

LAB 5
Socket Programming



Họ tên và MSSV: Lý Đại Phát

Nhóm học phần: CT293 Nhóm 02

- *Các sinh viên bị phát hiện sao chép bài của nhau sẽ nhận 0đ cho tất cả bài thực hành của môn này.*
- *Bài nộp phải ở dạng PDF, hình minh họa phải rõ ràng chi tiết. Hình minh họa chỉ cần chụp ở nội dung thực hiện, không chụp toàn màn hình.*

Github repo: https://github.com/xiaojiudev/CT293_Lab5

- Chứa các code minh họa, hình ảnh kết quả thực thi code.

1. Bài 1

Viết chương trình nhận đối số là một URL (Ví dụ: www.cit.ctu.edu.vn). Sử dụng TCP socket nối kết đến web server trong URL để lấy file HTML về, và hiển thị nội dung file HTML đó ra màn hình.

(Chụp hình minh họa code bài làm và kết quả thực thi; đính kèm tập tin code khi nộp bài.)

```
import socket

# Define function
1 usage  1dphat99
def get_and_display_html(url):
    # Extract the host and path from the URL
    if "://" in url:
        parts = url.split('/')
        host = parts[2]
        path = '/' + '/'.join(parts[3:])
    else:
        # Assuming HTTP if no protocol is specified
        parts = url.split('/')
        host = parts[0]
        path = '/' + '/'.join(parts[1:])

    # Create a TCP socket
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    try:
        # Connect to the web server on port 80 (HTTP)
        client_socket.connect((host, 80))

        # Send an HTTP GET request for the specified path
        request = f"GET {path} HTTP/1.1\r\nHost: {host}\r\n\r\n"
        client_socket.send(request.encode())

        # Receive and display the response
        while True:
            data = client_socket.recv(1024)
            if not data:
                break
            print(data.decode(), end='')

    except socket.error as e:
        print(f"Error: {e}")

    finally:
        client_socket.close()

# Run function
get_and_display_html("http://streamhd4k.com")
```

Hình 1 Mã nguồn ex1.py

```
"C:\Program Files\Python36\python.exe" D:\Study\NTT\A.MTTDL\Practice\Lab5\ex1\ex1.py
HTTP/1.1 200 OK
date: Tue, 31 Oct 2023 04:20:46 GMT
content-type: text/html; charset=utf-8
content-length: 1017
x-request-id: 8abae3a-7bd9-4fc7-9dec-9cc7a4db0329
cache-control: no-store, max-age=0
accept-ch: sec-ch-prefers-color=scheme
critical-ch: sec-ch-prefers-color=scheme
vary: sec-ch-prefers-color=scheme
x-adblock-key: MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBANDrp2lz7A0mADaNs8tA50LsWcjLFyQFcb/P2Txc58oY0eILb3vBw7J6f4pamkAQVSQuqYsKx3YzdUHCvbVZvFUsCAwEAAQ==_SIhd6z+488pu+HcF7LP/WH7ynT5EHdq
set-cookie: parking_session=8abae3a-7bd9-4fc7-9dec-9cc7a4db0329; expires=Tue, 31 Oct 2023 04:35:47 GMT; path=/

<!doctype html>
<html data-adblockkey="MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBANDrp2lz7A0mADaNs8tA50LsWcjLFyQFcb/P2Txc58oY0eILb3vBw7J6f4pamkAQVSQuqYsKx3YzdUHCvbVZvFUsCAwEAAQ==_SIhd6z+488pu+HcF7LP/WH7
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="icon" href="data:image/png;base64,iVBORw0KGgoAAAANSUHEUgAAAAEAAAABCAIAAACQd1PeAAAAAE1EQVQ1I2P4//8/AAx+Av7czFnnAAAAAE1FTkSuQmCC">
  <link rel="preconnect" href="https://www.google.com" crossorigin>
</head>
<body>
<div id="target" style="opacity: 0"></div>
<script>window.park = "eyJldWlkIjo1OGFiYWVlM2EtN2JkOj0S00ZmM3LTlkZWMT0WNjN2E0ZG1wMzI5Iiwic6FnZV90aWw1Ijoxhjk4NzI2MDQ3LjCjwYwDlX3VybCI6Imh0dHA6Ly9zdHJlYW1oZDRrLmNvbS8iLCJwYXVldlX211
<script src="/b0IAwqAkW.js"></script>
</body>
</html>
```

