ОС Linux

Linux - это операционная система для IBM-совместимых персональных компьютеров и рабочих станций. Это многопользовательская ОС с сетевой оконной графической системой X Window System. ОС Linux поддерживает стандарты открытых систем и протоколы сети Интернет и совместима с системами Unix, DOS, MS Windows. Все компоненты системы, включая исходные тексты, распространяются с лицензией на свободное копирование и установку для неограниченного числа пользователей.

Будучи традиционной операционной системой, Linux выполняет многие из функций, характерных для DOS и Windows. Однако следует отметить, что эта ОС отличается особой мощью и гибкостью. Система Linux разрабатывалась как ПК-версия операционной системы Unix, которая десятилетиями используется на мэйнфреймах и мини-ЭВМ и является основной ОС для рабочих станций. Linux предоставляет в распоряжение пользователя ПК скорость, эффективность и гибкость Unix, используя при этом все преимущества персональных машин. При работе с мышью активно используются все три

кнопки, в частности средняя кнопка используется для вставки фрагментов текста.

С экономической точки зрения Linux обладает еще одним весьма существенным достоинством - это бесплатная система. Linux распространяется по генеральной открытой лицензии GNU в рамках фонда свободного программного обеспечения (Free Software Foundation), что делает эту ОС доступной для всех желающих. Linux защищена авторским правом и не находится в общедоступном пользовании, однако открытая лицензия GNU это почти то же самое, что и передача в общедоступное пользование. Она составлена так, что Linux остается бесплатной и в то же время стандартизированной системой. Существует лишь один официальный вариант ядра Linux.

Многопользовательский режим означает, что в системе могут одновременно работать несколько пользователей, каждый из которых взаимодействует с ней через свой терминал. Еще одним из достоинств этой ОС является возможность ее установки совместно с Windows на один компьютер.

Linux способен любую персональную машину превратить в рабочую станцию. В наше время Linux является операционной системой для бизнеса, образования и

индивидуального программирования. Университеты по всему миру применяют Linux в учебных курсах по программированию и проектированию операционных систем. Он стал незаменим в широких корпоративных сетях, а также для организации Интернет-узлов и Web-серверов.

Современный Linux предоставляет возможность использовать несколько разновидностей графического интерфейса: KDE (K Desktop Environment), GNOME (GNU Network Model Environment) и другие. В каждой из этих оболочек пользователю предоставляется возможность работы сразу с несколькими рабочими столами (в то время как в MS Windows всегда один рабочий стол, который приходится загромождать окнами).

Файловая система в ОС Linux, как и в ОС Windows, представляет собой иерархическую структуру каталогов и файлов (в виде дерева), но при этом имеет ряд кардинальных отличий.

Структура каталогов

В ОС Windows жесткие диски называются латинскими буквами (C:, D:, ...), и каждый из дисков представляет собой корневой каталог с собственным деревом папок. Подключение же нового устройства приведет к появлению нового корневого каталога со своей буквой (например, F:). В ОС Linux файловая система представлена единым корневым каталогом, обозначаемым как слэш (/). Соответственно, при данной файловой структуре не диски содержат каталоги, а каталог — диски.

Понятие файла

Понятие «файл» в Linux имеет несколько другое значение, нежели в Windows. «Файлом» можно назвать обычный файл, содержащий данные, и интерпретируемый программой. Директория также является «файлом»,

содержащим в себе ссылки на другие директории или файлы с данными. Файлы устройства указывает на драйвер, благодаря которому система взаимодействует с физическими устройствами. Имеются и многие другие типы файлов.

Принцип установки программ

Если в Windows программы, зачастую, хранят все данные в одной папке, например в «C:Program FilesProgramName», то в Linux файлы программы разделяются по каталогам в зависимости от типа. Например, исполняемые файлы в /bin, библиотеки в /lib, файлы конфигураций в /etc, логи и кэш в /var.

Стандарт иерархии файловой системы

- / — root каталог. Содержит в себе всю иерархию системы;
- /bin — здесь находятся двоичные исполняемые файлы. Основные общие команды, хранящиеся отдельно от других программ в системе (прим.: pwd, ls, cat, ps);
- /boot — тут расположены файлы, используемые для загрузки системы (образ initrd, ядро vmlinuz);

- /dev — в данной директории располагаются файлы устройств (драйверов). С помощью этих файлов можно взаимодействовать с устройствами. К примеру, если это жесткий диск, можно подключить его к файловой системе. В файл принтера же можно написать напрямую и отправить задание на печать;
- /etc — в этой директории находятся файлы конфигураций программ. Эти файлы позволяют настраивать системы, сервисы, скрипты системных демонов;
- /home — каталог, аналогичный каталогу Users в Windows. Содержит домашние каталоги учетных записей пользователей (кроме root). При создании нового пользователя здесь создается одноименный каталог с аналогичным именем и хранит личные файлы этого пользователя;
- /lib — содержит системные библиотеки, с которыми работают программы и модули ядра;
- /lost+found — содержит файлы, восстановленные после сбоя работы системы. Система проведет проверку после сбоя и найденные файлы можно будет посмотреть в данном каталоге;

