# ОГЛАВЛЕНИЕ

[ФОРМУЛИРОВКА ЗАДАНИЙ 3](#_kao2wmz5e4yk)

[ВЫПОЛНЕНИЕ ЗАДАНИЙ 4](#_dx3vw11t8svt)

[Часть 1 4](#_hek9xwymqpqi)

[Часть 2 9](#_yrkfpsv8je8c)

[ВЫВОД 12](#_6xovosg9zgwg)

# ФОРМУЛИРОВКА ЗАДАНИЙ

Задания на выполнение **1 части** практической работы:

1. Напишите сценарий, который выводит дату, время, список зарегистрировавшихся пользователей, и uptime системы и сохраняет эту информацию в файл.

2. Напишите сценарий, который выводит содержимое любого каталога или сообщение о том, что его не существует.

3. Напишите сценарий, который с помощью цикла прочитает файл и выведет его содержимое.

4. Напишите сценарий, который с помощью цикла выведет список файлов и директорий из текущего каталога, укажет, что есть файл, а что директория.

5. Напишите сценарий, который подсчитает объем диска, занимаемого директорией. В качестве директории можно выбрать любую директорию в системе.

6. Напишите сценарий, который выведет список всех исполняемых файлов в директории, для которых у текущего пользователя есть права на исполнение.

Задания на выполнение **2 части** практической работы:

1. Определение зависимостей проекта.

2. Создание виртуального окружения.

3. Написание скрипта запуска приложения на новой системе.

# ВЫПОЛНЕНИЕ ЗАДАНИЙ

# Часть 1

Напишем сценарий, который выводит дату, время список зарегистрировавшихся пользователей, и uptime системы и сохраняет эту информацию в файл.

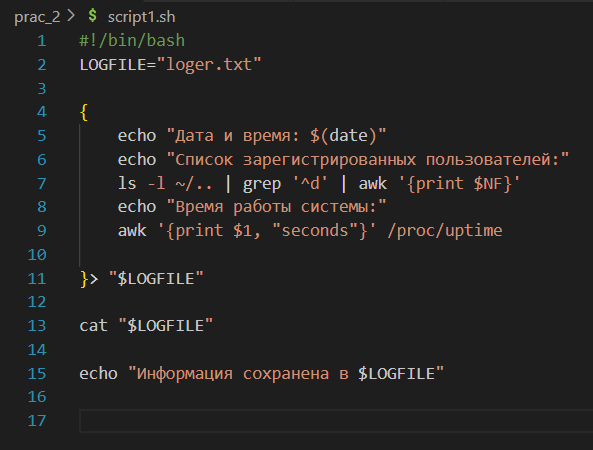


Рисунок 1 – Исходный код

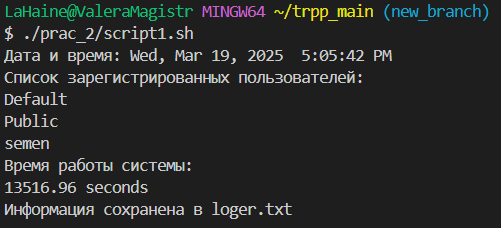


Рисунок 2 – Пример выполнения

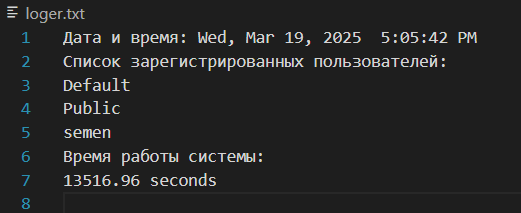


Рисунок 3 – Содержание файла с результатом программы

Напишем сценарий, который выводит содержимое любого каталога или сообщение о том, что его не существует.

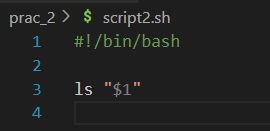


Рисунок 4 – Исходный код

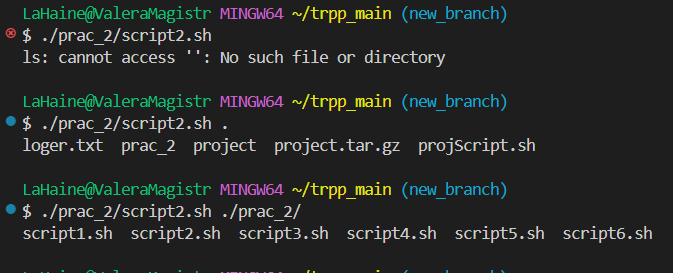


Рисунок 5 – Пример выполнения

Напишем сценарий, который с помощью цикла прочитает файл и выведет его содержимое.

