

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
Кафедра програмування

Звіт
до лабораторної роботи №10
з теми
“Програмування сокетів”

Підготував:
студент ПМІ-31
Процьків Назарій

Львів 2023

Мета: здобути знання і практичні навички, необхідних для програмування простих мережових клієнт-серверних додатків з використанням сокетів.

Хід роботи

1. Опрацював теоретичний матеріал
2. Ознайомився з наведеними в інструкції прикладами коду клієнтської та серверної частин мережевого застосунку.
3. Використовуючи бібліотеку `socket`, яка вміє працювати з сокетами, написав клієнт-серверний застосунок мовою програмування Python, де клієнтська програма відправляє логін і пароль та отримує від сервера вітання з моїм цим логіном. Реалізував вибір режиму роботи сервера під час його запуску — послідовна/паралельна обробка запитів.
4. Код програм:

Клієнт:

```
from socket import *

address = "localhost"
port = 5555
ip_end = (address, port)
s = socket(AF_INET, SOCK_STREAM)
s.connect(ip_end)
print("Connected to server!")

username = input("Enter username: ")
s.send(username.encode("utf-8"))

password = input("Enter password: ")
s.send(password.encode("utf-8"))
```

```
print("Waiting for response...")
response = s.recv(1024).decode("utf-8")
print(response)
s.close()
```

Сервер:

```
from socket import *
import threading

users = {
    "Rosaka": {
        "Name": "Nazarii",
        "Surname": "Protskiv",
        "Age": 19,
        "Password": "myPassword"
    },
    "dixen18": {
        "Name": "Roman",
        "Surname": "Dix",
        "Age": 21,
        "Password": "dixenCoolGuy1234"
    }
}

address = "localhost"
port = 5555
ip_end = (address, port)
server = socket(AF_INET, SOCK_STREAM)
server.bind(ip_end)
server.listen()
print("Server is listening...")
```

```
def handle_client(client_socket, client_address):
    print(f"New connection: {client_address}")

    print("Waiting for username...")
    username = client_socket.recv(1024).decode("utf-8")

    print("Waiting for password...")
    password = client_socket.recv(1024).decode("utf-8")

    if username in users:
        if users[username]["Password"] == password:
            print("User is logged in!")
            client_socket.send(f"Hello {users[username]['Name']} +
users[username['Surname']]}.encode("utf-8"))

        else:
            print("User entered wrong password!")
            client_socket.send("False".encode("utf-8"))
    else:
        print("User is not registered!")
        client_socket.send("False".encode("utf-8"))

    client_socket.close()
    print(f"Connection closed: {client_address}")

def sequential_server():
    while True:
        client_socket, client_address = server.accept()
        handle_client(client_socket, client_address)
```

```

def threaded_server():
    while True:
        client_socket, client_address = server.accept()
        client_thread = threading.Thread(target=handle_client,
args=(client_socket, client_address))
        client_thread.start()

mode = input("Choose mode (sequential/threaded): ").lower()

if mode == "sequential":
    print("Server is running in sequential mode...")
    sequential_server()
elif mode == "threaded":
    print("Server is running in threaded mode...")
    threaded_server()
else:
    print("Invalid mode!")

```

5. Перевірів вхід користувача:

Запустив клієнт і сервер і ввів **правильні** дані, про які сервер знає:

<pre> D:\Study\Комп'ютерні мережі\ЛР10\server>python server.py Server is listening... Choose mode (sequential/threaded): sequential Server is running in sequential mode... New connection: ('127.0.0.1', 58985) Waiting for username... Waiting for password... User is logged in! </pre>	<pre> D:\Study\Комп'ютерні мережі\ЛР10\server>python client.py Connected to server! Enter username: Rosaka Enter password: myPassword Waiting for response... Hello Nazarii Protskiv D:\Study\Комп'ютерні мережі\ЛР10\server> </pre>
---	---

Ввів **неправильні** вхідні дані (отримав помилку):

<pre>D:\Study\Комп'ютерні мережі\ЛР10\server>python server.py Server is listening... Choose mode (sequential/threaded): sequential Server is running in sequential mode... New connection: ('127.0.0.1', 59080) Waiting for username... Waiting for password... User is not registered! Connection closed: ('127.0.0.1', 59080)</pre>	<pre>D:\Study\Комп'ютерні мережі\ЛР10\server>python client.py Connected to server! Enter username: myPassword Enter password: Rosaka Waiting for response... False D:\Study\Комп'ютерні мережі\ЛР10\server>_</pre>
--	---