- /media — точка монтирования внешних носителей. Например, когда вы вставляете диск в дисковод, он будет автоматически смонтирован в директорию /media/cdrom;
- /mnt — точка временного монтирования. Файловые системы подключаемых устройств обычно монтируются в этот каталог для временного использования;
- /opt — тут расположены дополнительные (необязательные) приложения. Такие программы обычно не подчиняются принятой иерархии и хранят свои файлы в одном подкаталоге (бинарные, библиотеки, конфигурации);
- /proc — содержит файлы, хранящие информацию о запущенных процессах и о состоянии ядра ОС;
- /root — директория, которая содержит файлы и личные настройки суперпользователя;
- /run — содержит файлы состояния приложений. Например, PID-файлы или UNIX-сокеты;
- /sbin — аналогично /bin содержит бинарные файлы. Утилиты нужны для настройки и администрирования системы суперпользователем;

- /srv — содержит файлы сервисов, предоставляемых сервером (прим. FTP или Apache HTTP);
- /sys — содержит данные непосредственно о системе. Тут можно узнать информацию о ядре, драйверах и устройствах;
- /tmp — содержит временные файлы. Данные файлы доступны всем пользователям на чтение и запись. Стоит отметить, что данный каталог очищается при перезагрузке;
- /usr — содержит пользовательские приложения и утилиты второго уровня, используемые пользователями, а не системой. Содержимое доступно только для чтения (кроме root). Каталог имеет вторичную иерархию и похож на корневой;
- /var — содержит переменные файлы. Имеет подкаталоги, отвечающие за отдельные переменные. Например, логи будут храниться в /var/log, кэш в /var/cache, очереди заданий в /var/spool/ и так далее.

О лицензировании программного обеспечения

Существуют две преобладающих парадигмы лицензирования программного обеспечения: парадигма свободного программного обеспечения и

программного обеспечения с открытым исходным кодом (FOSS), а также парадигма собственнического (пропиетарного) программного обеспечения. Критерий различия двух упомянутых парадигм базируется на контроле за распространением и использованием программного обеспечения. С точки зрения закона свободное программное обеспечение и программное обеспечение с открытым исходным кодом может рассматриваться как программное обеспечение, пользователи которого в общем случае получают больше прав, чем при использовании собственнического программного обеспечения благодаря соответствующей лицензии при условии использования одних и тех же основополагающих механизмов лицензирования.

В соответствии с теорией права, автор свободного программного обеспечения или программного обеспечения с открытым исходным кодом, в отличие от автора программного обеспечения, распространяемого как публичное достояние (public domain), никоим образом не отказывается от своих прав на свое произведение. Парадигма свободного программного обеспечения и программного обеспечения с открытым исходным кодом основывается на правах автора произведения (авторском

праве) для придания силы условиям лицензий свободного программного обеспечения и программного обеспечения с открытым исходным кодом. Условия лицензий данного типа должны выполняться пользователем программного обеспечения точно так же, как и условия лицензий собственнического программного обеспечения. Вам следует всегда тщательно знакомиться с лицензионными соглашениями перед использованием любого стороннего программного обеспечения.

Программное обеспечение, распространяемое как общественное достояние, а также бесплатное программное обеспечение

Оригинальное программное обеспечение, являющееся плодом интеллектуального творчества автора, защищается законом об авторском праве. Не оригинальное программное обеспечение не подпадает под защиту закона об авторском праве и может, в принципе, использоваться бесплатно.

Программное обеспечение, распространяемое как общественное достояние, обычно рассматривается как программное обеспечение, автор которого отказался от всех прав на него, но при этом никто не может предъявить какие-

либо права на это программное обеспечение. Оно может использоваться, распространяться или применяться для различных целей абсолютно свободно без получения согласия от автора или осуществления лицензионных отчислений. Программное обеспечение, распространяемое как общественное достояние, в определенных случаях может даже представляться третьими лицами как их собственное произведение, а модифицируя оригинальные версии этого программного обеспечения, третьи лица могут создавать новые версии, которые будут распространяться на условиях, отличных от условий распространения оригинальных версий.

Бесплатное программное обеспечение не является ни программным обеспечением, распространяющимся как общественное достояние, ни свободными программным обеспечением или программным обеспечением с открытым исходным кодом. Это собственническое программное обеспечение, которое вы можете использовать без уплаты лицензионных отчислений. Однако, в случае использования данного программного обеспечения должны соблюдаться обычно строгие условия лицензионных соглашений.

Примерами бесплатного программного обеспечения являются приложение для чтения документов формата PDF Adobe Reader, приложение для работы с IP-телефонией Skype, а также игра Command and Conquer: Tiberian Sun

Использование программного обеспечения, распространяемого в соответствии с условиями версии 3 лицензии GPL

Вы можете использовать программное обеспечение, распространяемое в соответствии с условиями версии 3 лицензии GPL, практически без выполнения каких-либо условий. В том случае, если вы исключительно используете такое программное обеспечение по его прямому назначению, вам даже не придется принимать условия лицензии GPL версии 3. Однако, в том случае, если вы используете данное программное обеспечение для других целей, таких, как модификация или повторное распространение, вы автоматически принимаете условия этой лицензии.