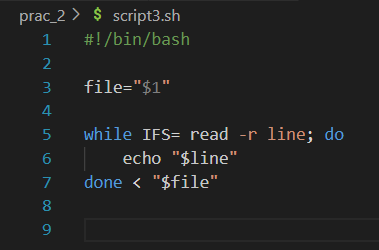


Рисунок 6 – Исходный код

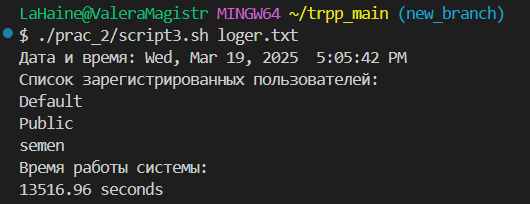


Рисунок 7 – Пример выполнения

Напишем сценарий, который с помощью цикла выведет список файлов и директорий из текущего каталога, укажет, что есть файл, а что директория.



Рисунок 8 – Исходный код

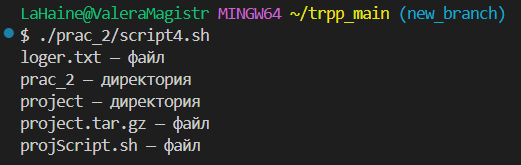


Рисунок 9 – Пример выполнения

Напишем сценарий, который подсчитает объем диска, занимаемого директорией. В качестве директории можно выбрать любую директорию в системе.

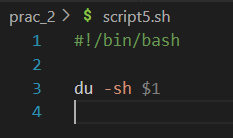


Рисунок 10 – Исходный код

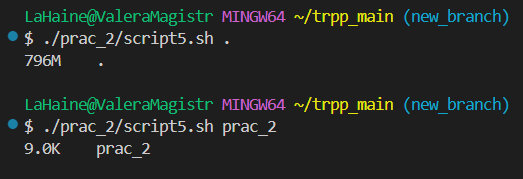


Рисунок 11 – Пример выполнения

Напишем сценарий, который выведет список всех исполняемых файлов в директории, для которых у текущего пользователя есть права на исполнение.

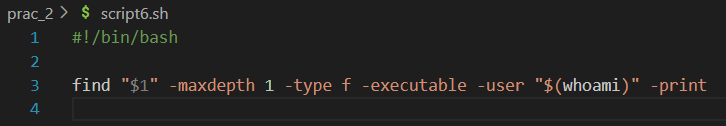


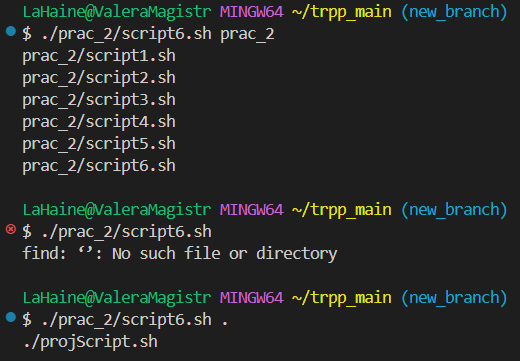
Рисунок 12 – Исходный код

Рисунок 13 – Пример выполнения

# Часть 2

Определим зависимости проекта.

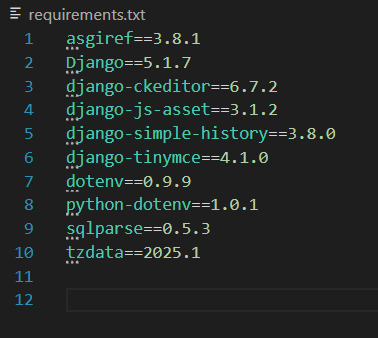


Рисунок 14 – Зависимости проекта

Напишем скрипт, который будет скачивать проект, создавать виртуальное окружение и настраивать его под проект.