Hình 2 Kết quả thu được khi thực thi ex1.py (URL được dùng http://streamhd4k.com)

2. Bài 2

Viết chương trình theo mô hình Client-Server sử dụng TCP Socket.

Trong đó:

- + Server lắng nghe ở cổng 8888, làm nhiệm vụ đọc một ký tự số từ '0' đến '9'.
(Ví dụ: nhận số 0, trả về "khong"; 1, trả về "mot" ; 9, trả về "chin"; nhận ký tự khác số thì trả về "Không phải số nguyên").
- + Client sẽ nhập vào 1 ký tự, gửi qua Server, nhận kết quả trả về từ Server và thể hiện lên màn hình.

(Chụp hình minh họa code bài làm và kết quả thực thi; đính kèm tập tin code khi nộp bài.)

```
import socket

1 usage  👤 ldphat99
def convert_to_word(data):
    conversion_dict = {
        '0': "zero",
        '1': "one",
        '2': "two",
        '3': "three",
        '4': "four",
        '5': "five",
        '6': "six",
        '7': "seven",
        '8': "eight",
        '9': "nine"
    }

    return conversion_dict.get(data, "Not an integer")

1 usage  👤 ldphat99
def main():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind(('0.0.0.0', 8888))
    server_socket.listen(1)
    print("Server is listening on port 8888...")

    while True:
        client_socket, client_address = server_socket.accept()
        print(f"Accepted connection from {client_address}")

        data = client_socket.recv(1).decode()

        response = convert_to_word(data)
        client_socket.send(response.encode())

        client_socket.close()

if __name__ == "__main__":
    main()
```

Hình 3 Mã nguồn serverTCP.py

```
import socket

1 usage  👤 ldphat99
def main():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(('localhost', 8888))

    user_input = input("Enter a numeric character (0-9): ")

    if len(user_input) == 1 and user_input.isnumeric():
        client_socket.send(user_input.encode())
        response = client_socket.recv(1024).decode()
        print(f"\nServer response: {response}")
    else:
        print("\nInvalid input. Please enter a single numeric character (0-9).")

    client_socket.close()

if __name__ == "__main__":
    main()
```

Hình 4 Mã nguồn clientTCP.py

```
"C:\Program Files\Python36\python.exe" D:\Study\CNTT\A.MTTDL\Practice\Lab5\ex2\serverTCP.py
Server is listening on port 8888...
Accepted connection from ('127.0.0.1', 35350)
```

Hình 5 Kết quả thực thi serverTCP.py và clientTCP.py (1)

```
"C:\Program Files\Python36\python.exe" D:\Study\CNTT\A.MTTDL\Practice\Lab5\ex2\clientTCP.py
Enter a numeric character (0-9): 1

Server response: one

Process finished with exit code 0
```

Hình 6 Kết quả thực thi serverTCP.py và clientTCP.py (2)

3. Bài 3

Viết chương trình theo mô hình Client-Server sử dụng UDP Socket.

