



pwd - вывести текущую директорию	date - вывести текущую дату и время	history - список команд за сессию
whoami - имя текущего юзера	clear - очистить консоль	man - мануал по команде
ls - вывести содержимое директории ls /folder - вывести содержимое указанной директории ls .. - вывести содержимое родительской директории (ls . - текущей директории) -l - вывод с дополнительной информацией) -a - показать скрытые файлы -t - сортировка по времени изменения -r - обратная сортировка (в сочетании с -t наиболее «свежие» файлы будут внизу) -h - вывод в формате human readable с размерами файлов (-lh используется вместе с list) -S - сортировка файлов по размеру -f - добавит символ «/» в конце каждого каталога -R - показать содержимое каждой подпапки		
cd /folder - перейти в директорию cd ~ - перейти в домашнюю директорию \$HOME (тоже самое, что cd без аргумента) cd - - вернуться в предыдущую директорию cd .. - подняться в директорию выше (родительский каталог) cd ../.. - подняться на 2 уровня выше cd / - перейти в корневой каталог		
mkdir folder - создать директорию mkdir {1..3} - создать 3 папки mkdir -p folder/sub_folder - создать директорию и поддиректорию		
rm file - удалить файл rm -r - удалить папку со всем содержимым (рекурсивно) rmdir folder - удалить папку rm folder/* - удалить все файлы в директории rm -f - игнорировать ошибки при удалении, если файл не существует		
cp file/folder - копировать файл или директорию cp folder/{file1,file2} /folder - копировать несколько файлов cp -r - копировать со всем содержимым cp -a - копировать со всеми атрибутами, правами и рекурсивно (как -r)		
mv file/folder/new_file - переместить/переименовать файл или директорию mv folder/{file1,file2} /folder - копировать несколько файлов		
cat file - вывести содержимое файла cat > file - создать файл cat >> file - добавить строку в файл (ввод текста, Enter, ctrl+c) cat > folder/file - создать файл не заходя в папку		
tail -n file - вывести n последних строк файла tail -f - вывод данных файла в реальном времени tail -F - вывод данных файла в реальном времени, даже если их переименуют или пересоздадут		tac file sed -n 1,2p - вывести 2 последние строки файла multitail log1 log2 - просмотр двух и более логов
head -n file - вывести n первых строк файла		cat file sed -n 1,2p - вывести 2 первые строки файла
less file - вывести содержимое большого файла (пейджер) f - прокрутить вперед b - прокрутить назад q - выход /текст - поиск слова вниз по тексту ?текст - поиск слова вверх по тексту +F - вывод в режиме реального времени (как tail -f) -N - вывести номера строк -s - удалить лишние пустые строки		
ps - вывести список процессов aux - вывести все процессы в системе (ps aux head -5 - вывести первые 5 процессов в системе) axjf - вывести дерево процессов		
kill PID - завершить процесс (PID - цифровой идентификатор процесса) kill -9 - завершить процесс с возможной потерей данных (SIGKILL)		
ping host.com - пинговать сайт (послать хосту ICMP пакет ECHO_REQUEST) ping -c 5 - количество посылаемых пакетов (ping -c 3 -i 5 host.com - послать 3 пакета с интервалом 5 секунд) ping -i 5 - интервал между посылками пакетов		
ssh user@185.104.114.90 - подключение к удаленному Серверу (далее ввести пароль) ssh 185.104.114.90 - указать IP-адрес Сервера ssh -p 150 185.104.114.90 - указать порт, если он не стандартный		
ssh-keygen - сгенерировать ключи, приватный (d_rsa) и публичный (id_rsa.pub) ssh-copy-id login@185.104.114.90 - скопировать публичный ключ на Сервер ssh user@185.104.114.90 - подключиться к Удаленному Серверу используя ключ		
scp - копировать файлы между хостами (с помощью ssh) scp user@11.11.22.22:/home/user/temp Downloads/temp копировать на свой хост (user@11.11.22.22) в папку (:/home/user/temp) папку (Downloads/temp) с другого хоста)		
rsync - синхронизация директорий между хостами (с помощью ssh) rsync -a - archive mode, копирует полностью все содержимое директории rsync -v - вывод на консоль дополнительной информации rsync -av user@testhost:/home/user Downloads находясь на хосте, указать хост с которым нужно синхронизироваться (user@testhost)		
echo 'text' - вывести на экран строку текста echo -n - не дополнять строку переносом строки в конце echo -e - включить интерпретацию экранирования с помощью \n echo \$? - посмотреть код последней завершенной команды echo -e '\n\n > file - создать файл с двумя пустыми строками echo -en '\n\n >> file - добавить в файл две пустые строки		
touch file - создать файл touch file{1..3}.txt file{1..5}.json - создать 3 txt файла и 5 json файлов		
vim file - открыть файл в редакторе vim vim i - режим редактирования dd - удалить строку vim /text - найти слово vim :u - отмена операции vim :wq - сохранить изменения и выйти vim :q! - выйти без сохранения		
nano file - открыть файл в редакторе nano nano ctrl+o - сохранить изменения nano ctrl+x - выход		

