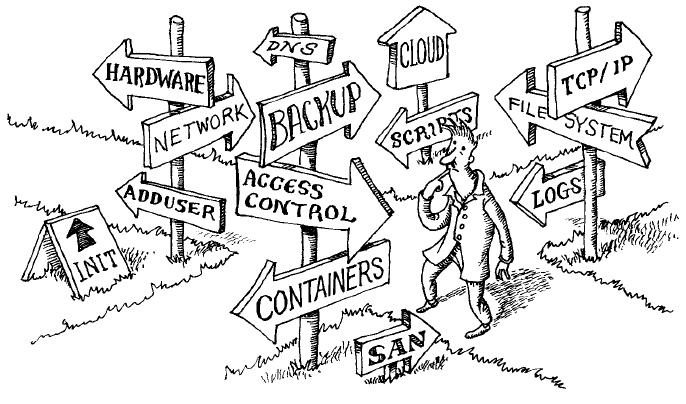


***С чего начать***

Мы разработали эту книгу, чтобы занять определенную нишу в обширной экосистеме справочных страниц (man-pages), блогов, журналов, книг и других справочных материалов, которые отвечают потребностям системных администраторов UNIX и Linux.

Во-первых, это справочник-путеводитель. В нем рассматриваются основные системы администрирования, определяются различные элементы каждой из них и объясняется, как они работают вместе. Во многих случаях, когда вам необходимо выбирать среди различных реализаций какой-либо идеи, мы описываем преимущества и недостатки наиболее популярных вариантов.

Во-вторых, это краткий справочник, в котором обобщается то, что вам нужно знать для выполнения типичных задач на различных системах UNIX и Linux. Например, команда **ps**, которая показывает состояние запущенных процессов, поддерживает более 80 параметров командной строки в системах Linux. Но всего лишь несколько комбинаций параметров удовлетворяют большинство потребностей системного администратора; мы обобщаем их в разделе *4.3 PS: Мониторинг процессов*.

Наконец, эта книга посвящена администрированию корпоративных серверов и сетей, то есть серьезному, профессиональному системному администрированию. Легко настроить один хост, труднее заставить распределенную, облачную платформу работать слаженно и устойчиво при распространенности вирусов, сетевом разделении и целевых атаках. Мы описываем методы и практические правила, которые помогут вам восстановить системы после серьёзных сбоев, и мы поможем вам выбрать решения, которые будут легко масштабироваться по мере роста вашей бизнес-империи в размере, сложности и неоднородности.

Мы не утверждаем что делаем всё это абсолютно объективно, но мы думаем, что изложили наши предубеждения довольно ясно во всей книге. Одна из интересных вещей в системном администрировании заключается в том, что здравомыслящие люди могут иметь совершенно разные мнения о том, что является наиболее подходящим решением. Мы предлагаем Вам наши субъективные мнения в качестве исходных данных. Решайте сами, что брать на вооружение и сколько наших комментариев соответствует вашей среде.

**1.1 Основные обязанности системного администратора**

В приведенных ниже разделах кратко излагаются некоторые основные задачи, которые должны выполнять администраторы. Вовсе необязательно эти задачи должны выполняться одним человеком. Во многих организациях эта работа распределяется между членами команды администрирования. Однако, по крайней мере хотя бы один человек должен понимать все компоненты и убедиться, что каждая задача выполнена корректно.

**Контроль доступа**

*Смотри Главы 8, 17 и 23 для получения большей информации о подготовке к работе аккаунтов пользователей.*

Системный администратор создает аккаунты для новых пользователей, удаляет аккаунты более неактивных пользователей и решает все возникающие проблемы, связанные с учетными записями (например, забытые пароли и потерянные пары ключей). Процесс фактического добавления и удаления учетных записей обычно автоматизирован системой управления конфигурациями или централизованной службой каталогов.

**Добавление оборудования**

Администраторы, работающие с физическим оборудованием (в отличие от облачных или систем хостинга), должны установить его и настроить для распознавания операционной системой. Работа по поддержке оборудования может варьироваться от простой задачи добавления карты сетевого интерфейса до настройки специализированного внешнего массива хранилищ данных.

**Задачи автоматизации**

*Смотри Главу 7, Скриптинг и shell, для получения большей информации о скриптинге и автоматизации.*

Использование инструментов для автоматизации повторяющихся и трудоемких задач повышает вашу эффективность, снижает вероятность возникновения человеческих ошибок и улучшает вашу способность быстро реагировать на изменяющиеся требования. Администраторы стремятся сократить объем ручного труда, необходимого для обеспечения бесперебойной работы систем. Ознакомление с языками сценариев и средствами автоматизации - это большая часть работы.

**Контроль резервных копий**

*Советы по выполнению резервного копирования см. в Главе 20.14 «Стратегии бэкапов данных.*

Резервное копирование данных и их успешное восстановление при необходимости являются важными задачами администрирования. Хотя резервное копирование занимает много времени и является скучным, частота катастроф в мире слишком высока, чтобы игнорировать эту работу. Операционные системы и некоторые отдельные пакеты программного обеспечения предоставляют хорошо зарекомендовавшие себя инструменты и методы для облегчения резервного копирования. Резервные копии должны выполняться по расписанию, и периодически проверяться, чтобы убедиться в их работоспособности.

**Установка и обновление ПО**

*См. Главу 6 Для получения информации об управлении программным обеспечением.*

Программное обеспечение должно быть выбрано, установлено и сконфигурировано, часто в различных операционных системах. По мере выпуска патчей и обновлений безопасности, они должны тестироваться, пересматриваться и включаться в локальную среду без угрозы стабильности production-систем.

*См. Главу 26 для получения информации о развертывании ПО и continuous delivery (непрерывная доставка).*

Термин "software delivery" относится к процессу выпуска обновленных версий ПО, особенно разработанного собственными силами, для последующего использования. "Continuous delivery" переводит этот процесс на следующий уровень автоматически выпуская програмное обеспечение потребителям на регулярной основе по мере его готовности. Администраторы помогают реализовать надежные процессы доставки, соответствующие требованиям предприятия.

**Мониторинг**

*См. Главу 28 для получения информации о мониторинге.*

Временное решение проблемы, как правило, быстрее, чем ее документирование и отчет, и пользователи внутри организации часто следуют по пути наименьшего сопротивления. Внешние пользователи с большей вероятностью озвучат свои жалобы открыто, чем сделают запрос в службу поддержки. Администраторы могут помочь предотвратить оба этих результата, обнаружив проблемы и исправив их до их возникновения.

Некоторые задачи мониторинга включают обеспечение быстрого и правильного ответа от веб-сервисов, сбор и анализ логов, а также отслеживание доступности ресурсов сервера, таких как дисковое пространство. Все эти задачи являются отличными возможностями для автоматизации, и множество платных и бесплатных систем мониторинга могут помочь системным администраторам в решении этих задач.

**Устранение проблем (troubleshooting)**

*См. Главу 13.12 для введения в устранение проблем в сети.*

Сетевые системы выходят из строя неожиданно и иногда очень эффектно. Работа администратора заключается в диагностировании проблем и вызове нужных специалистов по мере необходимости. Поиск источника проблемы часто является более сложной задачей, чем ее решение.

**Ведение локальной документации**

*Предложения по ведению документации см. в Главе 31.3*

Администраторы выбирают поставщиков, пишут скрипты, развертывают ПО и принимают множество других решений, которые не всегда очевидны или понятны другим. Тщательная и точная документация является благословением для членов команды, которые в противном случае должны были бы перерыть всю систему для решения проблемы в середине ночи. Любовно созданная схема сети полезнее многих абзацев текста при описании проекта.

**Неусыпный контроль безопасности**

*Дополнительные сведения о безопасности см. в главе 27.*

Администраторы являются первой линией защиты для сетевых систем. Администратор должен реализовать политику безопасности и настроить процедуры для предотвращения нарушения работы систем. Эта ответственность может включать только несколько основных проверок на несанкционированный доступ или сложную сеть ловушек и программ аудита, в зависимости от ситуации. Системные администраторы осторожны по своей природе и часто являются основными защитниками безопасности в организации.

**Настройка производительности**

*Дополнительные сведения о производительности см. в главе 29.*

Unix и Linux - универсальные операционные системы, которые хорошо подходят для почти любых мыслимых и немыслимых задач вычисления. Администраторы могут настраивать системы для оптимальной производительности в соответствии с потребностями пользователей, доступной инфраструктуры и сервисами, которые предоставляет система. Когда сервер работает плохо, задача администратора - исследовать его работу и определить области, которые нуждаются в улучшении.