Trong đó:

- + Server hoạt động ở cổng 8888, sẽ nhận từ Client các yêu cầu là một chuỗi có khuôn dạng như sau:

"OP Operant1 Operant2\n" Ví dụ: *" + 100 200 "*

Trong đó:

- OP là một ký tự chỉ phép toán muốn thực hiện: '+', '-', '*', '/'
- Operant1, Operant2 là đối số của phép toán.
- Các thành phần trên cách nhau bởi 1 ký tự trắng ' '
- Mỗi khi server nhận được một thông điệp nó sẽ thực hiện phép toán "Operant1 OP Operant2" để cho ra kết quả sau đó đổi kết quả thành chuỗi và gửi về Client.
- + Client cho phép người dùng nhập các phép toán muốn tính theo cách thức thông thường, ví dụ: 100 + 200. Client tạo ra thông điệp yêu cầu theo đúng dạng do Server quy định rồi gửi sang Server, chờ nhận kết quả trả về và in ra màn hình.
- + Kết thúc yêu cầu bằng ký tự xuống dòng '\n'

(Chụp hình minh họa code bài làm và kết quả thực thi; đính kèm tập tin code khi nộp bài.)

```
import socket

localIP = "127.0.0.1"
localPort = 20001
bufferSize = 1024

UDPServerSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)
UDPServerSocket.bind((localIP, localPort))

print("UDP server up and listening")

while True:
    data, address = UDPServerSocket.recvfrom(bufferSize)
    message = data.decode('utf-8').strip()

    try:
        parts = message.split()

        print("Client request: ", parts) # format: ['+', '100', '200']

        result = 0

        if (len(parts) == 3
            and parts[0] in ('+', '-', '*', '/')
            and parts[1].isdigit()
            and parts[2].isdigit()):
            operand1, operator, operand2 = int(parts[1]), parts[0], int(parts[2])

            if operator == '+':
                result = operand1 + operand2
            elif operator == '-':
                result = operand1 - operand2
            elif operator == '*':
                result = operand1 * operand2
            elif operator == '/':
                if operand2 != 0:
                    result = operand1 / operand2
                else:
                    UDPServerSocket.sendto("Division by zero error".encode('utf-8'), address)
                    continue

            UDPServerSocket.sendto(str(result).encode('utf-8'), address)
        else:
            UDPServerSocket.sendto("Invalid format. Use 'Operand1 Operator Operand2'.".encode('utf-8'),
                                    address)
    except (ValueError, IndexError):
        UDPServerSocket.sendto("Invalid format. Use 'Operand1 Operator Operand2'.".encode('utf-8'),
                                address)
```

Hình 7 Mã nguồn serverUDP.py

```
import socket

serverAddressPort = ("127.0.0.1", 20001)
bufferSize = 1024

UDPClietSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)

while True:
    user_input = input("Enter a calculation (e.g., 100 + 200) or 'q' to quit: ")

    if user_input.lower() == 'q':
        break

    try:
        operator, operands = user_input.split()[1], user_input.split()[2:]
        formatted_message = f"{operator} {' '.join(operands)}"

        message = formatted_message.encode('utf-8')
        UDPClietSocket.sendto(message, serverAddressPort)

        response, _ = UDPClietSocket.recvfrom(bufferSize)
        print(f"\nServer response: {response.decode('utf-8')}\n")
    except ValueError:
        print("Invalid input. Use 'Operand1 Operator Operand2' format.\n")
```

Hình 8 Mã nguồn clientUDP.py

```
"C:\Program Files\Python36\python.exe" D:\Study\CNTT\A.MTTDL\Practice\Lab5\ex3\serverUDP.py
UDP server up and listening
Client request: ['+', '100', '2']
Client request: ['-', '100', '2']
Client request: ['*', '100', '2']
Client request: ['/', '100', '2']
Client request: ['@', '100', '2']
```

Hình 9 Kết quả thực thi serverUDP.py và clientUDP.py (1)


```
"C:\Program Files\Python36\python.exe" D:\Study\CNTT\A.MTTDL\Practice\Lab5\ex3\clientUDP.py
Enter a calculation (e.g., 100 + 200) or 'q' to quit: 100 + 2
Server response: 102

Enter a calculation (e.g., 100 + 200) or 'q' to quit: 100 - 2
Server response: 98

Enter a calculation (e.g., 100 + 200) or 'q' to quit: 100 * 2
Server response: 200

Enter a calculation (e.g., 100 + 200) or 'q' to quit: 100 / 2
Server response: 50.0

Enter a calculation (e.g., 100 + 200) or 'q' to quit: 100 @ 2
Server response: Invalid format. Use 'Operand1 Operator Operand2'.

Enter a calculation (e.g., 100 + 200) or 'q' to quit: q

Process finished with exit code 0
```

Hình 10 Kết quả thực thi serverUDP.py và clientUDP.py (2)

4. Bài 4

Viết chương trình theo mô hình Client-Server sử dụng TCP socket.