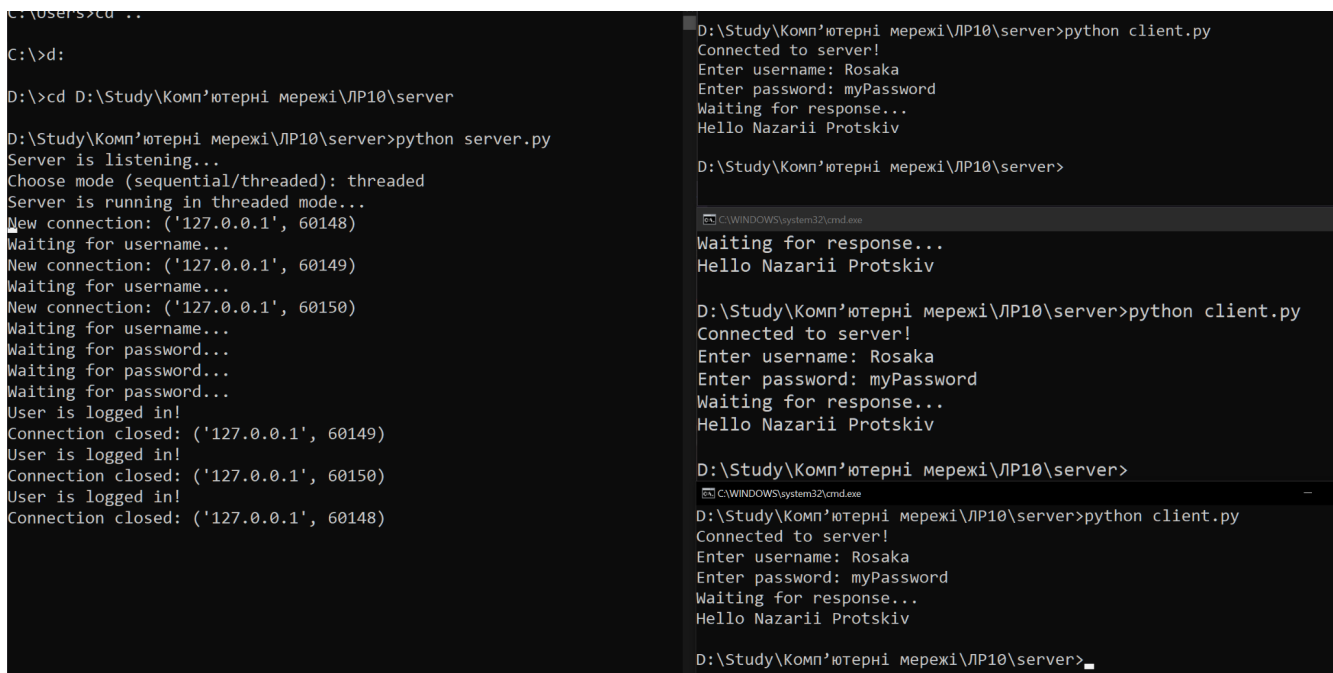
Порівняв роботу паралельного і послідовного серверів.

Послідовна робота (ліворуч сервер і його логи, праворуч три термінали):

<pre>D:\>cd D:\Study\Комп'ютерні мережі\ЛР10\server D:\Study\Комп'ютерні мережі\ЛР10\server>python server.py Server is listening... Choose mode (sequential/threaded): sequential Server is running in sequential mode... New connection: ('127.0.0.1', 59710) Waiting for username... Waiting for password... User is logged in! Connection closed: ('127.0.0.1', 59710) New connection: ('127.0.0.1', 59717) Waiting for username... Waiting for password... User is logged in! Connection closed: ('127.0.0.1', 59717) New connection: ('127.0.0.1', 59718) Waiting for username... Waiting for password... User is logged in! Connection closed: ('127.0.0.1', 59718)</pre>	<pre>D:\Study\Комп'ютерні мережі\ЛР10\server> D:\Study\Комп'ютерні мережі\ЛР10\server>python client.py Connected to server! Enter username: Rosaka Enter password: myPassword Waiting for response... Hello Nazarii Protskiv C:\WINDOWS\system32\cmd.exe Waiting for response... False D:\Study\Комп'ютерні мережі\ЛР10\server>python client.p Connected to server! Enter username: Rosaka Enter password: myPassword Waiting for response... Hello Nazarii Protskiv D:\Study\Комп'ютерні мережі\ЛР10\server>_ D:\Study\Комп'ютерні мережі\ЛР10\server> D:\Study\Комп'ютерні мережі\ЛР10\server>python client.py Connected to server! Enter username: Rosaka Enter password: myPassword Waiting for response... Hello Nazarii Protskiv D:\Study\Комп'ютерні мережі\ЛР10\server></pre>
--	---

З логів сервера видно, що запити були оброблені послідовно, один за одним.

Паралельна робота (ліворуч сервер і його логи, праворуч три термінали):



```
C:\Users>cd ..
C:\>d:
D:\>cd D:\Study\Комп'ютерні мережі\ЛР10\server
D:\Study\Комп'ютерні мережі\ЛР10\server>python server.py
Server is listening...
Choose mode (sequential/threaded): threaded
Server is running in threaded mode...
New connection: ('127.0.0.1', 60148)
Waiting for username...
New connection: ('127.0.0.1', 60149)
Waiting for username...
New connection: ('127.0.0.1', 60150)
Waiting for username...
Waiting for password...
Waiting for password...
Waiting for password...
User is logged in!
Connection closed: ('127.0.0.1', 60149)
User is logged in!
Connection closed: ('127.0.0.1', 60150)
User is logged in!
Connection closed: ('127.0.0.1', 60148)

D:\Study\Комп'ютерні мережі\ЛР10\server>python client.py
Connected to server!
Enter username: Rosaka
Enter password: myPassword
Waiting for response...
Hello Nazarii Protskiv

D:\Study\Комп'ютерні мережі\ЛР10\server>

C:\WINDOWS\system32\cmd.exe
Waiting for response...
Hello Nazarii Protskiv

D:\Study\Комп'ютерні мережі\ЛР10\server>python client.py
Connected to server!
Enter username: Rosaka
Enter password: myPassword
Waiting for response...
Hello Nazarii Protskiv

D:\Study\Комп'ютерні мережі\ЛР10\server>

C:\WINDOWS\system32\cmd.exe
Waiting for response...
Hello Nazarii Protskiv

D:\Study\Комп'ютерні мережі\ЛР10\server>
```

З логів сервера видно, що запити були оброблені паралельно, тобто в такій послідовності, в якій я вводив дані з трьох терміналів.

Висновок: Під час виконання цієї лабораторної роботи здобув знання і практичні навички, необхідні для програмування простих мережевих клієнт-серверних додатків з використанням сокетів.