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:**

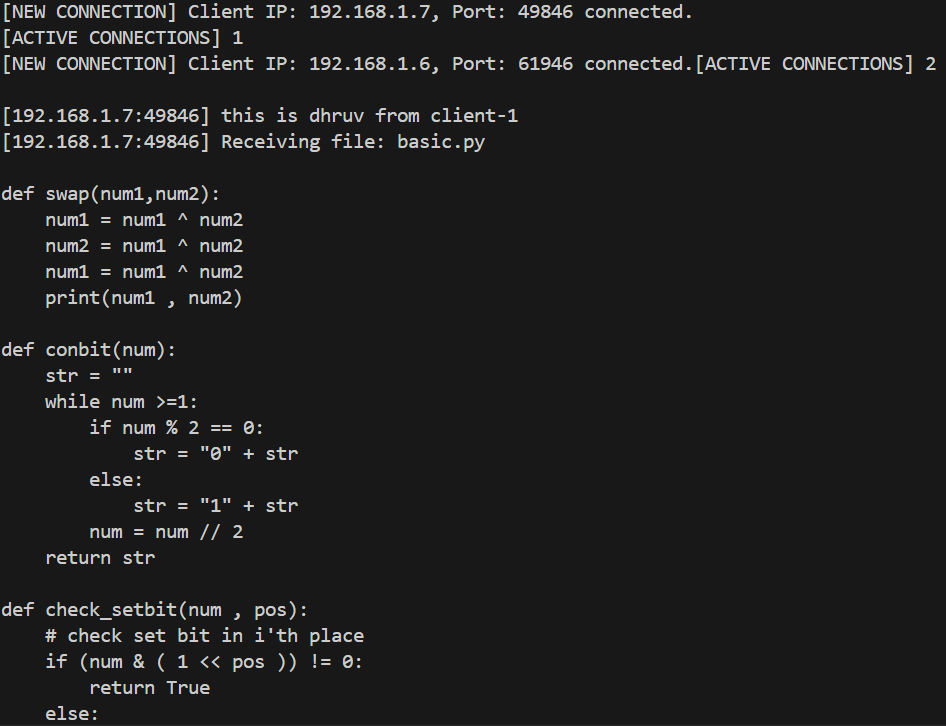
Client-1: 192.168.1.7

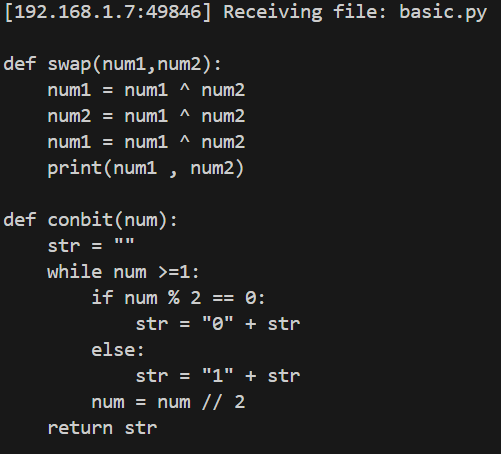
Client-2: 192.186.1.6

**IP address of server: 192.168.1.6**

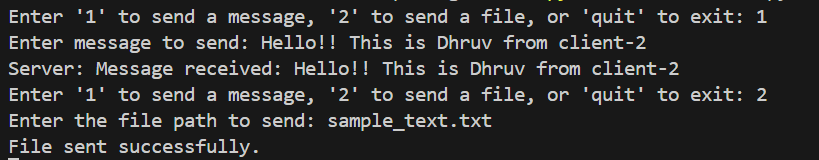
1. **Client-1 sending greetings and file basic.py**

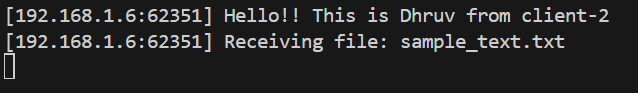






1. **Client -2 sending message and sample\_text.txt file to server**

****

****

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