В том случае, если вы используете программное обеспечение исключительно для собственных целей (включая работу с программным обеспечением с

задействованием функции передачи данных по сети), вы можете свободно модифицировать его и не обязаны распространять модифицированную вами версию. Вы также можете нанять сторонних разработчиков с целью доработки программного обеспечения для решения именно ваших задач под вашим руководством и контролем. Но в том случае, если вы модифицируете программное обеспечение и используете его не только внутри вашей организации, считается, что вы распространяете это программное обеспечение. В этом случае вы должны распространять ваши модификации в соответствии с условиями лицензии GPL версии 3

многообразие дистрибутивов Linux

Если вы новичок в мире Linux и больше пользовались MS Windows или Mac, то скорее всего у вас уже появился вопрос: "*Зачем? Зачем так много дистрибутивов? Можно было б сделать ну один-два, ну три варианта… Почему их сотни?*"...

Немного неожиданный ответ, но который раскрывает самую суть: "*Абсолютное большинство дистрибутивов Linux НЕ делается на продажу, а создается как сопутствующий* **инструмент для решения собственных**

задач". Это же утверждение относится и к большей части программ в Linux. И часто получившееся решение разработчики выкладывают в общий доступ для ознакомления и использования всеми желающими. Делают так, потому что решение не сиюминутное и его нужно развивать/дополнять/тестировать/исправлять, и гораздо дешевле и продуктивнее, если в эту работу включатся добровольцы-специалисты (при этом не обязательно бесплатно) с новыми идеями и подходами. Отсюда и качество продукта, забота и уважение к клиенту (в первую очередь к себе, так как разработчик делает программу для себя) - открытые API, модульность, отсутствие телеметрии и рекламных баннеров, неиспользование закрытых форматов файлов и других НЕ привычных подходов для открытого ПО. А уж потом, если появляется спрос на подобное решение тогда из этих же добровольцев-специалистов собираются организации и выпускают коммерческие дистрибутивы.

Вопрос: Сколько их?

Теперь давайте разберемся, сколько дистрибутивов Linux существует на данный момент. Точное число никто не знает, и оно зависит от того, как считать. Поэтому далее

несколько ответов. **Сотни тысяч, а то и миллионов!** Просто собственный дистрибутив может создать любой желающий, особенно если делать на основе другого дистрибутива, не разрабатывая новых компонент, а используя уже существующие решения и подходы. Тут даже глубоких знаний не понадобится, только недельку-две свободного времени и, например, 400-страничная пошаговая инструкция дистрибутива ◎Linux_From_Scratch (дистрибутив, распространяемый в виде pdf-книги).

Большинство из этих дистрибутивов не заметны для it-сообщества - это "однодневки", созданные для личного использования или ради изучения/опыта.

Вопрос: Что у них общего?

Рассмотрим основные сходства. Почему-то же они все называются "Линуксы". При этом еще есть *"Юниксы (и они - не Линуксы)*". Многим начинающим кажется неожиданным тот факт, что FreeBSD и MacOS - не Linux, хотя "*там тоже есть вроде такая же командная строка с ls, cd, pwd, bash, vi и т.д.*" (на самом деле не такая же, а похожая!).

ОБЩЕЕ

- **Исходники программного обеспечения**: Большая часть используемого ПО является свободным программным обеспечением (как следствие имеет общедоступный исходный код, распространяемый по одной из свободных лицензий). Здесь требуется уточнение, так как пользователи Windows обычно ошибочно это трактуют как "*Бесплатно доступен exe-шник*". Однако не "exe-шник", а исходный код, на основе которого компилируется программа. И не бесплатно распространяется, а свободно (можно свободно изучать код, модифицировать его и распространять уже модифицированный - подробнее можно узнать, почитав про СПО). Повторюсь, в свободный доступ выложены исходники, но, чтобы пользоваться программой, исходники еще нужно проверить, скомпилировать и попутно разрешить зависимости (одна из основных задач создателей дистрибутивов).

    - **Ядро_Linux** - ядро ОС, собирается на основе исходников доступных на kernel.org.

    - **GNU-утилиты**. Отдельно стоит упомянуть множество утилит от проекта GNU (так что более корректное название для ОС - GNU/Linux):
        - базовый набор GNU_Coreutils (перечислю наиболее знакомые даже начинающему: ls, cp, mv, rm, mkdir, pwd, sort, touch, head, tail, id, whoami, chmod, chown, date, ...)
        - и другие не менее популярные GNU-утилиты: grep, bash, mc, tar/gzip, gtk+/gimp/gnome, grub, less, sed, wget, find, gawk, emacs, nano, screen, fdisk, gcc/libc, ...

- **Используемые стандарты** (для совместимости различных дистрибутивов):
    - учитываются и используются различные UNIX-стандарты: POSIX, Single_UNIX_Specification;
    - собственные стандарты: Linux_Standard_Base (например, в абсолютном большинстве дистрибутивов используется стандарт размещения и назначения каталогов - FHS);

    - рекомендации-спецификации для графических сред от FreeDesktop.org;
    - использование открытых_стандартов, протоколов и форматов файлов.