Trong đó,

- + Server hoạt động ở 2 cổng (2 TCP socket) 8001 và 8002:
 - Cổng 8000: sử dụng để nhận các lệnh từ Client và gửi phản hồi cho Client.
 - Cổng 8001: sử dụng để gửi dữ liệu tập tin/thư mục tới Client.
- + Ở cổng 8000, Server sẽ nhận từ Client các câu lệnh có dạng như sau:
 - "GET <Tên file>".
 - Nếu file yêu cầu tồn tại: Server sẽ gửi trả cho Client chuỗi "OK\n". Sau đó là nội dung file sẽ được gửi tới Client qua cổng 8001.
 - Nếu file không tồn tại: Server sẽ gửi trả cho Client chuỗi "ERROR\n".
 - Client nhận nội dung tập tin từ Server và lưu lại.
 - "DELETE <Tên file>".
 - Nếu file yêu cầu tồn tại: Server sẽ gửi trả cho Client chuỗi "OK\n". Xóa file đó trên thư mục hiện hành của Server.
 - Nếu file không tồn tại: Server sẽ gửi trả cho Client chuỗi "ERROR\n".

- “LIST <Tên thư mục>”.
 - Nếu thư mục yêu cầu tồn tại: Server sẽ gửi trả cho Client chuỗi “OK\n”. Sau đó là nội dung thư mục sẽ được gửi tới Client qua cổng 8001.
 - Nếu thư mục không tồn tại: Server sẽ gửi trả cho Client chuỗi “ERROR\n”.
 - Client nhận nội dung thư mục từ Server và in ra màn hình.
 - + Client sẽ cho phép người dùng gửi các lệnh qua Server và nhận phản hồi ở cổng 8000.
 - + Kết thúc yêu cầu bằng ký tự xuống dòng '\n'
- (Chụp hình minh họa code bài làm và kết quả thực thi; đính kèm tập tin code khi nộp bài.)

```
import ...

HOST = "localhost"
PORT = 8000
PORT1 = 8001

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((HOST, PORT))

server1 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server1.bind((HOST, PORT1))

server.listen(1)
server1.listen(1)

print("Server on port 8000 is ready to accept connections")

while True:
    conn, add = server.accept()
    conn1, add1 = server1.accept()

    request = conn.recv(1024).decode("utf-8")
    filename = request.split()[1]
    method = request.split()[0]

    if method == "GET":
        if os.path.isfile("server_storage/" + filename):
            conn.send(b"OK\n")
            with open("server_storage/" + filename, "r") as f:
                data = f.read()
                conn1.send(data.encode("utf-8"))
            print("File sent to", add)
            print("Content:", data)
        else:
            print("File not found:", filename)
            conn.send(b"ERROR\n")

    if method == "DELETE":
        file_path = os.path.join("server_storage", filename)
        if os.path.isfile(file_path):
            conn.send(b"OK\n")
            os.remove(file_path)
            print("File", filename, "has been deleted from the server's server_storage directory")
        else:
            conn.send(b"ERROR\n")
            print("File not found:", filename)

    if method == "LIST":
        if os.path.isdir(filename):
            conn.send(b"OK\n")
            files = os.listdir(filename)
            print(files)
            data = (str(files).encode("utf-8"))
            conn1.send(data)
        else:
            conn.send("ERROR\n".encode("utf-8"))
            print("Folder not found:", filename)

    conn.close()
    conn1.close()
```

