

РАБОТА В ТЕРМИНАЛЕ LINUX ДЛЯ НАЧИНАЮЩИХ

ЧТО ТАКОЕ ТЕРМИНАЛ LINUX?

Применение терминала началось очень давно. Еще до того как была создана DOS и не существовало никакого графического интерфейса. В далеких восьмидесятих годах операционная система Unix только начинала развиваться. Пользователям нужно каким-то образом взаимодействовать с системой и самый простой способ - это использование команд. Вы вводите команду, система вам возвращает ответ.

С тех пор, такой способ ввода использовался во многих системах, в том числе DOS и OS/2 от Apple, пока не был придуман графический интерфейс. Затем текстовый режим терминала был успешно вытеснен, но его до сих пор продолжают использовать там, где это нужно.

Выше, под терминалом мы понимали то место, где можно вводить команды и получать на них ответ от компьютера. Это может быть текстовый режим Linux или же открытое в графическом режиме окно терминала. В Linux часто встречаются слова: консоль, терминал, командная строка, командная оболочка, tty, эмулятор терминала. Все они относятся к терминалу, но означают немного разные вещи. Перед тем как перейти дальше давайте разберемся с терминами, чтобы называть все своими именами.

Под **терминалом** принято понимать окружение, где можно вводить команды и получать на них ответ, это может быть физический терминал или терминал на компьютере.

Консоль - это физическое оборудование для управления сервером. Когда к серверу нет доступа из сети, для управления им можно использовать только консоль.

TTY - это файл устройства, который создается ядром и предоставляет доступ к терминалу для программ. Это могут быть файлы /dev/tty для постоянных текстовых терминалов и /dev/pts/* для эмуляторов терминалов. Вы можете выполнить команду или отправить сообщение просто записав данные в этот файл, и также получить результат, прочитав данные из этого файла.

Эмулятор терминала - это графическая программа, которая предоставляет вам доступ к tty или pts терминалу. Например, Gnome Terminal, Konsole, Terminix, Xterm и многие другие.

Командная оболочка - устройство tty занимается только передачей и приемом данных, но все эти данные должен еще кто-то обрабатывать, выполнять команды, интерпретировать их синтаксис. Командных оболочек достаточно много, это bash, sh, zsh, ksh и другие, но чаще всего применяется Bash.

Ну и **командная строка** - это то место куда вы будете вводить свои команды, приглашение терминала для ввода.

Теперь, когда мы разобрались что такое терминал Linux и знаем все основные принципы, перейдем к практике работы с ним.

КАК ОТКРЫТЬ ТЕРМИНАЛ LINUX?

Есть несколько способов получить доступ к терминалу. Ваша система инициализации по умолчанию создает 12 виртуальных терминалов. В одном из них - обычно седьмом, запущена ваша графическая оболочка, но все другие могут быть свободно использованы.

Для переключения между терминалами можно использовать сочетания Ctrl+Alt+F1-F12. Для авторизации нужно будет ввести логин и пароль.

Это текстовые терминалы без графического интерфейса, в них может быть не совсем удобно работать, но, зато такие терминалы будут полезны, если графический интерфейс не работает.

Второй способ позволяет открыть виртуальный терминал прямо в графическом интерфейсе с помощью эмулятора терминала. Эмулятор терминала linux работает с файлами в каталоге /dev/pts/* и еще называется псевдотерминалом, потому что не использует tty.

ВЫПОЛНЕНИЕ КОМАНД В ТЕРМИНАЛЕ

Рассмотрим более подробно терминал Linux для начинающих. Как я уже говорил, терминал и файлы устройств tty отвечают только за передачу данных. За обработку команд отвечает командная оболочка, которой и передаются полученные от пользователя данные.

Вы можете набрать что-либо и убедиться, что это работает:

Чтобы выполнить команду достаточно написать ее и нажать Enter.

Более того, командная оболочка Bash поддерживает автодополнение, поэтому вы можете написать половину команды, нажать TAB и если на такие символы начинается только одна команда, то она будет автоматически дополнена, если же нет, то вы можете нажать два раза TAB, чтобы посмотреть возможные варианты.

