

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп'ютерних систем

КУРСОВА РОБОТА

з дисципліни "Компоненти програмної інженерії"

Виконала: Нагорна Софія Олегівна

Група: КП-03

Допущено до захисту

1 семестр 2022/2023

Опис розробленого програмного засобу

In-memory file system була реалізована у вигляді HTTP додатку з консольним інтерфейсом користувача (CLI), який був реалізований за допомогою пакету мови програмування Python Click.

Також був застосований мікро-фреймворк Flask для реалізації клієнт-серверної взаємодії за протоколом HTTP.

Для тестування функціоналу додатку був застосований фреймворк PyTest, а для тестування роботи консольного інтерфейсу також був застосований модуль пакету Click click.testing, а саме функція - CliRunner().

Додаток було запущено у контейнері фреймворку Docker.

Результати роботи програми

1. Перш за все, створимо докер образ за допомогою команди:

```
sudo docker build -t ssepiphany/file-system .
```

Та отримаємо наступний результат:

```
Sending build context to Docker daemon 21.83MB
Step 1/6 : FROM python:3.8
--> 51a078947558
Step 2/6 : WORKDIR /usr/src/app
--> Using cache
--> ef3fba09a72e
Step 3/6 : COPY . .
--> 9f65fdb064b2
Step 4/6 : RUN pip install --no-cache-dir -r requirements.txt
--> Running in df9a2d0ba8cf
Collecting Flask==2.0.2
```

```
Successfully built f1212dc8075c
Successfully tagged ssepiphany/file-system:latest
```

Перевіримо створення образу за допомогою команди:

```
sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ssepihany/file-system	latest	f1212dc8075c	3 hours ago	943MB

2. Запустимо додаток в докер контейнері за допомогою команди:

```
sudo docker run -p 8888:5000 ssepihany/file-system
```

Та отримаємо наступний результат:

```
* Serving Flask app 'application' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://172.17.0.2:5000
Press CTRL+C to quit
```

3. Переглянемо список усіх доступних команд за допомогою команди:

```
python3 cli.py --help
```

Та отримаємо наступний результат:

```
Usage: cli.py [OPTIONS] COMMAND [ARGS]...

Options:
  --help  Show this message and exit.

Commands:
  appendlog
  consumebuf
  createbin
  createbuf
  createdir
  createlog
  deletebin
  deletebuf
  deletedir
  deletelog
  listdir
  movebin
  movebuf
  movedir
  movelog
  pushbuf
  readbin
  readlog
```

4. Запустимо деякі з команд:

a) Створення директорії:

```
python3 cli.py createdir /my/hello
```

```
b'Directory [/my/hello] was created.'
```

b) Створення директорії з некоректною назвою:

```
python3 cli.py createdir /he.log
```

```
b'Invalid directory name.'
```

c) Створення файлу з некоректною назвою:

```
python3 cli.py createlog /my/file.lig "some log text"
```

```
b'Wrong file name format [/my/file.lig] (filename.log)'
```

d) Створення log text файлу:

```
python3 cli.py createlog /my/file.log "some log text"
```

```
b'File [/my/file.log] was created.'
```

e) Створення buffer файлу:

```
python3 cli.py createbuf /my/file.buf
```

```
b'File [/my/file.buf] was created.'
```

f) Перегляд вмісту директорії:

```
python3 cli.py listdir /my
```

```
b'hello, file.log, file.buf'
```

g) Додавання нового рядку тексту до log text файлу:

```
python3 cli.py appendlog /my/file.log "second line"
```

```
b'Line [second line] was appended.'
```

h) Переміщення log text файлу:

```
python3 cli.py movelog /my/file.log /my/hello/file.log
```

```
b'File was moved from /my/file.log to /my/hello/file.log'
```

i) Перегляд вмісту log text файлу:

```
python3 cli.py readlog /my/hello/file.log
```

```
b'some log text\r\nsecond line'
```

j) Додавання нового об'єкту до неіснуючого buffer файлу:

```
python3 cli.py pushbuf /my/hello/file.buf
```

```
b"File doesn't exist."
```

к) Додавання нового об'єкту до buffer файлу:

```
python3 cli.py pushbuf /my/file.buf
```

```
b'Element was pushed to the file.'
```

l) Видалення log text файлу:

```
python3 cli.py deletelog /my/hello/file.log
```

```
b'File [/my/hello/file.log] was deleted.'
```

5. Переглянемо отримані сервером запити:

```
172.17.0.1 - - [09/Jan/2023 00:29:38] "GET / HTTP/1.1" 404 -
172.17.0.1 - - [09/Jan/2023 00:29:50] "GET / HTTP/1.1" 404 -
172.17.0.1 - - [09/Jan/2023 00:38:08] "POST /directory/create HTTP/1.1" 201 -
172.17.0.1 - - [09/Jan/2023 00:38:29] "POST /directory/create HTTP/1.1" 400 -
172.17.0.1 - - [09/Jan/2023 00:39:18] "POST /logtextfile/create HTTP/1.1" 400 -
172.17.0.1 - - [09/Jan/2023 00:39:35] "POST /logtextfile/create HTTP/1.1" 201 -
172.17.0.1 - - [09/Jan/2023 00:39:58] "POST /bufferfile/create HTTP/1.1" 201 -
172.17.0.1 - - [09/Jan/2023 00:40:09] "GET /directory/list/my HTTP/1.1" 201 -
172.17.0.1 - - [09/Jan/2023 00:40:22] "PUT /logtextfile/append HTTP/1.1" 200 -
172.17.0.1 - - [09/Jan/2023 00:40:41] "PUT /logtextfile/move HTTP/1.1" 200 -
172.17.0.1 - - [09/Jan/2023 00:41:07] "GET /logtextfile/read/my,hello,file.log HTTP/1.1" 200 -
172.17.0.1 - - [09/Jan/2023 00:41:28] "PUT /bufferfile/push HTTP/1.1" 400 -
172.17.0.1 - - [09/Jan/2023 00:41:39] "PUT /bufferfile/push HTTP/1.1" 200 -
172.17.0.1 - - [09/Jan/2023 00:42:00] "DELETE /logtextfile/delete HTTP/1.1" 404 -
172.17.0.1 - - [09/Jan/2023 00:42:33] "DELETE /logtextfile/delete HTTP/1.1" 200 -
```

6. Перевіримо виконання усіх команд додатку за допомогою PyTest:

`pytest test_file_system.py`

```
ssepiphany@softias-lap:~/my/projects/qa-kp03-nahorna/sem_1_lab_3$ pytest test_file_system.py
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.2.0, pluggy-1.0.0
rootdir: /home/ssepiphany/my/projects/qa-kp03-nahorna/sem_1_lab_3
collected 34 items

test_file_system.py .....

===== 34 passed in 0.17s =====
ssepiphany@softias-lap:~/my/projects/qa-kp03-nahorna/sem_1_lab_3$
```

`pytest test_cli.py`

```
ssepiphany@softias-lap:~/my/projects/qa-kp03-nahorna/sem_1_lab_3$ pytest test_cli.py
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.2.0, pluggy-1.0.0
rootdir: /home/ssepiphany/my/projects/qa-kp03-nahorna/sem_1_lab_3
collected 18 items

test_cli.py .....

===== 18 passed in 0.13s =====
ssepiphany@softias-lap:~/my/projects/qa-kp03-nahorna/sem_1_lab_3$
```

Висновки

Під час виконання курсової роботи був створений HTTP додаток in-memory файлової системи, досліджено створення Restful сервісу за допомогою Flask фреймворку.

Також було використано пакети мови програмування Python Click та requests при створенні command line interface та надсилання запитів до серверу відповідно.

Для тестування функціоналу додатку було досліджено фреймворк PyTest, а для тестування роботи консольного інтерфейсу також - модуль пакету Click click.testing, а саме функція - CliRunner().

Також була проведена робота з фреймворком Docker, а саме: створення файлів Dockerfile та requirements.txt для створення образу додатку. Далі відповідно створення образу та запуск додатку в контейнері.