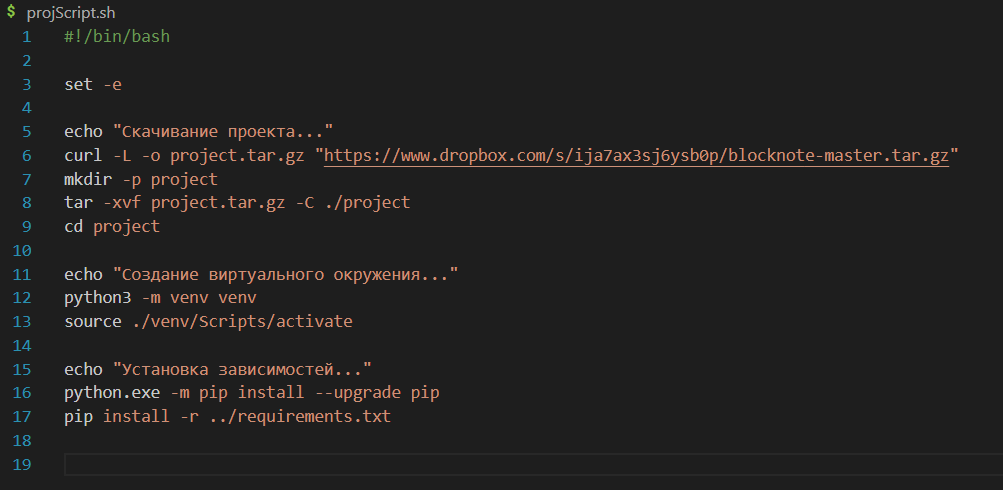


Рисунок 15 – Исходный код

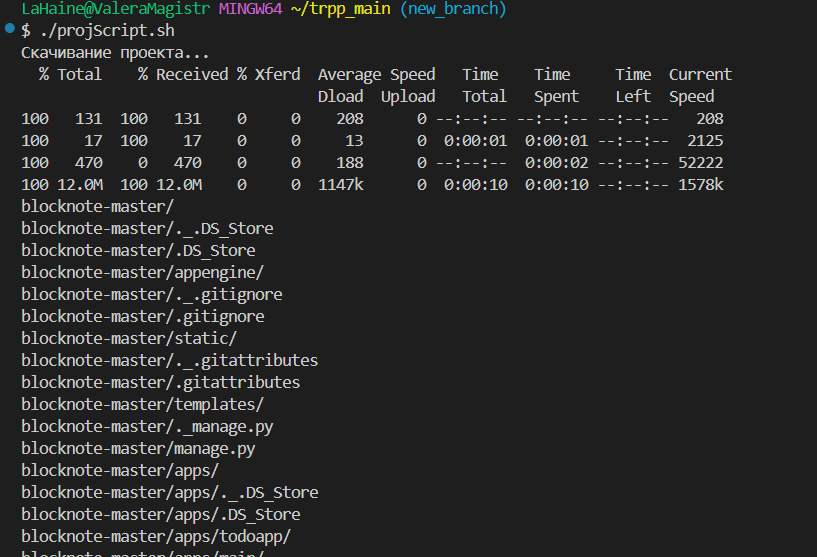


Рисунок 16 – Пример части выполнения

Улучшим имеющийся скрипт, до скрипта, который может запускаться на новой системе.

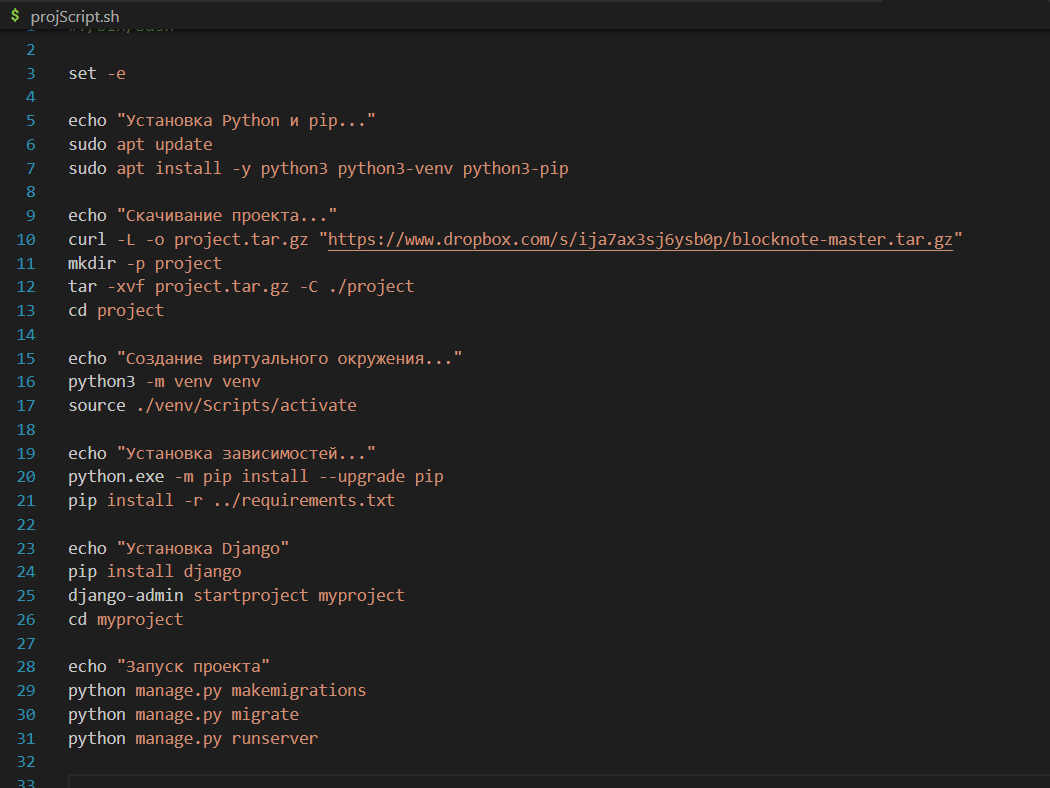


Рисунок 17 – Исходный код

# ВЫВОД

В ходе практической работы были изучены основы Bash-скриптов и их применение для автоматизации развертывания и запуска Django-проекта. Рассмотрены основные команды Bash, такие как работа с файлами и папками, управление переменными, проверка условий и использование циклов.

Практическое применение Bash-скриптов позволило автоматизировать процесс развертывания проекта и сделать запуск веб-приложения более удобным и быстрым.