**Компания A1 Project в городе Душанбе**

**(Направление: DevOps и администрирование БД)**

**Отчет по практическому заданию: "Работа с терминалом и написание BASH-скриптов"**

**(Итог первого месяца)**

Выполнил: стажёр Олимов Амир

Руководитель: менеджер по направлению DevOps и администрированию БД Темур Паллаев

**Душанбе – 2024**

**Введение:**

Этот отчёт посвящён практическому заданию по работе в терминале Linux и созданию BASH-скриптов. В нём пошагово показано, как создавать и управлять папками и файлами, настраивать права доступа, использовать разные команды для работы с данными и автоматизировать задачи с помощью скриптов. Каждое действие подробно объясняется, а для наглядности приведены примеры и результаты выполнения команд.

Все команды, представленные в отчёте, были выполнены в приложении MobaXterm и были скопированы и вставлены в данный отчёт. Также результаты команд были скопированы оттуда (без каких-либо изменений вручную).

Примеры применения скриптов были приведены в отчёт для убеждения в том, что скрипты работают правильно.

**Важные моменты и форматирование текста в отчёте**

Тексты, выделенные в отчёте, можно описать следующим образом:

Серый цвет в тексте обозначает команды, которые пользователь вводил для выполнения задачи, а также текст, который он вводил вручную в процессе решения. Это помогает выделить шаги работы и делает их понятными.

Тексты в прямоугольных рамках показывают результаты выполнения команд. Они включают информацию, которую выводит терминал в ответ на вводимые команды. Это помогает визуально разделить вводимые команды и их вывод, что делает документ более читабельным.

Красным цветом выделены ключевые части результатов. Это помогает быстро обратить внимание на важные моменты и избежать путаницы в больших объёмах вывода.

Обычный текст используется для описания задачи, объяснения команд и их шагов, а также для пояснения процесса решения.

Такое форматирование делает документ структурированным и лёгким для восприятия.

**Задание 1**

Под своей учеткой создать директорию **lab1**. В ней создать три файла - **file1.txt, file2.txt, file3.txt**. Зафиксировать текущие права на файлы и на каталог (скрином или текстом). Установить следующие права на файлы:

**file1.txt:** только владелец может читать и писать.

**file2.txt:** только владелец и группа могут читать и писать, остальные не имеют доступа.

**file3.txt:** все пользователи могут читать, но только владелец может писать.

Создайте новую группу - **lab\_group**. Добавьте своего пользователя в эту группу. Измените группу для каталога **lab1** и всех файлов внутри него на **lab\_group**. Зафиксируйте текущие права на файлы и каталог (скрином или текстом).

В отчете надо отразить команды, которыми вы осуществили задание 1, описать, что команды делают, и приложить сделанные скрины или текст.

**Решение:**

Создаём директорию lab1:

[amir@amir final\_project]$ mkdir lab1

Заходим в каталог:

[amir@amir final\_project]$ cd lab1

Смотрим абсолютный путь:

[amir@amir lab1]$ pwd

/home/amir/tasks\_first\_month/final\_project/lab1

Создаём файлы file1.txt, file2.txt и file3.txt внутри данного каталога:

[amir@amir lab1]$ touch file1.txt file2.txt file3.txt

Просматриваем список файлов и их права доступа ((r – права на чтение, w – права на запись, x – права на выполнение); (начиная со второй позиции первые 3 – права пользователя, последующий 3 – права группы, последние 3 – права остальных пользователей)):

[amir@amir lab1]$ ls -l

total 0

-rw-rw-r--. 1 amir amir 0 Nov 14 07:09 file1.txt

-rw-rw-r--. 1 amir amir 0 Nov 14 07:09 file2.txt

-rw-rw-r--. 1 amir amir 0 Nov 14 07:09 file3.txt

Изменяем права доступа для каждого файла внутри директории lab1 (**file1.txt:** только владелец может читать и писать (600 – -rw-------); **file2.txt:** только владелец и группа могут читать и писать, остальные не имеют доступа (660 – -rw-rw----) ; **file3.txt:** все пользователи могут читать, но только владелец может писать (644 – -rw-r--r--)):