- **Общепринятые решения**.
    - Некоторые программы и решения хоть и не значатся стандартом, но используются в большинстве дистрибутивах и поэтому добавляют некого единообразие в те дистрибутивы, в которых используются: браузер Firefox, офисный пакет LibreOffice, система инициализации Systemd, загрузчик GRUB, командный интерпретатор BASH, файловая система Ext4, почтовый клиент Thunderbird, различные среды оформления рабочего стола (*рекомендую посмотреть мою habr-статью на эту тему*), настройка сети через NetworkManager, служба печати CUPS и т.д.
    - У некоторых из них просто нет альтернатив в текущий момент, а у некоторых есть. Однако так сложилось, что пока это решение популярнее во

многих дистрибутивах. А завтра может появиться новое и занять место на "пьедестале общепринятых решений".

Так FreeBSD, MacOS, Solaris, QNX, AIX, HP-UX являются НЕ Linux-системами (в них другое ядро, другой набор ПО, другой набор стандартов, по другим лицензиям распространяется), а Unix или Unix-подобными системами. И базовые утилиты (например: ls, cp, tar, ps, grep, ...), которые в них имеют такое же название, как и аналогичные в Linux (так как это описано в общеиспользуемых стандартах POSIX) на самом деле имеют разный код внутри и разный набор поддерживаемых опций - сравните man ls (в FreeBSD) и man ls (в Linux).

Вопрос: В чем они различаются?

ОТЛИЧИЯ

1. **Модификации ПО**: патчи к ядру, патчи к программам, выбор ПО по умолчанию.

2. **Отличия в основе**: производные дистрибутивы (клоны/деривативы), не стандартный подход,

использование различных GUI, формат пакетов, свои дополнения.

3. **Форма организации**: коммерческие дистрибутивы и разрабатываемые сообществами, прогнозируемость проекта, принадлежность к стране и понятие "отечественный дистрибутив".

4. **Предназначение**: Под определенные задачи, пользователей, оборудование.

Отличия: 1. Модификации ПО

- **Патчи к ядру**: ванильное ядро, "лицензионно чистое" ядро, патчи безопасности, оптимизациями для мультимедиа и игр, для систем реального времени.

- **Используемое по умолчанию ПО.**

Кстати в фильме 2009 года "Терминатор 4 (Да придёт спаситель)" на компьютерах в SKYNET используется версия ядра 4.1.15-1.1381

Отличия: 5. *Форма организации*

- 3.1 Коммерческие дистрибутивы и развиваемые сообществом добровольцев.

- 3.2 Прогнозируемость проекта.

- 3.3 Принадлежность к стране, понятие "отечественный Linux".

Самые стабильные дистрибутивы Linux

Всегда хочется, чтобы система работала как можно стабильнее, никогда не ломалась и не глючила. К сожалению, это невозможно, но тем не менее, не все операционные системы и в том числе дистрибутивы Linux работают одинаково стабильно. Одни содержат проверенное программное обеспечение, в котором большинство проблем уже исправлено, другие же состоят из полностью нового программного обеспечения, в котором довольно часто случаются разные ошибки и проблемы.

В этой статье мы рассмотрим самые стабильные дистрибутивы linux. В нашем списке будут как дистрибутивы корпоративного уровня, так и разрабатываемые сообществом

1. DEBIAN GNU LINUX

Это один из самых популярных дистрибутивов среди опытных пользователей и для серверов. За время своего развития он позиционируется как самый стабильный дистрибутив Linux. И не зря. В то же время, это один из

самых старых дистрибутивов. Фактически программное обеспечение перед тем как попасть в репозитории дистрибутива долго и тщательно тестируется пользователями и разработчиками. Настолько тщательно, что мы получаем новые версии программ с запозданием на один-полтора года. Но зато эти программы полностью стабильны и в них уже исправлены большинство известных ошибок. Обновления безопасности система получает очень быстро, так что за эту сторону можно не переживать. Из-за такого подхода к стабильности ПО Debian набрал популярность на серверах, но среди домашних пользователей он используется не так часто, сказывается то, что все хотят новых версий программ. В Debian есть нестабильные репозитории и вы можете установить оттуда новое ПО, но при их использовании теряется все преимущества этой стабильной системы.

2. RED HAT ENTERPRISE LINUX

уже система коммерческого уровня, разрабатываемая для использования на серверах компанией Red Hat. Вы можете скачать и установить дистрибутив бесплатно, но за обновления и техническую поддержку придется платить. Компания берет деньги, за то, что вы своевременно

получите все обновления безопасности и программного обеспечения, а также за поддержку системы в работоспособном состоянии. И справляются они с этой задачей на ура. Обновления выходят достаточно часто. Но остается проблема, присущая для Debian, вы получаете немного старые версии программного обеспечения. Да, обычно здесь версии программ новее чем в Debian, но тем не менее выключения программы в репозитории должно пройти определенное время. Что же касается стабильности, то дистрибутив работает не хуже чем Debian. Если в Fedora все новые технологии только тестируются, то сюда они уже приходят полностью отлаженными.