Точно такая же схема работает для путей к файлам и параметров команд:

В Windows вы о таком и мечтать не могли. Чтобы выполнить команду можно указать имя ее исполняемого файла или полный путь к нему, относительно корневой или любой другой папки. Важно заметить, что командная оболочка Linux, в отличие от Windows, чувствительна к регистру, а поэтому будьте внимательны при вводе команд и их параметров.

По умолчанию работа в командной строке linux может выполняться с помощью большого количества команд, многие из них, например, для перемещения по каталогам, просмотра содержимого, установки ПО поставляются вместе с системой.

Экземпляр запущенной команды называется процесс. Когда в терминале Linux выполняется одна команда нам нужно подождать ее завершения, чтобы выполнить следующую.

Команды могут выполняться без параметров, что мы видели выше, или же с параметрами, которые позволяют указать данные, с которыми будет работать программа, также есть опции, с помощью которых можно настроить поведение. Большинство стандартных утилит придерживаются такого синтаксиса:

\$ команда **опции** параметр1 параметр2...

Справочник базовых команд в терминале Linux с примерами

16 Фев 2019

Admin

Bash, Linux

1 090

Работа со справочными страницами

`apropos`

- поиск справочных страниц по ключевому слову.

`info`

- аналог команде `man`, выводит справочные страницы в другом формате.

`man`

- вызов страниц из указанного раздела. Вызов справочной страницы о команде.

`manpath`

- отображение путей хранения справочных страниц.

`whatis`

- информация о команде одной строкой.

`type`

- определяет тип команды (выполняемая программа, внутренняя команда, функция оболочки, алиас).

`help`

- короткая справка по встроенным в оболочку командам (builtin).

команда `--help`

- показать доступные флаги для команды.

Работа с историей команд

`history`

- вывод истории команд (хранится в файле `~/.bash_history`)

`!!`

- повтор предыдущей команды

`!число`

- повтор команды под определенным номером. Номер можно узнать командой `history`.

`!строка`

- повтор последней команды, которая начинается с слов указанных в строке.

```
! ? строка
```

- повтор последней команды, которая содержит слова указанные в строке.

```
script
```

- запись сессии терминала в файл. Для остановки записи введите - `exit`

Навигация по файловой системе

```
cd
```

- перейти в каталог.

```
pwd
```

- вывод полного пути до текущего каталога.

Управление каталогами

```
mkdir
```

- создание нового каталога.

```
rmdir
```

- удаление пустого каталога.

```
rm -r каталог
```

- удаление непустых каталогов.

Управление файлами

Просмотр свойств файлов и каталогов

```
ls
```

- просмотр содержимого и атрибутов (свойств) каталога.

```
file
```

- просмотр типа файлов.

```
du
```

- просмотр, того сколько памяти занимают файлы и каталоги.

```
stat
```

- показать детальную информацию о свойствах файла.

```
sort
```

- сортировка строк.

Создание файлов

```
> имя файла
```

- создание нового файла. Также очистка файла если он уже существует.

```
ln
```

- создание жесткой или символической ссылки для файла.

`mkfifo`

- создание именного канала, к которому могут обратиться два различных процесса. Именованные каналы FIFO позволяют обмениваться данными с абсолютно чужим процессом.

`mknod`

- создание файла символьного/блочного устройства или именованного канала FIFO.

`touch`

- изменение времени редактирования файла (часто используется для создания файлов).

Перенос, копирование и удаление файлов

`mv`

- переименование файлов. Перенос файлов в другие каталоги.

`cp`

- копирование файла (с переименованием).

`rename`

- переименование файлов с использованием масок.

`rm`

- удаление файлов и каталогов.

`scp`

- копирование файлов через зашифрованное сетевое соединение между двумя хостами.

Управление доступом к файлам

`chmod`

- изменение битов режима доступа к файлу.

`chown`

- изменение владельца файлов и каталогов.

`chgrp`

- смена группы пользователей для файла.

`getfacl`

- вывод списка контроля доступа (ACL) для файла.

`sudo`

- запуск команды от имени другого пользователя (root по умолчанию)

`umask`

- изменение маски прав доступа по умолчанию при создании нового файла.