[amir@amir lab1]$ chmod 600 file1.txt

[amir@amir lab1]$ chmod 660 file2.txt

[amir@amir lab1]$ chmod 644 file3.txt

Просматриваем список файлов и их права доступа ((r – права на чтение, w – права на запись, x – права на выполнение); (начиная со второй позиции первые 3 – права пользователя, последующий 3 – права группы, последние 3 – права остальных пользователей)):

[amir@amir lab1]$ ls -l

total 0

-rw-------. 1 amir amir 0 Nov 14 07:09 file1.txt

-rw-rw----. 1 amir amir 0 Nov 14 07:09 file2.txt

-rw-r--r--. 1 amir amir 0 Nov 14 07:09 file3.txt

Создаём группу lab\_group:

[amir@amir lab1]$ sudo groupadd lab\_group

Так как ввели эту команду от имени администратора поэтому нас спрашивает пароль. Введём пароль:

[sudo] password for amir:

Добавляем пользователя amir в группу lab\_group:

[amir@amir lab1]$ sudo usermod -aG lab\_group amir

Выходим из каталога lab1 обратно:

[amir@amir lab1]$ cd ..

Меняем группу для каталога lab1 и всех его внутренних файлов на lab\_group:

[amir@amir final\_project]$ sudo chgrp -R lab\_group lab1

Так как ввели эту команду от имени администратора поэтому нас спрашивает пароль. Введём пароль:

[sudo] password for amir:

Проверяем, что каталог теперь принадлежит группе lab\_group:

[amir@amir final\_project]$ ls -l

total 4

drwxrwxr-x. 2 amir lab\_group 4096 Nov 14 07:09 lab1

Заходим в директорию:

[amir@amir final\_project]$ cd lab1

Проверяем, что файлы теперь принадлежат группе lab\_group:

[amir@amir lab1]$ ls -l

total 0

-rw-------. 1 amir lab\_group 0 Nov 14 07:09 file1.txt

-rw-rw----. 1 amir lab\_group 0 Nov 14 07:09 file2.txt

-rw-r--r--. 1 amir lab\_group 0 Nov 14 07:09 file3.txt

**Задание 2**

Под своей учеткой создать директорию **lab2**. В ней создать файл **input.txt**, ввести в него 6-7 разных строк текста. Также, перенаправьте содержимое директории **lab2** в файл **directory\_list.txt**. Добавьте текущее время и дату в конец файла **directory\_list.txt**. Отобразить содержимое, зафиксировать его скрином или текстом.

Отсортируйте строки в файле **input.txt** и выведите результат на экран (для этого используйте команду **sort** с пайпом). Отфильтруйте строки, содержащие букву **'a'**, из файла **input.txt**, используя команду **grep** и **пайп**. Подсчитайте количество строк в файле **input.txt** используя команду **wc -l**. Зафиксировать результаты скрином или текстом.

Отсортируйте строки в файле **input.txt**, выберите строки, содержащие букву **'e'**, и подсчитайте их количество, используя комбинацию команд **sort**, **grep** и **wc -l** с пайпами. Зафиксировать результаты скрином или текстом.

Выполните команду **ls non\_existent\_file** и перенаправьте сообщение об ошибке в файл **errors.txt**. Просмотрите содержимое файла **errors.txt**. Зафиксировать результаты скрином или текстом.

В отчете надо отразить команды, которыми вы осуществили задание 2, описать, что команды делают, и приложить сделанные скрины или текст.

**Решение:**

Создаём директорию lab2:

[amir@amir final\_project]$ mkdir lab2

Заходим в директорию lab2:

[amir@amir final\_project]$ cd lab2

Одновременно создание и добавление значения в файле input.txt с помощью команды nano:

[amir@amir lab2]$ nano input.txt

Просматриваем содержимое файла input.txt:

[amir@amir lab2]$ cat input.txt

My name is Olimov Amir. I am a third-year student in Applied Mathematics

and Informatics at the MSU branch in Dushanbe. I am passionate about data

analysis, algorithm development, and software engineering, and I am actively

building skills in these areas to prepare for a career in technology and

research. My studies focus on applying mathematical methods and programming

to solve complex problems, and I am eager to contribute to innovative projects

in the field.

