

Основные команды Linux

Команда	Для чего нужна
ls	Работа с содержимым каталога
alias	Задаёт псевдонимы
unalias	Удаляет псевдоним
pwd	Вывод пути рабочего каталога
cd	Переход в каталог
cp	Копирование файлов и каталогов
rm	Удаление файлов и каталогов
mv	Перемещение/переименований файлов или каталогов в файловой системе
mkdir	Создание каталогов
man	Справочник для любой команды
chmod	Изменение прав доступа файла (много флагов)
touch	Обновляет время обращения к файлам
exit	Завершение сессии оболочки и закрытие терминала
sudo	Действуйте как суперпользователь
shutdown	Выключение, остановка или перезагрузка компьютера
htop	Управление ресурсами машины прямо из терминала
unzip	Распаковка ZIP-файлов
apt, yum и pacman	Доступ к менеджерам пакетов
echo	Вывод информации на экран
cat	Создание, просмотр и объединение файлов
ps	Монитор запущенных процессов
kill	Убийца процессов
ping	Проверка подключения
vim	Текстовый редактор из 90-х
history	История ввода команд
passwd	Изменение паролей от аккаунтов пользователей
which	Выводит полный путь к командам оболочки
less	Просмотр файлов вперёд/назад
tail	Как <code>cat</code> , только выводит последние 10 строк
head	Выводит первые 10 строк
grep	Мощный поисковик по регулярным выражениям
whoami	Отвечает на вопрос: «Кто я, текущий пользователь?»
wc	Возвращает количество строк, слов, байтов и имя файла
uname	Выводит информацию об операционной системе
find	Поиск файлов в директории
wget	Извлекает контент из интернета

1. ls — работа с содержимым каталога

Команда `ls` позволяет перечислить содержимое нужного каталога (по умолчанию — текущий), включая файлы и другие вложенные каталоги. Может быть много опций для `ls`, поэтому не стесняйтесь обращаться к справке через `--help`, чтобы получить все доступные флаги для этой команды.

К примеру, раскрасим вывод команды `ls` :

```
ls --color=auto
```

2. alias — задаёт псевдонимы

Определяет временные псевдонимы в сеансе оболочки. При создании псевдонима вы заменяете слово серией команд. Вместо того чтобы каждый раз задавать цвет для `ls` и не вводить `--color`, используем псевдоним `ls="ls --color=auto"`.

Команда `alias` принимает один параметр пары «ключ-значение»: `alias NAME="VALUE"`, и значение должно быть заключено в кавычки. Если вы хотите перечислить все псевдонимы, которые у вас есть в сеансе оболочки, можете запустить команду `alias` без аргументов.

```
alias
```

3. unalias — удаляет псевдоним

Как можно догадаться, если `alias` создаёт псевдоним, то `unalias` удаляет его из уже определённых. Чтобы удалить предыдущий псевдоним `ls`, вбиваем:

```
unalias ls
```

4. pwd — вывод пути рабочего каталога

Если разбирать `pwd` дословно, то мы получим нечто вроде «распечатай рабочий каталог» и выведи абсолютный путь к каталогу, в котором сейчас находимся. Например, если ваше имя пользователя «Arseniy», и вы находитесь в каталоге «Документы», его абсолютный путь будет таким: `/home/Arseniy/Documents`. Просто введите `pwd` в терминале:

```
pwd
```

Результат: `/home/Arseniy/Documents/linux-commands`.

5. cd — переход в каталог

`cd` — одна из самых популярных команд. Она переключает в каталог, к которому вы хотите получить доступ.

Например, если вы находитесь в каталоге «Documents» и пытаетесь получить доступ к одной из его подпапок с названием «Reports», вводите:

```
cd Reports
```

Также не возбраняется прописывать абсолютный путь к папке:

```
cd /home/Arseniy/Documents/Reports
```

Несколько полезных трюков с командой `cd`

- Переход к домашней директории: `cd`
- На уровень выше: `cd ..`
- К предыдущей директории: `cd -`

6. cp — копирование файлов и каталогов

Отличный способ копировать папки и файлы прямо в терминале Linux! Чтобы использовать команду `cp`, достаточно прописать её вместо с исходным и целевым файлами:

```
cp file_to_copy.txt new_file.txt
```

Чтобы копировать целые директории, не забудьте применить «слэши»:

```
cp -r dir_to_copy/ new_copy_dir/
```

Директории в Linux всегда заканчиваются `/`.

7. rm — удаление файлов и каталогов

Создана для уничтожения! Чтобы избавиться от файлов или каталогов, дважды подумайте, а затем пропишите одну из версий.

