|  |
| --- |
|  |
| |  |  |  | | --- | --- | --- | |  |  |  | | МИНОБРНАУКИ РОССИИ | | | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |   Институт Информационных технологий  Кафедра Математического обеспечения и стандартизации информационных технологий  **Отчет по практической работе №3**  по дисциплине «Технология разработки программных приложений»  по теме «Основы работы с Bash Scriptами»   |  |  | | --- | --- | | **Выполнил:**  Студент группыИКБО-20-22 | Шумахер М. Е. | | **Проверил:** | преподаватель Мельников Д. А. |   МОСКВА 2024 г. |

**СОДЕРЖАНИЕ**

[Задание 3](#_Toc160900217)

[Выполнение заданий 4](#_Toc160900218)

[Вывод 8](#_Toc160900219)

Задание

**Цель работы**: получить навыки по работе с Bash Script’ами.

**Задание для выполнения**

Часть 1. Базовые Bash скрипты

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.

Часть 2. Развертка и запуск проекта при помощи Bash Script

1. Определение зависимостей проекта

2. Создание виртуального окружения

3. Написание скрипта запуска приложения на новой системе

Выполнение заданий

Напишем сценарий, который выводит дату, время, список зарегистрировавшихся пользователей, и uptime системы и сохраняет эту информацию в файл (рис. 1-2).

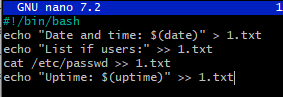


Рисунок 1 – Скрипт для вывода даты, времени, списка пользователей

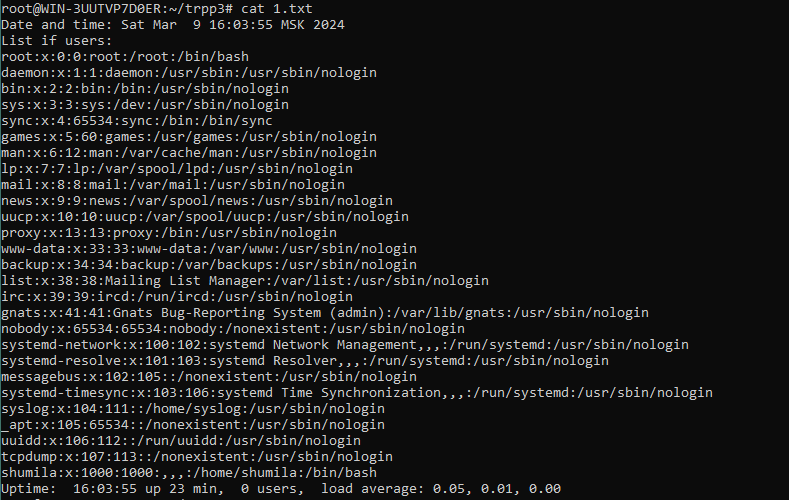


Рисунок 2 – Вывод работы скрипта

Напишем сценарий, который выводит содержимое любого каталога или сообщение о том, что его не существует (рис. 3-4).

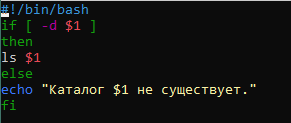


Рисунок 3 – Скрипт для вывода содержимого каталога

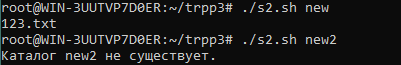


Рисунок 4 – Вывод работы скрипта

Напишем сценарий, который выводит содержимое любого каталога или сообщение о том, что его не существует (рис. 5-6).

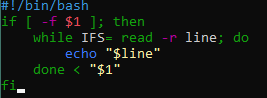


Рисунок 5 – Скрипт для считывания содержимого файла

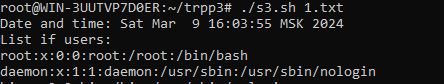


Рисунок 6 – Вывод работы скрипта

Напишем сценарий, который с помощью цикла выведет список файлов и директорий из текущего каталога, укажет, что есть файл, а что директория (рис. 7-8).

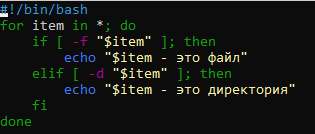


Рисунок 7 – Скрипт для вывода фалов и директорий

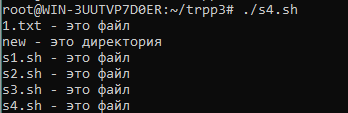


Рисунок 8 – Вывод работы скрипта

Напишем сценарий, который подсчитает объем диска, занимаемого директорией. В качестве директории можно выбрать любую директорию в системе (рис. 9-10).