Создаём список файлов в директории и сохраняем его в файл directory\_list.txt:

[amir@amir lab2]$ ls > directory\_list.txt

Просматриваем содержимое файла directory\_list.txt:

[amir@amir lab2]$ cat directory\_list.txt

directory\_list.txt

input.txt

Добавляем текущую дату и время в файл directory\_list.txt с помощью команды date:

[amir@amir lab2]$ date >> directory\_list.txt

Просматриваем содержимое файла directory\_list.txt:

[amir@amir lab2]$ cat directory\_list.txt

directory\_list.txt

input.txt

Fri Nov 15 14:57:49 MSK 2024

Сортируем содержимое файла input.txt с помощью команды sort:

[amir@amir lab2]$ sort input.txt

analysis, algorithm development, and software engineering, and I am actively

and Informatics at the MSU branch in Dushanbe. I am passionate about data

building skills in these areas to prepare for a career in technology and

in the field.

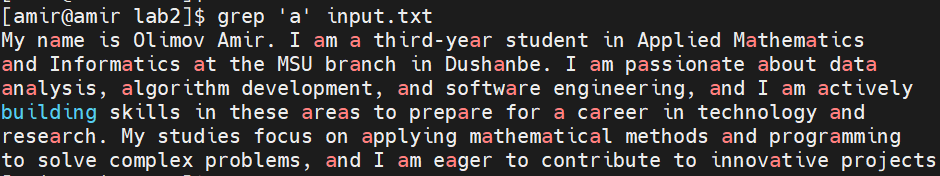
My name is Olimov Amir. I am a third-year student in Applied Mathematics

research. My studies focus on applying mathematical methods and programming

to solve complex problems, and I am eager to contribute to innovative projects

Ищем все строки, содержащие букву 'a' в файле input.txt (только последняя строка в тексте не содержит ‘a’):

[amir@amir lab2]$ grep 'a' input.txt



С помощью команды man изучаем команду wc:

[amir@amir lab2]$ man wc

Подсчитываем количество строк в файле input.txt с помощью команды wc (word count, а флажок l указывает на то, что надо посчитать только строки):

[amir@amir lab2]$ wc -l input.txt

7 input.txt

Сортируем содержимое input.txt с помощью команды sort, ищем строки с буквой 'e' используя команду grep и подсчитываем их количество с помощью команды wc, и для объединение комнад используем знак пайп – |, в результате получим количество отсортированных строк содержащие букву 'e'в файле input.txt:

[amir@amir lab2]$ sort input.txt | grep 'e' | wc -l

7

Обратимся к несуществующему файлу и записываем ошибки в файл errors.txt:

[amir@amir lab2]$ ls non\_existent\_file 2> errors.txt

Просматриваем содержимое файла errors.txt:

[amir@amir lab2]$ cat errors.txt

ls: cannot access non\_existent\_file: No such file or directory

**Задание 3**

BASH-скрипты (все скрипты надо снабдить подробными комментариями). Заметьте, что тут могут быть использованы концепции, которые мы ранее не проходили — это намеренно, надо будет поискать и имплементировать найденное в свои скрипты.

**Задание 3.1**

Напишите скрипт, который запрашивает имя пользователя и выводит приветственное сообщение с введенным именем.

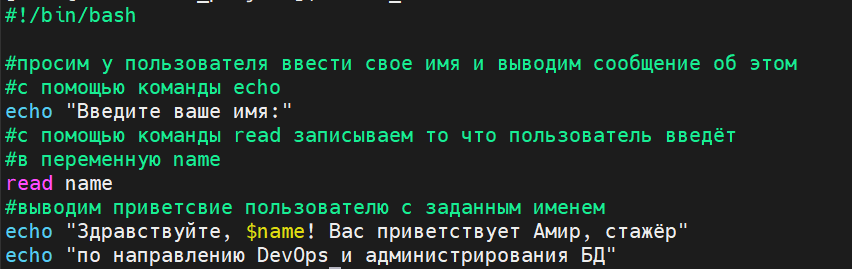
**Решение:**

Одновременно создаём и добавляем значения в файле 3\_1.sh с помощью команды nano:

[amir@amir final\_project]$ nano 3\_1.sh

Просматриваем содержимое файла 3\_1.sh после написание скрипта:

[amir@amir final\_project]$ cat 3\_1.sh



Просмотр список файлов в текущей директории (заметим, что у файла 3\_1.sh ни у кого нет права доступа на выполнение файла):

