**Домашняя работа**

**Часть 1. Linux terminal (GitBash) commands:**

1) Посмотреть, где я:

pwd

2) Создать папку:

mkdir fruits

3) Зайти в папку:

cd fruits

4) Создать 3 папки:

mkdir {apples,pears,oranges}

5) Зайти в любую папку:

cd apples

6) Создать 5 файлов (3 txt, 2 json):

touch sale.txt buy.txt stock.txt 1.json 2.json

7) Создать 3 папки:

mkdir tropical\_fruits rare\_fruits season\_fruits

8) Вывести список содержимого папки:

ls -la

9) + Открыть любой txt файл:

cat apples/sale.txt

10) + написать туда что-нибудь, любой текст:

cat > apples/sale.txt

1 Robert 25 apples

2 Maggie 10 oranges

3 Tom 0 bananas

11) + сохранить и выйти:

Ctrl+C

12) Выйти из папки на уровень выше:

cd ../

13) переместить любые 2 файла, которые создали, в любую другую папку:

mv fruits/apples/sale.txt fruits | mv fruits/apples/buy.txt fruits

14) скопировать любые 2 файла, которые создали, в любую другую папку:

cp fruits/sale.txt fruits/peares/sale.txt

cp fruits/buy.txt fruits/peares/buy.txt

15) Найти файл по имени:

find -name "2.json"

16) просмотреть содержимое в реальном времени:

tail -f log\_file | grep -C 3 search\_term

Пояснение:

* *опция -f (follow) постоянно выводит изменения на экран, т.е. показывает новые строки, добавленные в файл. Для остановки используется Ctrl +C.*
* *соединяем с командой grep для поиска конкретного термина при мониторинге файла журнала, т.к. визуально обнаружить искомое в потоке входящих новых строк практически невозможно.*
* *опция -С в grep , чтобы показать несколько строк до и после искомого термин, т.к. только строки с искомыми словами не раскрывают необходимых деталей.*

17) вывести несколько первых строк из текстового файла:

head *log\_file*

*Пояснение: команда head отображает по умолчанию первые 10 строк текстового файла. Опция -n выдает заданное количество строк. -n5 например*

18) вывести несколько последних строк из текстового файла:

tail log\_file

Пояснение: *команда tail отображает по умолчанию первые 10 строк текстового файла. Опция -n выдает заданное количество строк.*

19) просмотреть содержимое длинного файла:

less

Комментарий: перечень всех опций и внутренних команд можно просмотреть в терминале, выполнив команду man less

20) вывести дату и время:

date

=========

**Часть 2 \***

**1 задание.** Отправить http запрос на сервер. <http://162.55.220.72:5006/terminal-hw-request>

**Решение:**

1. В терминале пишим запрос:

$ curl <http://162.55.220.72:5006/terminal-hw-request>

Получаем ответ:

{

"Intro": "Hello!! This is your the first response from server",

"Tasks": {

"Task\_1": "Send the next URL in terminal: http://162.55.220.72:5005/get\_method?name=(set\_your\_String)&age=(set\_your\_number)",

"result": [

"Your\_String",

"Your\_number"

]

}

}

1. Отправляем запрос:

Jiokko@Jiokko MINGW64 /g/My Drive/QA/Ksendzov Cours/Terminal/fruits

$ curl ‘[http://162.55.220.72:5005/get\_method?name=Nata&age=37](http://162.55.220.72:5005/get_method?name='Nata'&age=37)’

Комментарий:

# тут подробно расписано <https://losst.pro/kak-polzovatsya-curl>

# Ставим в кавычки, так как аперсанд & считывается, как оператор. Чтобы цельный запрос был, заключаем в кавычки

**Задание 2**

**Написать скрипт, который выполнит автоматически команды 3, 4, 5, 6, 7, 8, 13**

**Решение:**

#!/bin/bash

mkdir fruits

cd fruits

mkdir {apples,pears,oranges}

cd apples

touch sale.txt buy.txt stock.txt 1.json 2.json

mkdir tropical\_fruits rare\_fruits season\_fruits

cd ../

ls -la

mv -i G:/"My Drive"/QA/Ksendzov\_Cours/Terminal/fruits/apples/sale.txt G:/"My Drive"/QA/Ksendzov\_Cours/Terminal/fruits/apples/buy.txt G:/"My Drive"/QA/Ksendzov\_Cours/Terminal/fruits

Комментарий: опция -i отвечает за подтверждение переноса, если уже есть файл с таким именем, чтобы не переписывал.

**2 вариант для команды mv.** Разница в том, что через опцию -t (target directory) указываю перемещение всех исходных файлов в директорию, которая указана в аргументе опции.

mv -i -t G:/"My Drive"/QA/Ksendzov\_Cours/Terminal/fruits G:/"My Drive"/QA/Ksendzov\_Cours/Terminal/fruits/apples/sale.txt G:/"My Drive"/QA/Ksendzov\_Cours/Terminal/fruits/apples/buy.txt

=====================

## **Как запустить sh скрипт из командной строки?**

Скриптовый файл должен быть c расширением script.sh.

