

Содержание:

- [1. В чем разница между символической и жесткой ссылкой](#)
- [2. В чем разница между adduser и useradd](#)
- [3. В чем разница между /etc/init и /etc/init.d](#)
- [4. В чем разница между /bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin, /usr/local/sbin](#)
- [5. В чем разница между sudo su, sudo -i, sudo /bin/bash](#)
- [6. В чем разница между терминами консоль, терминал, командная строка](#)
- [7. В чем разница между dpkg и apt-get](#)
- [8. В чем разница между service и /etc/init.d](#)
- [9. В чем разница между #!/bin/bash и #!/bin/sh](#)
- [10. В чем разница между пакетами i386 и amd64](#)
- [11. В чем разница между killall и pkill](#)
- [12. В чем разница между PGP и GPG](#)
- [13. В чем разница между режимом сна и гибернации](#)
- [14 В чем разница между Ctrl+C, Ctrl+Z, Ctrl+D](#)
- [15. В чем разница между gksu и pkexec](#)

1. В чем разница между символической и жесткой ссылкой

Как вы знаете команда ln позволяет создавать ссылки в Linux. Она может создавать два типа ссылок — обычные (жесткие) и символические если передать опцию -s. Обе они указывают на файл. Разница лишь в виде этого указателя.

Символическая ссылка указывает на файл по имени. В файле ссылки устанавливается специальный бит, с помощью которого система определяет, что этот файл нужно открывать, прочитав имя и запросив нужный файл. Вы можете менять содержимое файла, но не его имя. Если изменить имя, ссылка станет нерабочей, ведь она будет указывать на имя файла которого уже не существует. Такой вид ссылок можно сравнить с ярлыками Windows, но это сравнение условное.

Жесткая ссылка основана на файловой системе. Она указывает на место, на диске, где расположены данные файла (номер иноды). Вы можете полностью изменять файл, включая его имя. По сути во время создания файла создается одна ссылка не него, если вы создаете жесткую ссылку с помощью ln, то их становится уже две. И даже если вы удалите первую, вторая все равно будет указывать на тот же файл, потому что эти ссылки равноправны и место на диске может быть очищено, только если ссылок не осталось вовсе.

2. В чем разница между adduser и useradd

Обе команды создают нового пользователя в системе. Но они очень сильно отличаются.

Adduser — утилита для добавления пользователей и групп с систему. Программа учитывает политики дистрибутива по желаемому UID и GID, создает домашний каталог, копирует туда файлы по умолчанию и запускает специальные сценарии.

Useradd- это низкоуровневая утилита для добавления пользователей. Создается только пользователь без каких-либо дополнительных настроек, вам придется все делать вручную в том числе создавать домашний каталог.

Рекомендуется использовать adduser.

3. В чем разница между /etc/init и /etc/init.d

Папка /etc/init.d содержит скрипты, используемые системой инициализации SysVinit. Это стандартная система инициализации для многих дистрибутивов Linux. Обычно, это сценарии оболочки, используемые для запуска, остановки и перезапуска сервисов.

Папка /etc/init содержит файлы конфигурации системы инициализации Upstart используемой в ubuntu. Поскольку Ubuntu постепенно переходит на Upstart, в системе есть и те и другие скрипты, многие /etc/init.d скрипты выполняются как слой совместимости с Upstart

4. В чем разница между /bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin, /usr/local/sbin

/bin — каталог для бинарных файлов, которые используются системой еще до того, как смонтирован раздел /usr на самом раннем этапе загрузки. Также здесь содержатся те команды, которые вы используете при запуске в одно-пользовательском режиме и режиме восстановления.

/sbin — то же самое, только для бинарных файлов, для выполнения которых требуются права суперпользователя.

/usr/bin — для общесистемных бинарных файлов

/usr/bin — то же самое, только для запуска которых нужны права суперпользователя.

/usr/local/bin и /usr/local/sbin — аналогично двум предыдущим, только для ваших собственных сценариев и программ, которые не устанавливаются с помощью менеджера пакетов.

5. В чем разница между sudo su, sudo -i, sudo /bin/bash

Команда su позволяет изменить текущего пользователя терминала на другого. Команда sudo выполняет команду от имени root пользователя. Теперь разберем сам вопрос.

sudo su — выполняет команду su от имени суперпользователя. Таким образом, сначала используется sudo чтобы получить права суперпользователя, а затем пользователь меняется на root с помощью su. При этом вы останетесь в той же директории потому, что выполняется ваш .bashrc. А также .profile пользователя root поэтому вы окажетесь в окружении root.

`sudo -i` — похожая команда, опция `-i` указывает утилите что нужно переключиться в консоль от имени суперпользователя. На этот раз вы уже окажетесь в домашней папке `root`, и будет выполнен его `.bashrc` и `.profile`.

`sudo /bin/bash` — запускает еще одну оболочку `bash` от имени суперпользователя. Файлы конфигурации не читаются, но выполняется только `.bashrc` вашего пользователя. Вы не окажетесь в окружении `root`, а просто останетесь в своем окружении с правами суперпользователя.

6. В чем разница между терминами консоль, терминал, командная строка

Понятия консоль и терминал тесно связаны между собой. Первоначально под ними подразумевалось оборудование с помощью которого, можно взаимодействовать с компьютером. Это было устройство, очень похожее на телетайп или печатную машинку.

