможете осуществить посимвольное выделение, а путем нажатия **Shift+V** — построчное. Вы увидите, что находитесь в визуальном режиме, поскольку в нижней части экрана будет отображаться следующий текст;

```
-- VISUAL --
```

На данном этапе можно использовать любые клавиши управления курсором (клавиши со стрелками, **Page Down, End** и т. д.) для его перемещения в конец текста, который необходимо выделить. По мере движения страницы и курсора вы будете видеть, как происходит выделение текста.

Когда весь нужный текст выделен, вы можете прибегнуть к клавишам для осуществления действий в отношении его. Например, нажатие D приводит к удалению текста, C позволяет изменить выделенный текст, : w /tmp/test. txt сохраняет выделенный текст в файл и т. д.

Приложение Б

Специальные символы и переменные интерпретатора команд

В этом приложении:

О использование специальных символов интерпретатора команд:

О использование переменных интерпретатора команд.

Интерпретатором команд по умолчанию в Ubuntu является bash. В гл. 3 вы нашли сведения, призванные помочь вам в освоении этого интерпретатора команд. В данном приложении содержится справочная информация о многочисленных символах и переменных, имеющих специальное значение для интерпретатора команд bash. Многие из этих элементов приведены в табл. Б.1 и Б.2.

Специальные символы интерпретатора команд

Вы можете задействовать специальные символы из интерпретатора команд, чтобы выявлять соответствия между несколькими файлами, нажимать меньше клавиш во время работы или выполнять особые операции. В табл. Б.1 приведены некоторые символы интерпретатора команд, которые могут оказаться полезными для вас.

Таблица Б.1. Специальные символы интерпретатора команд

Символ	Описание
*	Найти любую соответствующую строку символов
?	Найти любой соответствующий символ
л\...	Удалить специальное значение символов в кавычках. Переменные не расширяются
« \\ ...	То же самое, что и одинарные кавычки, за исключением того, что символы управляющих последовательностей ($, ' и \) сохраняют свое специальное значение. Переменные расширяются

Продолжение &

Таблица Б.1 *(продолжение)*

Символ	Описание
\	Символ управляющей последовательности для удаления специального значения указанного далее символа
~	Ссылка на каталог $HOME
~+	Значение переменной интерпретатора команд PWD (рабочий каталог)
~-	Ссылка на предыдущий рабочий каталог
.	Ссылка на текущий рабочий каталог
..	Ссылка на каталог над текущим каталогом. Может использоваться повторно для ссылки на несколько расположенных выше каталогов
$параметр	Используется для расширения параметров переменных интерпретатора команд
команда 1 'команда 2' или команда 1 $(команда 2)	команда 2 выполняется первой. Затем вывод команды 2 используется как ввод для команды 1
команда 1 > файл	Перенаправляет стандартный вывод команды в файл
команда 1 < файл	Перенаправляет стандартный ввод из файла к команде
команда 1 >> файл	Добавляет стандартный вывод команды в файл, не удаляя при этом его текущее содержимое
команда 1 | команда 2	Передает по каналу стандартный вывод одной команды в качестве ввода для следующей команды
команда &	Запускает выполнение команды в фоновом режиме
команда 1 && команда 2	Запускает выполнение первой команды; затем, если возвращается нулевой статус выхода (успех), запускает выполнение второй команды
команда 1 || команда 2	Запускает выполнение первой команды; затем, если возвращается ненулевой статус выхода (неудача), запускает выполнение второй команды
команда 1 ; команда 2	Запускает выполнение первой команды, а когда оно завершается, запускает выполнение второй команды

Переменные интерпретатора команд

Чтобы обозначить строку символов как переменную, необходимо указать перед ней $ (например, `$HOME`). Переменные среды интерпретатора команд могут содержать информацию, используемую самим интерпретатором команд, а также командами, выполнение которых вы запускаете из него.

Многие команды проверяют, заданы ли значения для определенных переменных. Не все переменные будут заполненными по умолчанию. Вы можете изменять значения некоторых из них (например, переменной `SPRINTER`, указывающей принтер по умолчанию, или `$PS1`, которая определяет приглашение командной строки).