**Разработка политик системы**

По юридическим причинам и в целях соблюдения требований законодательства, большинство систем нуждаются в политиках, которые регулируют допустимое использование компьютерных систем, управление и хранение данных, конфиденциальность и безопасность сетей и систем, а также другие области, представляющие интерес для регулирования. Системные администраторы часто помогают организациям разрабатывать разумные политики, которые соответствуют букве и цели закона, и все же содействуют прогрессу и производительности.

**Работа с поставщиками**

Большинство организаций полагаются на третьи стороны для предоставления различных дополнительных услуг и продуктов, связанных с их вычислительной инфраструктурой. Эти поставщики могут включать разработчиков программного обеспечения, поставщиков облачных услуг, магазины, использующие программное обеспечение как услуга (SaaS), сотрудников службы поддержки, консультантов, подрядчиков, экспертов по безопасности и поставщиков платформ или инфраструктур. Администраторам может быть поручено выбирать поставщиков, помогать в переговорах по контрактам и внедрять решения после завершения оформления документов.

**Тушение пожаров**

Хотя помощь другим людям с их различными проблемами редко включается в описание работы системного администратора, эти задачи часто занимают большую часть рабочего дня администратора. Системные администраторы бомбардируются проблемами, начиная от "Оно работало вчера, а сейчас нет! Что вы сделали?" до "Я пролил кофе на мою клавиатуру! Могу ли я налить на нее воду, чтобы почистить?"

В большинстве случаев ваш ответ на эти вопросы влияет на вашу ценность как администратора гораздо больше, чем любые фактические технические навыки, которыми вы можете обладать. Вы можете либо взвыть от несправедливости, либо насладиться фактом, что один хорошо обработанный запрос на исправление ошибки заработает вам больше очков в карму, чем пять часов полуночной отладки. Выбор за вами!

**1.2 Предполагаемый опыт**

Мы предполагаем, что у читателей есть определенный опыт работы с Linux или UNIX. В частности, вы должны иметь общее представление о том, как выглядит система с точки зрения пользователя, поскольку мы не будем на этом останавливаться. Книги, перечисленные в Главе 1.13, помогут заложить необходимый фундамент знаний.

Мы любим хорошо продуманные графические интерфейсы. К сожалению, GUI инструменты для системного администрирования UNIX и Linux остаются рудиментарными по сравнению с богатством базового программного обеспечения. В реальном мире администраторы должны хорошо себя чувствовать при использовании командной строки.

Для редактирования текста, мы настоятельно рекомендуем изучить **vi** (в настоящее время чаще встречается в его расширенной форме, **vim**), который является стандартным для всех систем. Он простой, мощный и эффективный. Освоение **vim**, возможно, является лучшим вариантом повышения производительности, доступным для администраторов. Используйте команду **vimtutor** для отличного интерактивного знакомства.

В качестве альтернативы предлагаем GNU-редактор **nano** - это простой редактор, который хорошо подходит новичкам, он также имеет экранные подсказки.

*Введение в скриптинг см. в главе 7.*

Хотя администраторы обычно не считаются разработчиками программного обеспечения, отраслевые тенденции размывают границы между ними. Способные администраторы, как правило программисты-полиглоты, не возражают против изучения нового языка программирования, когда возникает такая необходимость.

Для новых проектов мы рекомендуем Bash (он же bash, sh), Ruby или Python. Bash является командной оболочкой по умолчанию в большинстве систем UNIX и Linux. Как язык программирования он примитивен, но является хорошим инструментом в арсенале администратора. Python - это умный язык с хорошо читаемым синтаксисом, большим сообществом разработчиков и библиотеками, которые облегчают решение многих общих задач. Разработчики Ruby описывают этот язык как "радость от работы" и "красивый для созерцания". Ruby и Python во многом похожи, и мы находим их одинаково функциональными для администрирования. Выбор между ними в основном зависит от личных предпочтений.

Мы также предлагаем Вам выучить **expect**, который представляет собой скорее интерфейс для управления интерактивными программами, а не язык программирования. По сути, это эффективная интегрирующая технология, которая может заменить некоторые сложные сценарии и проста в освоении.

В Главе 7, Скриптинг и shell, обобщены наиболее важные сведения о скриптинге в Bash, Python и Ruby. В ней также рассматриваются регулярные выражения (шаблоны, задающие правила поиска текста) и некоторые идиомы оболочки, полезные для сисадминов.

**1.3 Дистрибутивы Linux**

Дистрибутив Linux включает в себя Linux kernel, который является ядром операционной системы, и пакеты, которые позволяют запускать все команды в системе. Все дистрибутивы используют одно семейство ядер, но формат, тип и количество пакетов могут существенно отличаться. Дистрибутивы также различаются по своей направленности, поддержке и популярности. По-прежнему существуют сотни независимых дистрибутивов Linux, но мы считаем, что дистрибутивы, производные от Debian и Red Hat, будут преобладать в production средах в ближайшие годы.

По большому счету, различия между дистрибутивами Linux не сильно значимы. По сути, остается загадка - почему существует так много различных дистрибутивов? Каждый из которых претендует на звание "самого легкого в установке" и "с самой большой библиотекой ПО". Трудно избежать вывода, что люди просто любят создавать новые дистрибутивы Linux.

Большинство основных дистрибутивов имеют относительно безболезненную процедуру установки, среду рабочего стола и некоторую форму управления пакетами. Их можно легко опробовать, запустив экземпляр в облаке или на локальной виртуальной машине.

*Дополнительные сведения о Docker и контейнерах см. в Главе 25, Контейнеры.*

В основном ненадежность и незащищенность операционных систем общего назначения во многом обусловлено их сложностью. Практически все популярные дистрибутивы загромождены десятками неиспользуемых пакетов; уязвимости в безопасности и страдания администраторов часто приходят вместе. В ответ на это набирают популярность относительно новые минимальные дистрибутивы. CoreOS проповедует свой подход и предпочитает запускать все ПО в контейнерах. Alpine Linux - это облегченный дистрибутив, который используется в качестве основы для многих общедоступных образов Docker. Учитывая эту тенденцию к сокращению, мы ожидаем, что размер Linux сократится в ближайшие годы.

Принимая решение в пользу того или иного дистрибутива, вы делаете поощряете стиль работы конкретного поставщика. Вместо того, чтобы рассматривать только функции установленного ПО, разумно принимать в рассчёт то, как ваша организация и этот поставщик будут работать друг с другом. Список важных вопросов, на которые нужно ответить:

- Будет ли дистрибутив существовать в ближайшие пять лет?

- Будут ли оперативно устраняться бреши в системе защиты?

- Есть ли у этого дистрибутива активное сообщество и достаточная документация?

- Если у меня возникнут проблемы, будет ли поставщик говорить со мной, и сколько это будет стоить?

В таблице 1.1 перечислены некоторые из наиболее популярных основных дистрибутивов.

**Таблица 1.1 Наиболее популярные дистрибутивы Linux общего назначения**

|  |  |  |
| --- | --- | --- |
| Дитсрибутив | Веб сайт | Комментарии |
| Arch | archlinux.org | Для тех, кто не боится командной строки |
| CentOS | centos.org | Бесплатный аналог Red Hat Enterprise |
| CoreOS | coreos.com | Контейнеры, контейнеры везде |
| Debian | debian.org | Полностью бесплатный, самый GNU’ишный дистрибутив |
| Fedora | fedoraproject.org | Тестовая площадка для Red Hat Enterprise |
| Kali | kali.org | Для тестов на вторжение |
| Linux Mint | linuxmint.com | На основе Ubuntu, удобный рабочий стол |
| openSUSE | opensuse.org | Бесплатный аналог SUSE Linux Enterprise |
| openWRT | openwrt.org | Linux для роутеров и интегрированных устройств |
| Oracle Linux | oracle.com | Поддерживаемая Oracle версия RHEL |
| RancherOS | rancher.com | 20 Мб, все в контейнерах |
| Red Hat Enterprise | redhat.com | Надежный, медленно меняется, платный |
| Slackware | slackware.com | Поседевший, долгоживущий дистрибутив |
| SUSE Linux Enterprise | suse.com | Сильно популярен в Европе, многоязычный |
| Ubuntu | ubuntu.com | "Очищенная версия" Debian |

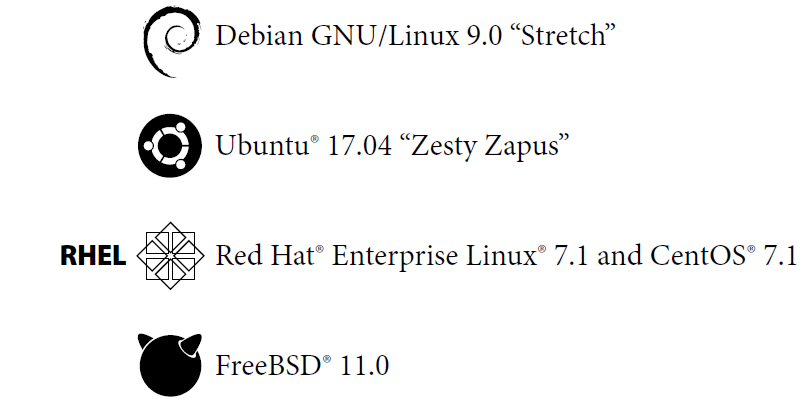
Наиболее жизнеспособные дистрибутивы не обязательно больше других используются в корпоративном секторе. Например, мы ожидаем, что Debian Linux (ОК, ОК Debian GNU/Linux!) будет оставаться жизнеспособным в течение длительного времени, несмотря на то, что Debian не является компанией, она ничего не продает и не предлагает поддержку корпоративного уровня. Debian выигрывает от приверженной группы спонсоров и от огромной популярности дистрибутива Ubuntu, который основан на нем.