[amir@amir final\_project]$ ls -l

total 16

-rw-rw-r--. 1 amir amir 300 Nov 15 17:33 3\_1.sh

-rw-rw-r--. 1 amir amir 58 Nov 15 14:57 directory\_list.txt

-rw-rw-r--. 1 amir amir 63 Nov 15 17:01 errors.txt

-rw-rw-r--. 1 amir amir 472 Nov 15 16:47 input.txt

Изменяем права доступа для файла 3\_1.sh для того чтобы у владельца были права на выполнения (запуска) файла (**7 (rwx)** – для владельца файла: чтение, запись и выполнение; **6 (rw-)** – для группы: чтение и запись; **4 (r--)** – для остальных: только чтение) чтобы можно было выполнять скрипты:

[amir@amir final\_project]$ chmod 764 3\_1.sh

Просматриваем обновлённый список файлов с изменёнными правами:

[amir@amir final\_project]$ ls -l

total 16

-rwxrw-r--. 1 amir amir 300 Nov 15 17:33 3\_1.sh

-rw-rw-r--. 1 amir amir 58 Nov 15 14:57 directory\_list.txt

-rw-rw-r--. 1 amir amir 63 Nov 15 17:01 errors.txt

-rw-rw-r--. 1 amir amir 472 Nov 15 16:47 input.txt

**Пример использования скрипта**:

Запуск скрипта 3\_1.sh:

[amir@amir final\_project]$ bash 3\_1.sh

Введите ваше имя:

Сино

Здравствуйте, Сино! Вас приветствует Амир, стажёр

по направлению DevOps и администрирования БД

**Задание 3.2**

Напишите скрипт, который автоматизирует установку прав доступа и изменение владельца для всех файлов и каталогов внутри заданного каталога. При выполнении скрипта, у всех файлов в каталоге должны устанавливаться владелец user1 и группа lab\_group. Также, все файлы должны получать права, при которых у владельца файла есть все права, у членов группы права на чтение и запись, у остальных - только на чтение.

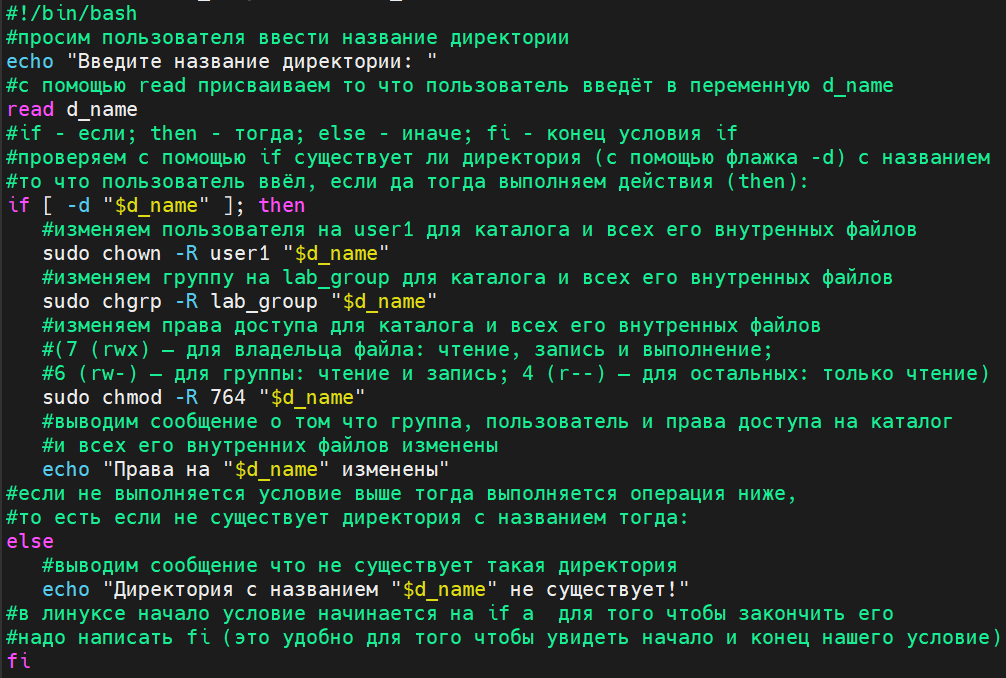
**Решение:**

Одновременно создаём и добавляем значения в файле 3\_2.sh с помощью команды nano:

[amir@amir final\_project]$ nano 3\_2.sh

Просматриваем содержимое файла 3\_2.sh после написание скрипта:

[amir@amir final\_project]$ cat 3\_2.sh



Изменяем права доступа для файла 3\_2.sh для того чтобы у владельца были права на выполнения (запуска) файла (**7 (rwx)** – для владельца файла: чтение, запись и выполнение; **6 (rw-)** – для группы: чтение и запись; **4 (r--)** – для остальных: только чтение) чтобы можно было выполнять скрипты:

[amir@amir final\_project]$ chmod 764 3\_2.sh

**Пример использования скрипта**:

Создаём директорию amir:

[amir@amir final\_project]$ mkdir amir

Переходим в директорию amir:

[amir@amir final\_project]$ cd amir

Создаём пустых файлов внутри директории amir:

[amir@amir amir]$ touch 1.txt 2.log 3.sh

Просматриваем содержимое директории amir:

[amir@amir amir]$ ls -l

total 0

-rw-rw-r--. 1 amir amir 0 Nov 16 09:32 1.txt

-rw-rw-r--. 1 amir amir 0 Nov 16 09:32 2.log

-rw-rw-r--. 1 amir amir 0 Nov 16 09:32 3.sh

Выходим из директории amir

[amir@amir amir]$ cd ..

Просмотр списка файлов:

[amir@amir final\_project]$ ls -l

Просматриваем содержимое директории:

total 28

-rwxrw-r--. 1 amir amir 273 Nov 15 17:43 3\_1.sh

-rwxrw-r--. 1 amir amir 356 Nov 16 06:47 3\_2.sh

drwxrw-r--. 2 amir amir 4096 Nov 16 09:32 amir

drwxrwxr-x. 2 amir lab\_group 4096 Nov 14 07:09 lab1

drwxrwxr-x. 2 amir amir 4096 Nov 16 05:08 lab2

Запуск скрипта 3\_2.sh:

[amir@amir final\_project]$ bash 3\_2.sh

Введите название директории:

amir

Права на amir изменены

Просмотр списка файлов директории amir после обновлёнными правами (Замечаем, что пользователь и группа для каталога amir изменился на user1 И lab\_group соответсвенно):

[amir@amir final\_project]$ ls -l

total 28

-rwxrw-r--. 1 amir amir 273 Nov 15 17:43 3\_1.sh

-rwxrw-r--. 1 amir amir 371 Nov 16 09:47 3\_2.sh

drwxrw-r--. 2 user1 lab\_group 4096 Nov 16 09:32 amir

drwxrwxr-x. 2 amir lab\_group 4096 Nov 14 07:09 lab1

drwxrwxr-x. 2 amir amir 4096 Nov 16 05:08 lab2

Попытка входа в директорию amir (так как мы поменяли пользователя файла на user1, поэтому у на (у пользователя amir) нет доступа на выполнение каталога (то есть не можем зайти в каталог и просматривать его содержимое)):

[amir@amir final\_project]$ cd amir

-bash: cd: amir: Permission denied

Просматриваем группу пользователя amir и убедимся что мы в этой группе, но так как мы измелили права доступа файла amir на 764 – rwxrw-r--, не смотря на то что мы в группе lab\_group, мы не можем зайти в этот каталог, так как у данного права на выполнение для групп нет:

[amir@amir final\_project]$ groups

amir wheel lab\_group

**Задание 3.3**

Напишите скрипт, который принимает имя файла в качестве аргумента, отсортировывает его содержимое и выводит результат в новый файл.

