



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РГУ МИРЭА

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных
технологий

Отчет по практической работе №2

по дисциплине «Технологии разработки программных приложений»

Выполнил:

Студент группы ИКБО-21-23

Зеленков Н. А.

Принял:

преподаватель

Чернов Е.А.

Москва 2024 г.

Оглавление

1 ЧАСТЬ. БАЗОВЫЕ BASH СКРИПТЫ	3
ХОД РАБОТЫ	4
2 ЧАСТЬ. РАЗВЕРТКА И ЗАПУСК ПРОЕКТА ПРИ ПОМОЩИ BASH SCRIPT	8
ХОД РАБОТЫ	10
ВЫВОД	13

1 ЧАСТЬ. БАЗОВЫЕ BASH СКРИПТЫ

Задание:

1. Написать сценарий, который выводит дату, время, список зарегистрировавшихся пользователей, и uptime системы и сохраняет эту информацию в файл.
2. Написать сценарий, который выводит содержимое любого каталога или сообщение о том, что его не существует.
3. Написать сценарий, который с помощью цикла прочитает файл и выведет его содержимое.
4. Написать сценарий, который с помощью цикла выведет список файлов и директорий из текущего каталога, укажет, что есть файл, а что директория.
5. Написать сценарий, который подсчитает объем диска, занимаемого директорией. В качестве директории можно выбрать любую директорию в системе.
6. Написать сценарий, который выведет список всех исполняемых файлов в директории, для которых у текущего пользователя есть права на исполнение.

ХОД РАБОТЫ

Первое задание:

Для первой задачи необходимо написать сценарий, который выводит дату, время, список зарегистрировавшихся пользователей, и uptime системы и сохраняет эту информацию в файл. Код показан на рисунке 1.

```
GNU nano 7.2
#!/bin/bash
echo $(date)
echo $(last)
echo $(uptime)
output_file="sysinfo.txt"
echo $(date) > $output_file
echo $(last) >> $output_file
echo $(uptime) >> $output_file
```

Рисунок 1 – Код для первой задачи

Результат работы программы 1 показан на рисунке 2.

```
akitapts@Obitel:~$ ./task1_1
Wed Mar 19 22:52:14 MSK 2025
reboot system boot 5.15.167.4-micro Wed Mar 19 21:17 still running wtmp begins Wed Mar 19 21:17:14 2025
22:52:14 up 1:29, 1 user, load average: 0.00, 0.00, 0.00
```

Рисунок 2 – Результат первой программы

Содержание файла с результатами на рисунке 3.

```
GNU nano 7.2                                         sysinfo.txt
Wed Mar 19 22:52:14 MSK 2025
reboot system boot 5.15.167.4-micro Wed Mar 19 21:17 still running wtmp begins Wed Mar 19 21:17:14 2025
22:52:14 up 1:29, 1 user, load average: 0.00, 0.00, 0.00
```

Рисунок 3 – Результат первой программы

Второе задание:

Для второй задачи необходимо написать сценарий, который выводит содержимое любого каталога или сообщение о том, что его не существует. Код показан на рисунке 4.

```
#!/bin/bash
read direct
if [ -d $direct ]
then
ls $direct
else
echo "This directory does not exist"
fi
```

Рисунок 4 – Код для второй задачи

Результат работы программы 2 показан на рисунке 5.

```
akitapts@Obitel:~$ ./task1_2
akitapts
This directory does not exist
akitapts@Obitel:~$ ls
myscript  sysinfo.txt  task1_1  task1_2
akitapts@Obitel:~$ ./task1_2
..
akitapts
```

Рисунок 5 – Результат второй программы

Третье задание:

Для третьей задачи необходимо написать сценарий, который с помощью цикла прочитает файл и выведет его содержимое. Код показан на рисунке 6.

```
GNU nano 7.2
#!/bin/bash
read file
for var in $(cat $file)
do
echo -n "$var "
done
echo
```

Рисунок 6 – Код для третьей задачи

Результат работы программы 3 показан на рисунке 7.

```
akitapts@Obitel:~$ ./task1_3
sysinfo.txt
Wed Mar 19 22:52:14 MSK 2025 reboot system boot 5.15.167.4-micro Wed Mar 19 21:17 still running wtmp begins Wed Mar 19 2
1:17:14 2025 22:52:14 up 1:29, 1 user, load average: 0.00, 0.00, 0.00
```

Рисунок 7 – Результат третьей программы

Четвертое задание:

Для четвертой задачи необходимо написать сценарий, который с помощью цикла выведет список файлов и директорий из текущего каталога, укажет, что есть файл, а что директория. Код показан на рисунке 8.

```
GNU nano 7.2                                         task1_4 *
#!/bin/bash
read direct
direct=${direct}/*
for file in $direct
do
if [ -d $file ]
then
echo "$file is a directory"
elif [ -f $file ]
then
echo "$file is a file"
fi
done
```

