



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 №2

по дисциплине «Технологии разработки программных приложений»

Тема практической работы: «Основы работы с Bash Scriptами»

Выполнил:

Студент группы ИКБО-14-22

Оганнисян Г.А.

Проверил:

Исобекова О.А.

Москва 2024

Оглавление

| | |
|---|----|
| Часть 1. Базовые Bash скрипты. | 3 |
| Задание 1..... | 3 |
| Задание 2..... | 4 |
| Задание 3..... | 5 |
| Задание 4..... | 6 |
| Задание 5..... | 7 |
| Задание 6..... | 8 |
| Часть 2. Развертка и запуск проекта при помощи Bash Script..... | 9 |
| 1. Определение зависимостей проекта | 9 |
| 2. Создание виртуального окружения..... | 10 |
| 3. Написание скрипта запуска приложения на новой системе | 12 |

Часть 1. Базовые Bash скрипты.

Задание 1.

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

```
#!/bin/bash

date
sed 's/:.*/' /etc/passwd
uptime
```

Рис. 1. Содержимое файла sc1

```
Practic_MIREA/semestr-4/TRPP/PR2/Part1$ ./sc1 > f.txt
Practic_MIREA/semestr-4/TRPP/PR2/Part1$
```

Рис. 1. Запуск скрипта sc1

f.txt – Блокнот

Файл Правка Формат Вид Справка

Thu Mar 14 21:21:18 MSK 2024

root

daemon

bin

sys

sync

games

man

lp

mail

news

uucp

proxy

www-data

backup

list

irc

gnats

nobody

systemd-network

systemd-resolve

systemd-timesync

messagebus

syslog

_apt

tss

uidd

tcpdump

sshd

landscape

pollinate

problema

21:21:18 up 21 min, 0 users, load average: 0.29, 0.22, 0.10

Рис. 2. Содержимое файла file.txt

Задание 2.

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

```
1  #!/bin/bash
2
3  if [ -d "$1" ]
4  then
5      echo "Содержимое каталога $1:"
6      ls "$1"
7  else
8      echo "Каталог $1 не существует"
9  fi
10
```

Рис. 4. Содержимое файла sc2

```
problema@DESKTOP-6H2LE3M:/mnt/c/Users/SystemX/Documents/GitHub/Practic_MIREA/semestr-4/TRPP/PR2/Part2$ ./sc2 /n
X/Documents
Содержимое каталога /mnt/c/Users/SystemX/Documents:
GitGarik  TgGarik  WindowsPowerShell  'Мои видеозаписи'  'Настраиваемые шаблоны Office'
GitHub    TgGrisha  desktop.ini         'Моя музыка'      'мои рисунки'
```

Рис. 5. Вывод отработки скрипта

Задание 3.

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

```
1  #!/bin/bash
2
3  IFS = $'\n'
4  for i in $(cat $1)
5  do
6  echo "$i"
7  done
```

Рис. 6. Содержимое файла sc3

```
Hello, world!!!
```

Рис. 7. Содержание файла rm.txt

```
problema@DESKTOP-6H2LE3M:/mnt/c/Users/!
R2$ ./sc3 rm.txt
Hello, world!!!
```

Рис. 8. Вывод отработки скрипта

Задание 4.

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

```
1  #!/bin/bash
2
3  for i in $(ls)
4  do
5      if [ -f "$i" ]
6      then
7          echo " $i - file "
8      elif [ -d "$i" ]
9      then
10         echo " $i - directory "
11     fi
12 done
```

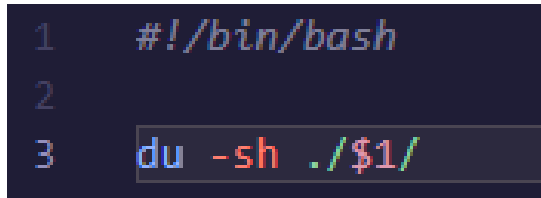
Рис. 9. Содержимое файла sc4

```
problema@DESKTOP-6H2LE3M:/mnt/c/Users/System
R2$ ./sc4
f.txt - file
rm.txt - file
sc1 - file
sc2 - file
sc3 - file
sc4 - file
```