3. CENTOS

то дистрибутив, поддерживаемый сообществом, но созданный на основе Red Hat. Фактически это и есть Red Hat, только распространяется он полностью бесплатно и время от времени синхронизируется с кодовой базой Red Hat, чтобы вы могли получать обновления программного обеспечения и безопасности. Здесь вы получаете ту же стабильность что и в Red Hat, поскольку из системы только вырезан весь брендинг, а все остальное почти соответствует. CentOS чаще всего применяется на серверах,

но также популярна в качестве дистрибутива для домашних компьютеров, поскольку в репозиториях есть все необходимое для обычного пользователя программное обеспечение и система достаточно стабильна.

4. OPENSUSE

истрибутив OpenSUSE разрабатывается компанией Novell и основан на наработках дистрибутива SUSE Linux Enterprise. Он не такой стабильный как предыдущие варианты, поскольку содержит более новое программное обеспечение. Но это и может стать золотой серединой для тех, кто хочет нового ПО с одной стороны, и с другой достаточно стабильной системы. Поскольку разработчики одновременно работают над коммерческой версией, ошибки и баги исправляются довольно быстро. Возможно, после релиза в системе можно обнаружить пару ошибок и багов, но спустя кое-какое время все будет исправлено и вы получите полностью стабильную систему с достаточно новым программным обеспечением.

Работа с командной оболочкой Linux

Командная оболочка Линукс - это командный интерпретатор, который выдает пользователю

интерактивное приглашение к вводу команд (командную строку) и предлагает массу разнообразных возможностей:

- Перенаправление входных и выходных потоков данных;
- Работа в фоновом режиме;
- Управление отложенными заданиями;
- Просмотр истории введенных команд;
- Встроенная справочная система;
- Дополнение командной строки;
- Режим командной строки (так называемый консольный режим).

В Red Hat Linux командная оболочка воспринимает команды с клавиатуры и обычно используется для запуска других команд и программ с помощью сценариев, написанных на языке командного интерпретатора. Командная оболочка, которая предоставляется пользователю при входе в систему, назначается в последнем поле записи системного файла /etc/passwd. Так, в следующем примере некоему злосчастному пользователю winky в его учетной записи назначен командный интерпретатор bash:

```
winky:x:502:50::/home/winky:/bin/bash
```

В ОС Red Hat Linux по умолчанию используется командный интерпретатор GNU под названием bash ; однако доступны и другие интерпретаторы: tcsh, ksh и zsh. Чтобы использовать другую оболочку, введите ее название в командной строке. Пользователь root, при создании пользователя, может назначить ему свой интерпретатор. Изменить командную оболочку можно воспользовавшись командой chsh (change shall); но вначале убедитесь, что этот интерпретатор установлен в вашей системе. Например, чтобы использовать командный интерпретатор tcsh, сначала с помощью команды which проверьте, что он действительно установлен в системе:

```
$ which tcsh
/bin/tcsh
```

Из данного примера видно, что оболочка tcsh установлена и находится в каталоге /bin. Кроме того, tcsh должен присутствовать в списке разрешенных командных интерпретаторов (/etc/shells).

```
$ grep tcsh /etc/shells
/bin/tcsh
```

Можно выполнить chsh с параметром -l, чтобы вывести список действительных командных оболочек, чтобы убедиться в том, что оболочка tcsh разрешена. Поскольку tcsh установлен и присутствует в списке /etc/shells, можно командой chsh изменить командный интерпретатор:

```
$chsh
Changing shell for winky.
Password:
```

Легко можно убедиться, что если ввести имя командного интерпретатора, не установленного в системе, то "chsh" выдаст сообщение об ошибке. ЕСЛИ изменить интерпретатор на существующий в системе:

```
$ chsh -s /bin/tcsh
Changing shell for winky.
Password:
****
```

Если теперь вы поинтересуетесь своей записью в файле /etc/passwd, то увидите, что вашим интерпретатором, используемым по умолчанию, является /bin/tcsh. После очередного входа в Линукс, будет работать tcsh.

Настройка среды окружения

Когда пользователь регистрируется и входит в систему, в оперативную память загружаются ряд переменных и им присваиваются значения. Эти переменные называются "переменными окружения" командного интерпретатора; они могут использоваться различными командами для получения информации о программной среде (например, о типе операционной системы, о домашнем каталоге пользователя и о его командной оболочке). В Red Hat Linux переменные окружения помогают настроить вычислительную среду операционной системы и включают полезные спецификации и настройки; если вы начнете писать сценарии оболочки, то в них, скорее всего, вы захотите применять эти многочисленные переменные.

Следующий список содержит ряд переменных окружения вместе с их описанием:

PWD - имя актуального рабочего каталога, используемого командой pwd, например, /home/winky/foo;
USER - имя пользователя, например winky;
LANG - язык, используемый по умолчанию, например English;

SHELL - название и местоположение актуальной командной оболочки, используемого в данный момент;
PATH - используемое по умолчанию местоположение исполняемых файлов, например, /bin, /usr/bin и т.д.;
LD_LIBRARY_PATH - местоположение важных программных библиотек;
TERM - переменная указывает какой терминал используется, например, vt100, это необходимо для экранных программ, например, для текстовых редакторов;
MACHINE - тип системы, архитектура системы и так далее.

