

Linux

Командный процессор (shell)

Пользовательский интерфейс для набора команд, их исполнения и отображения результатов.

X

Это графическая система, которая обеспечивает поддержку окон, меню, иконок, мыши и других известных элементов GUI - графического интерфейса пользователя. На основе X строятся более сложные графические среды; наиболее популярные из них - KDE и GNOME.

Что такое команда?

Команда в Linux, как правило, состоит из названия программы, за которым следуют опции и аргументы, которые набираются в командном процессоре. Название программы ссылается на программу, расположенную где-то на диске (которую командный процессор найдет и выполнит). Опция, которая, как правило, начинается со знака минус, определяет действие программы, а аргументы, как правило, представляют собой входные и выходные данные. Например, следующая команда, которая считает количество строк в файле:

\$ wc -l myfile

состоит из имени программы (wc - программа "word count"), опции (-l), которая говорит, что программа должна подсчитать количество строк, и аргумента (myfile), задающего файл, над которым нужно проделать эту операцию (значок доллара - это приглашение на ввод команды в командном процессоре, указывающее на то, что он ждет вашей команды). Опции можно задавать по отдельности,

\$ myprogram -a -b -c myfile

либо их можно объединять после одного минуса.

\$ myprogram -abc myfile

То же самое, что -a -b -c хотя некоторые программы ведут себя своеобразно и не распознают объединенные опции.

Получение справки

Используйте команду man

Команда man выводит на экран страницу оперативной справки по заданной команде.

Например, для того чтобы посмотреть на документацию по команде вывода списка файлов ls, выполните следующую команду:

\$ man ls

Используйте команду info

Команда info - это расширенная гипертекстовая справочная система, охватывающая многие Linux-программы.

\$ info ls

Если для интересующей вас программы документации нет, то info выведет man-страницу для нее.

Используйте опцию --help (если она имеется) Многие Linux-команды отвечают на опцию --help

выводом короткого справочного сообщения.

\$ ls --help

"Перемещаться" из одной директории в другую можно с помощью команды cd.

\$ cd /one/two/three

***\$ pwd* Эта команда печатает рабочую директорию**

cp [опции] источник цель

Команда cp копирует файл:

\$ cp file1 file2

Используя опции -a или -R, вы также можете рекурсивно копировать директории.

mv [опции] источник цель

Команда mv может либо переименовывать файлы:

\$ mv file1 file2

либо перемещать файлы и директории в заданную директорию:

\$ mv file1 file2 destination_directory

Команда rm может либо удалять файлы:

\$ rm file1 file2 file3

либо рекурсивно удалять директории:

\$ rm -R dir1 dir2

Команда rmdir (удалить директорию) удаляет одну или более пустых директорий

Команда mkdir создает одну или несколько директорий.

\$ mkdir d1 d2 d3

Просмотр файлов

cat - Просмотреть файлы целиком

less - Просмотреть файлы постранично

more - Выводит файл постранично

head - Просмотреть первые строки файла

tail - Просмотреть последние строки файла

\$ cat file1

\$ less file1

\$ more file1

\$ head file1

\$ tail file1

Текстовый редактор nano

Позволяет создать/редактировать файлы.

\$ nano file1

Выделение Alt+A

Копирование AtI+6

Вставить F10

Вырезать Ctrl+K

Команда chown (от англ. change ownership) устанавливает владельца файлов или директорий.

\$ chown user myfile myfile2 mydir

\$ chown user:group myfile myfile2 mydir

Опция -R выполняет операцию рекурсивно

Команда `chmod` (от англ. *change mode*) устанавливает права доступа к файлам и директориям.

Символьные	RWX	RWX	RWX
	Пользователь	Группа	Все остальные
В двоичной СС	111	101	000
В восьмеричной СС	7	5	0

Команда "+" - добавить права, "-" - удалить права, "=" -установить абсолютные права, игнорируя существующие.

```
$ chmod +x myfile myfile2 mydir
```

```
$ chmod -R 640 mydir
```

Опция -R выполняет операцию рекурсивно

Команда **`chgrp`** позволяет изменить группу-владельца файла даже если нет прав суперпользователя

```
chgrp группа файл
```

```
$ chgrp users myfile
```

Команда `find` ищет в одной или более директориях (и рекурсивно в их поддиректориях) файлы, соответствующие определенному критерию. Найти определенный файл с именем `myfile`.

```
$ find /home -name myfile
```

Опции:

-name Поиск по имени

-type {d | f} Поиск по типу d-директория, f-файл.

