

**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

```
Is - вывести содержимое директории
```

Is /folder - вывести содержимое указанной директории

Is .. - вывести содержимое родительской директории (Is . - текущей директории)

- -І вывод с дополнительной информацией)
- -а 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

mkdir folder - создать директорию

cd ~ - перейти в домашнюю директорию \$HOME (тоже самое, что cd без аргумента)

- cd - вернуться в предыдущую директорию
- сф .. подняться в директорию выше (родительский каталог)
- сф ../.. подняться на 2 уровня выше
- сф / перейти в корневой каталог

# folder{1..3} - создать 3 папки

-p folder/sub\_folder - создать директорию и поддиректорию

**rm** file - удалить файл -r - удалить папку со всем содержимым (рекурсивно)

rmdir folder - удалить папку

rm folder/\* - удалить все файлы в директории

-f - игнорировать ошибки при удалении, если файл не существует

ср file /folder - копировать файл или директорию cp folder/{file1,file2} /folder - копировать несколько файлов

-r - копировать со всем содержимым

- -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

-f - вывод данных файла в реальном времени

tac file | sed -n 1,2p - вывести 2 последние строки файла multitail log1 log2 - просмотр двух и более логов

-F - вывод данных файла в реальном времени, даже если их переименуют или пересоздадут 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

аих - вывести все процессы в системе (ps aux | head -5 - вывести первые 5 процессов в системе) axif - вывести дерево процессов

рѕ - вывести список процессов

-9 - завершить процесс с возможной потерей данных (SIGKILL)

kill PID - завершить процесс (PID - цифровой идентификатор процесса)

ping host.com - пинговать сайт (послать хосту ICMP пакет ECHO\_REQUEST)

- -c 5 количество посылаемых пакетов (ping -c 3 -i 5 host.com послать 3 пакета с интервалом 5 секунд) -і 5 - интервал между посылками пакетов
- ssh user@185.104.114.90 подключение к удаленному Серверу (далее ввести пароль)

**185.104.114.90** - указать IP-адрес Сервера

-р 150 - 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)

-a - archive mode, копирует полностью все содержимое директории -у - 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

-n - не дополнять строку переносом строки в конце

- -е включить интерпретацию экранирования с помощью \ \$? - 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

i - режим редактирования dd - удалить строку /text - найти слово :u - отмена операции