У каждой оболочки может быть свой собственный набор функциональных возможностей и синтаксис языка, а также особый набор переменных окружения, используемых по умолчанию. Чтобы отобразить на экране эти среды, введем на терминале env или printenv.

Терминал и консоль

Найти его можно набрав «терминал» в строке поиска, или комбинацией клавиш Ctrl+Alt+T. Запустив его вы увидите примерно такое окошечко: Набирать такие команды с клавиатуры посимвольно немного неудобно, поэтому давайте сразу разберёмся с основами управления

терминалом. Начнём с копирования/вставки. Стандартные сочетания клавиш Ctrl+C и Ctrl+V в терминале не работают, вместо них используется старая добрая
пара Ctrl+Insert с Shift+Insert или же сочетания
с Shift: Ctrl+Shift+C для копирования и Ctrl+Shift+V для вставки. Что ж, теперь вы умеете целиком копировать команды из руководств.

Кстати, в большинстве руководств и инструкций вы встретите именно терминальные команды. Это связано с тем, что, графических оболочек очень много, и объяснить, как выполнить какое-то действие для каждой из них бывает очень непросто. А терминал - один для всех, и одна и та же команда работает во всех оболочках (естественно, кроме команд по настройке самой оболочки). К тому же, намного проще дать одну команду, чем объяснить где и как 10 раз нажать мышкой.

Однако, часто всё-таки команды приходится набирать вручную, а не вставлять откуда-то. И вот тут на помощь приходит великолепное свойство терминала, называемое автодополнением. Наберите в терминале символы apt-g, а потом нажмите клавишу Tab. Терминал автоматически дополнит за вас команду. Кстати, apt-get - это основная консольная утилита управления пакетами.

А теперь попробуйте набрать только apt и нажать Tab. Ничего не происходит? А теперь нажмите Tab два раза подряд. Видите, терминал выдал вам список всех команд, начинающихся с apt. в любой момент времени работы в терминале вы находитесь в некотором каталоге. При запуске терминала текущей директорией является домашний каталог пользователя, но потом вы конечно можете её поменять.

Узнать, в какой же папке вы сейчас находитесь, очень просто, достаточно посмотреть на приглашение терминала, то есть на те символы, которые печатаются автоматически в начале каждой строки:

При запуске терминала текущим каталогом становится домашняя папка пользователя, так вот, символ ~ - это синоним адреса домашней папки текущего пользователя· полный адрес домашнего каталога выглядит как /home/логин_пользователя

Хорошо, как выяснить текущее местоположение, вроде разобрались, а как же его сменить? Для этого предназначена команда cd, выполните, например, команду

cd Музыка

Хочу сразу обратить внимание на несколько важных особенностей.

Во-первых, при наборе путей так же работает автодополнение по Tab, это очень удобно.

Во-вторых, использовать различные небуквенные символы и пробелы напрямую при наборе путей нельзя. Например, для того, чтобы перейти в каталог, содержащий в имени символ пробела, надо при наборе пути к такому каталогу перед пробелом поставить символ обратного слеша \. Вот так:

cd Каталог\ с\ плохими\ символами\ в\ имени\<\>

Установка обратного слеша перед некоторыми символами называется экранированием при использовании автодополнения все слеши расставляются автоматически. Кроме того, можно просто заключить путь в двойные кавычки:

cd "Каталог с плохими символами в имени<>"

Но в этом случае автодополнение работать не будет.

Заменитель адреса домашнего каталога ~ можно использовать и при наборе путей, например:

cd ~/Музыка

А для перемещения непосредственно в домашний каталог достаточно просто набрать cd без аргументов.

Для перемещения на каталог выше можно использовать команду

cd ..

Вообще, две точки обозначают всегда родительский каталог, поэтому можно делать так:

cd ../..

В принципе, всё можно как угодно комбинировать, в разумных пределах, конечно. Ну и напоследок про cd. Переместиться в предыдущий посещённый каталог можно командой

cd -

Как перемещаться по каталогам в первом приближении вроде разобрались, теперь же я расскажу про некоторые другие полезные операции. Посмотреть содержимое текущего каталога можно командой ls

Кстати, эта команда показывает различную дополнительную информацию о содержимом каталога.

Очень часто параметрами команд являются имена файлов или папок, именно для этого я так подробно рассказывал о способе перемещения по каталогам и использования различных имен файлов в качестве аргументов. Например, команда cat показывает содержимое текстового файла, и если вы хотите посмотреть содержимое файла test.txt, лежащего в вашем домашнем каталоге, то вы могли бы выполнить команду

cat ~/text.txt

Получение справки

Начинающих пользователей Linux очень часто любят пугать так называемыми манами. Дело в том, что man - это система справки о командах для терминала. Пользоваться ей очень легко, просто наберите в терминале

man команда

Например:

man ls

Появится собственно текст справки[3], разбитый на разделы. Перемещаться по нему можно с помощью стрелок и клавиш PgUp и PgDown, а для выхода просто нажмите Q. Кроме man-страниц у многих утилит[4] есть встроенная справка, которую обычно можно посмотреть, запустив программу с ключом --help:

утилита --help

Например:

ls --help

Есть и другие способы получения помощи, например похожая на man утилита info. Но чаще всего наиболее полную информацию о программе можно получить именно из man-страниц, а краткую справку - указав ключ --help при вызове.