Рисунок 8 – Код для четвертой задачи

Результат работы программы 4 показан на рисунке 9.

```
akitapts@Obitel:~$ ./task1_4
.
./myscript is a file
./sysinfo.txt is a file
./task1_1 is a file
./task1_2 is a file
./task1_3 is a file
./task1_4 is a file
akitapts@Obitel:~$ ./task1_4
..
.../akitapts is a directory
```

Рисунок 9 – Результат четвертой программы

Пятое задание:

Для пятой задачи необходимо сценарий, который подсчитает объем диска, занимаемого директорией. В качестве директории можно выбрать любую директорию в системе. Код показан на рисунке 10.

```
GNU nano 7.2
#!/bin/bash
read direct
du -sh $direct
```

Рисунок 10 – Код для пятой задачи

Результат работы программы 5 показан на рисунке 11.

```
akitapts@Obitel:~$ ./task1_5
.
68K .
```

Рисунок 11 – Результат пятой программы

Шестое задание:

Для шестой задачи необходимо сценарий, который выведет список всех исполняемых файлов в директории, для которых у текущего пользователя есть права на исполнение. Код показан на рисунке 12.

```
GNU nano 7.2
#!/bin/bash
read direct
direct=$direct/*
for file in $direct
do
if [[ -f "$file" && -x "$file" ]]
then
echo "$file"
fi
done
```

Рисунок 12 – Код для пятой задачи

Результат работы программы 6 показан на рисунке 13.

```
akitapts@0bitel:~$ ./task1_6
.
./myscript
./task1_1
./task1_2
./task1_3
./task1_4
./task1_5
./task1_6
```

Рисунок 13 – Результат пятой программы

2 ЧАСТЬ. РАЗВЕРТКА И ЗАПУСК ПРОЕКТА ПРИ ПОМОЩИ BASH SCRIPT

Задание:

1. Определение зависимостей проекта Любой проект зависит от ряда библиотек, которые предоставляют тот или иной функционал. Для развертывания приложения необходимо, чтобы данные библиотеки были установлены в том окружении, где предполагается это самое развертывание.

На основании этого для начала необходимо определить, какие зависимости имеет проект. По ссылке <https://www.dropbox.com/s/ija7ax3sj6ysb0p/blocknote-master.tar.gz> расположен проект для скачивания. Будет скачан архив с непонятным названием, распаковать его можно при помощи команды tar -xvf имя_архива имя_директории_для_распаковки. Проект написан на языке программирования Python. Необходимо составить список зависимостей проекта в виде requirements.txt файла. Данный файл содержит в себе список библиотек, которые необходимо установить в окружение для запуска приложения. Подробнее про составление данного файла можно почитать по ссылке https://semakin.dev/2020/04/requirements_txt/.

Зависимости в Python можно определить по import'ам в файлах, однако некоторые библиотеки включены в стандартную библиотеку языка, поэтому также необходимо будет определить, является ли библиотека внешней или же встроенной в язык.

2. Создание виртуального окружения Python позволяет создавать так называемое виртуальное окружение. Данное окружение представляет из себя отдельную копию Python с собственным набором библиотек. Оно позволяет работать с проектами, не загрязняя основной интерпретатор ненужными глобально, то есть для всей системы, библиотеками. Подробнее про создание такого рода окружений можно прочитать <https://ru.hexlet.io/courses/python-virtual-environments>

setup environment.

Необходимо на основании составленного в прошлом шаге списка команд написать скрипт скачивания указанного в прошлом шаге проекта с последующим созданием виртуального окружения и настройкой его под проект, то есть установкой всех необходимых библиотек.

3. Написание скрипта запуска приложения на новой системе Bash скрипты позволяют создать с нуля все необходимое окружение в системе, начиная с установки самого python-а и всего необходимого ПО для запуска приложения и заканчивая запуском самого приложения.

Для начала необходимо установить python. Сделать это можно при помощи команды `sudo apt install python3`.

Далее необходимо загрузить к себе на машину собственный проект.

