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:

cтудент ПМІ-31

Процьків Назарій

Львів 2023

Мета: здобути знання і практичні навички, необхідних для програмування простих мережевих 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...")

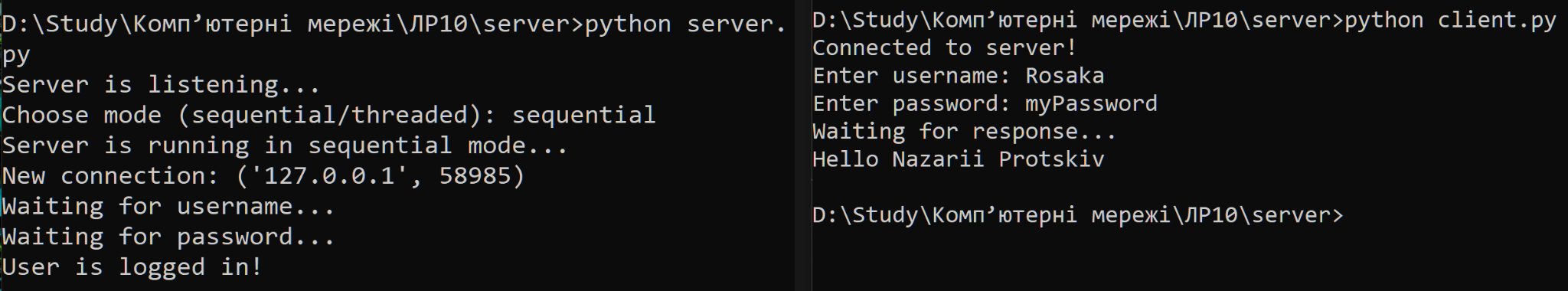
threaded\_server()

*else*:

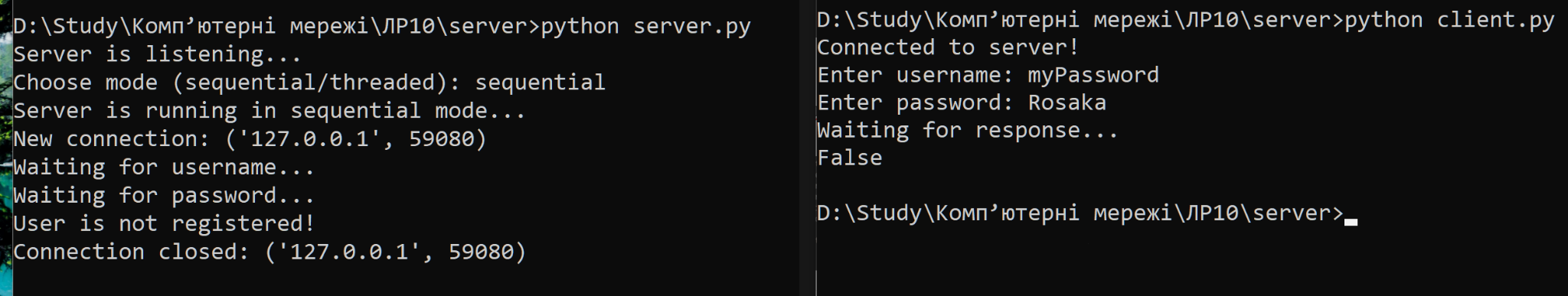
*print*("Invalid mode!")

1. Перевірив вхід користувача:

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

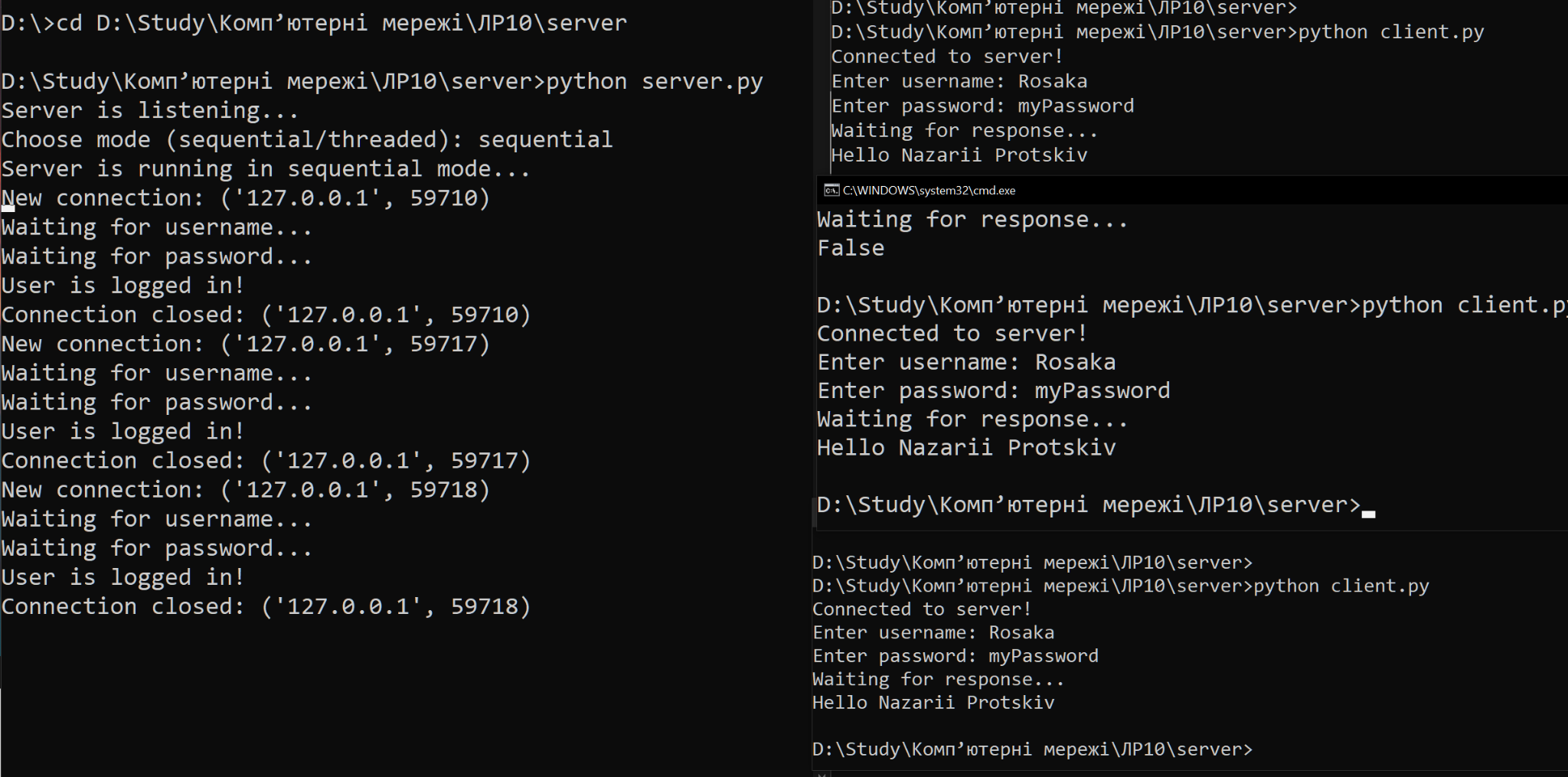


Ввів **неправильні** вхідні дані (отримав помилку):

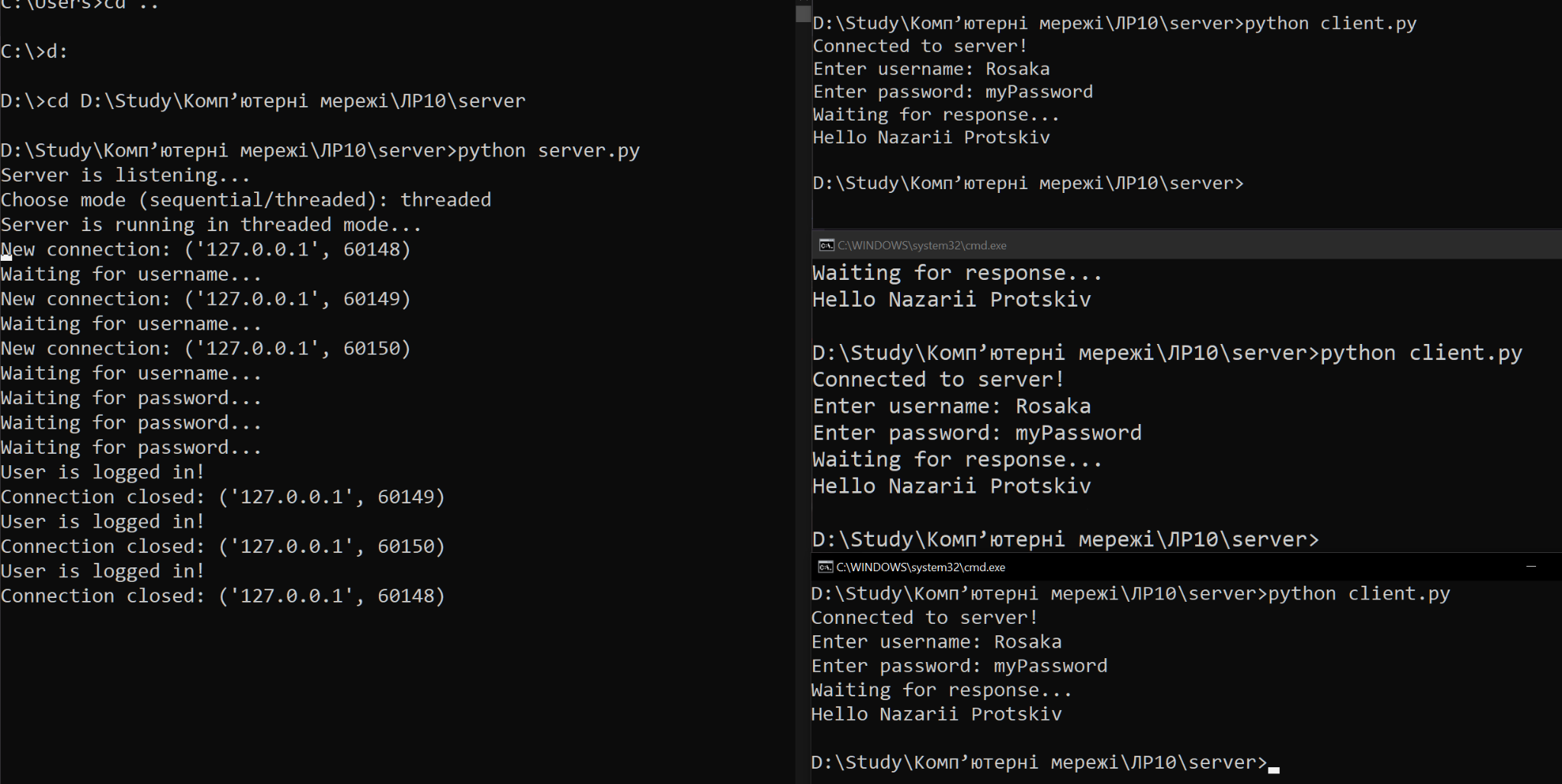


Порівняв роботу паралельного і послідовного серверів.

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



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

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

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.