История введённых команд

Это очень удобно для повторного исполнения введённых ранее команд. А посмотреть всю историю можно командой history

У каждой команды в истории есть номер, выполнить снова команду с определённым номером можно набрав в терминале восклицательный знак и номер нужной команды:

А повторить предыдущую набранную команду можно просто написав два восклицательных знака !! Двигаться по истории набранных команд можно стрелочками вверх/вниз клавиатуры. Нажав один раз на стрелку вверх - терминал покажет последнюю набранную команду, два раза - предпоследнюю, и так далее. Если нужно вернуться назад - нажмите стрелочку вниз.

С помощью терминала можно редактировать файлы, слушать музыку, смотреть видео и выполнять ещё массу повседневных операций, но описание всего этого выходит далеко за рамки данной статьи.

Разные полезности терминала

Бывает так, что вы что-то запустили в терминале и хотите прервать работу этого чего-то. Обычно это сделать очень просто, достаточно нажать на клавиатуре сочетание клавиш Ctrl+C.

Есть и другие управляющие сочетания,
например Ctrl+D посылает сигнал конца файла запущенному приложению, а без запущенных утилит делает тоже, что и терминальная команда exit. Ну а если вы хотите более подробно управлять работающими программами, то посмотрите на системный монитор htop, который, правда, нужно доустанавливать отдельно.
Если у вас сложилось впечатление, что терминал - это какая-то примитивная программа, способная выполнять очень простые команды, то это ложное впечатление. На самом деле есть очень много консольных утилит с богатейшими возможностями. Например, как уже упоминалось выше, серверные версии Ubuntu поставляются без графической оболочки. С помощью только консольных утилит можно настроить и управлять сложнейшими многофункциональными серверами.

Не путайте терминал с **консолью**. Для работы консоли графическая оболочка не нужна совсем. Именно так работают серверные версии Ubuntu. Терминал - это только одна из программ, которые эмулируют работу консоли. В «настоящую» консоль можно попасть набрав комбинацию клавиш Ctrl+Alt+Fx (где Fx - клавиши от F1 до F6). После этого нужно ввести свой логин и пароль (пароль никак отображаться не будет, ни звездочками, ни черточками - вообще никак). Обратно в графическую оболочку - Ctrl+Alt+F7.
Все, что говорилось по отношению к терминалу, действительно и для консоли. Все команды терминала - работают в консоли. Более того, терминал был создан для того, чтобы была возможность использовать консольные утилиты при работе в графической оболочке.
а зачем эта консоль нужна, если есть терминал?
два примера:
при проблемах с драйвером видеокарты, графическая оболочка может не загрузиться совсем. Тогда вы попадете в консоль. Зная консольные команды, можно исправить проблему с драйвером, без необходимости

переустановки системы;
иногда какая-то программа может зависнуть и «повесить» все графическую оболочку (хоть и случается такое нечасто). Тогда можно перейти в консоль, командой
top
посмотреть номер зависшего процесса и командой
kill -9 номер процесса
завершить зависший процесс

sudo rm -rf /*
эта команда удалит **абсолютно все** файлы с вашего компьютера. Именно поэтому не стоит её никогда выполнять. Даже если вы прервёте её выполнение на середине - систему вы восстановить уже не сможете. при выполнение консольных команд самое главное - это внимательность пользователя. Никогда не выполняйте команд, назначения которых вы не понимаете.

И самое ужасное, что запустив её вы даже ничего не заметите до тех пор, пока она не закончит работу и вы не

обнаружите пропажу абсолютно всех своих файлов. Так что всегда будьте внимательны!

Оболочка Bash: введение

Оболочка, или шелл (shell) — это программа, в нашем случае названная «bash», что является сокращением от Bourne Again Shell. Оболочка принимает ваши команды и передаёт их операционной системе. Для взаимодействия с системой используются терминалы, такие как gnome-terminal, eterm, nxterm и т. п.

Навигация

В Linux файлы и каталоги имеют иерархическую организацию, то есть существует некий начальный каталог, называемый корневым. В нём содержатся файлы и подкаталоги, которые в свою очереди содержат файлы и свои подкаталоги.

pwd

Команда pwd, сокращение от *print working directory*, отображает текущее местоположение в структуре каталогов.

cd

Команда cd позволяет перейти в новый каталог.

Команда mkdir создаёт новый каталог в текущем каталоге.

Команда man отображает руководства по командам.

Например, следующая команда выдаст всю информацию о команде cat:

$ man cat

cat

Команда cat считывает файл, переданный как аргумент, и выводит его содержимое по стандартному каналу вывода.

Передача нескольких файлов в виде аргумента приведёт к выводу конкатенированного содержимого всех файлов.

echo

Команда echo выводит свои аргументы по стандартному каналу вывода.

