

Linux

Существует многообразие дистрибутивов Linux. Каждый дистрибутив обладает особенными характеристиками, которые оптимизированы под конкретные потребности пользователя или требования проекта. То есть большинство дистрибутивов не делаются на продажу, а создается как сопутствующий инструмент для решения собственных задач.

Все дистрибутивы отличаются друг от друга настольными средами, менеджерами пакетов, дисплей сервером, целями и задачами.

Одна общая черта всех дистрибутивов Linux - ядро Linux, то есть ядро ОС, которое соединяет ПО системы с базовым оборудованием.

Структура каталогов LINUX

В Linux нет привычных для Windows-пользователя C:\ , но есть каталоги “ / ” и каталоги с трехбуквенными именами.

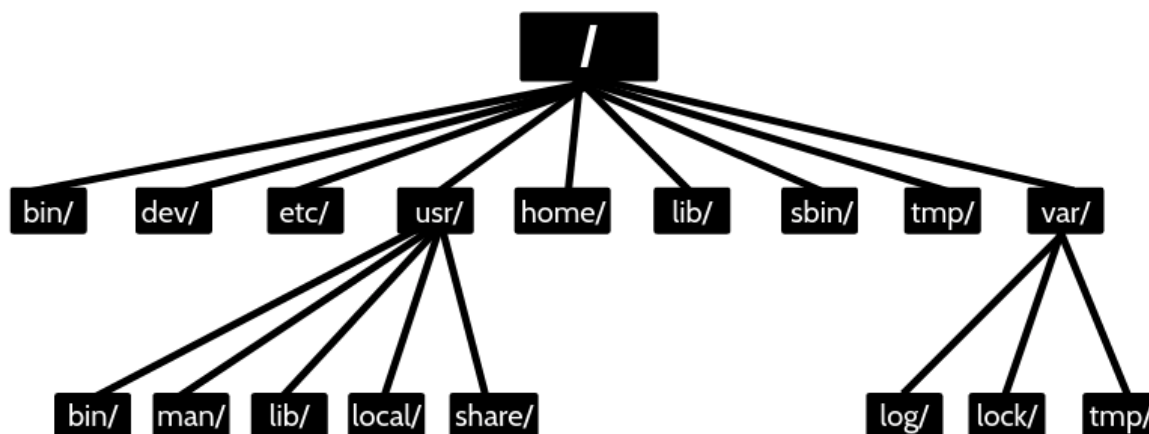
Стандарт иерархии файловой системы (**FHS - Filesystem Hierarchy Standard**) определяет структуру файловых систем в Linux и других UNIX-подобных операционных системах. Однако файловые системы Linux также содержат некоторые каталоги, которые еще не определены стандартом.

Основные типы содержимого, хранящегося в файловой системе Linux:

- Persistent (Постоянный) - это содержимое, которое должно быть постоянным после перезагрузки, например, параметры конфигурации системы и приложений.
- Runtime (Время выполнения) - контент, созданный запущенным процессом, обычно удаляется после перезагрузки.
- Variable / Dynamic (Переменный / динамический) - это содержимое может быть добавлено или изменено процессами, запущенными в системе Linux.
- Static (Статический контент) - остается неизменным до тех пор, пока не будет явно отредактирован или перенастроен.

/ - ROOT - Корневой каталог

Все в системе Linux находится в каталоге “ / ”.



/BIN - основные пользовательские двоичные файлы

Приложения, например такие как браузер Firefox, хранятся в `/usr/bin`, а важные системные программы и утилиты, такие как оболочка **bash**, находятся в `/bin`. Каталог `/usr` может храниться в другом разделе - размещение этих файлов в каталоге `/bin` гарантирует, что в системе будут эти важные утилиты, даже если другие файловые системы не смонтированы.

`/bin` непосредственно содержит исполняемые файлы многих основных команд оболочки, таких как `ps`, `ls`, `ping`, `grep`, `cp`.