Полный список дистрибутивов, включая дистрибутивы на иностранных языках, можно найти по адресу lwn.net/Distributions или distrowatch.com.

**1.4 Примеры систем, используемых в этой книге**

Мы выбрали три популярных дистрибутива Linux и один UNIX в качестве главных примеров для этой книги: Debian GNU/Linux, Ubuntu Linux, Red Hat Enterprise Linux (и ее двойник CentOS) и FreeBSD. Эти системы являются репрезентативными для всего рынка и в совокупности составляют значительную часть установленных систем, используемых сегодня в крупных организациях.

Информация в этой книге, как правило, относится ко всем нашим примерам систем, если не дается уточнение. Подробности, касающиеся определенной системы, отмечены логотипом:



Большинство этих товарных знаков принадлежат поставщикам, которые выпускают соответствующее ПО и используются с любезного разрешения соответствующих владельцев. Однако поставщики не рассмотрели и не одобрили содержание этой книги.

Мы неоднократно пытались и не смогли получить разрешение от Red Hat на использование их знаменитого логотипа красная шапка, поэтому вам придется читать еще одну техническую аббревиатуру. По крайней мере, только на полях.

Ниже описывается немного более подробно каждая из систем.

**Примеры дистрибутивов Linux**

Информация, относящаяся к Linux, но не к какому-либо конкретному дистрибутиву, отмечена логотипом пингвина Такса (Tux), показанным слева.

Debian (произносится как *deb-ian*, названный в честь покойного основателя Иэна Мердока и его жены Дебры) является одним из старейших и наиболее признанных дистрибутивов. Это некоммерческий проект с тысячами контрибьюторами по всему миру. Debian поддерживает идеологическую приверженность развитию сообщества и открытому доступу, поэтому никогда не возникает вопросов о том, какие части дистрибутива являются свободными для использования.

Debian определяет три версии, которые поддерживаются одновременно: стабильный, ориентированный на рабочие серверы; нестабильный, с текущими пакетами, которые могут иметь ошибки и уязвимости безопасности; и тестовый, который находится где-то посередине.

Ubuntu основана на Debian и поддерживает приверженность Debian свободному и открытому программному обеспечению. Бизнес, который стоит за разработкой Ubuntu является Canonical Ltd., основанная предпринимателем Марком Шаттлвортом (Mark Shuttleworth).

Canonical предлагает множество выпусков Ubuntu, ориентированных на работу в облаках, рабочих компьютерах и "голом железе". Есть даже релизы, предназначенные для телефонов и планшетов. Номера версии Ubuntu производятся от года и месяца выпуска, так версия 16.10 выпущена в октябре 2016 года. Каждый выпуск также имеет аллитеративное кодовое имя, такое как Vivid Vervet (Вертлявая Верветка) или Wily Werewolf (Хитрый Оборотень).

В год выпускается две версии Ubuntu: в апреле и в октябре. В четные годы в апреле выпускаются версии долгосрочной поддержки (LTS), для которых напротяжении пяти лет выпускаются обновления. Эти выпуски рекомендуются к использованию в производственной среде.

Red Hat являлась доминирующей силой в мире Linux на протяжении более двух десятилетий, и его дистрибутивы широко используются в Северной Америке и за ее пределами. Если прибегнуть к цифрам, Red Hat, Inc. является самой успешной компанией, которая работает с открытым исходным кодом, в мире.

Red Hat Enterprise Linux, часто сокращаемая до RHEL, ориентирована на производственные среды на крупных предприятиях, которым требуется поддержка и консультационные услуги для бесперебойной работы их систем. Как это ни парадоксально, хоть RHEL и распространяет свободное ПО, но его использование требует лицензии. Если вы не хотите платить за лицензию, тогда вы не сможете использовать Red Hat.

Red Hat также спонсирует Fedora, дистрибутив разрабатываемый сообществом, который служит инкубатором для новейшего программного обеспечения, которое не считается достаточно стабильным для RHEL. Fedora используется в качестве начальной тестовой площадки для программного обеспечения и конфигураций, которые позже могут попасть в RHEL.

CentOS практически идентичен Red Hat Enterprise Linux, но является бесплатным. Проект CentOS (centos.org) принадлежит Red Hat, который использует своих ведущих разработчиков для его развития. Однако, они работают отдельно от команды Red Hat Enterprise Linux. Дистрибутиву CentOS не хватает продвижения Red Hat и нескольких фирменных утилит, но в остальном они похожи.

CentOS - это отличный выбор для организаций, которые хотят развернуть дистрибутив, ориентированный на производство, но не платить при этом отчисления Red Hat. Возможен также гибридный подход: серверы на передней линии могут работать под управлением Red Hat Enterprise Linux и пользоваться отличной поддержкой Red Hat, в то время когда непроизводственные системы работают под управлением CentOS. Этот механизм охватывает важные основы с точки зрения риска и поддержки, а также минимизирует затраты и сложность в администрировании.

CentOS стремится к полной бинарной совместимости и совместимости в ошибках с Red Hat Enterprise Linux. Вместо того, чтобы повторять "Red Hat и CentOS" до тошноты, мы обычно упоминаем только одно или другое название в этой книге. Текст в равной степени применим к Red Hat и CentOS, если мы не указываем иное.

Другие популярные дистрибутивы также являются потомками Red Hat. Oracle продает переименованную и переделанную версию CentOS клиентам своей корпоративной базы данных. Amazon Linux, доступный пользователям Amazon Web Services, первоначально произошел из CentOS и до сих пор разделяет многие из его конвенций.

Большинство администраторов столкнутся с системой, подобной Red Hat, в какой-то момент своей карьеры, и знакомство с ее нюансами будет полезным, даже если она не является системой №1 в вашей организации.

**Примеры дистрибутивов UNIX**

Популярность UNIX в течение некоторого времени снижается, и большинство ранее непоколебимых дистрибутивов UNIX (например, Solaris, HP-UX и AIX) больше не пользуются популярностью. Потомки BSD с открытым исходным кодом являются исключениями из этой тенденции и продолжают пользоваться популярностью, особенно среди экспертов по операционным системам, евангелистов свободного программного обеспечения и администраторов, ориентированных на безопасность. Другими словами, некоторые из ведущих мировых операционных систем полагаются на различные дистрибутивы BSD. Apple macOS имеет наследие BSD.

FreeBSD, впервые выпущенная в конце 1993 года, является наиболее широко используемой из производных BSD. Он занимает 70% рынка среди вариантов BSD в соответствии с некоторой статистикой использования. Пользователями FreeBSD являются крупные интернет-компании, такие как WhatsApp, Google и Netflix.

В отличие от Linux, FreeBSD - это полноценная операционная система, а не только ядро. Как ядро, так и программное обеспечение пользователя лицензируются по разрешительной лицензии BSD, что стимулирует его развитие и разработку дополнений со стороны бизнес-сообщества.

**1.5 Ньюансы и условные обозначения**

Имена файлов, команды и аргументы команд выделены жирным шрифтом. Заполнители (например, аргументы команд, которые не следует воспринимать буквально) выделены курсивом. Например, в команде

**ср** *файл каталог*

вы должны заменить *файл* и *каталог* с именами реальных файлов и каталогов.

Фрагменты конфигурационных файлов и сеансы терминалов отображаются нормальным шрифтом. Иногда мы даем комментарии к сессии символом комментария **bash** # и курсивом. Например:

$ **grep Bob /pub/phonelist** *# Найти номер телефона Боба*

Bob Knowles 555-2834

Bob Smith 555-2311

Мы используем **$** для обозначения работы в оболочке обычного непривилегированного пользователя и **#** для пользователя root. Если команда относится к конкретному дистрибутиву или семейству дистрибутивов, то перед вызовом команды ставится имя дистрибутива. Например:

$ **sudo su - root** *# Стать суперпользователем root*

# **passwd** *# Изменить пароль суперпользователя root*

debian# **dpkg -l** *# Вывести список установленных пакетов в Debian и Ubuntu*

Эти обозначения соответствуют тем, которые используются стандартными оболочками UNIX и Linux.