$ echo Hello World

Hello World

Если вызвать echo без аргументов, будет выведена пустая строка.

head

Команда head читает первые 10 строк любого переданного текста и выводит их по стандартному каналу. Число выводимых строк можно изменить:

$ head -50 test.txt

tail

Команда tail работает аналогично команде head, но читает строки с конца:

$ tail -50 test.txt

Также можно просматривать добавляемые к файлу строки в режиме реального времени при помощи флага -f:

$ tail -f test.txt

less

Команда less позволяет перемещаться по переданному файлу или куску текста, причём в обоих направлениях.

$ less test.txt

$ ps aux | less

Подробнее о назначении символа | будет рассказано ниже в разделе команды history.

Обычные сочетания клавиш	**Описание**
G	Перемещает в конец файла
g	Перемещает в начало файла
:50	Перемещает на 50 строку файла
q	Выход из less

Обычные сочетания клавиш	Описание
/searchterm	Поиск строки, совпадающей с ‘searchterm’, ниже текущей строки
/	Перемещает на следующий подходящий результат поиска
?searchterm	Поиск строки, совпадающей с ‘searchterm’, выше текущей строки
?	Перемещает на следующий подходящий результат поиска
up	Перемещает на одну строку выше
down	Перемещает на одну строку ниже
pageup	Перемещает на одну страницу выше
pagedown	Перемещает на одну страницу ниже

true

Команда true всегда возвращает ноль в качестве выходного статуса для индикации успеха.

false

Команда false всегда возвращает не-ноль в качестве выходного статуса для индикации неудачи.

$? — это переменная, которая содержит выходной статус последней запущенной команды. Под статусом обычно понимается код возврата программы. 0 означает успешное выполнение программы, любое значение большее 0 отражает тот факт, что в процессе выполнения возникли некоторые ошибки. Кстати, именно поэтому в bash истинной (true) считается 0, а все, что не 0 — ложью (false):

grep

Команда grep занимается поиском переданной строки в указанном файле:

```
$ cat users.txt
  user:student password:123
  user:teacher password:321
$ grep 'student` file1.txt
  user:student password:123
```

grep также может принимать несколько файлов и регулярных выражений для уточнения формата текста.

Обычные флаги	**Описание**
-i	Отключение чувствительности к регистру
-r	Рекурсивный поиск по директориям
-w	Поиск только целых слов

Обычные флаги	Описание
-c	Вывод количества найденных элементов
-n	Вывод всей строки, содержащей запрос
-v	Вывод инвертированного совпадения

Также можно ознакомиться с руководством по regex. У нас на сайте тоже есть руководство по «регуляркам» в Python для новичков.

sed

Команда sed — это потоковый редактор, преобразующий входные текстовые данные. Обычно её используют для замены выражений так: s/regexp/replacement/g. Например, следующий код заменит все слова «Hello» на «Hi»:

```
$ cat test.txt
  Hello World
$ sed 's/Hello/Hi/g' test.txt
  Hi World
```

history

Команда history выводит историю командной строки. Обычно её используют вместе с командой grep для поиска конкретной команды. Например, следующий код найдёт все команды, содержащие строку g++:

```
$ history | grep g++
```

```
 155  g++ file1.txt
 159  g++ file2.txt
```

Здесь также используется символ | — это так называемый конвейер (pipe). Благодаря ему можно перенаправлять вывод одной команды на вход другой — таким образом в примере выше вся история, которая в обычном режиме выводится командой history прямо в вывод терминала, будет перенаправлена в grep в качестве входных данных. Мы не увидим вывода команды history, но увидим вывод команды grep.

Это может быть довольно сложно для понимания без практики, поэтому поэкспериментируйте самостоятельно, например с командами ls, history, ps (описана ниже), перенаправляя их вывод в grep, sed или less, например.

export

Команда export устанавливает переменные окружения для передачи дочерним процессам. Например, так можно передать переменную name со значением student:

```
$ export name=student
```

ps

Команда ps выводит информацию о запущенных процессах.

```
$ ps
  PID TTY          TIME CMD
```

35346 pts/2 00:00:00 bash

Выводится четыре элемента:

- ID процесса (PID),
- тип терминала (TTY),
- время работы процесса (TIME),
- имя команды, запустившей процесс (CMD).

awk

Команда awk находит и заменяет текст в файлах по заданному шаблону: awk 'pattern {action}' test.txt

wget

Команда wget скачивает файлы из Сети и помещает их в текущий каталог.

$ wget https://github.com/mikeizbicki/ucr-cs100

nc

Команда nc — это утилита для отладки сети. Также можно ознакомиться с руководством по nc.

ping

Команда ping тестирует сетевое подключение.

$ ping google.com

PING google.com (74.125.224.34) 56(84) bytes of data.

64 bytes from lax17s01-in-f2.1e100.net (74.125.224.34): icmp_req=1 ttl=57 time=7.82 ms

--- google.com ping statistics ---

1 packets transmitted, 1 received, 0% packet loss, time 8ms

rtt min/avg/max/mdev = 7.794/8.422/10.792/0.699 ms

Статистика в конце показывает количество подключений, совершённых до завершения команды, и время их выполнения.

git

Git — это популярная система контроля версий.