CS 461

Lab Assignment 3

Name: Gandhi Dhruv Vipulkumar

Institute ID: 202151053

Date: 23-9-2024

Q. Implement multiple client and multiple server architecture using Java.

MultiServer.java

```
import java.io.*;
import java.net.ServerSocket;
import java.net.Socket;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

public class MultiServer {

    // Each server listens on a different port, same Wi-Fi IP
address
    private static final String SERVER_IP = "192.168.1.14";
    private static final int PORT = 5557; // Change for each server
instance like 5555, 5556, 5557, etc.
    private static final int THREAD_POOL_SIZE = 10; // Fixed thread
pool size

    public static void main(String[] args) {
        ExecutorService clientHandlerPool =
Executors.newFixedThreadPool(THREAD_POOL_SIZE);

        try (ServerSocket serverSocket = new ServerSocket(PORT)) {
            System.out.println("Server is listening on " + SERVER_IP
+ ":" + PORT);

            while (true) {
                Socket clientSocket = serverSocket.accept();
                System.out.println("New client connected: " +
clientSocket.getInetAddress());
```

```
                clientHandlerPool.execute(new
ClientHandler(clientSocket)); // Handle each client in a separate
thread
            }

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}

class ClientHandler implements Runnable {

    private final Socket clientSocket;

    public ClientHandler(Socket socket) {
        this.clientSocket = socket;
    }

    @Override
    public void run() {
        try (BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
                PrintWriter out = new
PrintWriter(clientSocket.getOutputStream(), true)) {

            String message;
            while ((message = in.readLine()) != null) {
                System.out.println("Received: " + message);
                out.println("Echo: " + message); // Echo back the
message
            }

        } catch (IOException e) {
            System.err.println("Error handling client: " +
e.getMessage());
        } finally {
            try {
                clientSocket.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Client.java

```
import java.io.*;
import java.net.Socket;

public class Client {

    public static void main(String[] args) {
        // Replace with the Wi-Fi IP address of the server and
respective port
        String SERVER_IP = "192.168.1.14"; // Same Wi-Fi IP address
for all servers
        int SERVER_PORT = 5557; // Change this to connect to
different servers (5555, 5556, etc.)

        try (Socket socket = new Socket(SERVER_IP, SERVER_PORT);
                BufferedReader in = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
                PrintWriter out = new
PrintWriter(socket.getOutputStream(), true);
                BufferedReader userInput = new BufferedReader(new
InputStreamReader(System.in))) {

            // Start a thread to read server responses
            Thread readThread = new Thread(() -> {
                String response;
                try {
                    while ((response = in.readLine()) != null) {
                        System.out.println("Server: " + response);
                    }
                } catch (IOException e) {
                    System.err.println("Error reading from server: "
+ e.getMessage());
                }
            });
            readThread.start();

            // Sending user input to the server
            String message;
            while ((message = userInput.readLine()) != null) {
                out.println(message); // Send message to server
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
```

```
    }
}
```

Code Explanation:

- The server is running on a machine with IP address assigned by Wifi router and multiple instance of server are created on different PORT, and each instances are running at a same time.
- The Clients are connecting with different IP addresses to different ports hosted by server.
- Multiple clients can send message to different server instances.

Testing Phase:

IP address of clients:

Client-1: 192.168.1.2

Client-2: 192.186.1.10

Client-3: 192.168.1.14

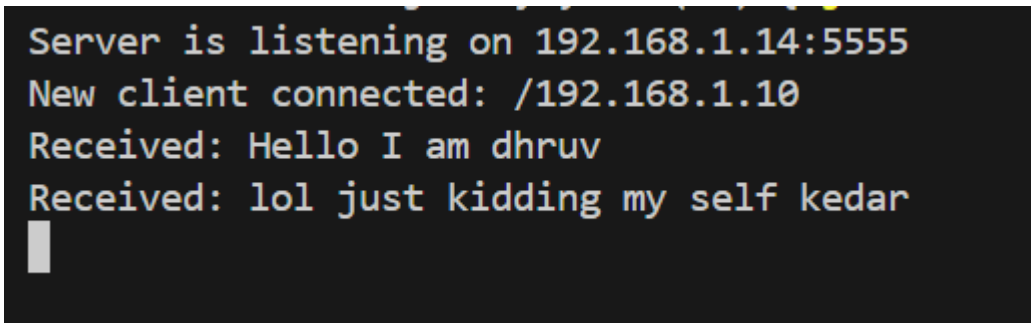
IP address of servers:

- 192.168.1.14:5555
- 192.168.1.14:5556
- 192.168.1.14:5557

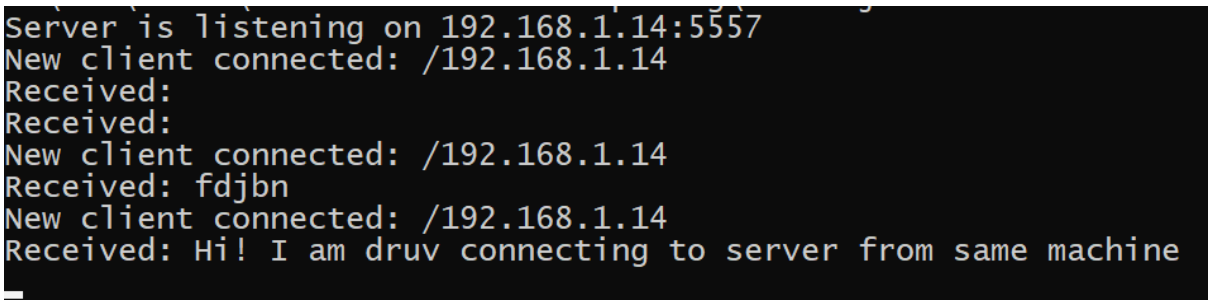
1) Client-1 sending greetings on server 192.168.1.14:5556

```
Server is listening on 192.168.1.14:5556
New client connected: /192.168.1.2
Received: koonichiwa wore wa kira
Received: hi iam hari
```

2) Client-2 sending message on server 192.168.1.14:5555



3) Client-3 sending message on server 192.168.1.14:5557



Note: All the communication is happening at same time.

Conclusion: Multiple clients can send messages to different servers simultaneously.