Вне этих конкретных примеров мы постарались свести к минимуму использование специальных шрифтов и форматирования, чтобы не нанести ущерб разборчивости. Например, мы часто говорим о таких сущностях, как группа процессов без специального форматирования.

Мы используем те же условные обозначения, какие находятся в справке для команд:

- Все, что находится в квадратных скобках (“[” и “]”), является необязательным;

- Все, за чем стоит многоточие (“...”), можно повторять;

- Фигурные скобки (“{” и “}”) указывают на то, что необходимо выбрать один из элементов, разделенных вертикальной чертой (“|”).

Например, спецификации

**bork** [**-х**] { **on** | **off** } *имя\_файла . . .*

соответствует любая из следующих команд:

**bork on /etc/passwd**

**bork -x off /etc/passwd /etc/smartd.conf**

**bork off /usr/lib/tmac**

В выражениях с шаблонами поиска используются следующие метасимволы:

- Звездочка (\*) обозначает нуль или более символов;

- Знак вопроса (?) обозначает один символ;

- Тильда (~) обозначает начальный каталог текущего пользователя;

- Выражение *~user* обозначает начальный каталог пользователя *user*.

Например, иногда мы обозначаем каталоги, где хранятся скрипты запуска **/etc/rc0.d**, **/etc/rc1.d** и т.д., сокращенным шаблоном **/etc/rc\*.d**.

Текст в кавычках часто имеет точное техническое значение. В этих случаях мы игнорируем обычные правила препинания и ставим знаки препинания за пределами кавычек, чтобы не было путаницы в отношении того, что включено, а что нет.

**1.6 Единицы измерения**

Такие метрические приставки, как кило-, мега- и гига-, определяются показателями степени числа 10; один мегабакс это $1,000,000. Эти приставки "позаимствованы" и для компьютерных типов данных с обозначением степеней числа 2. Например, один "мегабайт" памяти составляет 220 или 1,048,576 байт. Ассоциация твердотельных технологий (Solid State Technology Association) Объединенного инженерного совета по электронным устройствам (Joint Electronic Device Engineering Council — JEDEC) даже превратила эти “позаимствованные” единицы измерения в такой официальный стандарт, как Standard 100В.01, который признает их степенями двойки (не без некоторых натяжек).

В попытке внести ясность Международная Электротехническая комиссия (International Electrotechnical Commission - IЕС) определила ряд числовых приставок (kibi-, mebi-, gibi- и т. д., сокращенно Ki, Mi и Gi), основанных именно на степени числа 2. Эти единицы всегда однозначны, но они только начинают широко использоваться. Оригинальные префиксы серии кило- мега- и т.д. по-прежнему используются в обоих смыслах (десятичном и двоичном).

Контекст помогает определить значение приставки. Объем ОЗУ всегда обозначается в степени 2, но пропускная способность сети всегда обозначается кратной 10. Объем жесткого диска обычно указывается в степени числа 10, но размеры блока и страницы в действительности указываются в степени 2.

В этой книге мы используем единицы МЭК (IEC) для степеней 2, метрические единицы для степеней 10. Метрические единицы мы применяем для грубых значений и случаев, в которых точное основание неясно, недокументировано или его невозможно определить. В выходных данных команд и в конфигурационных файлах, или там, где разграничение не важно, мы оставляем исходные значения и обозначения единиц измерения. Мы указываем бит как *b* а байт как *B*. Таблица 1.2 показывает некоторые примеры интерпретации единиц измерения.

Аббревиатура *K*, как в "8KB оперативной памяти!" не является частью какого-либо стандарта. Это компьютеризированная адаптация метрической аббревиатуры *k* для кило-, и первоначально означала 1,024 а не 1,000. Но поскольку аббревиатуры для метрических префиксов уже имеют верхний регистр, аналогия не масштабируется. Позже люди запутались в различии и начали использовать *K* для обозначения коэффициента 1000.

Большинство стран мира не считают это важным делом и, как использование имперских единиц в США, метрические приставки могут быть использованы в обозримом будущем. Ubuntu поддерживает полезную политику единиц, хотя мы подозреваем, что она не была широко принята даже в Canonical (см. wiki.ubuntu.com/UnitsPolicy для получения дополнительных деталей).

**Таблица 1.2 примеры интерпретации единиц измерения**

|  |  |
| --- | --- |
| **Пример** | **Значение** |
| 1kB файл | Файл, который содержит 1,000 байт |
| 4KiB SSD страницы | SSD страницы, которые содержат 4,096 байт |
| 8KB памяти | Не используется в этой книге, см пример на этой странице |
| Ограничение размера файла 100MB | Неоднозначно, необходим контекст. Номинально 108 байт |
| Размер диска 100MB | Номинально 108 байт. В определенном контексте возможно 99,999,744 bytesa |
| 1 GiB оперативной памяти | 1,073,741,824 байт памяти |
| 1 Gb/s Ethernet | Сеть, которая передает 1,000,000,000 бит в секунду |
| Жесткий диск на 6TB | Жесткий диск, на котором хранится 6,000,000,000,000 байт |

а. В этом случае число 108 округлено в меньшую сторону к ближайшему целому блоку размером 512 байт.

**1.7 man-страницы и другая онлайн-документация**

Страницы руководства, обычно называются "man-страницами", поскольку они просматриваются с помощью команды **man**, представляют собой традиционную "сетевую" документацию (в наши дни вся документация в той или иной форме находится онлайн.) При установке новых пакетов программного обеспечения вместе с ними устанавливаются их справочные страницы. Даже в эпоху Google мы продолжаем обращаться к man-страницам как к авторитетному ресурсу, потому что они доступны из командной строки, обычно включают в себя полную информацию о параметрах программы и показывают полезные примеры и связанные с ними команды.

Man-страницы - это краткие описания отдельных команд, драйверов, форматов файлов или программных библиотек. Они не затрагивают более общие темы, такие как "Как мне установить новое устройство?" или "Почему эта система такая чертовски медленная?"

**Организация справочных страниц**

FreeBSD и Linux делят страницы на разделы. В таблице 1.3 показана базовая схема. Другие варианты UNIX иногда определяют разделы немного по-другому.

|  |  |
| --- | --- |
| **Раздел** | **Содержание** |
| 1 | Команды уровня пользователя и приложений |
| 2 | Системные вызовы и коды ошибок ядра |
| 3 | Вызовы библиотек |
| 4 | Драйверы устройств и сетевые протоколы |
| 5 | Стандартные форматы файлов |
| 6 | Игры и демонстрации |
| 7 | Различные файлы и документы |
| 8 | Команды системного администрирования |
| 9 | Скрытые спецификации ядра и интерфейсы |

Точная структура разделов не важна для большинства тем, потому что **man** найдет соответствующую страницу везде, где бы она не хранилась. Просто помните об определениях разделов, когда тема с тем же именем появляется в нескольких разделах. Например, **passwd** – можеть быть как командой так и файлом конфигурации, поэтому она имеет записи в разделах 1 и 5.

**Команда man: чтение страниц руководства**

*См. стр. 193, чтобы узнать о переменных окружения.*

Команда **man** *title* форматирует определенную справочную страницу и отправляет ее на ваш терминал используя команды **more**, **less** или любую другую, указанную в переменной среды PAGER. *Title* обычно представляет собой команду, устройство, имя файла или имя программной библиотеки. Поиск по разделам руководства осуществляется в числовом порядке, хотя в первую очередь обычно ищутся разделы, содержащие описание команд (разделы 1 и 8).

Команда **man** *section title* представляет вам справочную страницу из указанного раздела. Таким образом, в большинстве систем команда **man sync** показывает справочную страницу для команды **sync**, а команда **man 2 sync** показывает man-страницу для системного вызова **sync**.

Команда **man -k** *ключевое\_слово* или **apropos** *ключевое\_слово* выводит список справочных страниц, которые содержат указанное *ключевое\_слово*. Например:

**$ man -k translate**

objcopy (1) - копировать и преобразовать файлы объектов

dcgettext (3) - преобразовать сообщение

tr (1) - преобразовать или удалить символы

snmptranslate (1) - преобразовать значения SNMP OID в полезную информацию

tr (1p) - преобразовать символы

...

База данных ключевых слов может устареть. Если вы добавляете в систему дополнительные справочные страницы, вам может потребоваться перестроить этот файл с помощью команды **makewhatis** (Red Hat и FreeBSD) или **mandb** (Ubuntu).