Удаление обычного файла:

```
rm file_to_copy.txt
```

Удаление пустой директории (можно взять флаг `-r`):

```
rm -r dir_to_remove/
```

Если нужно избавиться от набитого данными каталога, используйте принудительные `-f` и рекурсивные флаги:

```
rm -rf dir_with_content_to_remove/
```

8. Команда `mv` — перемещение/переименований файлов или каталогов в файловой системе

Чтобы использовать эту команду, нужно ввести `mv` с исходным и целевым файлами:

```
mv source_file destination_folder/  
mv command_list.txt commands/
```

Для использования абсолютных путей введите:

```
mv /home/kinsta/BestMoviesOfAllTime ./
```

На всякий случай, `./` — это текущая директория.

Команда `mv` справляется и с переименованием файлов, сохраняя их в том же каталоге:

```
mv old_file.txt new_named_file.txt
```

9. `mkdir` — создание каталогов

Просто укажите имя нового каталога, которого ещё не существует:

```
mkdir pictures/
```

Чтобы создать поддиректорию, добавьте флаг `-p`:

```
mkdir -p music/jazz/
```

10. man — справочник для любой команды

Представим, что вы захотели узнать всё о команде `mkdir`:

```
man mkdir
```

И сама команда `man` не станет исключением:

```
man man
```

11. chmod — изменение прав доступа файла. Команда поддерживает достаточно опций, но основные разрешения выглядят так:

- `r` — чтение
- `w` — запись
- `x` — выполнение

Одним из самых популярных сюжетов для `chmod` — создание исполняемого файла пользователем. Для этого нужно ввести `chmod` и флаг `+x`, а затем файл, для которого вы хотите изменить разрешения:

```
chmod +x script
```

Чтобы сделать сценарии исполняемыми и запускать их напрямую, нужно использовать `./`.

12. touch — обновляет время обращения к файлам

Допустим, у нас есть файл, который запрашивали неделю назад. Чтобы проставить для него актуальную дату, нужно ввести команду с флагом `-m`:

```
touch -m old_file
```

Теперь дата будет совпадать с текущей. Но скорее всего вы будете использовать команду `touch` для создания новых файлов, а не обновления времени:

```
touch new_file_name
```

13. exit — выход есть выход

Команда означает ровно то, что вы подумали — завершение сессии `shell` и в большинстве случаев закрытие терминала:

```
exit
```

14. sudo — действуйте как суперпользователь

Эта команда превращает вас в суперпользователя (или `root`) в процесс выполнения какой-либо команды. `sudo` обычно используют для установки ПО или редактирования файлов за пределами домашнего каталога пользователя:

```
sudo apt install gimp  
sudo cd /root/t
```

Без пароля администратора здесь не обойтись — позаботьтесь о нём.

15. shutdown — выключение, остановка или перезагрузка компьютера

Выключение машины прямо сейчас:

```
shutdown now
```

Если хотите задать время выключения в 24-часовом формате:

```
shutdown 20:40
```

Чтобы отменить предыдущий вызов `shutdown`, нужно использовать флаг `-c`:

```
shutdown -c
```

16. htop — управление ресурсами машины прямо из терминала

```
htop
```

В большинстве случаев `htop` не установлен по умолчанию, поэтому скачайте установщик с [официального сайта](#).

17. unzip — распаковка ZIP-файлов

```
unzip some_scary_files.zip
```

Команда извлекает содержимое файлов `.zip`. Однако, инструмент может быть не установлен по умолчанию — проверьте менеджер пакетов.

18. apt , yum и pacman — доступ к менеджерам пакетов

Рано или поздно вам придётся обратиться к менеджерам пакетов, чтобы забрать какую-нибудь утилиту. В зависимости от ОС команды отличаются.

Установка GIMP на Debian-системах (Ubuntu, Linux Mint):

```
sudo apt install gimp
```

То же самое на Fedora и CentOS (Red Hat):

```
sudo yum install gimp
```

Наконец, Manjaro и Arco Linux (Arch):

```
sudo pacman -S gimp
```

19. echo — вывод информации на экран

Если вам нужно что-либо вывести на экран, то используйте `echo`:

```
echo "Somebody, help my pl..."
```

Обычно `echo` применяют для вывода переменных окружений, например:

```
echo "Hey $USER"
```

```
# Hey Arseniy
```


20. cat (от “concatenate”) — создание, просмотр и объединение файлов

Зачем лишний раз открывать текстовый редактор с интерфейсом, когда можно вбить `cat` ?

```
cat text_file_with_evidence.txt
```

21. ps — монитор запущенных процессов

```
ps
```

И тут же перед вами развернутся процессы, запущенные в текущем сеансе, а также полезная информация вроде ID процесса, TTY (TeleTYpewriter), время и имя команды.