Чтобы его запустить, надо зайти в каталог, где расположен скрипт, набрать название интерпретатора sh и первым параметром указать ваш файл hello.sh:

sh script.sh

Еще есть запуск командой из текущей папки:

./skript.sh

Чтобы каждый раз не указывать интерпретатор в терминале, можно сделать скрипт исполняемым. Для этого необходимо дать права на запуск из консоли через команду chmod +x и имя файла скрипта (опция -х отвечает за то, что дал права всем, кто будет запускать, а не только создавший юзер):

chmod +x script.sh

Можно указать интерпретатор внутри файла.  
На первой строчке после #! прописывам путь к bash-интерпретатору, по-умолчанию это /bin/bash (посмотреть можно командой whereis bash). Содержмиое скрипта hello.sh получается таким:  
#!/bin/bash (на первой строке)

Тут хорошо описано<https://www.youtube.com/watch?v=qZjnr66Cys4>

=====================

**Справка- команды**

<https://losst.pro/42-komandy-linux-kotorye-vy-dolzhny-znat>

Навигация

1. **cd (Change Directory) - сменить каталог**
2. **cd . — выход в текущую папку**
3. **cd .. — выход во внешнюю (родительскую) директорию**
4. **cd ../.. — Выйти на 2 уровня вверх**
5. **cd !$ — Перейти в только что созданную папку**
6. **pwd (Print Working Directory) — вывести рабочую директорию**
7. **ls (List) — вывести список содержимого, кроме скрытых**

С помощью флага **-l** мы можем вывести список прав доступа к файлам и каталогам, а также другие атрибуты (названия папок, размеры файлов и каталогов, дата и время изменения). Для просмотра скрытых файлов добавьте флаг **-a**

1. **find - поиск** [**https://losst.pro/komanda-find-v-linux**](https://losst.pro/komanda-find-v-linux)

#### **head / tail.** Ещё одна пара, но здесь у каждой команды своя область применения. Утилита [head](https://losst.pro/komanda-head-linux) выводит несколько первых строк из файла (голова), а [tail](https://losst.pro/komanda-tail-linux) выдает несколько последних строк (хвост). **По умолчанию каждая утилита выводит десять строк. Но это можно изменить с помощью опции -n (отменяет это ограничение). Ещё один полезный параметр -f, это сокращение от follow (следовать). Утилита постоянно выводит изменения в файле на экран.** Например, если вы хотите следить за лог файлом, вместо того, чтобы постоянно открывать и закрывать его, используйте команду tail -f.

про отслеживание журналов в реальном времени <https://andreyex.ru/linux/3-sposoba-prosmotra-zhurnalov-v-realnom-vremeni-v-linux/>

Работа с каталогами:

1. **mkdir (Make Directory) — создать новую директорию**
2. **mkdir {dir1,dir2,dir3,dir4} — Создает несколько каталогов в текущем местоположении**
3. **mkdir -p directory/path/newdir — Наиболее удобная опция -p (Parents), позволяет создать всю структуру подкаталогов одной командой, даже если они ещё не существуют.**
4. **rmdir — Удалить пустую папку**
5. **rm -r 1 — Удалить папку test с файлами внутри неё**
6. **rm -r name\_directory — удаление директории и все, что в ней находится**

Работа с файлами:

Создание:

1. **touch directory/test.txt — позволяет создавать файлы**
2. **touch dir/1.txt dir/2.txt — позволяет создавать несколько файлов**

Запись:

Cat - Печатает содержимое файла, переданного в параметре, в стандартный вывод. Если передать несколько файлов, команда склеит их. Также можно перенаправить вывод в ещё один файл с помощью символа '>'. Если нужно вывести только определенное количество строк, используйте опцию -n (Number). Подробнее [тут](https://losst.pro/komanda-cat-linux).

1. **cat — запись в файл**
2. **cat > directory/test.txt** пишем нужный текст

Ctrl+C — выход из режима редактирования

1. **cat directory/test.txt — вывести содержимое файла**
2. **cat >> directory/test.txt — запись в конце файла**

Удаление:

1. **rm Name\_directory/{1.txt,2.txt,3.txt} — удаляет несколько файлов**

Перемещение файлов:

1. **mv test1.txt test2.txt Directory — перенести несколько файлов из одной папки в другую**
2. **mv Directory1/test.txt Directory2 (назначение просто директория, если не надо переписывать название файла)**

Опция -i или --interactive при перезаписи существующих файлов спрашивать разрешение пользователя на подтверждение операции.

Опция -t или --target-director означает переместить все исходные файлы в директорию, которая указана в аргументе опции.

1. **mv app1/\*.\* app2 — Переместить все файлы из папки app1 в папку app2**
2. **mv filename1.json {путь к каталогу} && mv filename2.json {путь к другому каталогу} - если надо в одной строчке одно перемещение и второе файлы из разных папок**

Копирование файлов:

1. **cp Directory1/test.txt Directory2/test.txt - копирование файла**

#### **cp -** [**Копирование файлов и каталогов**](https://losst.pro/kopirovanie-fajlov-v-linux)**.** Она не копирует каталоги по умолчанию рекурсивно (то есть все поддиректории и все файлы в поддиректориях), поэтому не забудьте добавить опцию -r (Recursive) или -a (Archive). Последняя включает режим сохранения атрибутов, владельца и временного штампа в дополнение к рекурсивному копированию.

1. **Clear — очистить консоль**
2. стрелка вверх/вниз - пролистывание команд
3. , то использовать кавычкиесли название папки с пробелами
4. ping IP. Для того чтобы пропинговать IP адрес через командную строку. Так можно посмотреть время, необходимое для передачи данных с вашего компьютера на сервер (или компьютер другого игрока), а затем обратно на ваш компьютере.
5. Оператор AND (&&) будет выполнять вторую команду только в том случае, если при выполнении первой команды SUCCEEDS, т.е. состояние выхода первой команды равно «0» — программа выполнена успешно. Эта команда очень полезна при проверке состояния выполнения последней команды. Например, я хочу посетить сайт sedicomm.com с помощью команды links в терминале, но перед этим мне нужно проверить, является ли хост доступным или нет.

**В WSL (Windows Subsystem for Linux)**

**sudo apt-get update** - апдейт всех доступных приложений

**sudo apt-get install** htop - установка программы диспетчера задач htop

sudo (англ. Substitute User and do, дословно «подменить пользователя и выполнить»)