ДЗ 1 к лекции Terminal 12.06.23

Первая часть первого ДЗ Linux terminal (GitBash) commands

1) Посмотреть где я

pwd

2) Создать папку

mkdir folder\_1

3) Зайти в папку

cd folder\_1

4) Создать 3 папки

mkdir folder\_2 folder\_3 folder\_4

5) Зайти в любую папку

cd folder\_2

6) Создать 5 файлов (3 txt, 2 json)

touch file1.txt file2.txt file3.txt file4.json file5.json

7) Создать 3 папки

mkdir folder\_5 folder\_6 folder\_7

8) Вывести список содержимого папки

Заметка: Нахожусь в папке folder\_2 , поэтому выводом будет ее содержимое. Команда ls применяется для той директории, в которой находишься.

ls -la

9) + Открыть любой txt файл

Заметка: т.к. затем в этот пустой файл буду вводить текст, сразу использую > .

cat > file1.txt

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

Hello, World

11) + сохранить и выйти.

Ctrl + c

12) Выйти из папки на уровень выше

cd ..

—

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

Заметка: могу использовать, находясь в папке folder\_1, для вложенных папок folder\_2 и folder\_3.

Работает только одна команда за раз, две строки вместе не работает.

mv folder\_2/file1.txt folder\_3/file1.txt

mv folder\_2/file2.txt folder\_3/file2.txt

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

cp folder\_2/file3.txt folder\_4/file3.txt

cp folder\_2/file4.json folder\_3/file4.json

15) Найти файл по имени

Заметка: команда find рекурсивно (т.е. вызывая сама себя) просматривает все поддиректории, начиная с указанной директории, и выводит пути к файлам, удовлетворяющим заданному имени, т.е. если указать folder\_1, поищет по всем вложенным в нее папкам.

find /Users/margaritapetrosan/folder\_1 -name "file1.txt” —— работает (прописала весь путь)

find folder\_3 -name file1.txt -- прописала только папку и конкретное имя файла, тоже работает

find . -name "file\*" -- с точкой и \* тоже работает

find . -name file1.txt —- с точкой и конкретное имя тоже работает

16) просмотреть содержимое в реальном времени (команда grep) изучите как она работает.

Заметка: просмотреть содержимое в реальном времени позволит команда **tail** (вывод нужного количества последних строк) с опцией **-f** (останется активной и продолжит отображать новые строки, добавленные в файл после начала выполнения команды).

Команда **grep** - фильтрация вывода, т.е. выведет строки, содержащие шаблон (слово);

При чем сработало как для шаблона с “”, так и без: **grep Hello file1.txt**

*Дополнительно: -i: регистронезависимый поиск; -r или -R: выполняет рекурсивный поиск в указанной директории и ее поддиректориях; -v: выводит строки, которые НЕ содержат совпадения с указанным шаблоном; -l: выводит только имена файлов, содержащих совпадения, но не строки самих совпадений. Пр:* ***grep -i hello file1.txt***

Чтобы просмотреть содержимое в реальном времени с помощью команды grep в терминале, можно использовать комбинацию команд tail и grep (использую **конвейер**). Теперь по мере добавления новых строк в файл, вывод tail будет отфильтрован так, чтобы отображались только строки, содержащие совпадения с указанным шаблоном (слово “Hello”).

Вопрос: при сохранении изменений в файле (file1.txt), они не отображаются в терминале, в котором запущена команда tail -f file1.txt , пока не введу команду повторно. И аналогично с конвейером tail -f file1.txt | grep Hello file1.txt. Правильно понимаю, что у меня какая-то ошибка, и изменения вносимые в файл руками, должны отображаться в терминале, если там запущена команда tail -f сразу же после сохранения изменений в файле (понимаю это как режим реального времени)?

Пробовала:

находясь в папке folder\_1 прописывать путь: tail -f folder\_3/file1.txt --- изменения не появляются пока не введу команду повторно.

находясь в папке folder\_3 прописывать путь: tail -f folder\_3/file1.txt --- это ошибка, вообще не работает.

находясь в папке folder\_1 прописывать только имя файла: tail -f file1.txt --- изменения тоже не появляются пока не введу команду повторно.

tail -f file1.txt | grep Hello file1.txt

выход: Ctrl + c

17) вывести несколько первых строк из текстового файла

head -n 2 file1.txt

18) вывести несколько последних строк из текстового файла

tail -n 2 file1.txt

19) просмотреть содержимое длинного файла (команда less) изучите как она работает.

less f3.txt

Заметка: команда **less** имя\_файла позволяет просматривать содержимое длинных файлов постранично и осуществлять поиск внутри файла. Появится просмотрщик, где можно прокручивать содержимое файла. Основные команды, которые можно использовать внутри less:

* Стрелки вверх и вниз: Прокрутка на одну строку вверх или вниз.
* Пробел: Прокрутка на одну страницу вперед.
* B: Прокрутка на одну страницу назад.
* G: Переход к концу файла.
* /поисковый\_запрос: Поиск по заданному тексту в файле. Например, чтобы найти строку "ключевое\_слово", введите /ключевое\_слово.
* q: Выход из просмотрщика less.

Это только некоторые из основных команд. Доп инфа - нажать на клавишу h внутри less.

20) вывести дату и время

date

=========

Задание \*

1) Отправить http запрос на сервер.

http://162.55.220.72:5006/terminal-hw-request

Отправила curl http://162.55.220.72:5006/terminal-hw-request

Получила ответ с заданием отправить новый запрос, внеся данные. Внесла свои данные и отправила (использовала для ссылки “”)

curl "http://162.55.220.72:5005/get\_method?name=Margarita&age=27"

Получила ответ:

[

"Margarita",

"27"

]

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

1. Создала файл с расширением sh: script.sh (в папке folder\_1):

touch script.sh

2. Открыла его в режиме редактирования (для vim — vim myscript.sh ):

nano script.sh

3. Внесла в него команды из пунктов 3, 4, 5, 6, 7, 8, 13 и первую строку #! /bin/bash руками:

#! /bin/bash

cd folder\_1

mkdir folder\_2 folder\_3 folder\_4

cd folder\_2

touch file1.txt file2.txt file3.txt file4.json file5.json

mkdir folder\_5 folder\_6 folder\_7

ls -la

mv folder\_2/file1.txt folder\_3/file1.txt

mv folder\_2/file2.txt folder\_3/file2.txt

4. Ctrl+о — сохранила файл (предложило имя для сохранения, нажала Enter)

5. Ctrl+х — вышла из nano (для vim :wq или :x — выйти и сохранить)

Cнова оказалась в обычном режиме в терминале.

6. Написала команду sh в терминале (в терминале можно вводить команды для выполнения, как делала выше, а можно sh имя\_скрипта (файла), тогда выполнится скрипт) и указала имя скрипта как аргумент после команды запуска оболочки:

sh script.sh

Заметка: т.к. такие папки уже существовали, то дубликаты не создались:

“mkdir: folder\_7: File exists”

Поэтому для проверки заменила в скрипте имя директории на несуществующее (folder\_74) -- директория folder\_74 создалась, скрипт работает.

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