Каталог `/sbin` аналогичен - он содержит важные двоичные файлы системного администрирования. `/sbin` содержит `iptables`, `reboot`, `fdisk`, `ifconfig`, `swapon`.

/BOOT - Статические загрузочные файлы

Каталог `/boot` содержит файлы, необходимые для загрузки системы - например, здесь хранятся файлы загрузчика **GRUB** и ваши ядра Linux. Однако файлы конфигурации загрузчика не находятся здесь - они находятся в `/etc` вместе с другими файлами конфигурации.

/DEV - Файл устройства

Linux представляет устройства в виде файлов, а каталог `/dev` содержит ряд специальных файлов, представляющих устройства. Это не настоящие файлы в том виде, в каком мы их знаем, но они отображаются как файлы например, `/dev/sda` представляет собой первый диск **SATA** в системе. Второй диск будет называться `/dev/sdb`. Если вы хотите его разбить, вы можете

запустить редактор разделов и указать ему отредактировать `/dev/sda`. В итоге получим что первым разделом этого диска будет `/dev/sda1`, а вторым `/dev/sda2`.

Этот каталог также содержит псевдоустройства, которые представляют собой виртуальные устройства, которые на самом деле не соответствуют оборудованию. Например, `/dev/random` производит случайные числа. `/dev/null` - это специальное устройство, которое не производит вывода и автоматически отбрасывает весь ввод - когда вы перенаправляете вывод команды на `/dev/null`, вы отбрасываете его.

/ETC - файлы конфигурации

Каталог `/etc` содержит файлы конфигурации, которые обычно можно редактировать вручную в текстовом редакторе. Каталог `/etc/` содержит общесистемные файлы конфигурации (например имя хоста) - пользовательские файлы конфигурации находятся в домашнем каталоге каждого пользователя.

/HOME - домашние папки

Каталог `/home` содержит домашнюю папку для каждого пользователя. Например, если ваше имя пользователя - bob, у вас есть домашняя папка, расположенная в `/home/bob`. Эта домашняя папка содержит файлы данных пользователя и пользовательские файлы конфигурации. Каждый пользователь имеет право записи только в свою домашнюю папку и должен получить повышенные права (стать пользователем **root**) для изменения других файлов в системе.

/LIB - основные общие библиотеки.

Каталог `/lib` содержит библиотеки, необходимые для основных двоичных файлов в папке `/bin` и `/sbin`. Библиотеки, необходимые для двоичных файлов в папке `/usr/bin`, находятся в `/usr/lib`.

Имена файлов библиотеки: `ld*` или `lib*.so.*`.

/LOST + FOUND - восстановленные файлы

В каждой файловой системе Linux есть каталог `/lost+found`. В случае сбоя файловой системы проверка файловой системы будет выполнена при следующей загрузке. Любые найденные поврежденные файлы будут помещены в каталог `lost+found`, чтобы вы могли попытаться восстановить как можно больше данных.

/MEDIA - съемный носитель

Каталог `/media` содержит подкаталоги, в которых монтируются съемные носители, вставленные в компьютер. Например, когда вы вставляете диск в свою систему Linux, внутри каталога `/media` автоматически создается каталог. Вы можете получить доступ к содержимому диска внутри этого каталога.

/MNT - временные точки монтирования

Исторически сложилось так, что каталог `/mnt` - это то место, где системные администраторы монтируют временные файловые системы во время их использования. Например, если вы монтируете раздел Windows для выполнения некоторых операций по восстановлению файлов, вы можете подключить его в `/mnt/windows`. Однако вы можете монтировать другие файловые системы в любом месте системы.

/OPT - дополнительные пакеты

Каталог `/opt` содержит подкаталоги для дополнительных пакетов программного обеспечения. Он обычно используется проприетарным программным обеспечением, которое не подчиняется стандартной иерархии файловой системы - например, проприетарная программа может выгружать свои файлы в `/opt/application` при ее установке.