После этого необходимо воссоздать полученное на прошлом этапе виртуальное окружение со всеми зависимостями. Затем необходимо запустить проект из виртуального окружения при помощи следующих команд:

`python manage.py makemigrations`

`python manage.py migrate`

`python manage.py runserver`

ХОД РАБОТЫ

Для начала необходимо скачать архив, он показан на рисунке 14.

```
akitapts@Obitel:~$ wget https://www.dropbox.com/s/ija7ax3sj6ysb0p/blocknote-master.tar.gz
--2025-03-26 14:44:56--  https://www.dropbox.com/s/ija7ax3sj6ysb0p/blocknote-master.tar.gz
Resolving www.dropbox.com (www.dropbox.com)... 162.125.70.18, 2620:100:6022:18::a27d:4212
Connecting to www.dropbox.com (www.dropbox.com)|162.125.70.18|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://www.dropbox.com/scl/fi/voifzl1zetz99d1u51pcj/blocknote-master.tar.gz?rlkey=following]
--2025-03-26 14:44:57--  https://www.dropbox.com/scl/fi/voifzl1zetz99d1u51pcj/blocknote-master.tar.gz?rlkey=following
nnr3cbdojbs9u
Reusing existing connection to www.dropbox.com:443.
```

Рисунок 14 – Команда для скачивания архива

Также нужно его разархивировать, результат представлен на рисунке 15.

```
akitapts@Obitel:~$ pwd
/home/akitapts
akitapts@Obitel:~$ tar -xvf blocknote-master.tar.gz -C /home/akitapts
blocknote-master/
blocknote-master/._DS_Store
blocknote-master/.DS_Store
blocknote-master/appengine/
blocknote-master/._gitignore
blocknote-master/.gitignore
blocknote-master/static/
blocknote-master/._gitattributes
blocknote-master/.gitattributes
blocknote-master/templates/
blocknote-master/._manage.py
blocknote-master/manage.py
blocknote-master/apps/
blocknote-master/apps/._DS_Store
blocknote-master/apps/.DS_Store
blocknote-master/apps/todoapp/
blocknote-master/apps/main/
blocknote-master/apps/main/migrations/
blocknote-master/apps/main/._services.py
blocknote-master/apps/main/services.py
blocknote-master/apps/main/._models.py
blocknote-master/apps/main/models.py
blocknote-master/apps/main/.___init__.py
blocknote-master/apps/main/_init__.py
blocknote-master/apps/main/._apps.py
blocknote-master/apps/main/apps.py
```

Рисунок 15 – Разархивация архива

Далее создаём файл requirements.txt на основе импортов в python файлах, содержимое файла показано на рисунке 16 и 17.

```
GNU nano 7.2
asttokens==3.0.0
attrs==25.3.0
backcall==0.2.0
beautifulsoup4==4.13.3
bleach==6.2.0
certifi==2025.1.31
charset-normalizer==3.4.1
decorator==5.2.1
defusedxml==0.7.1
docopt==0.6.2
executing==2.2.0
fastjsonschema==2.21.1
idna==3.10
ipython==8.12.3
jedi==0.19.2
Jinja2==3.1.6
jsonschema==4.23.0
jsonschema-specifications==2024.10.1
jupyter_client==8.6.3
jupyter_core==5.7.2
jupyterlab_pygments==0.3.0
MarkupSafe==3.0.2
matplotlib-inline==0.1.7
mistune==3.1.3
nbclient==0.10.2
nbconvert==7.16.6
nbformat==5.10.4
packaging==24.2
pandocfilters==1.5.1
parso==0.8.4
pexpect==4.9.0
pickleshare==0.7.5
pipreqs==0.5.0
platformdirs==4.3.7
prompt_toolkit==3.0.50
ptyprocess==0.7.0
pure_eval==0.2.3
Pygments==2.19.1
python-dateutil==2.9.0.post0
```

Рисунок 16 – Содержание файла requirements.txt

```
typing==3.7.4.3
pathlib==1.0.1
django==5.1.6
asgiref==3.8.1
sqlparse==0.5.3
django-simple-history==3.8.0
django-tinymce==4.1.0
django-ckeditor==6.7.2
django-js-asset==3.0.1|
```

Рисунок 17 – Содержание файла requirements.txt

После всех действий нужно создать виртуальное окружение и запустить проект, чтобы это автоматизировать создадим файл setup, его содержание показано на рисунке 18.

```
GNU nano 7.2                                         setup *
#!/bin/bash
sudo apt update
sudo apt install -y python3
pip3 install virtualenv
wget https://www.dropbox.com/s/ija7ax3sJ6ysb0p/blocknote-master.tar.gz
tar -xvf blocknote-master.tar.gz -C
cd blocknote-master
virtualenv venv
python3 -m venv venv
source venv/bin/activate
pip install -r requirements.txt
python3 manage.py makemigrations
python3 manage.py migrate
python3 manage.py runserver|
```

Рисунок 18 – Файл setup

Файл запущен результат выполнения на рисунке 19.

```
System check identified 4 issues (0 silenced).
February 24, 2025 - 15:00:46
Django version 5.1.6, using settings 'appengine.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Рисунок 19 – Результат работы файла setup

ВЫВОД

В ходе выполнения данной практической работы, были изучены основы работы с bash на примере нескольких задач, также был запущен проект с помощью bash.