**Хранение страниц интерактивного руководства**

Неформатированная информация для справочных страниц (входные данные команды **nroff**) обычно хранится в каталоге **/usr/share/man**, сжатых с помощью **gzip** для экономии места на диске. Команда **man** знает как разархивировать их налету. Команда **man** поддерживает кеш отформатированных страниц в каталогах **/var/cache/man** или **/usr/share/man**, если соответствующие каталоги доступны для записи, но эти операции рискованны с точки зрения безопасности. В большинстве систем предварительное форматирование справочных страниц выполняется однократно во время инсталляции (см. команду **catman**) или не выполняется вовсе.

Команда **man** может искать страницы руководства в нескольких каталогах, содержащих эти самые страницы. В системах Linux вы можете узнать какой путь поиска установлен по умолчанию позволяет команда **manpath**. Обычный вывод данного пути (в системе Ubuntu):

ubuntu$ **manpath**

/usr/local/man:/usr/local/share/man:/usr/share/man

Если необходимо, вы можете переназначить переменную среды MANPATH:

**$ export MANPATH=/home/share/localman:/usr/share/man**

Некоторые системы позволяют вам устанавливать любой путь поиска справочных страниц по умолчанию для всей системы, который может оказаться полезным в случае, если вам придется поддерживать параллельное дерево справочных страниц, например, сгенерированных кроссплатформенной системой управления пакетами OpenPKG. Для распространения локальной документации в виде man-страниц, проще всего использовать стандартный системный механизм упаковки и положить справочные страницы в стандартные справочные каталоги. Подробнее об этом написано в Главе 6, *Установка и управление программным обеспечением*.

**1.8 Другие виды официальной документации**

man-страницы - это лишь малая часть официальной документации. Большая ее часть, к сожалению, разбросана по сети.

**Руководства для конкретных систем**

Крупные поставщики имеют свои собственные проекты по подготовке и выпуску документации. Многие из них продолжают выпускать полезные руководства, включая руководства по администрированию и установке. Они, как правило, доступны онлайн для скачивания в формате PDF. В таблице 1.4 показывает где ее можно найти.

Хотя эта документация полезна, но она не относится к литературе, которую можно почитать перед сном. Сначала мы обычно гуглим ответы, а только позже обращаемся к официальной документации.

**Таблица 1.4 Где найти официальную документацию поставщиков ОС**

|  |  |  |
| --- | --- | --- |
| **ОС** | **Ссылка** | **Описание** |
| Debian | debian.org/doc | Руководство администратора отстает от текущей версии |
| Ubuntu | help.ubuntu.com | Наиболее дружественная к пользователю; см, раздел "server guide" для версий LTS |
| RHEL | redhat.com/docs | Исчерпывающая документация для администраторов |
| CentOS | wiki.centos.org | Включает в себя советы, Howto и часто задаваемые вопросы |
| FreeBSD | freebsd.org/docs.html | Смотрите *FreeBSD Handbook* для получения информации по администрированию |

**Документация по конкретным пакетам**

Большинство важных пакетов ПО в мире UNIX и Linux разрабатываются отдельными лицами или третьими сторонами, такими как Internet Systems Consortium и Apache Software Foundation. Эти группы пишут свою собственную документацию. Диапазон качества находится в пределах от сомнительного до впечатляющего, но жемчужины вроде *Pro git* от git-scm.com/book заставляют попотеть в их поисках, но оно того стоит.

Дополнительные документы включают в себя технические отчеты, проектные обоснования и книги или брошюры по конкретным темам. Эти дополнительные материалы не ограничиваются описанием только одной команды, поэтому они могут применять методический или процедурный подход. Многие пакеты ПО имеют в своем составе как man-страницы так и целые статьи. Например, на справочной странице для **vim** рассказывается об аргументах командной строки, которые понимает **vim**, но вам нужно перейти к углубленному изучению, чтобы узнать, как на самом деле редактировать файлы.

Большинство программных продуктов имеют списки рассылки пользователей и разработчиков и каналы IRC. Эти источники должны быть первыми, куда вам следует обращаться при возникновении специфического вопроса по настройке или если вы столкнулись с ошибкой.

**Книги**

Книги издательнства О'Рейли (O’Reilly) являются одними из лучших в индустрии высоких технологий. Бизнес начался с книги *UNIX in a  
Nutshell* и теперь включает в себя отдельный том почти для каждой важной подсистемы Unix и Linux и команды. O'Reilly также публикует книги по сетевым протоколам, языкам программирования, Microsoft Windows и другим темам, не связанным с UNIX. Все книги продаются по разумным ценам, публикуются своевременно и специализируются на определенной теме.

Многие читатели обращаются к сервису онлайн-подписки O'Reilly's Safari Books Online, который предлагает неограниченный доступ к электронным книгам, видео и другим учебным ресурсам. Содержимое сервиса представлено не только O’Reilly, многие издатели также включены в его состав и вы можете выбирать что угодно из огромной библиотеки материалов.

**Публикации RFC**

Документы "Request for Comments" - рабочее предложение - содержат описание протоколов и процедур, используемых в Интернете. Большинство из них являются достаточно подробными и техническими, но некоторые из них написаны в виде обзоров. Фраза "эталонная реализация", применяемая к программному обеспечению, обычно переводится как "реализованная доверенным источником в соответствии со спецификацией RFC".

RFC являются абсолютно авторитетными, и многие из них весьма полезны для системных администраторов. Более полное описание этих документов в Главе 13.1. Мы обращаемся к различным RFC везде в этой книге.

**1.9 Другие источники информации**

Источники, рассмотренные в предыдущем разделе, рецензируются и пишутся авторитетными организациями, но они едва ли являются последним словом в администрировании UNIX и Linux. Бесчисленные блоги, дискуссионные форумы и новостные ленты доступны в Интернете и часто содержат самую свежую информацию.

Само собой разумеется, что Google - лучший друг системного администратора. Если вам необходима подробная информация о конкретной команде или формате файла, Google или эквивалентная поисковая система должна быть первым ресурсом, с которым вы консультируетесь по любому вопросу системного администрирования. Выработайте в себе привычку при возникновении затруднений обращаться к поисковым системам. Так вы избежите задержек и унижений, когда на Ваши вопросы в онлайн-форуме будет дан ответ со ссылкой на Google1. *Когда застрял, ищи в Интернете*.

1. Или еще хуже, ссылка на Google через lmgtfy.com

**Актуальность информации**

Операционные системы, инструменты и методы, которые их поддерживают, меняются достаточно быстро. Ознакомьтесь с сайтами, представленными в таблице 1.5 чтобы быть в курсе тенденций отрасли.

**Таблица 1.5 Ресурсы, содержащие актуальную информацию**

|  |  |
| --- | --- |
| **Веб сайт** | **Описание** |
| darkreading.com | Новости безопасности, тенденции и обсуждения |
| devopsreactions.tumblr.com | Юмор сисадминов в виде анимированных GIF файлов |
| linux.com | Сайт Linux Foundation; форум, хорош для новичков |
| linuxfoundation.org | Некоммерческое содействие OSS, работодатель Линуса Торвальдса |
| lwn.net | Качественные, своевременные статьи по Linux и OSS |
| lxer.com | Агрегатор новостей Linux |
| securityfocus.com | Отчеты об уязвимостях и списки рассылки, связанные с безопасностью |
| @SwiftOnSecurity | Мнение Infosec от Taylor Swift (пародийный аккаунт) |
| @nixcraft | Твиты об администрировании UNIX и Linux |
| everythingsysadmin.com | Блог Томаса Лимончелли, уважаемого сисадминаа |
| sysadvent.blogspot.com | Рождественский пост для системных администраторов со статьями каждый декабрь |
| oreilly.com/topics | Обучающие ресурсы от O'Reilly по многим темам |
| schneier.com | Блог Брюса Шнайера, эксперта по конфиденциальности и безопасности |

а. См. также коллекцию Тома первоапрельских розыгрышей RFC на rfc-humor.com

Социальные медиа также полезны. Twitter и reddit, в частности, имеют большие и активные сообщества, которые могут многое вам предложить, хотя соотношение хорошего материала к плохому иногда может быть довольно низким. Присоединяйтесь к сообществам sysadmin, linux, linuxadmin и netsec в reddit.

**Инструкции и справочные сайты**

Сайты, перечисленные в таблице 1.6, содержат руководства, учебные пособия и статьи о том, как выполнять конкретные задачи в UNIX и Linux.