/PROC - файл ядра и процессов

Каталог `/proc` похож на каталог `/dev`, потому что он не содержит стандартных файлов. Он содержит специальные файлы, которые представляют информацию о системе и процессе.

Тут можно получить текстовую информацию о системных ресурсах. Например узнать аптайм `/proc/uptime`, проверить информацию о процессоре `/proc/cpuinfo` или проверить использование памяти вашей системой Linux `/proc/meminfo`.

/ROOT - корневой домашний каталог

Каталог `/root` - это домашний каталог пользователя **root**. Вместо того, чтобы находиться в `/home/root`, он находится в `/root`. Он отличается от `/`, который является корневым каталогом системы, важно не путать их.

/RUN - файлы состояния приложения

Каталог `/run` является довольно новым и предоставляет приложениям стандартное место для хранения необходимых им временных файлов, таких как сокеты и идентификаторы процессов. Эти файлы нельзя хранить в `/tmp`, потому что файлы в `/tmp` могут быть удалены.

/sbin - двоичные файлы системного администрирования

Каталог `/sbin` аналогичен каталогу `/bin`. Он содержит важные двоичные файлы, которые обычно предназначены для запуска пользователем `root` для системного администрирования.

/SELINUX - виртуальная файловая система Selinux

Если ваш дистрибутив Linux использует **SELinux** для обеспечения безопасности (например, Fedora и Red Hat), каталог `/selinux` содержит специальные файлы, используемые SELinux. Это похоже на `/proc`. Ubuntu не использует SELinux, поэтому наличие этой папки в Ubuntu кажется ошибкой.

/SRV - сервисные данные

Каталог `/srv` содержит «данные об услугах, предоставляемых системой». Если вы использовали HTTP-сервер Apache для обслуживания веб-сайта, вы, вероятно, сохранили бы файлы своего веб-сайта в каталоге внутри каталога `/srv`.

/TMP - временные файлы

Приложения хранят временные файлы в каталоге `/tmp`. Эти файлы обычно удаляются при перезапуске вашей системы и могут быть удалены в любое время с помощью таких утилит, как `tmpwatch`.

/USR - пользовательские двоичные файлы и данные только для чтения

Каталог `/usr` содержит приложения и файлы, используемые пользователями, в отличие от приложений и файлов, используемых системой. Например, второстепенные приложения расположены в каталоге `/usr/bin` вместо каталога `/bin`, а второстепенные двоичные файлы системного администрирования расположены в каталоге `/usr/sbin` вместо каталога `/sbin`. Библиотеки для каждого из них находятся в каталоге `/usr/lib`.

Каталог `/usr` также содержит другие каталоги - например, файлы, не зависящие от архитектуры, такие как графика, находятся в `/usr/share`.

Каталог `/usr/local` - это место, куда по умолчанию устанавливаются локально скомпилированные приложения - это не позволяет им испортить остальную часть системы.

/VAR - файлы переменных данных

`/var` это место, где программы хранят информацию о времени выполнения, такую как системный журнал, отслеживание пользователей, кэши и другие файлы, которые системные программы создают и управляют.

Каталог `/var` является записываемым аналогом каталога `/usr`, который при нормальной работе должен быть доступен только для чтения. Файлы логов и все остальное, что обычно записывается в `/usr` во время нормальной работы, записывается в каталог `/var`. Например, вы найдете файлы логов в `/var/log`. Помимо логов тут можно найти пакеты и файлы базы данных `/var/lib`, электронные письма `/var/mail`, очереди печати `/var/spool`, файлы блокировки `/var/lock`, временные файлы, необходимые при перезагрузке `/var/tmp`.

Распространение Linux

Система **Linux** распространяется на условиях лицензии особого рода, известной под названием GNU-лицензии (GNU General Public License, или GPL). GNU-лицензия была разработана для проекта GNU организацией Free Software Foundation, и эта лицензия устанавливает несколько условий на распространение и изменение свободного программного обеспечения.