Рис. 10. Вывод отработки скрипта

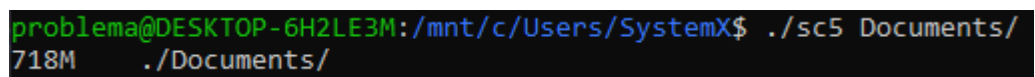
Задание 5.

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



```
1  #!/bin/bash
2
3  du -sh ./ $1/
```

Рис. 11. Содержимое файла sc5



```
problema@DESKTOP-6H2LE3M:/mnt/c/Users/SystemX$ ./sc5 Documents/
718M    ./Documents/
```

Рис. 12. Вывод отработки скрипта

Задание 6.

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

```
1  #!/bin/bash
2
3  files=$(find $1 -type f -executable)
4
5  for file in $files; do
6      if [ -x $file ]; then
7          echo $file
8      fi
9  done
```

Рис. 13. Содержимое файла sc6

```
problema@DESKTOP-6H2LE3M:/mnt/c/Users/SystemX/Do
R2$ ./sc6
./f.txt
./rm.txt
./sc1
./sc2
./sc3
./sc4
./sc5
./sc6
```

Рис. 14. Вывод отработки скрипта

Часть 2. Развертка и запуск проекта при помощи Bash Script.

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. 11 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.

```
C:\Users\SystemX\Documents\GitHub\Practic_MIREA\semestr-4\TRPP\PR2\part2>tar -xvf blocknote.tar.gz
x blocknote-master/
x blocknote-master/.DS_Store
x blocknote-master/.DS_Store
```

Рис. 15. Распаковка архива `blocknote-master.tar.gz`

```
semestr-4 > TRPP > PR2 > part2 > blocknote-master > requirements.txt
1  asgiref==3.7.2
2  Django==5.0.3
3  django-ckeditor==6.7.1
4  django-js-asset==2.2.0
5  django-simple-history==3.5.0
6  django-tinymce==1.0.1
7  python-dotenv==1.0.1
8  sqlparse==0.4.4
9  tzdata==2024.1
10
```

Рис. 16. Содержимое файла `requirements.txt`

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 по ссылке <https://ru.hexlet.io/courses/python-setupenvironment>.

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.

```
1  #!/bin/bash
2
3  cd blocknote-master
4  echo -n > requirements.txt
5  for i in $(find -name '*.py')
6  do
7      grep -oP '(?<=^from )\w+|(?<=^import )(\w+)' $i >> requirements.txt
8  done
9
10 sort -u -o requirements.txt requirements.txt
11
12 apt install python3
13 apt install python3-pip
14 apt install python3-virtualenv
15
16 virtualenv venv
17 source venv/bin/activate
18
19 for i in $(cat requirements.txt)
20 do
21     pip install $i
22 done
23
24 deactivate
25
26 cd ..
27
```

Рис. 17. Скрипт p2sc2

```
1  anyio==4.2.0
2  argon2-cffi==23.1.0
3  argon2-cffi-bindings==21.2.0
4  arrow==1.3.0
5  astroid==3.1.0
6  asttokens==2.4.1
7  async-lru==2.0.4
8  attrs==23.2.0
9  Babel==2.14.0
10 beautifulsoup4==4.12.3
11 bleach==6.1.0
12 certifi==2023.11.17
13 cffi==1.16.0
14 charset-normalizer==3.3.2
15 colorama==0.4.6
16 comm==0.2.1
17 debugpy==1.8.0
18 decorator==5.1.1
19 defusedxml==0.7.1
20 dill==0.3.8
21 distlib==0.3.8
22 executing==2.0.1
23 fastjsonschema==2.19.1
24 filelock==3.13.1
25 fqdn==1.5.1
26 idna==3.6
27 ipykernel==6.29.0
28 ipython==8.20.0
29 isoduration==20.11.0
```

Рис. 18. Список всех библиотек

3. Написание скрипта запуска приложения на новой системе

Bash-скрипты позволяют создать с нуля всё 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 3. Сделать это можно при помощи команды `sudo apt install python3`.