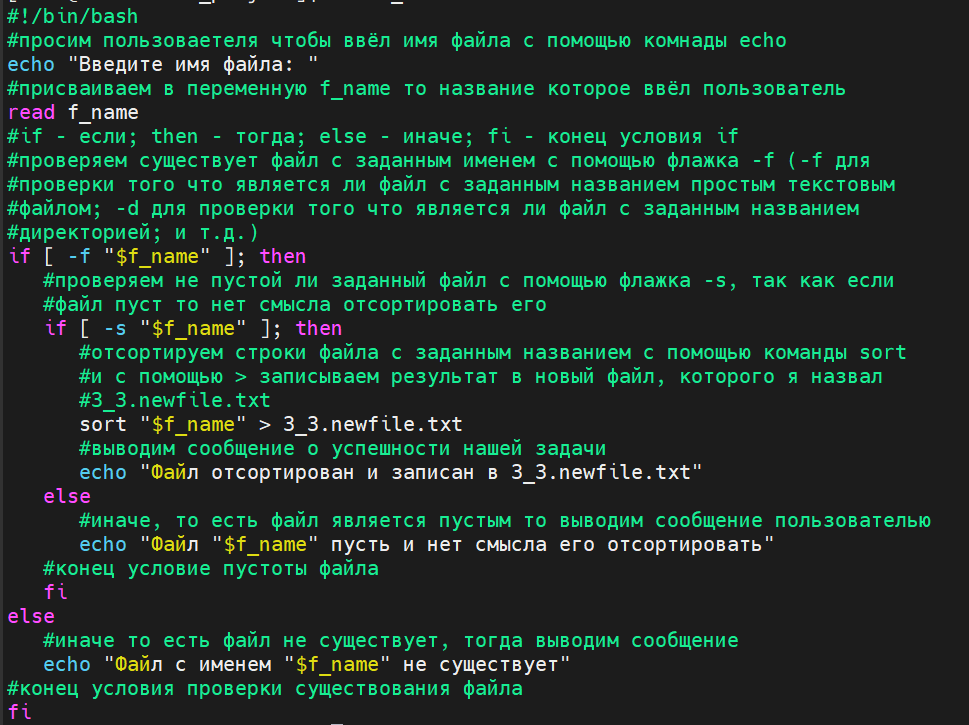
**Решение:**

Одновременно создание и добавление значения в файле 3\_3.sh с помощью команды nano:

[amir@amir final\_project]$ nano 3\_3.sh

Просмотр содержимого файла 3\_3.sh после написание скрипта:

[amir@amir final\_project]$ cat 3\_3.sh



Изменение прав доступа для файла 3\_3.sh (**7 (rwx)** – для владельца файла: чтение, запись и выполнение; **6 (rw-)** – для группы: чтение и запись; **4 (r--)** – для остальных: только чтение) чтобы можно было выполнять скрипты:

[amir@amir final\_project]$ chmod 764 3\_3.sh

Просматриваем список после обновления права доступа на файл 3\_3.sh:

[amir@amir final\_project]$ ls -l

total 16

-rwxrw-r--. 1 amir amir 273 Nov 15 17:43 3\_1.sh

-rwxrw-r--. 1 amir amir 356 Nov 16 04:47 3\_2.sh

-rwxrw-r--. 1 amir amir 122 Nov 16 05:36 3\_3.sh

drwxrwxr-x. 2 amir lab\_group 4096 Nov 14 07:09 lab1

drwxrwxr-x. 2 amir amir 4096 Nov 16 05:08 lab2

**Пример использования скрипта**:

Запускаем скрипт 3\_3.sh для несуществующего файла:

[amir@amir final\_project]$ bash 3\_3.sh

Введите имя файла:

f.txt

Файл с именем f.txt не существует

Создаём пустой файл для убеждения того, что скрипт которую мы написали корректно работает:

[amir@amir final\_project]$ touch f.txt

Запускаем скрипт 3\_3.sh для существующего файла f.txt с пустым значением:

[amir@amir final\_project]$ bash 3\_3.sh

Введите имя файла:

f.txt

Файл f.txt пусть и нет смысла его отсортировать и вывести его значение в новый файл

Запускаем скрипт 3\_3.sh для существующего файла file.txt с непустым значением:

[amir@amir final\_project]$ bash 3\_3.sh

Введите имя файла:

file.txt

Файл отсортирован и записан в 3\_3.newfile.txt

Просматриваем содержимое файла 3\_3.newfile.txt после создания и записи отсортированными значениями строк заданного файла:

[amir@amir final\_project]$ cat 3\_3.newfile.txt

1

2

4

4

5

6

6

7

8

**Задание 3.4**

Напишите скрипт, который запрашивает у пользователя число и проверяет, является ли оно положительным, отрицательным или нулем. Выведите соответствующее сообщение.