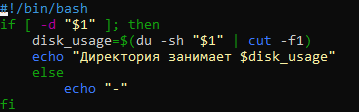


Рисунок 9 – Скрипт для подсчета объема диска, занимаемый директорией



Рисунок 10 – Вывод работы скрипта

Напишем сценарий, который выведет список всех исполняемых файлов в директории, для которых у текущего пользователя есть права на исполнение (рис. 11-12).

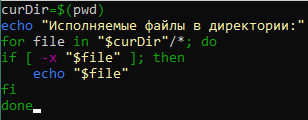


Рисунок 11 – Скрипт для вывода исполняемых файлов

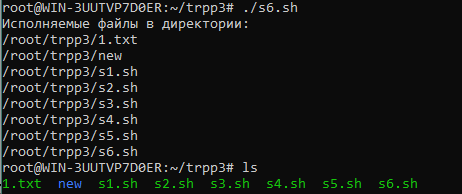


Рисунок 12 – Вывод работы скрипта

По ссылке https://www.dropbox.com/s/ija7ax3sj6ysb0p/blocknote-master.tar.gz скачаем проект. Распакуем архив при помощи команды tar -xvf имя\_архива имя\_директории\_для\_распаковки. Составим список зависимостей проекта в виде requirements.txt файла (рис. 13).

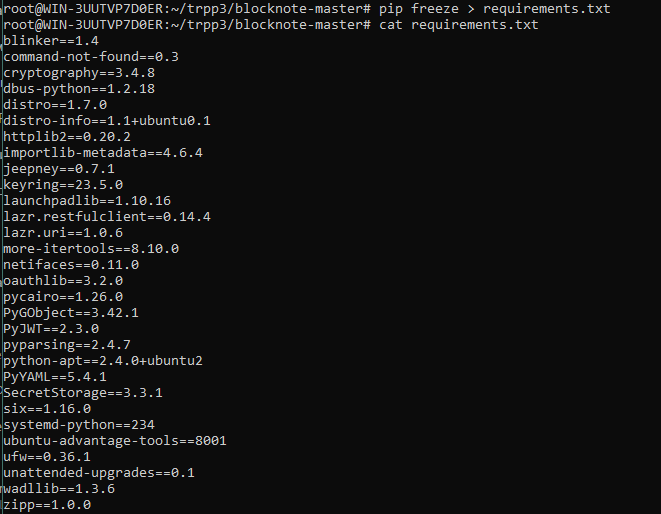


Рисунок 13 – Скрипт для заполнения зависимостей в файл

Напишем скрипт скачивания указанного в прошлом шаге проекта с последующим созданием виртуального окружения и настройкой его под проект, то есть установкой всех необходимых библиотек (рис. 14).