Прежде всего, нужно объяснить, что свободное программное обеспечение, на которое распространяется GNU-лицензия, не является общедоступным (public domain) программным обеспечением. По определению, общедоступные программы принадлежат обществу и не защищаются никакими авторскими правами. Напротив, программы, на которые распространяется GNU-лицензия, защищены авторским копирайтом. У таких программ есть автор в юридическом понимании этого слова, и права авторов этих программ защищаются обычными международными законами, относящимися к авторскому праву. Таким образом, GNU-лицензия делает программное обеспечение свободно распространяемым, но не передаёт его в пользование обществу.

Также GNU-лицензия позволяет получать, модифицировать и распространять собственные модифицированные версии программного обеспечения. Однако на все продукты, разработанные на основе программ, защищённых GNU-лицензией, также должна быть поставлена GNU-лицензия. Другими словами, организация не может взять систему **Linux**, изменить её и начать продавать, поставив на неё лицензию, которая будет ограничивать распространение. Если некоторое программное обеспечение получено с помощью модификации программного обеспечения системы **Linux**, на него также должна быть поставлена GNU-лицензия.

GNU-лицензия позволяет распространять и использовать свободное программное обеспечение бесплатно. Но она также позволяет частным лицам и организациям распространять эти программы за плату и даже извлекать

прибыль из продажи и распространения. При этом распространитель программного обеспечения, защищённого GNU-лицензией, не может отобрать эти же права у покупателя. Если вы купили это программное обеспечение у некоторого третьего лица, вы можете его как самостоятельно продавать, так и распространять бесплатно.

Сказанное выше может казаться противоречивым. Зачем, казалось бы, продавать программное обеспечение, когда GNU-лицензия позволяет получить его бесплатно? Представим себе однако, что некоторая организация решила собрать большое количество свободного программного обеспечения, записать его на лазерный диск и распространять этот диск. Для того, чтобы покрыть расходы по производству и распространению лазерных дисков, компания будет вынуждена требовать некоторую оплату, и может даже извлечь из этой деятельности прибыль. Подобное не запрещается GNU-лицензией.

Организации, которые продают свободное программное обеспечение, должны следовать некоторым ограничениям, которые накладывает GNU-лицензия. Продавцы не могут ограничивать права пользователей, которые покупают программное обеспечение. Если вы купили диск CD-ROM, который содержит программы, защищённые GNU-лицензией, вы можете копировать программы с этого диска и распространять их бесплатно, или же сами можете их продавать. Продавцы программ должны чётко объяснить покупателям, что программы защищены GNU-лицензией. Продавцы также обязаны бесплатно распространять полный набор исходных текстов распространяемого программного обеспечения. Это позволяет каждому покупателю программного обеспечения, защищённого GNU-лицензией, производить модификации этих программ.

Такой подход, когда организации разрешено распространять и продавать свободное программное обеспечение, даёт положительные результаты. Не у каждого пользователя есть доступ к Интернету и возможность скопировать через него программное обеспечение. Кроме того, большое количество организаций продаёт систему **Linux** на дискетах, лентах и лазерных дисках по заказам через почту и от этих продаж получает прибыль. Разработчики системы **Linux** могут никогда не увидеть этих денег; когда на программное обеспечение накладывается GNU-лицензия, это понимают и автор, и распространитель программ. Другими словами, Линус Торвалдс знал, что организации могут захотеть продавать его систему **Linux**, и что он за это может не получить ни цента, и сознательно шёл на это.

Дело в том, что в мире свободного программного обеспечения важны не деньги. Цель создания и распространения свободных программ всегда будет состоять в том, чтобы дать всем возможность получать и использовать хорошие программы.