**Решение:**

Одновременно создание и добавление значения в файле 3\_4.sh с помощью команды nano:

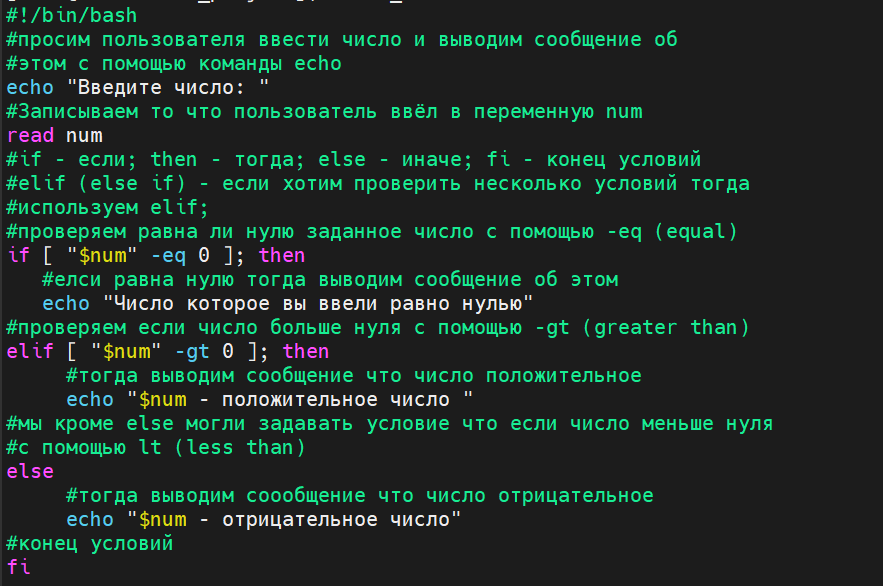
[amir@amir final\_project]$ nano 3\_4.sh

Изменение прав доступа для файла 3\_4.sh (**7 (rwx)** – для владельца файла: чтение, запись и выполнение; **6 (rw-)** – для группы: чтение и запись; **4 (r--)** – для остальных: только чтение) чтобы можно было выполнять скрипты:

[amir@amir final\_project]$ chmod 764 3\_4.sh

Просмотр содержимого файла 3\_4.sh после написание скрипта:

[amir@amir final\_project]$ cat 3\_4.sh



**Пример использования скрипта**:

Запускаем скрипт 3\_4.sh в случае положительных чисел:

[amir@amir final\_project]$ bash 3\_4.sh

Введите число:

4

4 - положительное число

Запускаем скрипт 3\_4.sh в случае отрицательных чисел:

[amir@amir final\_project]$ bash 3\_4.sh

Введите число:

-6

-6 - отрицательное число

Запускаем скрипт 3\_4.sh в случае 0:

[amir@amir final\_project]$ bash 3\_4.sh

Введите число:

0

Число которое вы ввели равно нулью

**Задание 3.5**

Напишите скрипт, который принимает имя файла в качестве аргумента и выводит информацию о файле (существует ли файл, является ли он каталогом или обычным файлом, права доступа и размер).

**Решение:**

Одновременно создание и добавление значения в файле 3\_5.sh с помощью команды nano:

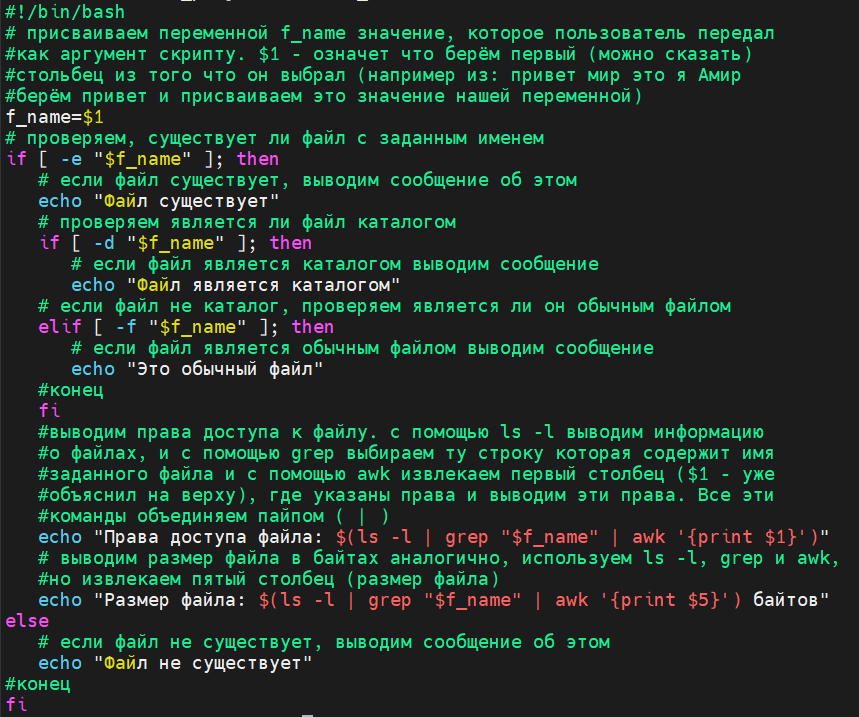
[amir@amir final\_project]$ nano 3\_5.sh