**Таблица 1.6 Специализированные форумы и справочные сайты**

|  |  |
| --- | --- |
| **Веб сайт** | **Описание** |
| wiki.archlinux.org | Статьи и руководства для Arch Linux; многие из них носят общий характер |
| askubuntu.com | Вопросы и ответы для пользователей и разработчиков Ubuntu |
| digitalocean.com | Учебники по многим темам OSS, разработки и системного администрированияа |
| kernel.org | Официальный сайт ядра Linux |
| serverfault.com | Совместно редактируемая база данных вопросов системного администрированияb |
| serversforhackers.com | Качественные видео, форумы и статьи по администрированию |

а. Смотри digitalocean.com/community/tutorials

b. Также смотри stackoverflow.com, сайт который посвящен программированию, но также полезен для сисадминов

Stackoverflow и ServerFault, оба перечислены в таблице 1.6 (являются членами группы сайтов Stack Exchange), заслуживают ближайшего рассмотрения. Если у вас возникли проблемы, скорее всего, кто-то уже столкнулся с ней и попросил о помощи на одном из этих сайтов. Формат вопросов и ответов, используемый сайтами StackExchange, хорошо подходит для проблем, с которыми сталкиваются системные администраторы и программисты. Вам стоит создать учетную запись и присоединиться к этому большому сообществу.

**Конференции**

Отраслевые конференции - это отличный способ пообщаться с другими профессионалами, следить за тенденциями в области технологий, проходить учебные занятия, получать сертификаты и узнавать о последних услугах и продуктах. В последние годы резко возросло число конференций по вопросам управления. В таблице 1.7, на следующей странице, приведены некоторые из наиболее заметных примеров.

Митапы (meetup.com) еще один способ общения и взаимодействия с единомышленниками. Самые большие города Соединенных Штатов и во всем мире имеют группу пользователей Linux или DevOps meetup, которые спонсируют спикеров, проведение дискуссий и hack days.

**1.10 Способы поиска и установки программного обеспечения**

В Главе 6, *Установка программного обеспечения и управление им*, подробно рассматривается разворачивание программного обеспечения. Но для нетерпеливых, мы представляем краткое руководство о том, как узнать, что установлено на вашей системе и как получить и установить новое программное обеспечение.

Современные операционные системы делят свое содержимое на пакеты, которые можно устанавливать независимо друг от друга. Базовая установка включает в себя ряд "стартовых" пакетов, который можно расширять или сокращать в соответствии с вашими потребностями. При добавлении программного обеспечения помните, что дополнительное программное обеспечение может принести с собой уязвимости в безопасности. Устанавливайте только то, что необходимо.

Дополнительное программное обеспечение часто предоставляется в виде предварительно скомпилированных пакетов, хотя такой подход широко варьируется в различных системах. Большая часть программного обеспечения разрабатывается независимыми группами, которые выпускают программное обеспечение в виде исходного кода. Затем репозитории пакетов собирают исходный код, компилируют его в соответствии с особенностями конкретной системы, и упаковывают полученные бинарные файлы. Как правило, намного проще инсталлировать бинарную версию пакета, чем искать и компилировать исходный код. Однако нужно учитывать, что версия ПО в пакетах иногда отстает от последних версий ПО на один-два релиза.

**Таблица 1.7 Конференции, полезные для системных администраторов**

|  |  |  |  |
| --- | --- | --- | --- |
| **Конференция** | **Место проведения** | **Даты проведения** | **Описание** |
| LISA | Варьируется | 4 квартал | Large Installation System Administration |
| Monitorama | Портланд | Июнь | Инструменты и методы мониторинга |
| OSCON | США/Европа | 2 или 3 квартал | Длительная конференция O'Reilly OSS |
| SCALE | Пасадена | Январь | Linux Expo южной Калифорнии |
| DefCon | Лас Вегас | Июль | Старейшее и крупнейшее собрание хакеров |
| Velocity | Глобально | Варьируется | Конференция O'Reilly по веб-операциям |
| BSDCan | Оттава | Май/Июнь | Все о BSD для новичков и гуру |
| re:Invent | Лас Вегас | 4 квартал | Конференция AWS по облачным вычислениям |
| VMWorld | США/Европа | 3 или 4 квартал | Виртуализация и облачные вычисления |
| LinuxCon | Глобально | Варьируется | Будущее Linux |
| RSA | Сан Франциско | 1 или 2 квартал | Корпоративная криптография и защита информации |
| DevOpsDays | Глобально | Варьируется | Ряд тем по преодолению разрыва между командами разработки и ops |
| QCon | Глобально | Варьируется | Конференция для разработчиков программного обеспечения |

Тот факт, что две системы используют один и тот же формат пакетов, не обязательно означает, что пакеты для двух систем взаимозаменяемы. Например, Red Hat и SUSE используют RPM, но их файловая система несколько отличается. Лучше всего использовать пакеты, предназначенные для вашей конкретной системы.

В приведенных нами системах используются отличные системы управления пакетами, которые включают инструменты для доступа и поиска размещенных в Интернете репозиториев программного обеспечения. Дистрибьюторы активно поддерживают эти хранилища в интересах сообщества, чтобы облегчить исправление и обновление программного обеспечения. Другими словами – жить становится проще.

Когда ПО в пакетах становится недостаточным, администраторы должны установить его по старинке: загрузив архив **tar** исходного кода и вручную сконфигурировав, скомпилировав и установив его. В зависимости от программного обеспечения и операционной системы этот процесс может варьироваться от обычного до кошмарного.

В этой книге мы предполагаем, что дополнительное программное обеспечение уже установлено и не мучает вас стандартными инструкциями по установке каждого отдельного пакета. Чтобы избежать путаницы, мы иногда будем упоминать точные имена пакетов, необходимых для завершения конкретного проекта. В большинстве случаев мы не повторяем инструкции по установке, так как они, как правило, одинаковы для всех пакетов.

**Определение факта установки программного обеспечения**

По ряду причин может быть немного сложно определить, какой пакет содержит необходимое вам программное обеспечение. Вместо того, чтобы начинать с уровня пакета, проще использовать команду оболочки **which**, чтобы узнать, находится ли уже соответствующий бинарный файл в пути поиска по умолчанию. Например, следующая команда показывает, что компилятор GNU C уже установлен на этом компьютере:

ubuntu$ **which gcc**

/usr/bin/gcc

Если команда **which** не может найти то что вы ищете, попробуйте **whereis**; она ищет в более широком диапазоне системных каталогов и не зависит от пути поиска вашей оболочки.

Другой альтернативой является невероятно полезная команда **locate**, которая просматривает предварительно скомпилированный индекс файловой системы, чтобы найти имена файлов, которые соответствуют определенному шаблону.

FreeBSD включает **locate** в часть базовой системы. В Linux текущая реализация **locate** находится в пакете **mlocate**. В Red Hat и CentOS вы можете установить пакет **mlocate** с помощью **yum**; см. стр. 174.

**locate** может найти файл любого типа, но он не ищет команды или пакеты. Например, если Вы не были уверены, где найти **signal.h**, вы можете попробовать:

freebsd$ **locate signal.h**

/usr/include/machine/signal.h

/usr/include/signal.h

/usr/include/sys/signal.h

...

База данных команды **locate** обновляется командой **updatedb** (в FreeBSD, locate.updatedb), которая периодически запускается из **cron**. Поэтому результаты поиска не всегда отражают последние изменения в файловой системе.

*Дополнительные сведения об управлении пакетами см. в главе 6.*

Если вы знаете имя пакета, который ищете, Вы также можете использовать утилиты упаковки в вашей системе, чтобы проверить наличие пакета непосредственно. Например, в системе Red Hat следующая команда проверяет наличие (и установленную версию) интерпретатора Python:

redhat$ **rpm -q python**

python-2.7.5-18.el7\_1.1.x86\_64

Вы также можете узнать, к какому пакету принадлежит конкретный файл:

redhat$ **rpm -qf /etc/httpd**

httpd-2.4.6-31.el7.centos.x86\_64

freebsd$ **pkg which /usr/local/sbin/httpd**

/usr/local/sbin/httpd was installed by package apache24-2.4.12

ubuntu$ **dpkg-query -S /etc/apache2**

apache2: /etc/apache2

**Добавление нового программного обеспечения**

Если Вам необходимо установить дополнительное программное обеспечение, сначала необходимо определить каноническое имя соответствующего пакета. Например, вам необходимо преобразовать "я хочу установить **locate**" в "мне нужно установить пакет **mlocate**" или "мне нужно имя" на "я должен установить BIND". Разнообразие системных индексов в Интернете может помочь в этом, особенно Google. Например, запрос вида "команда **locate**" приведет вас непосредственно к нескольким соответствующим обсуждениям.

Следующие примеры демонстрируют установку **tcpdump** на каждую из представденных нами систем. **tcpdump** - это инструмент захвата пакетов, который позволяет просматривать необработанные пакеты, отправляемые в/из системы в сети.

Debian и Ubuntu используют APT, Расширенный инструмент упаковки:

ubuntu# **sudo apt-get install tcpdump**

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following NEW packages will be installed:

tcpdump

0 upgraded, 1 newly installed, 0 to remove and 81 not upgraded.

Need to get 0 B/360 kB of archives.

After this operation, 1,179 kB of additional disk space will be used.