Прочие переменные управляются интерпретатором команд (например, S0LDPWD). В табл. Б.2 приведен список полезных переменных интерпретатора команд.

Таблица Б.2. Переменные интерпретатора команд

Переменная интерпретатора команд	Описание
BASH	Показывает имя пути к команде bash (/bin/bash)
BASH_COMMAND	Команда, выполняемая в данный момент
BASH_VERSION	Номер версии команды bash
COLORS	Путь к конфигурационному файлу, касающемуся цветов, отображаемых командой ls
COLUMNS	Ширина строки терминала (в символах)
DISPLAY	Идентифицирует экран X для отображения команд, выполнение которых будет запускаться из текущего интерпретатора команд (например, :0.0)
EUID	Эффективный идентификатор пользователя. Основывается на записи в /etc/passwd для вошедшего в систему пользователя
FCEDIT	Определяет текстовый редактор, который будет использоваться командой fc для редактирования команд history. Команда vi используется по умолчанию
GROUPS	Показывает группы (согласно их идентификаторам), членом которых является текущий пользователь
HISTCMD	Показывает номер текущей команды в файле истории
HISTFILE	Показывает местоположение вашего файла истории (обычно он располагается в $HOME/.bash_history)
HISTFILESIZE	Общее количество записей в файле истории, которое будет сохраняться (по умолчанию равно 1000). По достижении этого количества данные о более старых командах будут отбрасываться
HISTCMD	Номер текущей команды в списке истории
HOME	Местоположение домашнего каталога текущего пользователя. Ввод команды cd без параметров возвращает интерпретатор команд в домашний каталог
HOSTNAME	Имя хоста текущего компьютера
HOSTTYPE	Содержит данные о компьютерной архитектуре, на которой работает система Linux (i386, i486, i586, i686, x86_64, ppc или ppc64)
LESSOPEN	Имеет значение в виде имени команды, которая конвертирует содержимое, отличное от обычного текста (изображения, пакеты RPM, сжатые файлы и т. д.), чтобы его можно было передать по каналу через команду less
LINES	Определяет количество строк в текущем терминале
LOGNAME	Содержит имя текущего пользователя

Продолжение &

Таблица Б.2 *(продолжение)*

Переменная интерпретатора команд	Описание
LS_COLORS	Соотносит цвета с расширениями файлов для указания, какие цвета команда ls будет отображать, когда столкнется с файлами соответствующих типов
MACHTYPE	Показывает информацию об архитектуре вычислительной машины, компании и операционной системе (например, x86_64-pc-linux-gnu)
MAIL	Указывает местоположение вашего файла почтового ящика (обычно это имя пользователя в каталоге /var/spool/mail)
MAILCHECK	Обеспечивает проверку на предмет поступления новой почты через интервалы, указанные в секундах (по умолчанию они равны 60 секундам)
OLD PWD	Каталог, который был рабочим до перехода в текущий рабочий каталог
OSTYPE	Имя, идентифицирующее текущую операционную систему (например, linux или linux-gnu)
PATH	Разделенный двоеточиями список каталогов, используемый для определения местоположения вводимых вами команд (/bin, /usr/bin и $HOME/bin обычно содержатся в PATH). При осуществлении поиска каталоги исследуются слева направо
PPID	Идентификатор процесса команды, запустившей текущий интерпретатор команд
PRINTER	Определяет принтер по умолчанию, который будет использоваться командами печати, например lpr и lpq
PROMPT_COMMAND	Имеет значение в виде имени команды, выполнение которой должно будет запускаться каждый раз перед отображением приглашения вашего интерпретатора команд (например, PROMPT_COMMAND=ls обеспечивает показ команд в текущем каталоге перед отображением приглашения)
PS1	Определяет приглашение интерпретатора команд. Приглашение может включать такие элементы, как дата, время, имя пользователя, имя хоста и др. Дополнительные приглашения можно определить посредством PS2, PS3 и т. д.
PWD	Каталог, назначенный в качестве вашего текущего каталога
RANDOM	При осуществлении доступа к этой переменной генерируется случайное число в диапазоне от 0 до 32 767
SECONDS	Количество секунд с момента запуска интерпретатора команд
SHELL	Содержит полный путь к текущему интерпретатору команд
SHELLOPTS	Показывает активизированные параметры интерпретатора команд (в случае с которыми задано значение on)

Приложение B

Получение информации из /proc

В этом приложении:

О просмотр информации из /proc;

О изменение информационных переменных /proc.