Источник: <https://docs.altlinux.org/ru-RU/archive/2.4/html-single/master/alt-docs-extras-linuxcovice/ch01s03.html>

Многопользовательский режим в Linux

Операционная система считается «многопользовательской», если позволяет нескольким людям использовать компьютер и не влияет на «вещи» друг друга (файлы, настройки и т. д.). В Linux несколько человек могут даже использовать компьютер одновременно.

Самые стабильные дистрибутивы Linux

1. Debian GNU Linux - один из самых старых и стабильных дистрибутивов. Программное обеспечение перед тем как попасть в репозитории дистрибутива долго и тщательно тестируется пользователями и разработчиками.
2. Red Hat Enterprise Linux - Это 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.
3. Centos - Это 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. Часто 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 - Дистрибутив 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 может стать золотой серединой для тех, кто хочет новое ПО с одной стороны, и с другой достаточно стабильную систему. 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.

Командная оболочка - SHELL

Shell или командная оболочка - это программа, которая организывает среду для выполнения других программ и команд. Командная оболочка имеет свои встроенные команды, арифметические операторы и другие синтаксические выражения, но основная её задача упрощать запуск других программ. Именно командная оболочка занимается поиском программ в текущем каталоге и в путях, указанных в переменной среды PATH, управляет сменой текущего каталога и переменными окружения. Таким образом, основная задача оболочки - интерпретировать команды пользователя и выполнять их независимо от того, внешние ли это программы или внутренние.

Чтобы понять с чем именно вы имеете дело, командой оболочки или внешней программой, можно воспользоваться командой "Whereis". Например, команда "cd" это не программа, а встроенная команда оболочки. А команда "ls" - это внешняя программа, исполняемый файл которой расположен в файловой системе.

Управление консолью Linux

Консоль позволяет управлять всей системой посредством коротких команд, причем, можно как устанавливать программы, так и изменять оболочки, что экономит много времени.

Часто пользователю необходимо запрашивать привилегии суперпользователя у системы, чтобы осуществить установку или изменение каких-либо файлов. Для этого системой предусмотрен механизм предоставления необходимого количества привилегий "sudo". Эта команда вводится перед другими и требует ввести пароль пользователя для получения разрешения. В целях безопасности консоль не отображает вводимый пароль, но на самом деле он вводится.

Помощь с работой в консоли можно получить двумя способами:

1. Функции любой команды можно посмотреть, используя команду `man <команда>`, что откроет мануал;
2. Также помощь можно получить в утилитах `ls --help ..`

Команды консоли

При помощи консоли, используя всего одну команду можно удалить все файлы (`sudo rm -rf /`) в системе, поэтому с командной строкой нужно быть очень осторожным и не использовать команды, назначения которых вы не понимаете.

Из командной строки можно изменять права доступа `sudo chmod`, посмотреть версию дистрибутива `cat /etc/*release`, посмотреть диски в системе `fdisk -l` или `sudo fdisk -l`, установить или удалить программы `sudo apt install -y name` (установить пакет), `sudo apt remove -y name` (удалить пакет), `sudo apt purge -y name` (удалить пакет вместе с его конфигурационными файлами), `sudo apt -y autoremove` (удалить явно указанные неиспользуемые зависимости). `-y` - позволяет соглашаться со всеми вопросами установщика, который часто выдает множество ненужных уточняющих деталей установки (типа “приложение займет столько-то места на диске”). Также можно установить сразу несколько пакетов, используя одну команду `sudo apt install -y name1 name2`
...