Далее необходимо загрузить к себе на машину собственный проект.

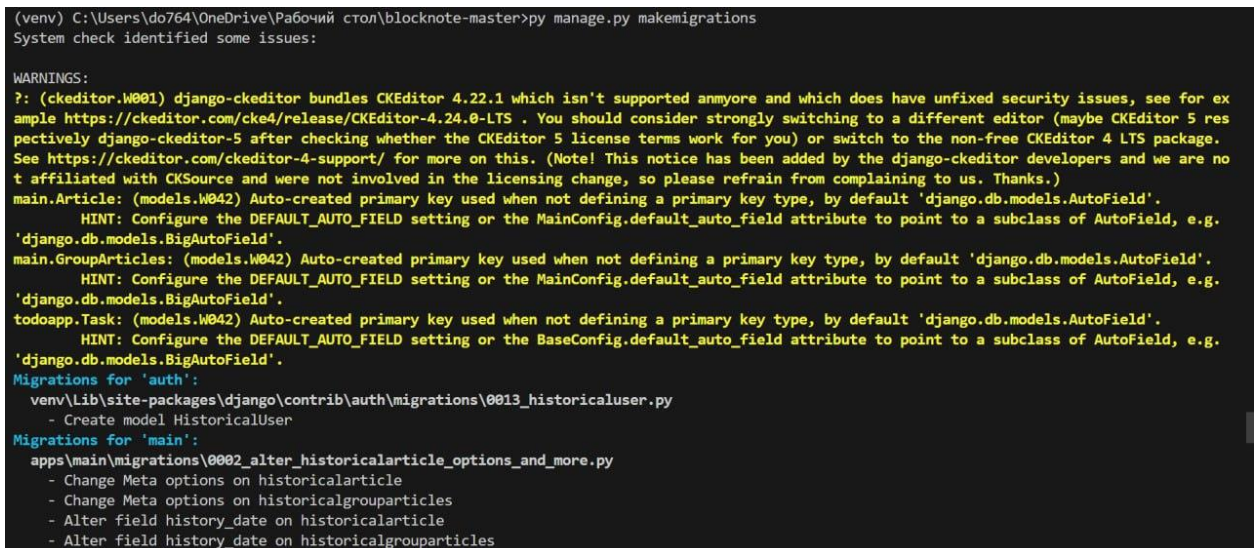
После этого необходимо воссоздать полученное на прошлом этапе виртуальное окружение со всеми зависимостями.

Затем необходимо запустить проект из виртуального окружения при помощи следующих команд:

```
python manage.py makemigrations
```

```
python manage.py migrate
```

```
python manage.py runserver
```



```
(venv) C:\Users\do764\OneDrive\Рабочий стол\blocknote-master>py manage.py makemigrations
System check identified some issues:

WARNINGS:
?: (ckeditor.W001) django-ckeditor bundles CKEditor 4.22.1 which isn't supported anymore and which does have unfixed security issues, see for example https://ckeditor.com/cke4/release/CKEditor-4.24.0-LTS . You should consider strongly switching to a different editor (maybe CKEditor 5 respectively django-ckeditor-5 after checking whether the CKEditor 5 license terms work for you) or switch to the non-free CKEditor 4 LTS package. See https://ckeditor.com/ckeditor-4-support/ for more on this. (Note! This notice has been added by the django-ckeditor developers and we are not affiliated with CKSource and were not involved in the licensing change, so please refrain from complaining to us. Thanks.)
main.Article: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
      HINT: Configure the DEFAULT_AUTO_FIELD setting or the MainConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
main.GroupArticles: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
      HINT: Configure the DEFAULT_AUTO_FIELD setting or the MainConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
todoapp.Task: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
      HINT: Configure the DEFAULT_AUTO_FIELD setting or the BaseConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
Migrations for 'auth':
  venv\Lib\site-packages\django\contrib\auth\migrations\0013_historicaluser.py
    - Create model HistoricalUser
Migrations for 'main':
  apps\main\migrations\0002_alter_historicalarticle_options_and_more.py
    - Change Meta options on historicalarticle
    - Change Meta options on historicalgrouparticles
    - Alter field history_date on historicalarticle
    - Alter field history_date on historicalgrouparticles
```