Опции можно сочетать.

Команда `grep` - одна из наиболее полезных и мощных в арсенале Linux. Ее предпосылки просты: если на вход подаются несколько файлов, то она печатает все строки в этих файлах, которые соответствуют определенному шаблону регулярного выражения.

```
$ grep "что ищем" myfile
```

Поиск в выводе другой команды:

```
$ ps -A | grep nano
```

Команда `mount` делает доступным физическое устройство хранения данных. Наиболее часто она используется с дисковыми устройствами (скажем, `/dev/sda/`), делая их доступными через существующую директорию (скажем, `/mnt/mydir`).

```
#mkdir /mnt/mydir
```

```
#mount /dev/hda1 /mnt/mydir
```

Команда `ps` выводит информацию о ваших работающих процессах и с опцией `-A` или `-e` о процессах других пользователей.

`$ ps`

`$ ps -A`

`$ ps -e`

Команда `kill` отправляет сигнал процессу. Он может завершить процесс (действие по умолчанию), прервать его, приостановить, аварийно завершить и т. д. Вы должны быть владельцем процесса или суперпользователем, чтобы выполнять над ним операции.

`$ kill 13243`

Завершение процесса по имени

`killall top`

Команда `free` выводит объем доступной и занятой виртуальной памяти в мегабайтах

`free -m`

в гигабайтах

`free -g`

в человекопонятном формате

`free -h`

Команда `df` выводит информацию о свободном дисковом пространстве

`df -H`

Команда **`du`** оценивает объем, занимаемый файлом или директорией

`du -h text.txt`

Сетевые команды

Маршрут до узла:

`traceroute 8.8.8.8`

Настройки IP

`ip addr show`

`ip a`

Файл для настройки свойств сетевого адаптера

`nano /etc/network/interfaces`

Файл настройки серверов DNS

`/etc/resolv.conf`

Обновление IP адреса

`dhclient eth0`

Вывод истории команд

`history`

Направить вывод команды в файл (с перезаписью существующих данных в файле)

`history > history.txt`

Направить вывод команды в файл (с добавлением в конец файла)

`history >> history.txt`

Текстовые редакторы.

nano

Система обозначений комбинаций клавиш следующая: Комбинации с Control обозначены символом (^) и вводятся при помощи нажатой кнопки Ctrl или двойном нажатии Escape (Esc); комбинации с Esc обозначены символом Meta (m) и могут быть введены при помощи кнопок Esc, Alt или Meta, в зависимости от используемой клавиатуры. Также, нажатие Esc дважды и дальнейший ввод трёхзначного числа от 000 до 255 введёт соответствующий символ. Следующие комбинации доступны в главном окне редактирования. Альтернативные комбинации показаны в скобках:

Ctrl+G или F1 - Показать справку по командам Ctrl+X
или F2 - Закрыть текущий буфер / Выйти из nano

Ctrl+W или F6 - Искать текст или регулярное выражение
Ctrl+K или F9 - Вырезать текущую строку и сохранить её в буфере обмена
Ctrl+U или F10 - Вставить содержимое буфера обмена в текущую строку
Ctrl+^ или m+Alt - Отметить текст в текущей позиции курсора
m+^ или m+6 - Копировать текущую строку и сохранить ее в буфере обмена
Alt+A - Начать выделение текста

vim

При запуске vim находится в командном режиме. Для перехода в режим редактирования необходимо нажать *i*

Для перехода обратно в командный режим – *Esc*

Командный режим. Любая команда в vim начинается с символа :

Выход

:q выход из vim, если файл не изменялся

:q! выход без сохранения изменений.

Сохранение

:w – записать или сохранить файл

:w filename – для сохранения под другим именем

:x или :wq – сохранить и выйти.

Конвейер

Простым примером перенаправления является `pipe` (конвейер). Обозначается он символом `|`.

Применение:

команда 1 | команда 2 | команда 3 ...

При таком вызове все данные, которые при обычном запуске *команды 1* выводились бы на экран будут поступать на стандартный ввод *команды 2*, как будто бы мы вводим эти данные с клавиатуры.

Пример:

ps -A | grep java