Чтение текстовых файлов

cat

- конкатенация (склеивание) файлов. Часто используется для простого просмотра текстового файла.

grep

- поиск заданной строки в текстовых файлах.

head

- вывод первых 10 строк файла.

less

- мощная утилита для просмотра содержимого файла, более функциональный вариант утилиты more.

more

- старая утилита просмотра содержимого файла. Не позволяет скролить вверх.

tac

- тоже самое, что и команда cat только вывод текста происходит снизу вверх (последняя строка становится первой, предпоследняя становится второй и т.д.).

tail

- вывод последних 10 строк файла.

tr

- осуществляет посимвольную замену, удаление, преобразование символов строки из стандартного ввода.

uniq

- исключает повторяющиеся строки из сортированного текста.

wc

- вывод количества строк, слов и символов в файле.

Поиск ПО, файлов и каталогов

locate

- поиск файлов и каталогов, используя базу данных имен файлов.

updatedb

- обновление базы данных имен файлов. По умолчанию запускается в cron, раз в сутки.

find

- расширенный поиск файлов и каталогов.

which

- показывает путь бинарного файла установленной программы (сообщает установлена ли программа).

whereis

- расширенный аналог which, показывает нахождение программы когда нет бинарников и мануалов.

Получение информации о системе

`df`

- отчет об использовании пространства дисковых накопителей.

`du`

- отображение дискового пространства, занятого файлами и каталогами.

`free`

- отображение занятого пространства оперативной памяти.

`hostname`

- имя текущего хоста.

`iostat`

- статистика использования CPU и операций ввода/вывода.

`screenfetch`

- информация о системе (не везде предустановлена).

`uptime`

- время прошедшее после запуска системы.

`uname`

- информация об ОС.

`vmstat`

- информация об использовании виртуальной памяти.

Управление процессами

`ps`

- выводит информацию о выполняемых в данный момент процессах.

`pstree`

- вывод дерева процессов.

`top`

- выводит информацию в реальном времени о запущенных процессах в системе.

`jobs`

- вывод запущенных задач, остановленных или выполняемых в фоновом режиме.

`bg`

- перевод задачи (остановленной) на выполнение в фоновом режиме.

`fg`

- вывод задач (остановленных) из фонового режима работы на передний план.

`процесс &`

- запуск процесса в фоновом режиме.

`kill`

- принудительное завершение работы процессов.

`killall`

- принудительное завершение работы для нескольких одноименных процессов.

Работа с сетью

`mtr`

- утилита для диагностики сети (улучшенный аналог `traceroute` и `ping`).

`traceroute`

- трассировка маршрута до указанного хоста.

`ip`

- показать или изменить сетевые настройки.

Установка имени хоста (hostname)

`hostnamectl`

- установить новое имя хоста.

`nmtui`

- утилита для смены hostname в псевдографическом режиме.

Управление учетными записями пользователей

`id`

- вывод идентификаторов пользователя и его групп.

`logname`

- выводит логин текущего пользователя.

`su`

- смена учетной записи.

`whoami`

- отображает имя текущего пользователя.

`who`

- показывает залогиненных пользователей.

`w`

- отображает залогиненных пользователей и их процессы.

Работа с утилитами *ctl

`systemctl`

- управляет работой системных сервисов и утилит.

```
loginctl
```

- менеджер учетных записей.

```
journalctl
```

- вывод журналов (логов) сервисов.

Работа с файловыми системами

```
cat /etc/fstab
```

- список смонтированных файловых систем.

```
fuser
```

- вывод идентификаторов процессов, использующих файловую систему или сокет.

Управление окном терминала

```
clear
```

- очистка содержимого на экране терминала.

```
reset
```

- перезапуск сессии терминала.

Переменные окружения

```
env
```

- вывод всех переменных окружения.

```
export переменная
```

- создание переменной окружения (будет видна всей системе).

Управление питанием компьютера

```
shutdown
```

- управление питанием компьютера.

```
poweroff
```

- выключение компьютера.

```
reboot
```

- перезагрузка компьютера.

Другие интересные и полезные команды

```
cal
```

- просмотр календаря.

```
date
```

- вывод текущей даты.

```
expr
```

- простой калькулятор (некоторые символы необходимо экранировать). Пробелы между знаками обязательны.

`factor`

- раскладывает число на множители.

`look`

- отобразить слова, начинающиеся с заданной строки. По умолчанию используется словарь /usr/share/dict/words.

`man ascii`

- шпаргалка кодовой таблица ASCII-символов.

`yes` строка

- бесконечный вывод строки в терминале.

`wall` строка

- выводит сообщение в терминалах всех залогиненных пользователей.