Файловая система /proc, первоначально предназначавшаяся для хранения информации, используемой запущенными процессами, в конечном итоге стала основным местом размещения всевозможной информации, которая используется ядром Linux. Несмотря на появление /sys для обеспечения более упорядоченного фреймворка, позволяющего отображать информацию о ядре, многие Linux-утилиты по-прежнему задействуются для сбора и представления данных о запущенной системе из /proc.

Если вы предпочитаете действовать без посредников, можете не прибегать к утилитам для чтения файлов /proc, осуществляя чтение этих файлов (а иногда даже и запись в них) напрямую. Заглянув в /proc, вы сможете получить информацию о состоянии процессов, аппаратных устройствах, подсистемах ядра и других атрибутах Linux.

Просмотр информации из /proc

Для просмотра информации, содержащейся в файлах из каталога /proc, можно использовать простую команду cat. В /proc имеется отдельный каталог для каждого запущенного процесса (с именем, соответствующим идентификатору связанного с ним процесса), содержащий информацию об этом процессе. Есть также файлы /proc, которые содержат другие всевозможные данные, например о центральном процессоре компьютера, уровне использования памяти, версиях программного обеспечения, дисковых разделах и т. д.

В приведенных далее примерах показаны некоторые сведения, которые вы можете получить из каталога /proc в своей системе Linux:

```
$ cat /proc/cmdline        Показывает параметры, указанные в приглашении
                           к загрузке
BOOT_IMAGE=/vmlinuz-3.2,0-37-generic root=/dev/mapper/ubuntutb-root ro
$ cat /proc/cpuinfo        Показывает информацию о вашем центральном
                           процессоре
```

```
processor    : 0
vendorj d    : Genuinelntel
cpu family   : 6
model        : 23
model name   : Pentium(R) Dual-Core CPU   E6300 @ 2.80GHz
steppi ng    : 10
microcode    : 0xa07
cpu MHz      : 1603.000
cache size   : 2048 KB
  ...
```

Как показано в приведенном выше примере, отображаемая тактовая частота центрального процессора, выраженная в мегагерцах, может оказаться намного ниже реальной, если запущен регулятор тактовой частоты центрального процессора, например cpuspeed. В следующем примере отображаются символьные и блочные устройства:

```
$ cat /proc/devices        Показывает существующие символьные и блочные устройства
Character devices:
  1 mem
  4  /dev/vc/O
  4 tty
  4 ttyS
  5 /dev/tty
    ...
Block devices:
  1 ramdisk
259 blkext
  7 loop
  8 sd
$ cat /proc/diskstats      Отобразить диски, разделы и статистику
    1    0 ramO 0 0 0 0 0 0 0 0 0 0 0
    1    1 rami 0 0 0 0 0 0 0 0 0 0 0
   ...
   8        0 sda 21883 11293 874900 194544 7813- 9966 932416 1073344
            0 120276 1267864
   8        1 sdal 278 25 2310 11636 8 0 16 476 0 11960 12112
   8        2 sda2 2 0 12 144 0 0 0 0 0 144 144
   ...
   7        0 loopO 0 0 0 0 0 0 0 0 0 0 0
```

В данном выводе di skstats вы можете видеть такие устройства, как диски в оперативной памяти (ramO, rami и т. д.), а также петлевые устройства (loopO, loopl и т. д.). Что касается дисковых разделов, то в этом примере показана статистика относительно всего жесткого диска (sda) и каждого его раздела (sdal, sda2 и т. д.).