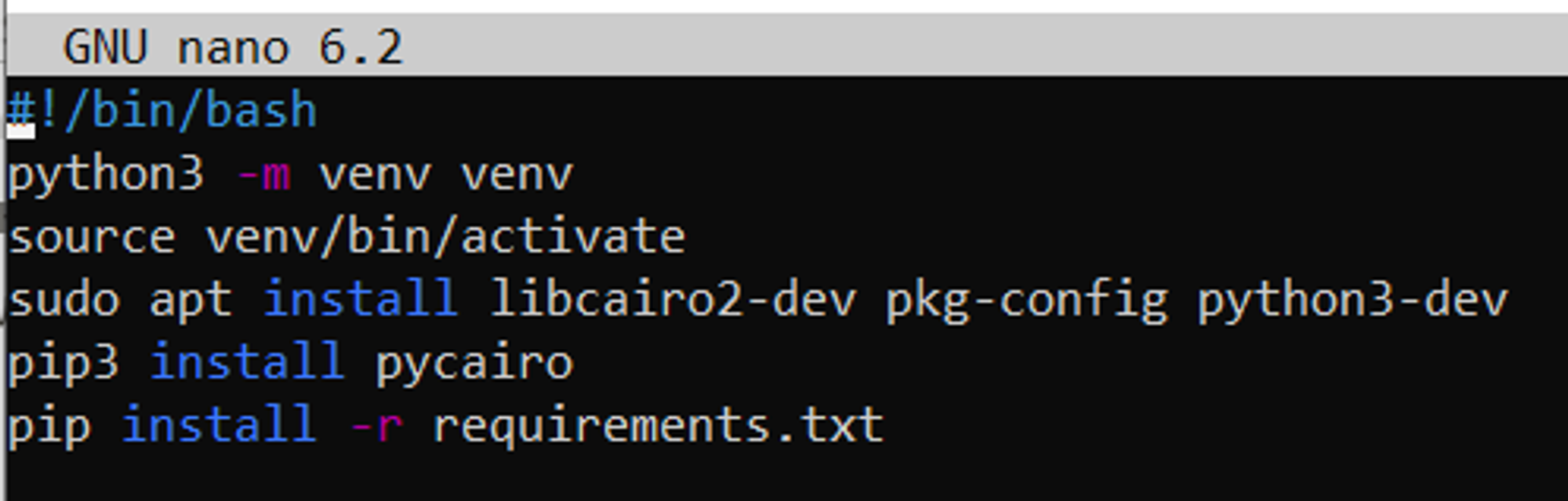


Рисунок 14 – Скрипт для создания виртуального окружения

Далее мы загрузим свой проект и проделаем ту же процедуру установки необходимых зависимостей и запуска виртуального окружения. Зайдем в корень проекта, войдем в виртуальное окружение, установим необходимые зависимости, и запустим проект (рис. 15).

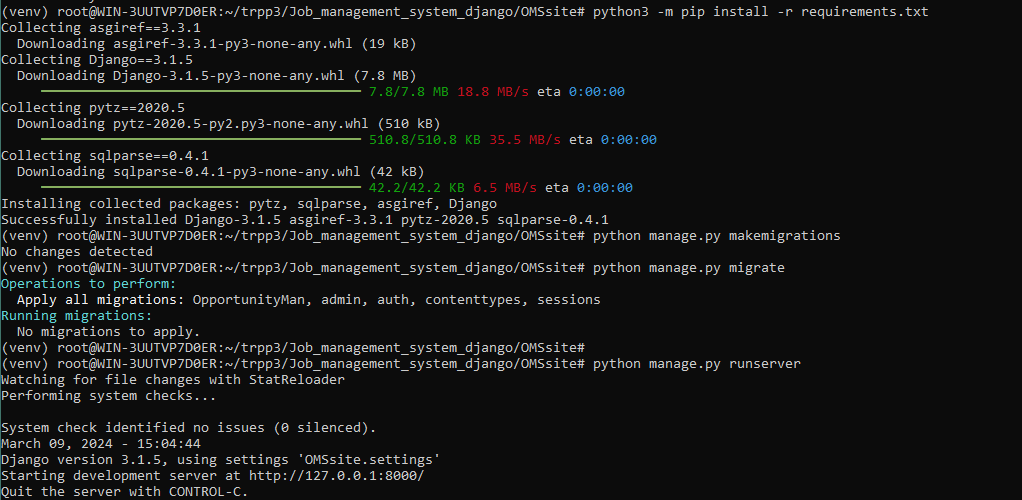


Рисунок 15 – Были установлены необходимые библиотеки и запущено приложение

Теперь убедимся что все работает перейдя по ссылке <http://127.0.0.1:8000/> (рис. 16)

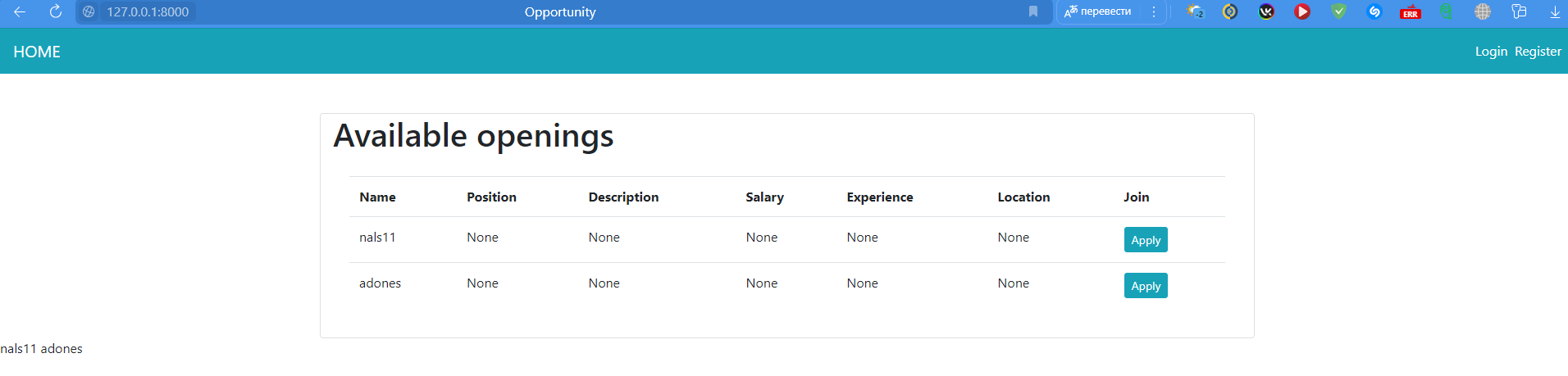


Рисунок 16 – Запущенное приложение

Вывод

В результате выполнения работы были получены навыки по работе с Bash Script’ами.