Hình 11 Mã nguồn server.py

```
import socket

HOST = "localhost"
PORT = 8000
PORT1 = 8001

1 usage  👤 ldphat99
def handle_get():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect((HOST, PORT))

    client1 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client1.connect((HOST, PORT1))

    filename = input("\nEnter the name of the file to retrieve: ")
    request = "GET " + filename
    client.send(request.encode("utf-8"))

    response = client.recv(1024).decode("utf-8")

    if response == "OK\n":
        with open(f"client_storage/{filename}", "w") as serverFile:
            while True:
                data = client1.recv(1024)
                if not data:
                    break
                serverFile.write(data.decode("utf-8"))
        print(f"\nFile received and saved in client_storage directory: {filename}")
        with open(f"client_storage/{filename}", "r") as clientFile:
            contents = clientFile.read()
            print(f"\nFile contents: {contents}")
    else:
        print("Server on port 8000 Response:", response)
```

Hình 12 Mã nguồn client.py (1)

```
def handle_delete():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect((HOST, PORT))

    client1 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client1.connect((HOST, PORT1))

    filename = input("\nEnter the name of the file to delete: ")
    request = "DELETE " + filename
    client.send(request.encode("utf-8"))

    response = client.recv(1024).decode("utf-8")

    if response == "OK\n":
        print("\nFile has been deleted on the server")
    else:
        print("Server on port 8000 Response:", response)

1 usage  👤 ldphat99
def handle_list():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect((HOST, PORT))

    client1 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client1.connect((HOST, PORT1))

    filename = input("\nEnter the folder name: ")
    request = "LIST " + filename
    client.send(request.encode("utf-8"))

    response = client.recv(1024).decode("utf-8")

    if response == "OK\n":
        file = client1.recv(1024)
        array = eval(file.decode("utf-8"))
        file_list = ', '.join(array)
        print(f'\nFiles in the {filename} directory are: {file_list}\n')
    else:
        print("Server on port 8000 Response:", response)
```

Hình 13 Mã nguồn client.py (2)

```
test = True
while test:
    print('\n' + '-' * 40 + '\n')
    print("Press 1 to choose the GET function\n")
    print("Press 2 to choose the DELETE function\n")
    print("Press 3 to choose the LIST function")
    print('\n' + '-' * 40 + '\n')
    a = (input("Choose one of the three functions to perform: "))
    if a == "1":
        handle_get()
    if a == "2":
        handle_delete()
    if a == "3":
        handle_list()
    if a == "":
        test = False
    print("End of work session")
```