Визуальный вывод будет довольно сдержанным, поэтому за впечатлениями обращайтесь к `htop`.

22. kill — убийца процессов

Программа не отвечает? Её не получается закрыть? Нужно быстро прервать действие? Используйте `kill` с указанием ID процесса (PID) или имени программы в бинарнике:

```
kill 533494  
kill firefox
```

Прежде чем запускать `kill` , убедитесь, что сохранили важные данные в этом процессе.

23. ping — проверка подключения

Пожалуй, это самая популярная утилита сетевого терминала, которая нужна для проверки подключения к сети.

```
ping google.com  
ping 8.8.8.8
```

У команды `ping` может быть масса опций, но в большинстве случаев вы будете вбивать домены или IP адреса.

24. vim — текстовый редактор из 90-х(от “concatenate”) — создание, просмотр и объединение файлов

Благодаря `vim` вы можете редактировать простые текстовые файлы прямо в терминале, жонглируя горячими клавишами. Поначалу будет непросто, но потом всё получится!

Для запуска VIM пропишите:

```
vim
```

25. history — история ввода команд

```
history
```

И вы получаете пронумерованный список команд, которые были использованы в прошлом.

26. — изменение паролей от аккаунтов пользователей

Когда вы пропишите эту команду:

```
passwd
```

Терминал первым делом попросит ввести текущий пароль и нажать Enter, а уже затем установить новый. Не шутите с паролями!

27. which — выводит полный путь к командам оболочки

К примеру, проверим абсолютные пути для Python и веб-браузера Brave.

```
which python  
# /usr/bin/python  
  
which brave  
# /usr/bin/brave
```

Если распознать команду не получится, терминал выдаст ошибку.

28. less — просмотр файлов вперёд/назад

Укажите этой команде файл, который хотите «прокручивать», и действуйте:

```
less best_file_to_scroll.txt
```

Фишка `less` в том, что она содержит в себе команды `vim` и `more`. Само собой, `less` похож по функционалу на `cat`, но выдаёт результат в более интерактивном виде.

29. tail — как cat, только выводит последние 10 строк

```
tail long_text.txt
```

С помощью флага `-n` вы можете указать количество последних строк:

```
tail -n 4 long_text.txt
```

30. head — выводит первые 10 строк

Полная противоположность `tail`, так как выводит первые строки файла:

```
head long_text.txt
```

И так же, как `tail` принимает флаг `-n`, чтобы вывести определённое количество:

```
head -n 7 long_text.txt
```

31. grep — мощный поисковик по регулярным выражениям

Одна из самых крутых команд в Linux, которая ведёт поиск в тексте на предмет соответствия регулярному выражению, а затем печатает их:

```
grep -c "linux" long_text.txt
```

```
#2
```

Флаг `-c` нужно для того, чтобы подсчитать количество повторяющихся результатов поиска.

32. whoami — отвечает на вопрос: «Кто я, текущий пользователь?»

На самом деле, `whoami` отображает имя текущего пользователя:

```
whoami
```

```
# Arseniy
```

Переменные окружения здесь также работают без проблем:

```
echo $USER
```

```
# Arseniy
```

33. wc — возвращает количество строк, слов, байтов и имя файла

```
wc long_text.txt
```

```
# 59 314 1500 long_text.txt
```

Если нужно получить только количество слов, примените флаг `-w`:

```
wc -w long_text.txt
```

```
314 long_text.txt
```

34. uname (сокр. от “Unix name”) — выводит информацию об операционной системе

Чаще всего эта команда используется с флагом `-a`, потому что вариант по умолчанию не слишком многословный:

```
uname

# Linux

uname -a

# Linux kinstamanjaro 5.4.138-1-MANJARO #1 SMP PREEMPT Thu Jul 5
10:15:21 UTC 2022 x86_64 GNU/Linux
```

35. find — поиск файлов в директории

Команда `find` занимается поиском файлов по каталогам на основе регулярного выражения. Стоит потренироваться с синтаксисом:

```
find [flags] [path] -name [expression]
```

Чтобы найти файл `long_text.txt` в текущей директории, введите:

```
find ./ -name "long_text.txt" # ./long_text.txt
```

Для указания определённых расширений, например `.py` из Python, введите:

```
find ./ -type f -name "*.py" ./get_keys.py ./github_automation.py
./binarysearch.py
```

36. wget — извлекает контент из интернета

Эта команда хранит самую большую коллекцию флагов, которую только можно представить. Чтобы просто скачать файл из репозитория GitHub, вбивайте:

```
wget https://raw.githubusercontent.com/DaniDiazTech/Object-Oriented-
Programming-in-Python/main/object_oriented_programming/cookies.py
```