:wq - сохранить изменения и выйти :q! - выйти без сохранения

nano file - открыть файл в редакторе nano ctrl+o - сохранить изменения ctrl+x - выход

```
find file - найти файл по имени

    -maxdepth - максимальная глубина поиска по поддиректориям (1 - если только в текущей директории)

        -depth - искать сначала в текущей директории, а затем в поддиректориях
        -print - выводить полные имена файлов
         -type d - искать только папки
         -type f - искать только файлы
        -name - с учетом регистрозависимости
         -iname - без учета регистрозависимости
        -not -name - найти файлы не соответствующие шаблону
        -user - поиск файлов по владельцу
        -size - поиск файлов по размеру
        -perm - поиск файлу по режиму доступа (разрешениям)
         -empty - поиск пустых папок или файлов
        -o - или (find -name '*.html' -o -name '*.php' - найдет все HTML файлы или PHP файлы)
        find 'fi*' - найти файл начинающийся на 'fi'
        find '*.txt' - поиск по расширению
        find . - показать все файлы в текущей директории
        find ./folder - поиск в указанной директории
        find '.*' - найти скрытые файлы (начинающиеся на .)
        find folder/ -type f -ctime +5 - искать файлы в папке folder, созданные более 5 дней назад
        find folder/ -type f -mtime -10 - искать файлы в папке folder, измененные за последние 10 дней
         -ехес - выполнение действий с найденными файлами
        find /tmp -type f -name "*.txt" -exec rm -f {} \ - удалить все txt файлы в папке tmp
grep text - поиск текста в файлах
          -і - отключить регистрозависимость

    -n - показать номера строк, содержащих образец

          -с - выдать только количество строк, содержащих образец
          -r - рекурсивный поиск по поддиректориям
          -І - выведет только названия файлов, содержащих образец
          -и - выведет все строки, не содержащие образец (исключить образец из поиска)
          -e - или-или (grep -e 'text 1' -e text - выведет или text 1, или text)
          ' ' - поиск пробела 'ctrl+v TAB' - поиск табуляции
          -E - регулярное выражение (grep -E p\{2} file.txt - найдет строки, в которых есть 2 подряд идущие р)
          -F - искать, как строка, а не как регулярное выражение (grep -F $ - найдет $ и его не нужно экранировать)
          [Tt]ext - поиск по образцу, где первая буква может быть как заглавной, так и строчной
          '\<text\>' - искать по точному совпадению
          -o - выберет только e-mail адреса, а не всю строку с e-mail (grep -o '[[:alnum:]+\.\_\-]*@[[:alnum:]+\.\_\-]*')
          | sort - вывод с сортировкой по алфавиту
sed - найти и заменить в файлах, директориях (потоковый редактор текста) (sed опции -е команды файл)
     -п - не выводить содержимое буфера шаблона в конце каждой итерации
     р - вывести содержимое буфера шаблона
     d - удаляет буфер шаблона и запускает следующую итерацию цикла
     g - заменить содержимое буфера шаблона, содержимым дополнительного буфера
     sed -n '5,10p' file - вывести из файла строки с 5 по 10
     sed '1,20d' file - вывести весь файл, кроме строк с 1 по 20
     sed 's/root/losst/g' file - заменить вхождения слова root на losst
chown user:user folder - изменить владельца файла или директории (новый владелец:группа (опционально))
                       -R - изменить владельца файла или директории у всех поддиректорий
chmod 760 file - изменить права доступа файла
      7 - все права для владельца (чтение) 6 - чтение и запись (запись) 0 - ничего для остальных (исполнение)
      -х - убрать права на исполнение
      +х - вернуть права на исполнение
hexdump file - вывести содержимое бинарного файла
              -b - однобайтовый восьмеричный
              - с - отображение однобайтовых символов
              -C - канонический шестнадцатеричный + ASCII
              -у - показать все содержимое
awk - вывести колонку из вывода
     ps ux | awk '{print $2}' - выведет вторую колонку вывода ps ux
telnet host.com 80 - коммуникация с другим хостом (Клиент для протокола TELNET)
openssI s_client -connect example.com:443 - коммуникация с хостом по протоколу TLS
host google.com - узнать IP адрес по имени хоста
nslookup google.com - узнать IP адрес по имени хоста
dig +short google.com - узнать IP адрес по имени хоста
ifconfig - сетевая информация
ip a - сетевая информация (ip -4 a - сетевая информация по IPv4)
netstat - посмотреть открытые порты
        netstat -lptnu - посмотреть все слушающие TCP и UDP порты с отображением PID'а процесса,
```

слушающего порт, и с числовым представлением порта

-а - напечатать всю информацию, кроме -р (процессор) и -і (тип аппаратной платформы), если они неизвестны

top - вывод информации о задачах и статистике по системе (топ процессов по использованию ОЗУ или проц. времени)

port 80 - какой порт дампить (вместо порта, можно указать host или и порт, и хост)

-X (request) - метод HTTP, который будет использоваться

-d (data) - данные, которые должны быть отправлены-H (header) - дополнительный заголовок для отправки

**uname** - посмотреть информацию о системе

-h - в удобочитаемом формате

-h - в удобочитаемом формате-T - выводит тип файловой системы

**-А** - вывод в ASCII

free - посмотреть, сколько оперативной памяти занято или свободно

df - информация о файловых системах (сколько места занято и свободно)

tcpdump - дамп (перехват) сетевого трафика (tcpdump -i any -A port 80)

-n - не конвертировать адреса в хостнеймы в выводе

-X - выводить данные в hex и ASCII (если трафик бинарный)

curl [options] [URL] - утилита для передачи данных на удаленный Сервер или с него

-i (include) - включить заголовки ответа

-i any - не интересует сетевой интерфейс

**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 АРІ. 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)

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

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

Выбрать ІР 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]?)" /var/log/auth.log

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

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