Hình 14 Mã nguồn client.py (3)

```
"C:\Program Files\Python36\python.exe" D:\Study\CNTT\A.MTTDL\Practice\Lab5\ex4\client.py

-----

Press 1 to choose the GET function

Press 2 to choose the DELETE function

Press 3 to choose the LIST function

-----

Choose one of the three functions to perform: 1

Enter the name of the file to retrieve: data.txt

File received and saved in client_storage directory: data.txt

File contents: Hello World

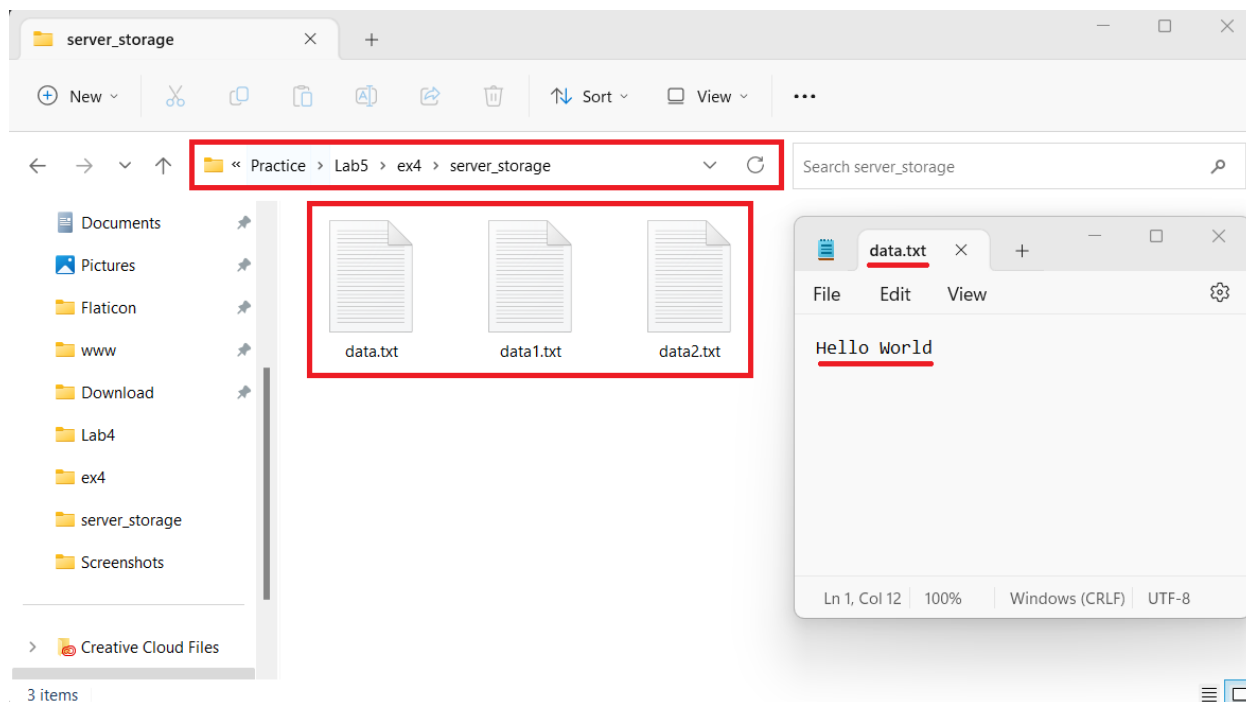
-----
```

Hình 15 Kiểm tra chức năng GET (1)

```
"C:\Program Files\Python36\python.exe" D:\Study\CNTT\A.MTTDL\Practice\Lab5\ex4\server.py
Server on port 8000 is ready to accept connections

File sent to ('127.0.0.1', 34734)
Content: Hello World
```

Hình 16 Kiểm tra chức năng GET (2)



Hình 17 Kiểm tra chức năng GET (3)

```
"C:\Program Files\Python36\python.exe" D:\Study\CNTT\A.MTTDL\Practice\Lab5\ex4\client.py

-----

Press 1 to choose the GET function

Press 2 to choose the DELETE function

Press 3 to choose the LIST function

-----

Choose one of the three functions to perform: 3

Enter the folder name: server_storage

Files in the server_storage directory are: data.txt, data1.txt, data2.txt

-----
```

Hình 18 Kiểm tra chức năng LIST (1)


```
"C:\Program Files\Python36\python.exe" D:\Study\CNTT\A.MTTDL\Practice\Lab5\ex4\server.py
Server on port 8000 is ready to accept connections
['data.txt', 'data1.txt', 'data2.txt']
```

