# CS 461

## Lab Assignment 2

Name: Gandhi Dhruv Vipulkumar

Institute ID: 202151053

Date: 9-9-2024

Q. Implement multiple client and single server architecture

**Server.py**

```
import socket
import threading

def handle_client(client_socket, client_address):
    client_ip, client_port = client_address
    print(
        f"[NEW CONNECTION] Client IP: {client_ip}, Port:
{client_port} connected.")

    connected = True
    while connected:
        try:
            # Receive the initial message from the client
            message = client_socket.recv(1024).decode("utf-8")

            if message == "FILE":
                # Receive the filename
                filename = client_socket.recv(1024).decode("utf-8")
                print(f"[{client_ip}:{client_port}] Receiving file:
{filename}")

                # Open a file with the received filename
                with open(f"received_{filename}", "wb") as f:
                    while True:
                        file_data = client_socket.recv(1024)
                        if not file_data:
                            break
                        f.write(file_data)

                print(f"[{client_ip}:{client_port}] File {filename}
```

```
received.")
                client_socket.send(
                    f"File {filename} received
successfully.".encode("utf-8"))
            else:
                print(f"[{client_ip}:{client_port}] {message}")
                client_socket.send(
                    f"Message received: {message}".encode("utf-8"))

        except Exception as e:
            print(f"Error: {e}")
            connected = False

    client_socket.close()
    print(
        f"[DISCONNECT] Client IP: {client_ip}, Port: {client_port}
disconnected.")

def start_server():
    server_ip = "192.168.1.6"  # Replace with your Wi-Fi IP address
    server_port = 5555

    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server.bind((server_ip, server_port))
    server.listen(5)
    print(f"[LISTENING] Server is listening on
{server_ip}:{server_port}")

    while True:
        client_socket, client_address = server.accept()
        thread = threading.Thread(
            target=handle_client, args=(client_socket,
client_address))
        thread.start()
        print(f"[ACTIVE CONNECTIONS] {threading.active_count() -
1}")

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

**Client.py**

```
import socket
import os

def send_message(client_socket):
    message = input("Enter message to send: ")
    client_socket.send(message.encode("utf-8"))
    response = client_socket.recv(1024).decode("utf-8")
    print(f"Server: {response}")

def send_file(client_socket):
    file_path = input("Enter the file path to send: ")
    filename = os.path.basename(file_path)

    # Notify the server that you're sending a file
    client_socket.send("FILE".encode("utf-8"))

    # Send the file name
    client_socket.send(filename.encode("utf-8"))

    # Send the actual file content
    with open(file_path, "rb") as f:
        while (file_data := f.read(1024)):
            client_socket.send(file_data)

    print("File sent successfully.")
    response = client_socket.recv(1024).decode("utf-8")
    print(f"Server: {response}")

def start_client():
    server_ip = "192.168.1.6"  # Replace with the server Wi-Fi IP
address
    server_port = 5555

    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client.connect((server_ip, server_port))

    connected = True
    while connected:
        option = input(
            "Enter '1' to send a message, '2' to send a file, or
'quit' to exit: ").strip()

        if option == "1":
            send_message(client)
```

```
        elif option == "2":
            send_file(client)
        elif option.lower() == "quit":
            connected = False
        else:
            print("Invalid option, please try again.")

    client.close()

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

**Features:**

- The server is running on a machine with IP address assigned by Wifi router.
- Multiple clients are connecting with different IP addresses
- Clients can send message and share files limited to 1024 bytes.

**Testing Phase:**

**IP address of clients:**

Client-1: 192.168.1.7

Client-2: 192.186.1.6

**IP address of server: 192.168.1.6**

1) **Client-1 sending greetings and file basic.py**

```
[LISTENING] Server is listening on 192.168.1.6:5555
[NEW CONNECTION] Client IP: 192.168.1.6, Port: 61913 connected.[ACTIVE CONNECTIONS] 1
```

```
[NEW CONNECTION] Client IP: 192.168.1.7, Port: 49846 connected.
[ACTIVE CONNECTIONS] 1
[NEW CONNECTION] Client IP: 192.168.1.6, Port: 61946 connected.[ACTIVE CONNECTIONS] 2

[192.168.1.7:49846] this is dhruv from client-1
[192.168.1.7:49846] Receiving file: basic.py
```

```
[192.168.1.7:49846] Receiving file: basic.py

def swap(num1,num2):
    num1 = num1 ^ num2
    num2 = num1 ^ num2
    num1 = num1 ^ num2
    print(num1 , num2)

def conbit(num):
    str = ""
    while num >=1:
        if num % 2 == 0:
            str = "0" + str
        else:
            str = "1" + str
        num = num // 2
    return str
```

2) **Client -2 sending message and sample_text.txt file to server**

```
Enter '1' to send a message, '2' to send a file, or 'quit' to exit: 1
Enter message to send: Hello!! This is Dhruv from client-2
Server: Message received: Hello!! This is Dhruv from client-2
Enter '1' to send a message, '2' to send a file, or 'quit' to exit: 2
Enter the file path to send: sample_text.txt
File sent successfully.
```

```
[192.168.1.6:62351] Hello!! This is Dhruv from client-2
[192.168.1.6:62351] Receiving file: sample_text.txt
```

**Conclusion:** Messages and files received successfully to the server