Изменение прав доступа для файла 3\_5.sh (**7 (rwx)** – для владельца файла: чтение, запись и выполнение; **6 (rw-)** – для группы: чтение и запись; **4 (r--)** – для остальных: только чтение) чтобы можно было выполнять скрипты:

[amir@amir final\_project]$ chmod 764 3\_5.sh

Просмотр содержимого файла 3\_5.sh после написание скрипта:

[amir@amir final\_project]$ cat 3\_5.sh



[amir@amir final\_project]$ bash 3\_5.sh lab1

Файл сущеcтвует

Файл является каталогом

Права доступа файла: drwxrwxr-x.

Размер файла: 4096 байтов

**Задание 3.6**

Напишите скрипт, который создает резервную копию заданного каталога, сохраняя ее в указанное место и добавляя текущую дату к имени архива. Выполните его, архивировав все скрипты и каталоги, использованные в данном практическом задании.

**Решение:**

Одновременно создаём и добавляем значения в файле 3\_6.sh с помощью команды nano:

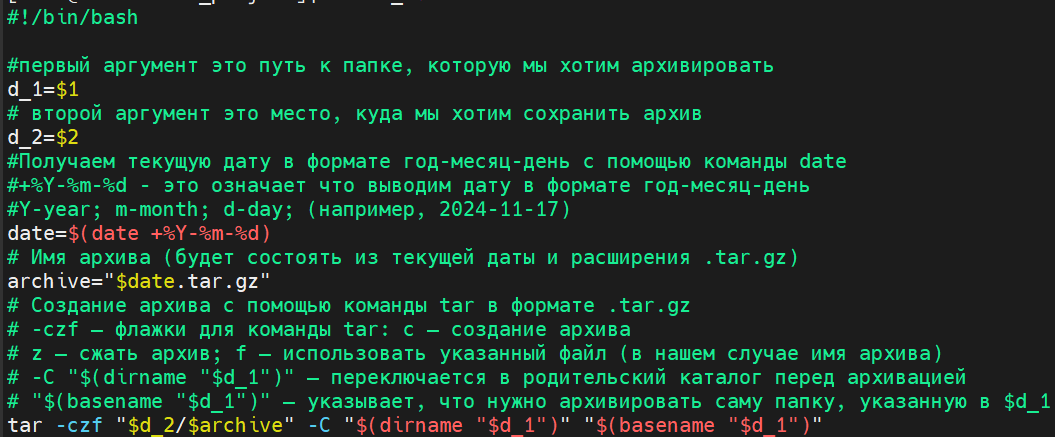
[amir@amir final\_project]$ nano 3\_6.sh

Изменяем права доступа для файла 3\_6.sh для того чтобы у владельца были права на выполнения (запуска) файла (**7 (rwx)** – для владельца файла: чтение, запись и выполнение; **6 (rw-)** – для группы: чтение и запись; **4 (r--)** – для остальных: только чтение) чтобы можно было выполнять скрипты:

[amir@amir final\_project]$ chmod 764 3\_6.sh

Просматриваем содержимое файла 3\_6.sh после написание скрипта:

[amir@amir final\_project]$ cat 3\_6.sh



**Задание:**

Все скрипты выложить на гитхаб, там же приложить архив, сделанный скриптом 3.6.

**Cсылка на гитхаб:**

[**https://github.com/AMIROLIMI/DevOps-first-month**](https://github.com/AMIROLIMI/DevOps-first-month)

(Нажмите на CTRL и нажмите ссылку)