Selecting previously unselected package tcpdump.

(Reading database ... 63846 files and directories currently installed.)

Preparing to unpack .../tcpdump\_4.6.2-4ubuntu1\_amd64.deb ...

Unpacking tcpdump (4.6.2-4ubuntu1) ...

Processing triggers for man-db (2.7.0.2-5) ...

Setting up tcpdump (4.6.2-4ubuntu1) ...

Вариант для Red Hat и CentOS:

redhat# **sudo yum install tcpdump**

Loaded plugins: fastestmirror

Determining fastest mirrors

\* base: mirrors.xmission.com

\* epel: linux.mirrors.es.net

\* extras: centos.arvixe.com

\* updates: repos.lax.quadranet.com

Resolving Dependencies

--> Running transaction check

---> Package tcpdump.x86\_64 14:4.5.1-2.el7 will be installed

--> Finished Dependency Resolution

tcpdump-4.5.1-2.el7.x86\_64.rpm | 387 kB 00:00

Running transaction check

Running transaction test

Transaction test succeeded

Running transaction

Installing : 14:tcpdump-4.5.1-2.el7.x86\_64 1/1

Verifying : 14:tcpdump-4.5.1-2.el7.x86\_64 1/1

Installed:

tcpdump.x86\_64 14:4.5.1-2.el7

Complete!

**pkg** - менеджер пакетов для FreeBSD:

freebsd# **sudo pkg install -y tcpdump**

Updating FreeBSD repository catalogue...

Fetching meta.txz: 100% 944 B 0.9kB/s 00:01

Fetching packagesite.txz: 100% 5 MiB 5.5MB/s 00:01

Processing entries: 100%

FreeBSD repository update completed. 24632 packages processed.

All repositories are up-to-date.

The following 2 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:

tcpdump: 4.7.4

libsmi: 0.4.8\_1

The process will require 17 MiB more space.

2 MiB to be downloaded.

Fetching tcpdump-4.7.4.txz: 100% 301 KiB 307.7kB/s 00:01

Fetching libsmi-0.4.8\_1.txz: 100% 2 MiB 2.0MB/s 00:01

Checking integrity... done (0 conflicting)

[1/2] Installing libsmi-0.4.8\_1...

[1/2] Extracting libsmi-0.4.8\_1: 100%

[2/2] Installing tcpdump-4.7.4...

[2/2] Extracting tcpdump-4.7.4: 100%

**Сборка программного обеспечения из исходного кода**

В качестве примера рассмотрим сборку **tcpdump** из исходного кода.

Первая задача - найти код. Разработчики программного обеспечения иногда выкладывают индекс выпусков на веб-сайте проекта, которые доступны для скачивания в **tar**-архиве. Для проектов с открытым исходным кодом вы, скорее всего, найдете код в репозитории Git.

Исходный код **tcpdump** хранится на GitHub. Клонируйте репозиторий в каталог **/tmp**, создайте ветку с тегами, которую вы хотите собрать, затем распакуйте, настройте, соберите и установите ее:

redhat$ **git clone https://github.com/the-tcpdump-group/tcpdump.git**

*<status messages as repository is cloned>*

redhat$ **cd tcpdump**

redhat$ **git checkout tags/tcpdump-4.7.4 -b tcpdump-4.7.4**

Switched to a new branch 'tcpdump-4.7.4'

redhat$ **./configure**

checking build system type... x86\_64-unknown-linux-gnu

checking host system type... x86\_64-unknown-linux-gnu

checking for gcc... gcc

checking whether the C compiler works... yes

...

redhat$ **make**

*<several pages of compilation output>*

redhat$ **sudo make install**

*<files are moved in to place>*

Эта последовательность **configure/make/make install** является общей для большинства программ, написанных на C, и работает на всех системах UNIX и Linux. Хорошей идеей является изучение файлов INSTALL или README для ознакомления с особенностями. Необходимо, чтобы в системе была установлена среда разработки и все необходимые компоненты, относящиеся к пакету (в случае с **tcpdump** обязательным условием является наличие **libpcap** и его библиотек).

Вам часто нужно будет настраивать конфигурацию сборки, поэтому используйте **./configure --help** для просмотра параметров, доступных для каждого конкретного пакета. Еще один полезный параметр команды **configure** является **--prefix**=*directory*, который позволяет компилировать программное обеспечение для установки где-либо кроме папки **/usr/local**, которая обычно используется по умолчанию.

**Установка с помощью веб-скрипта**

Кросс-платформенные пакеты программного обеспечения все чаще предлагают ускоренный процесс установки, который представляет собой shell скрипт, который можно загрузить из Интернета с помощью команд **curl, fetch** или **wget2**. Например, чтобы настроить компьютер в качестве клиента Salt, можно выполнить следующие команды:

*2. Все они являются простыми HTTP-клиентами, которые загружают содержимое URL-адреса в локальный файл или, при необходимости, печатают содержимое в STDOUT.*

$ **curl -o /tmp/saltboot -sL https://bootstrap.saltstack.com**

$ **sudo sh /tmp/saltboot**

Скрипт автозагрузки исследует локальную среду, затем загружает, устанавливает и настраивает соответствующую версию программного обеспечения. Этот тип установки особенно распространен в тех случаях, когда сам процесс несколько сложен, но поставщик сильно заинтересован в его облегчении для пользователей (хороший пример - RVM; см. стр. 232).

*Дополнительные сведения об установке пакетов см. в главе 6.*

Этот метод установки очень хорош, но он поднимает несколько вопросов, достойные внимания. Для начала, он не оставляет надлежащей записи установки для обращения к ней в будущем. Если ваша операционная система предлагает пакетную версию программного обеспечения, предпочтительно установить пакет вместо использования веб-установщика. Пакеты легко отслеживать, обновлять и удалять (с другой стороны, большинство пакетов уровня ОС уже устарели. Вы вероятно не захотите работать с самыми последними версиями ПО).

*Подробные сведения о цепочке доверия Https см. на стр. 1007.*

Будьте бдительны, если URL-адрес скрипта загрузки не является безопасным (то есть он не начинается с https). Незащищенный HTTP очень прост для перехвата, а URL-адреса установки представляют особый интерес для хакеров, потому что они знают, что вы, вероятно, запустите любой код, который вам предоставят, из-под root. Для сравнения, HTTPS проверяет подлинность сервера с помощью криптографической цепочки доверия. Не сложно, но достаточно надежно.

Несколько поставщиков публикуют HTTP URL, которые автоматически перенаправляется на версию HTTPS. Это глупо и на самом деле не более безопасно, чем прямой доступ по HTTP. Не существует ничего для защиты от перехвата первоначального обмена по HTTP, поэтому вы можете и не достичь перенаправления на HTTPS поставщика. Тем не менее, существование таких перенаправлений означает, что стоит попробовать свою собственную замену https на http в небезопасных URL. Чаще всего это отлично срабатывает.

Shell принимает текст скрипта на свой стандартный ввод, и эта функция позволяет проводить аккуратную процедуру установки в одну строку, такую как следующая:

$ **curl -L https://badvendor.com | sudo sh**

Однако, существует потенциальная проблема с этой конструкцией в том, что root оболочка останется активной, даже если **curl** частично отрабатывает, а затем фейлится - скажем, из-за временного сбоя сети. Конечный результат непредсказуем и потенциально не дает хорошего результата.

Нам не известны задокументированные случаи, относящихся к данной проблеме. Тем не менее, такие ошибки вполне вероятны. Более того, конвейерезация вывода **curl** в оболочку вошла в среду системного администрирования неосознанно и является типичной грубой ошибкой новичка, поэтому, если вам придётся поступить так, держите это в тайне.

Исправить это легко: просто сохраните скрипт во временный файл, а затем запустите его на отдельном шаге после успешного завершения скачивания.

**1.11 Хостинг**

Операционные системы и программное обеспечение могут размещаться в частных датацентрах, на оборудовании co-location, в облачной платформе или на некоторой их комбинации. Большинство быстрорастущих стартапов выбирают облака. Уже укоренившиеся энтерпрайзы, вероятно, предпочтут иметь существующие датацентры и запустить в них свое частное облако.

Наиболее практичным выбором и нашей рекомендацией для новых проектов является использование публичного облачного провайдера. Такие ресурсы имеют многочисленные преимущества над датацентрами:

- Нет капитальных затрат и низкие первоначальные эксплуатационные расходы

- Нет необходимости в установке, защите и управлении оборудованием

- Изменение объемов хранилища, пропускной способности сети и вычислительной мощности по требованию

- Готовые решения для дополнительных потребностей, таких как базы данных, балансировщики нагрузки, очереди, мониторинг и многое другое

- Более дешевая и простая реализация высокодоступных / избыточных систем

Ранние облачные системы приобрели свою репутацию из-за слабой безопасности и производительности, сейчас эти вопросы больше не вызывают опасений. В наши дни большая часть административной работы перенесена в облака. Общие сведения об этой теме см. в главе 9.

