Linux terminal (GitBash) commands
Окружение: windows10x64, gitbash с правами root

1) **Посмотреть где я**
pwd

2) **Создать папку**
mkdir folder (создать папку в текущей папке);
mkdir folder/folder2 (создаёт последнюю папку по указанному пути, если путь уже существует)
mkdir –p folder1/folder2 (флаг -p позволяет создать путь из нескольких папок, если их не существует)

3) **Зайти в папку**
cd folder или cd folder/folder2 (относительный путь, указывается путь ОТ текущей папки);
cd /b/folder (абсолютный путь, переход в конкретную папку из любого места)

4) **Создать 3 папки**
mkdir folder1 folder2 folder3;
mkdir folder{1..3} (создаёт любое количество папок с именем folder1, folder2 и т.д.)

5) **Зайти в любоую папку**
cd /b/folder/folder2/folder3/folder4 (из любого места - указать абсолютный путь, из текущей папки и далее из неё - относительыний путь)

6) **Создать 5 файлов (3 txt, 2 json)**
touch 1.txt 2.txt 3.txt 4.json 5.json

7) **Создать 3 папки** - см. п.4)

8) **Вывести список содержимого папки**
ls (только названия файлов и папок);
ls -la (подробный табличный вид всех файлов и папок в текущей папке)
ls -t (сортирует по времени создания от нового к старому)
ls -r (сортирует по времени создания от старого к новому)
ls -R (показывает содержимое текущего каталога, а затем рекурсивно содержимое каждой вложенной папки с файлами)
ls -latrR (всё вышеперечисленное сразу)

9) **+ Открыть любой txt файл**
vim 1.txt или vi 1.txt (файл должен существовать. Если файла не существует и в редакторе нажать сохранить - создаётся новый файл)
nano 1.txt (второй встроенный редактор в терминал. Отличие от vim - в окне терминала внизу написаны команды: как сохранить, как выйти, и т.д.: Ctrl+x сохранить? Y)

10) **+ написать туда что-нибудь, любой текст.**
vim 1.txt затем нажать i и редактировать текст

11) **+ сохранить и выйти.**
vim 1.txt + нажать i + редактировать + нажать ESC + ввести :wq
:q закрыть
:q! закрыть без сохранения
:w сохранить
:wq сохранить и выйти
i войти в режим редактирования
ESC выйти из режима редактирования

12) **Выйти из папки на уровень выше**
cd .. (на папку выше)
cd ../.. (на 2 папки выше)

13) **переместить любые 2 файла, которые вы создали, в любую другую папку.**
mv 1.txt 2.txt folder (если в текущей папке есть папка folder - переместит в неё)
mv 1.txt 2.txt folder/folder1 (переместит по указанному относительному пути)
mv 1.txt 2.txt /b/folder/folder1 (переместит по указанному абсолютному пути)

14) **скопировать любые 2 файла, которые вы создали, в любую другую папку.**
cp 1.txt 2.txt folder (если в текущей папке есть папка folder - скопирует в неё)
cp 1.txt 2.txt folder/folder1 (скопирует по указанному относительному пути)
cp 1.txt 2.txt /b/folder/folder1 (скопирует по указанному абсолютному пути)

15) **Найти файл по имени**
find . -name 1.txt (рекурсивный поиск от текущей папки)
find . -name "название с пробелами.txt" (берем в кавычки)
find . -iname a.txt (i делает поиск регистронезависимым, найдет и a.txt и A.txt)
find 1.txt (поиск в текущей папке)
find /home -name 1.txt (рекурсивный поиск от указанной папки)
find . -name 1.txt (рекурсивный поиск от текущей папки)
find . -name "*.txt" (* - любое имя с расширением txt)
find . -name "1.*" (* - любое расширение с именем 1)
find . -empty (покажет все пустые файлы и папки)
find . -empty -delete (найдет пустые файлы и папки и удалит их)
find . -name "*.jpg" -exec cp {} /b/folder/fotos \; (найдет все файлы jpg и скопирует их по указанному пути. -exec позволяет выполнить команды rm, mv, cp; {} означает результаты поиска; затем можно указать путь; \; означает конец команды)

16) **просмотреть содержимое в реальном времени (команда grep) изучите как она работает. Посмотреть содержимое В РЕАЛЬНОМ ВРЕМЕНИ:**
tail -f 1.txt (показывает последние 10 строк файла в терминале, если в файл вносятся изменения - показывает изменения в реальном времени)
Просто посмотреть содержимое файла:
cat 1.txt (выводит всё содержимое файла в окно терминала)
nl 1.txt (выводит содержимое в терминал и ставит номера строк, исп. для просмотра файлов с кодом или конфигурационных файлов)
less 1.txt (выводит содержимое только в рамках текущего размера окна терминала в режиме просмотра. После выхода в терминале нет текста файла)
head 1.txt (выводит первые 10 строк в окно терминала)
tail 1.txt (выводит последние 10 строк в окно терминала)

17) **вывести несколько первых строк из текстового файла**
head -2 1.txt (-2 означает количество строк, которые покажутся)

18) **вывести несколько последних строк из текстового файла**
tail -2 1.txt (-2 означает количество строк, которые покажутся)

19) **просмотреть содержимое длинного файла (команда less) изучите как она работает.**
less 1.txt (покажет текст только в рамках текущего размера окна терминала
- стрелки вверх-вниз - прокручивать содержимое вверх-вниз
- w, z - листать страницы
- / - поиск текста по файлу
- h - открывает справку по горячим клавишам
- q - выйти из режима просмотра (содержимое файла не сохраняется в окне терминала)

20) вывести дату и время
date (покажет Wed Jul 28 22:02:56 RTZ 2021);
date +"%H:%M %d/%m/%Y" (покажет 22:02 28/07/2021)
==========

Задание *
1) Отправить http запрос на сервер.
https://api.quarantine.country/api/v1/regions

curl GET https://api.quarantine.country/api/v1/regions

2) Написать скрипт который выполнит автоматически пункты 3, 4, 5, 6, 7, 8, 13

создал файл: myFirstScript.sh
ввел текст:

```
#!/bin/bash
cd /b/folder
mkdir 1 2 3
cd /b/folder/3
touch 1.txt 2.txt 3.txt 4.json 5.json
mkdir folder{01..03}
ls -la
mv 1.txt 2.txt /b/folder/1/
```

в терминале ввел команду: ./myFirstScript.sh