В терминологии UNIX, терминал — это любой файл устройства, который реализует ряд специальных команд для чтения и записи. Под эмуляторами терминалов подразумеваются программы, обеспечивающие доступ к этим Unix терминалам через некоторую прослойку ядра. Это могут быть такие программы как `Xterm`, `Terminator`, программы, обеспечивающие изоляцию между приложениями — `screen`, `tmux`, программы удаленного доступа — `ssh` и т.д.

Консоль — это, как правило, терминал в физическом смысле. Главный терминал, реализуемый ядром, находящийся непосредственно на машине. В Linux консолью называют несколько терминалов TTY, между которыми можно переключаться с помощью сочетаний клавиш `Ctrl+Alt+F(n)`

Командная строка — это интерфейс, где пользователь вводит свою команду и нажимает `Enter` чтобы ее выполнить.

7. В чем разница между `dpkg` и `apt-get`

`dpkg` — это утилита для установки `deb` пакетов. То есть при выполнении команды `dpkg -i file.deb`, этот пакет будет установлен, если ему нужны какие-либо зависимости, утилита уведомит вас об этом. Но сами зависимости установлены не будут и пакет не будет установлен если зависимости не разрешены.

`apt-get` — это система управления пакетами, для дистрибутивов на базе Debian. Вы можете очень просто устанавливать, обновлять и удалять пакеты.

Здесь есть еще один инструмент — `gdebi`, это как бы среднее звено между `dpkg` и `apt-get`. Во время установки `deb` пакета утилита может разрешать и сама устанавливать зависимости.

8. В чем разница между `service` и `/etc/init.d`

Как мы уже говорили, `/etc/init.d` — содержит скрипты инициализации SysVinit. Это старый способ запускать сервисы. Эти сервисы запускаются в определенной

последовательности и не могут быть настроены более сложные зависимости.

Upstart была разработана с целью заменить все `/etc/init.d` скрипты на более совершенные конфигурационные файлы `/etc/init`.

`service` позволяет плавно перейти от скриптов `/etc/init.d` к конфигурационным файлам `upstart`. Эта утилита позволяет работать с обоими типами сервисов.

9. В чем разница между `#!/bin/bash` и `#!/bin/sh`

`bash` и `sh` это две разные оболочки. `Bash` — более новая и функциональная, с расширенными возможностями синтаксиса. Большинство команд работают одинаково, но они разные.

В современных системах чаще всего `/bin/sh`, это всего лишь ссылка на `/bin/bash`.

10. В чем разница между пакетами `i386` и `amd64`

Эти пакеты собраны для разных архитектур. `i386` — для 32 битной версии, а `amd64` (`x86_64`) — для 64 разрядной версии процессора Intel или AMD

Intel 80386, еще известный как `i386`, это 32 битный процессор, выпущенный компанией Intel в 1985 году. Эта архитектура еще называется `x86`, `IA-32` или просто `i386`.

`x86-64` — расширение набора инструкций для `x86`. Здесь поддерживаются гораздо большее адресное пространство, чем в `x86`, что позволяет программистам намного легче работать с большим количеством данных.

Даже если у вас процессор Intel, вы будете использовать `amd64` пакеты для установки 64 битной системы, здесь используются те же наборы инструкций, это стандарт.

11. В чем разница между `killall` и `kill`

Обе эти утилиты делают почти то же самое — уничтожают процессы, но по-разному.

`kill` находит процесс по разным параметрам, например, `pid`, `ppid`, `cmd`, `uid`, и посылает ему сигнал завершения.

`killall` находит процесс по имени и завершает все совпавшие процессы, есть режим регулярных выражений и интерактивный режим, которых нет в `kill`.

12. В чем разница между `PGP` и `GPG`

`PGP` — формат ключей и зашифрованных сообщений, оформленный и стандартизированный как `OpenPGP`. А также программное обеспечение `Pretty Good Privacy`, для шифрования, написанное Филом Зумерманом и сейчас принадлежащее компании `Symantec`.

GPG- независимая программная реализация стандарта OpenPGP, вы можете использовать ее для обмена сообщениями с людьми, даже теми, кто использует ПО от Symantec.

13. В чем разница между режимом сна и гибернации

В режиме сна ваш компьютер не отключается полностью. Система оставляет все периферийные устройства в режиме энергосбережения. Если батарея разрядится или компьютер отключится по каким-либо другим причинам текущая сессия и настройки будут потеряны.

Гибернация сохраняет состояние вашего компьютера на жесткий диск и полностью выключает. При включении сохраненное состояние восстанавливается.

14 В чем разница между Ctrl+C, Ctrl+Z, Ctrl+D

Сочетание клавиш Ctrl+C (intr) отправляет сигнал SIGINT приложению. Обычно после этого приложение завершает свою деятельность и закрывается.

Ctrl+Z или susp, отправляет сигнал SIGSTP, приложение приостанавливает свою деятельность и сворачивается в фоновый режим. Потом его можно восстановить с помощью команды fg. Посмотреть свернутые процессы можно командой jobs.

Ctrl+D — полностью завершает сеанс терминала.

15. В чем разница между gksu и pkexec

Обе программы позволяют выполнять графические приложения от имени суперпользователя.

Pkexec работает на основе PolicyKit. PolicyKit более настраиваемая, показывает полный путь к программе, и позволяет использовать политики для настройки поведения программы, например, настройки запроса пароля и т.д.

—

Источник: <https://losst.ru/top-voprosov-o-linux-v-chem-raznitsa>

Материал распространяется под лицензией CC-BY-SA