GET (метод по умолчанию при использовании **curl**)

запрос GET к API JSONPlaceholder к представлению JSON всех сообщений

curl https://jsonplaceholder.typicode.com/posts

с фильтрацией результатов

curl https://jsonplaceholder.typicode.com/posts?userId=1

POST

создать новую запись, используя данные, указанные в опции -d

curl -X POST -d "userId=5&title=Hello World&body=Post body." https://jsonplaceholder.typicode.com/posts

Для отправки данных в формате JSON нужно установить тип тела application/json (Тип тела запроса указывается с помощью заголовка Content-Type. По умолчанию, когда этот заголовок не указан, curl использует Content-Type: application/x-www-form-urlencoded)

curl -X POST -H "Content-Type: application/json" \ -d '{"userId": 5, "title": "Hello World", "body": "Post body."}' \ https://jsonplaceholder.typicode.com/posts

PUT

curl -X PUT -d "userId=5&title=Hello World&body=Post body." https://jsonplaceholder.typicode.com/posts/5

PATCH

curl -X PUT -d "title=Hello Universe" https://jsonplaceholder.typicode.com/posts/5

DELETE

curl -X DELETE https://jsonplaceholder.typicode.com/posts/5

Аутентификация

Если конечная точка API требует аутентификации, необходимо получить ключ доступа. В противном случае сервер API ответит сообщением «Access Forbidden» или «Unauthorized».

Процесс получения ключа доступа зависит от используемого API. Получив токен доступа, можно отправить его в шапку:

curl -X GET -H "Authorization: Bearer {ACCESS_TOKEN}" "https://api.server.io/posts"

Bash script

touch file.sh - создать файл с расширением .sh

#!/bin/bash - в первой строке написать шебанг (указываем какую оболочку будем использовать)

cd folder; mkdir folder - написать необходимые команды

chmod +x ./file.sh - сделать файл исполняемым

./file.sh - запустить скрипт

Bash script, который выводит лог и подсвечивает красным, желтым и синим, если сообщения имеют ключевое слово ERROR, WARN, DEBUG)

```
tailf() {
    tail -f $1 | sed --unbuffered
    -e 's/\(. *FATAL.*\)/\o033[1;31m\1\o033[0;39m/' -e 's/\(. *ERROR.*\)/\o033[31m\1\o033[39m/'
    -e 's/\(. *WARN.*\)/\o033[33m\1\o033[39m/' -e 's/\(. *DEBUG.*\)/\o033[34m\1\o033[39m/'
    -e 's/\(. *TRACE.*\)/\o033[30m\1\o033[39m/' -e 's/\(. *[Ee]xception.*\)/\o033[1;39m\1\o033[0;39m/'
}
```

Заменить все вхождения строки 'string1', на 'string2' во всех файлах в текущей директории

grep -rl 'string1' ./ | xargs sed -i 's/string1/string2/g'

Выбрать IP адреса

grep -E -o "(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)" /var/log/auth.log

Удалить все пустые строки или строки с комментариями из конфига Apache:

sed '/^#\|^\$\|^#/' /etc/apache2/apache2.conf