

Міністерство Освіти І НАУКИ України
Національний університет "Львівська політехніка"

Інститут ІКНІ
Кафедра ПЗ

ЗВІТ

До лабораторної роботи № 9

На тему: “Організація взаємодії між процесами”

З дисципліни: “*Операційні системи*”

Лектор:

Старший викладач ПЗ
Грицай О.Д.

Виконав:

ст. гр. ПЗ-22
Солтисюк Д.А.

Прийняв:

Старший викладач ПЗ
Грицай О.Д.

« ____ » _____ 2022 р.

$\Sigma =$ ____

Львів – 2022

Тема роботи: Організація взаємодії між процесами

Мета роботи: Ознайомитися зі способами міжпроцесної взаємодії. Ознайомитися з класичним прикладом взаємодії між процесами на прикладі задачі «виробник – споживач». Навчитися працювати із процесами з використанням способів міжпроцесної взаємодії, синхронізувати їхню роботу. .

Індивідуальне завдання

Завдання.

1. Реалізувати алгоритм моделювання заданої задачі за допомогою окремих процесів згідно індивідуального завдання.
2. Реалізувати синхронізацію роботи процесів.
3. Забезпечити зберігання результатів виконання завдання.
4. Результати виконання роботи відобразити у звіті.

Варіант 4. Створити програму, що моделює наступну ситуацію: Модератори форуму. Користувач реєструється на форумі і його ім'я записується у базу-даних (файл). Після того він може написати **повідомлення**. Викликаються модератори форуму. Кожен з яких слідкує за певним забороненим словом. **Повідомлення** може бути виведене на екран, якщо загальна сума заборонених слів не перевищує певне число. Кількість заборонених слів заноситься в базу даних. На форумі можуть працювати кілька користувачів.

Хід роботи

Код серверної частини

main.py

```
import json
import socket
from threading import Thread
```



```

except Exception as e:
    print("Error: ", address, e.with_traceback(None))

def main():
    engine = init_db_engine()
    with socket.socket() as server:
        server.bind((LEADER_HOST, LEADER_PORT))
        # configure how many client the server can listen simultaneously
        server.listen(100)
        while True:
            sock, address = server.accept() # accept new connection
            Thread(target=on_new_client, args=(engine, sock, address)).start()

if __name__ == "__main__":
    main()

```

Код клієнтської частини

main.py

```

import pytermgui as ptg

from client.views.login import client_login_form, login

def main():
    with ptg.WindowManager() as ui:
        client_login_form(
            ui,
            login,
        )

if __name__ == "__main__":
    main()

```

login.py

```

import pytermgui as ptg

from client.socket_client import socket_client
from client.views.view_posts import client_view_posts

PROMT_TO_KEY = {"Name: ": "name", "Password: ": "password"}

```

```

def client_login_form(ui, callback):
    def submit(window: ptg.Window) -> None:
        output = {}
        for widget in window:
            if isinstance(widget, ptg.InputField):
                key = PROMT_TO_KEY[widget.prompt]
                output[key] = widget.value
            continue
        callback(ui, output)
        ui.remove(window)

    inputs = [ptg.InputField(prompt=prompt) for prompt in PROMT_TO_KEY.keys()]
    window = ptg.Window(
        "[secondary>Login",
        "",
        *inputs,
        "",
        ["Submit", lambda _: submit(window)],
    ).center()
    ui.add(window)

```

```

def login(ui, creds):
    res = socket_client.send_request(
        ui=ui,
        method="login",
        payload=creds,
    )

```

```

    if res is None:
        # retry login
        client_login_form(ui, login)
        return

```

```

    socket_client.set_token(res)

```

```

    client_view_posts(ui)

```

socket_client.py

```

import json
import socket

```

```

from constants import LEADER_HOST, LEADER_PORT, MAX_PAYLOAD_SIZE

```

```

class SocketClient:
    def __init__(self):
        self.client_token = None
        self.client = socket.socket()
        self.client.connect((LEADER_HOST, LEADER_PORT))

    def set_token(self, token):
        self.client_token = token

    def get_token(self):
        return self.client_token

    def send_request(self, method, payload, ui):
        self.client.send(
            json.dumps(
                {
                    "auth": {"token": self.client_token},
                    "method": method,
                    "payload": payload,
                }
            ).encode()
        )
        raw_res = self.client.recv(MAX_PAYLOAD_SIZE)
        res = json.loads(raw_res.decode())
        message = res["message"]

        match res["type"]:
            case "error":
                ui.toast(
                    f"[bold red]{message}",
                    delay=6 * 10**3,
                    slot="Alert",
                    animate=False,
                )
                return None
            case "ok":
                ui.toast(
                    f"[bold green]{message}",
                    delay=6 * 10**3,
                    slot="Alert",
                    animate=False,
                )

                return res["payload"]

socket_client = SocketClient()

```

Приклад використання

Login

Name:

Password:

```
2022-11-30 19:23:08,313 INFO sqlalchemy.engine.Engine BEGIN (implicit)
2022-11-30 19:23:08,313 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("bad_words")
2022-11-30 19:23:08,313 INFO sqlalchemy.engine.Engine [raw sql] ()
2022-11-30 19:23:08,313 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("posts")
2022-11-30 19:23:08,313 INFO sqlalchemy.engine.Engine [raw sql] ()
2022-11-30 19:23:08,313 INFO sqlalchemy.engine.Engine PRAGMA main.table_info("users")
2022-11-30 19:23:08,313 INFO sqlalchemy.engine.Engine [raw sql] ()
2022-11-30 19:23:08,313 INFO sqlalchemy.engine.Engine COMMIT
Listening on: ('127.0.0.1', 55095)

```

Логін форма

Forums Feed

▶ Me (nick)

▶ dima

▶ Me (nick)

▶ Me (nick)

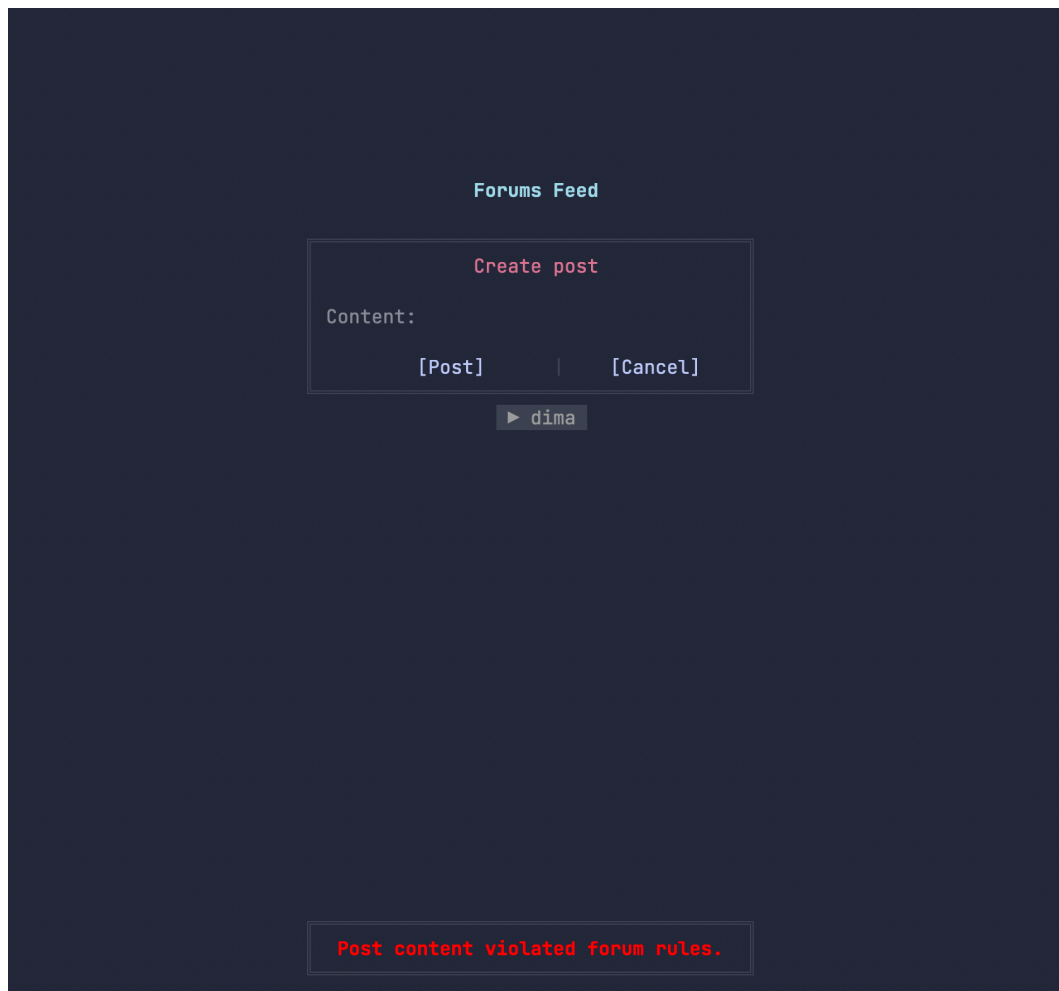
▶ dima

Posts were successfully retrieved

```
ame AS users_name, users.password AS users_password, users.token AS users_token
FROM users
WHERE users.token = ?
LIMIT ? OFFSET ?
2022-11-30 19:23:57,235 INFO sqlalchemy.engine.Engine [generated in 0.00011s] ('859823a1-d55e-4778-88a0-c8de485c2063', 1, 0)
2022-11-30 19:23:57,236 INFO sqlalchemy.engine.Engine SELECT posts.id AS posts_id, posts.content AS posts_content, posts.user_id AS posts_user_id
FROM posts JOIN users ON users.id = posts.user_id
2022-11-30 19:23:57,236 INFO sqlalchemy.engine.Engine [generated in 0.00006s] ()
2022-11-30 19:23:57,236 INFO sqlalchemy.engine.Engine SELECT users.id AS users_id, users.name AS users_name, users.password AS users_password, users.token AS users_token
FROM users
WHERE users.id = ?
2022-11-30 19:23:57,236 INFO sqlalchemy.engine.Engine [generated in 0.00006s] (2,)

```

Перегляд постів на форумі



Створення посту який порушує правила форуму

Висновок

Під час виконання лабораторної роботи я ознайомився зі способами міжпроцесної взаємодії. Та створив програмний комплекс з використанням бази даних