Наш выбор пал на лидера в области облачных вычислений: Amazon Web Services (AWS). Gartner, ведущая технологическая исследовательская фирма, выяснила, что AWS в десять раз больше всех конкурентов вместе взятых. AWS быстро внедряет инновации и предлагает гораздо более широкий спектр услуг, чем любой другой провайдер. Она также имеет хорошую репутацию из-за превосходного обслуживания клиентов и поддержки большого и активного сообщества. AWS предлагает ряд бесплатных услуг, чтобы попрактиковаться, в их числе год использования маломощного облачного сервера.

Google Cloud Platform (GCP) активно совершенствует и продает свои продукты. Некоторые утверждают, что его технология не имеет аналогов у других поставщиков. Рост GCP был медленным, отчасти из-за репутации Google отказываться от поддержки популярных предложений. Однако его приятные цены и уникальные функции являются отличительными и привлекательными чертами.

DigitalOcean является более простым сервисом с заявленной целью в обеспечении высокой производительности. Его целевой рынок - разработчики, которых он обеспечивает понятным API, низкой ценой и чрезвычайно быстрой загрузкой. DigitalOcean является сильным сторонником свободного программного обеспечения, и его учебники и руководства по популярным интернет-технологиям являются лучшими из доступных.

**1.12 специализация и смежные дисциплины**

Системные администраторы не существуют в вакууме; для создания и обслуживания сложной сети требуется целая команда экспертов. В этом разделе описываются некоторые роли, с которыми системные администраторы пересекаются в навыках и знаниях. Некоторые администраторы предпочитают специализироваться в одной или нескольких из этих областей.

Ваша цель как системного администратора или как профессионала, работающего в любой из этих смежных областей, заключается в достижении целей организации. Не позволяйте политике или иерархии мешать прогрессу. Лучшие администраторы решают проблемы и свободно обмениваются информацией с другими.

**DevOps**

*Дополнительные комментарии по DevOps см. на стр. 1106.*

DevOps-это не столько конкретная функция, сколько культура или философия эксплуатации. Она направлена на повышение эффективности создания и доставки программного обеспечения, особенно в больших системах, которые имеют много взаимосвязанных служб и команд. Организации использующие DevOps способствуют интеграции между инженерными группами и могут практически не проводить различия между development и operations. Эксперты, работающие в этой области, ищут неэффективные процессы и заменяют их небольшими скриптами или громоздкими репозиториями Chef.

**Инженеры по обеспечению надёжности (SRE)**

SR-инженеры прежде всего ценят uptime (время безотказной работы) и корректность в работе. Мониторинг сетей, развертывание ПО в продакшн-среде, дежурство, планирование будущего расширения и отладка простоев - все это находится в сфере ответственности этих рыцарей надёжности. Одиночные точки отказа являются заклятыми врагами SR-инженеров.

**Инженеры информационной безопасности**

Инженеры-безопасники сосредоточены на практической, повседневной стороне программы информационной безопасности. Эти люди устанавливают и управляют инструментами, которые ищут уязвимости и отслеживают атаки в сети. Они также участвуют в моделировании нападений для оценки эффективности своих методов предотвращения и обнаружения.

**Сетевые администраторы**

Сетевые администраторы разрабатывают, устанавливают, настраивают и работают с сетью. В организациях, которые содержат свои датацентры, чаще всего работают сетевые администраторы, потому что эти объекты имеют множество физических коммутаторов, маршрутизаторов, брандмауэров и других устройств, требующих управления. Облачные платформы также предлагают различные сетевые возможности, но они обычно не требуют специального администратора, поскольку большая часть работы выполняется поставщиком.

**Администраторы баз данных**

Администраторы баз данных (иногда называемые DBA) являются экспертами по установке и управлению программным обеспечением баз данных. Они управляют схемами баз данных, выполняют установки и обновления, настраивают кластеризацию, настраивают параметры для оптимальной производительности и помогают пользователям формулировать эффективные запросы. DBA – волшебники, которые знают один или несколько языков запросов и имеют опыт работы с реляционными и нереляционными базами данных (NoSQL).

**Инженеры центра управления сетью (NOC)**

NOC инженеры контролируют в реальном масштабе времени состояние больших систем и отслеживают инциденты и сбои. Они устраняют неполадки по запросам пользователей, занимаются установкой обновлений и координируют действия других команд. Их чаще всего можно увидеть наблюдающими за стеной мониторов, на которых показаны графики и различные измерения.

**Технические специалисты датацентра**

Специалисты центра обработки данных работают с оборудованием. Они получают новое оборудование, проводят инвентаризацию оборудования и его жизненные циклы, устанавливают серверы в стойках, прокладывают кабели, поддерживают питание и кондиционирование воздуха, а также ежедневно управляют работой датацентра. Как системному администратору, в ваших интересах подружиться с этими техниками и подкупить их безалкогольными напитками, кофе и печеньками.

**Архитекторы**

Системные архитекторы имеют глубокие знания в более чем одной области. Они используют свой опыт для проектирования распределенных систем. Их должностные обязанности могут включать определение зон безопасности и их сегментацию, устранение отдельных точек отказа, планирование будущего роста, обеспечение связи между несколькими сетями и третьими сторонами, а также принятие других решений, затрагивающих все предприятие. Хорошие архитекторы технически подкованны и, как правило, предпочитают внедрять и тестировать свои собственные проекты.

**1.13 Рекомендуемая литература**

Abbott, Martin L., and Michael T. Fisher. *The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise (2nd Edition).* Addison-Wesley Professional, 2015.

Gancarz, Mike. *Linux and the Unix Philosophy.* Boston: Digital Press, 2003.

Limoncelli, Thomas A., and Peter Salus. *The Complete April Fools’ Day RFCs.* Peer-to-Peer Communications LLC. 2007. Engineering humor. You can read this collection on-line for free at rfc-humor.com.

Raymond, Eric S. *The Cathedral & The Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary.* Sebastopol, CA: O’Reilly Media, 2001.

Salus, Peter H. *The Daemon, the GNU & the Penguin: How Free and Open Software is Changing the World.* Reed Media Services, 2008. Эта увлекательная история изменения открытого ПО от самого известного историка UNIX также доступна по адресу groklaw.com под лицензией Creative Commons. URL-адрес самой книги довольно длинный; найдите текущую ссылку на groklaw.com или используйте этот сжатый эквивалент: tinyurl.com/d6u7j.

Siever, Ellen, Stephen Figgins, Robert Love, and Arnold Robbins. *Linux in a Nutshell (6th Edition).* Sebastopol, CA: O’Reilly Media, 2009.

**Системное администрирование и DevOps**

Kim, Gene, Kevin Behr, and George Spafford. *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win.* Portland, OR: IT Revolution Press, 2014. Руководство по философии и мышлению, необходимое для управления современной ИТ-организацией, написанное как повествование. Мгновенно стала классикой.

Kim, Gene, Jez Humble, Patrick Debois, and John Willis. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations.* Portland, OR: IT Revolution Press, 2016.

Limoncelli, Thomas A., Christina J. Hogan, and Strata R. Chalup. *The Practice of System and Network Administration (2nd Edition).* Reading, MA: Addison- Wesley, 2008. Это хорошая книга с особенно сильным освещением политических и процедурных аспектов системного администрирования. Авторы поддерживают блог для системных администраторов everythingsysadmin.com.

Limoncelli, Thomas A., Christina J. Hogan, and Strata R. Chalup. *The Practice of Cloud System Administration.* Reading, MA: Addison-Wesley, 2014. От тех же авторов, что и предыдущее издание, теперь с акцентом на распределенные системы и облачные вычисления.

**Основной инструментарий**

Blum, Richard, and Christine Bresnahan. *Linux Command Line and Shell Scripting Bible (3rd Edition).* Wiley, 2015.

Dougherty, Dale, and Arnold Robins. *Sed & Awk (2nd Edition).* Sebastopol, CA: O’Reilly Media, 1997. Классическая книга О'Рейли о мощных, незаменимых текстовых процессорах **sed** и **awk**.

Kim, Peter. *The Hacker Playbook 2: Practical Guide To Penetration Testing.* CreateSpace Independent Publishing Platform, 2015.

Neil, Drew. *Practical Vim: Edit Text at the Speed of Thought.* Pragmatic Bookshelf, 2012.

Shotts, William E. *The Linux Command Line: A Complete Introduction.* San Francisco, CA: No Starch Press, 2012.

Sweigart, Al. *Automate the Boring Stuff with Python: Practical Programming for Total Beginners.* San Francisco, CA: No Starch Press, 2015.