В полях 11 касательно всего жесткого диска показано (слева направо) количество:

О операций чтения (общее количество);

О объединенных операций чтения;

О прочитанных секторов;

О миллисекунд, затраченное на все операции чтения;

О завершенных операций записи;

О объединенных операций записи;

О секторов, куда была осуществлена запись;

О миллисекунд, затраченное на операции записи;

О запросов на ввод/вывод, выполняемых в настоящее время;

О миллисекунд, затраченное на осуществление ввода/вывода;

О миллисекунд, затраченное на осуществление ввода/вывода (среднее количество).

В полях касательно определенного раздела показано (слева направо): количество проведенных операций чтения, прочитанных секторов, проведенных операций записи и количество секторов, куда была осуществлена запись.

```
$ cat /proc/filesystems          Показать типы файловых систем в текущем ядре
nodev sysfs                      nodev означает, что тип не используется ни на одном
                                 из устройств
nodev rootfs
   ...
        ext4                     На смонтированном блочном устройстве
                                 используется ext4
        iso9660                  На смонтированном блочном устройстве
                                 используется iso9660
$ cat /proc/i interrupts         Просмотр распределения каналов IRQ
           CPU0       CPU1       CPU2       CPU3
  0:        126          0          0          0   IO-APIC-edge      timer
  1:       3815       2223       4786       2596   IO-APIC-edge      18042
  8:          1          0          0          0   IO-APIC-edge      rtcO
  9:      15351      14722       5296       4894   I0-APIC-fasteoi   acpi
 12:     268497     419436     322826     145849   IO-APIC-edge      18042
   ...
$ cat /proc/iomem                Показать адреса физической памяти
OOOOOOOO-OOOOffff reserved
00010000-0009efff : System RAH
0009f000-0009ffff : reserved
000a0000-000bffff : PCI Bus 0000:00
000c0000-000c7fff : Video ROM
OOOdOOOO-OOOdffff PCI Bus 0000:00
000e0000-000fffff : reserved
  OOOfOOOO-OOOffffSystem ROM
00100000-bdd9ffff : System RAH
  01000000-0166a78b: Kernel code
...
$ cat /proc/ioports              Показать адреса виртуальной памяти
0000-001f : dmal
0020-0021 : picl
0040-0043 : timerO
0050-0053 : timerl
0060-0060 : keyboard
```

```
0064-0064 : keyboard
0070-0071 : rtcO
0080-008f : dma page reg
00a0-00al : pi c2
00c0-00df : dma 2
00f0-00ff : fpu
...
```

$ cat /proc/loadavg
Показывает среднюю нагрузку за 1, 5 и 15 минут, запущенные процессы/общее количество и наибольший идентификатор процесса

```
1.77 0.56 0.19 2/247 1869
```

$ cat /proc/meminfo
Показывает доступную оперативную память и область подкачки

```
MemTotal:      4014504 kB
MemFree:       3202120 kB
Buffers:         90048 kB
Cached:         492060 kB
SwapCached:          0 kB
Active:         377604 kB
Inactive:       274020 kB
Active(anon):    70136 kB
Inactive(anon):  25448 kB
Acti ve(file):  307468 kB
...
```

$ cat /proc/misc
Показывает имена/младшие номера устройств. зарегистрированных в случае со старшим устройством misc (10)

```
229 fuse

236 device-mapper
173 agpgart
...
```

$ cat /proc/modules
Показывает загруженные модули, объем памяти. загруженные экземпляры, состояние загрузки зависимостей и память ядра

```
Ьпер 18281 2 - Live 0x0000000000000000
rfcomm 47604 0 - Live 0x0000000000000000
bluetooth 180153 10 bnep,rfcomm. Live 0x0000000000000000
parport_pc 32866 0 - Live 0x0000000000000000
ppdev 17113 0 - Live 0x0000000000000000
ext2 73795 1 - Live 0x0000000000000000
snd_hda_codec_realtek 224173 1 - Live 0x0000000000000000
sndjidajntel 33773 3 - Live 0x0000000000000000
...
```