Рис. 19. Выполнение команды `python manage.py makemigrations`


```
(venv) C:\Users\do764\OneDrive\Рабочий стол\blocknote-master>py manage.py migrate
System check identified some issues:

WARNINGS:
?: (ckeditor.W001) django-ckeditor bundles CKEditor 4.22.1 which isn't supported anymore and which does have unfixed security issues, see for example https://ckeditor.com/cke4/release/CKEditor-4.24.0-LTS . You should consider strongly switching to a different editor (maybe CKEditor 5 respectively django-ckeditor-5 after checking whether the CKEditor 5 license terms work for you) or switch to the non-free CKEditor 4 LTS package. See https://ckeditor.com/ckeditor-4-support/ for more on this. (Note! This notice has been added by the django-ckeditor developers and we are not affiliated with CKSource and were not involved in the licensing change, so please refrain from complaining to us. Thanks.)
main.Article: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
      HINT: Configure the DEFAULT_AUTO_FIELD setting or the MainConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
main.GroupArticles: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
      HINT: Configure the DEFAULT_AUTO_FIELD setting or the MainConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
todoapp.Task: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
      HINT: Configure the DEFAULT_AUTO_FIELD setting or the BaseConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, main, sessions, todoapp
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
```

Activate Windows
Go to Settings to activate Windows.

Рис. 20. Выполнение команды python manage.py migrate

```
(venv) C:\Users\do764\OneDrive\Рабочий стол\blocknote-master>py manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified some issues:

WARNINGS:
?: (ckeditor.W001) django-ckeditor bundles CKEditor 4.22.1 which isn't supported anymore and which does have unfixed security issues, see for example https://ckeditor.com/cke4/release/CKEditor-4.24.0-LTS . You should consider strongly switching to a different editor (maybe CKEditor 5 respectively django-ckeditor-5 after checking whether the CKEditor 5 license terms work for you) or switch to the non-free CKEditor 4 LTS package. See https://ckeditor.com/ckeditor-4-support/ for more on this. (Note! This notice has been added by the django-ckeditor developers and we are not affiliated with CKSource and were not involved in the licensing change, so please refrain from complaining to us. Thanks.)
main.Article: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
      HINT: Configure the DEFAULT_AUTO_FIELD setting or the MainConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
main.GroupArticles: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
      HINT: Configure the DEFAULT_AUTO_FIELD setting or the MainConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.
todoapp.Task: (models.W042) Auto-created primary key used when not defining a primary key type, by default 'django.db.models.AutoField'.
      HINT: Configure the DEFAULT_AUTO_FIELD setting or the BaseConfig.default_auto_field attribute to point to a subclass of AutoField, e.g. 'django.db.models.BigAutoField'.

System check identified 4 issues (0 silenced).
March 09, 2024 - 14:25:55
Django version 5.0.3, using settings 'appengine.settings'
Starting development server at http://127.0.0.1:8000/
```

Рис. 21. Выполнение команды python manage.py runserver

Вывод.

По итогам выполнения данной практической работы были получены навыки по работе Bash Scriptами и виртуальным окружением.