Базовые команды для перемещения и управления файлами:

- `pwd` - показывает директорию, в которой находится пользователь
- `cd` - переходит в указанную папку (`cd /home/$USER`)
- `ls` - выводит список файлов и папок в текущей директории. Более подробную информацию о директории или папке можно получить, добавив `-l` (показывает права доступа) или `-a` (показывает скрытые файлы).
- `du *` - считает размер директории (`du -hs *` или `du -hc /` путь)
- `find` - поиск файлов и директорий
- `cat` - выводит содержимое файла, также можно перезаписать или склеить файлы, используя “>” или “>>”
- `mkdir` - создает директорию
- `cp` - копирует файл/папку
- `mv` - перемещает или переименовывает каталоги и файлы

- `rm` - удаляет файл/папку

BASH

Это командная оболочка для UNIX-подобных ОС (UNIX, GNU/Linux, MacOS). Она дает пользователю систему команд для работы с файлами и папками, поиском, настройкой окружения и позволяет управлять ОС прямо из командной строки.

Что именно делает bash? Эта оболочка принимает команды, которые вводит пользователь в командную строку, и переводит их на машинный код. ОС получает код в виде инструкций и выполняет их.

Зачем bash? Те же самые действия с файлами, папками и поиском можно выполнить с помощью графического интерфейса ОС, но это дольше, неудобнее и сложнее. Программисты пользуются bash или shell, чтобы упростить и ускорить работу с системой. Например, чтобы скопировать файл с помощью графического интерфейса, нужно открыть папку, где он расположен, кликнуть на файл правой кнопкой мыши, вызвать контекстное меню и выбрать «Скопировать». А если использовать командную строку и bash — потребуется ввести одну команду `cp <файл1> <файл2>`.

Как запустить? Оболочка встроена в систему и включается автоматически, нужно просто открыть терминал.

Bash для тестировщика

Первое, для чего тестировщик откроет терминал и начнет в нем работать - это логи.

Если нужно просмотреть весь журнал (который может быть внушительных размеров) и найти определенные события, то можно использовать `-grep`

```
grep -i 'error' console.log # где i - это регистронезависимый поиск
```

в данном случае будут найдены все строки с ошибками.

`tail` - команда показывает окончание файла (с ключом `-n` можно увидеть заданное количество строк)

```
tail -n 300 console.log
```

`less` - также является способом чтения логов (очень похожа на vim, но с возможностью только чтения и поиска по файлу)

```
less console.log # откроет файл на просмотр
```

Тестирование интеграции

Прежде чем проводить тестирование, необходимо провести проверку связности. Начать можно с проверки доступности сервиса, с которым мы интегрируемся. Действия производятся с машины, на которой поднят интегрирующийся сервис.

`ping <host>` # если пинг проходит, значит на той стороне как минимум есть рабочий стенд, однако, обратный результат еще ни о чем не говорит.

Затем можно проверить открыты ли нужные порты:

`telnet <ip> <port>` # пример: `telnet chekip.dyndns.org 80`

Если соединение открыто, то остается последний шаг перед началом тестирования интеграции. С помощью утилиты **curl** можно проверить возможность представленного в спецификации запроса:

`curl <host> | jq` # **jq** для структурированного просмотра ответа, если используется форма **json**

При получении ответа 200 можно приступать к тестированию.

Перезагрузка приложений и изменений настроек

Иногда возникает потребность временно изменить настройки на тестовом стенде (например изменить максимальный размер загружаемых файлов или поправить ссылку интеграционного стенда). Настройки обычно лежат в файлах `application.properties`. Чтобы найти их и открыть, необходимо воспользоваться командой `locate`

`locate application.properties`

Данная команда проводит поиск по специальной базе данных, которая периодически обновляется через планировщик. Для немедленного обновления нужно запустить команду `updatedb` с администраторскими полномочиями `sudo`.

Сами настройки можно открыть в VIM или другом текстовом редакторе:

`vim <path> /application.properties`

После изменения и сохранения настроек нужно перезагрузить приложение:

`sudo systemctl restart <serviceName>.service`

После рестарта необходимо сразу посмотреть логи, и еще проверить статус приложения спустя несколько минут после рестарта:

`sudo systemctl status <serviceName>.service`

Приложение должно запуститься. Если же нет, то нужно вернуть старые настройки и снова перезапустить приложение.