$ cat /proc/mounts
Показать информацию о смонтированных локальных/удаленных файловых системах

```
rootfs / rootfs rw 0 0
sysfs /sys sysfs rw.nosuid,nodev,noexec.relatime 0 0
proc /proc proc rw.nosuid,nodev,noexec.relatime 0 0
udev /dev devtmpfs rw,relatime,size=19981k,nr_inodes=499543,mode=755 0 0
/dev/sdal /boot ext2 rw,relatime,errors=continue 0 0
...
```

$ cat /proc/partitions *Показать смонтированные локальные дисковые разделы*

```
major minor #b!ocks name
   8        0   156290904 sda
   8        1      248832 sdal
   8        2           1 sda2
   8        5   156039168 sda5
 252        0    93495296 dm-0
 252        1     4157440 dm-1
  11        0     1048575 srO
```

$ cat /proc/mdstat *Если используется программный RAID-массив. то показать информацию о его состоянии*

```
Personalities : [raidl]
read_ahead 1024 sectors
Event: 1
mdO : active raidl sdbl[l] sda2[0]
      69738048 blocks [2/2] [UU]
unused devices: <none>
```

Файл /proc/mdstat содержит подробную информацию о состоянии ваших устройств в программном RAID-массиве, если они у вас установлены. В этом примере mdO является RAID1-массивом (с функцией зеркалирования), включающим разделы /dev/sdbl и /dev/sdal. В следующей строке в случае с каждым рабочим членом RAID-массива указан символ U. Если бы вы лишились диска, то вывод выглядел бы как [UJ.

$ cat /proc/stat *Показывает статистику касаемо ядра с момента загрузки системы*

```
cpu 58394 4008 18635 50620848 9216 0 151 0 0 0
Cpu0 29693 2416 9551 25311091 4459 0 87 0 0 0
cpul 28701 1592 9084 25309756 4757 0 64 0 0 0
intr 7720709 889 11 0 0      ...
ctxt 11180402
btime 1361187396
processes 5107
procs_running 1
procs_b1ocked 0
```

Файл /proc/stat включает статистику касаемо активности центральных процессоров и процессов. В строке cpu показаны суммарные данные относительно всех центральных процессоров, в то время как в отдельной строке для каждого центрального процессора (cpuO, cpul и т. д.), который имеется в компьютере, отображается касающаяся его статистика. Есть семь полей (слева направо) с информацией о центральном процессоре, такой как количество нормальных процессов, выполняющихся в пользовательском режиме; количество процессов с измененными значениями nice, выполняющихся в пользовательском режиме; количество процессов в режиме ядра; количество простаивающих процессов, количество процессов, ожидающих (завершения выполнения запросов) ввода/вывода; количество обслуживаемых прерываний (IRQ) и количество обслуживаемых программных IRQ-прерываний.

```
$ cat /proc/swaps                    Показать информацию о пространстве подкачки
Filename        Type        Size    Used    Priority
/dev/sda2       partition 4157436 0         -1
$ cat /proc/uptime                   Количество секунд с момента загрузки системы/
                                     общее количество секунд простоя
2300251.03 2261855.31
$ cat /proc/version                  Показать номер версии ядра и соответствующего
                                     компилятора
Linux version 3.2.0-37-generic (buildd@allspice) (gcc version 4.6.3
(Ubuntu/Linaro 4.6.3-lubuntu5) ) #58-Ubuntu SMP
Thu Jan 24 15:28:10 UTC 2013
```

Изменение информации из /proc

В отдельных версиях Linux некоторые данные в каталоге /proc/sys в действительности можно изменять на лету. В Linux-системах, которые позволяют делать это, вы могли бы просто добавить значение в любой файл, который хотите модифицировать, после чего изменение незамедлительно вступило бы в силу.

Предпочтительный метод изменения информации из /proc/sys на лету заключается в использовании команды sysctl. Для изменения соответствующих настроек на более постоянной основе вам потребуется добавить записи в файл /etc/sysctl .conf. Вот примеры использования команды sysctl:

```
$ sudo sysctl -A | less              Отобразить все параметры ядра
$ sudo sysctl -w net.ipv4.ip_forward=l Активизировать пересылку пакетов IPV4
```

Дополнительную информацию можно найти в гл. 10, а также на MAN-страницах, посвященных sysctl и sysctl .conf.