Hình 19 Kiểm tra chức năng LIST (2)

```
"C:\Program Files\Python36\python.exe" D:\Study\CNTT\A.MTTDL\Practice\Lab5\ex4\client.py

-----

Press 1 to choose the GET function

Press 2 to choose the DELETE function

Press 3 to choose the LIST function

-----

Choose one of the three functions to perform: 2

Enter the name of the file to delete: data.txt

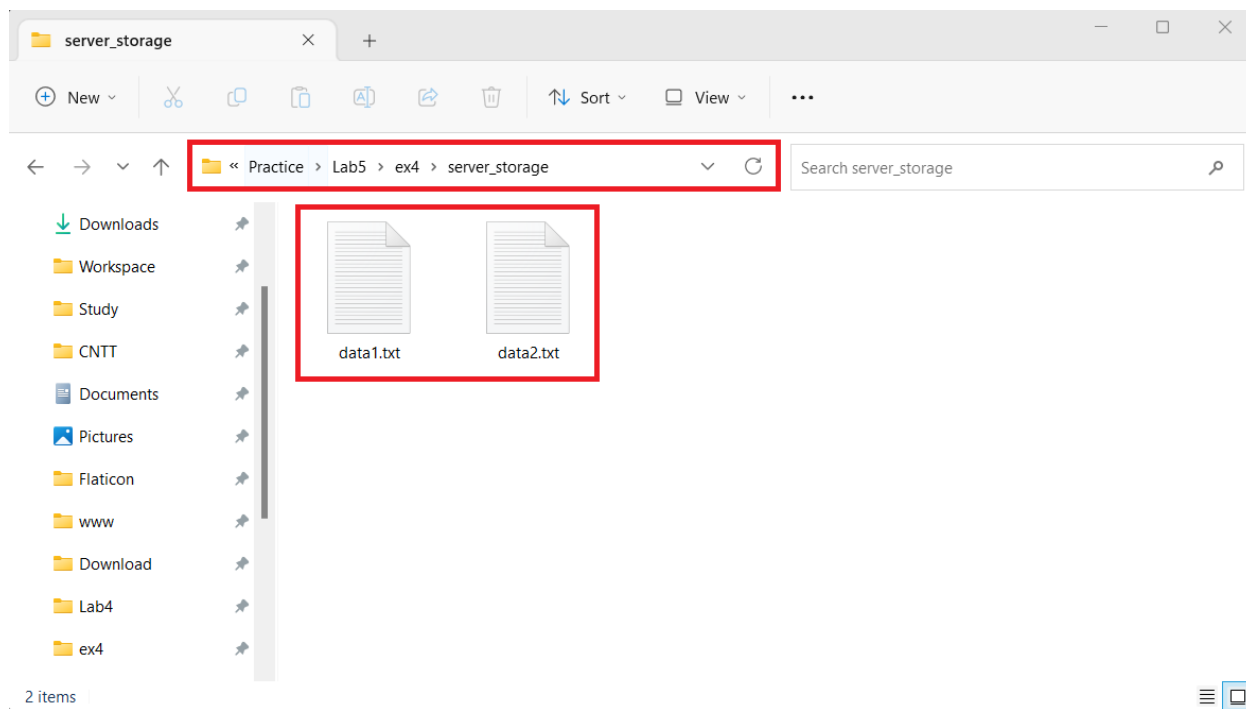
File has been deleted on the server

-----
```

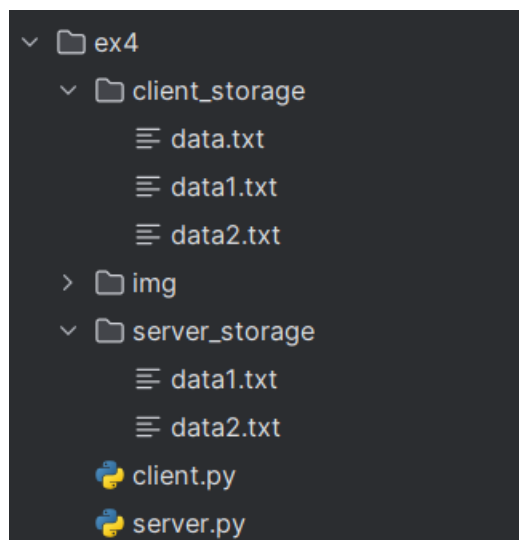
Hình 20 Kiểm tra chức năng DELETE (1)

```
"C:\Program Files\Python36\python.exe" D:\Study\CNTT\A.MTTDL\Practice\Lab5\ex4\server.py
Server on port 8000 is ready to accept connections
File data.txt has been deleted from the server's server_storage directory
```

Hình 21 Kiểm tra chức năng DELETE (2)



Hình 22 Kiểm tra chức năng DELETE (3)



Hình 23 Cấu trúc thư mục bài tập 4

--- Hết ---