Multi-threaded TCP Server

CS F303

BITS Pilani
Dubai Campus

Dr. Pranav M. Pawar

Problem Statement

- Develop a multithreaded TCP server and verify the same.

Multithreaded TCP server (1)

- Step 1: Write Multithreaded TCP server program using Socket programming.
  - Create Server.java file and write following code in it.

```
import java.io.*;
import java.net.*;
class Server {
    public static void main(String[] args)
    {
        ServerSocket server = null;
        try {
            // server is listening on port 8009
            server = new ServerSocket(8009);
            //setReuseAddress () method of Java Socket class
            //enables or disables the SO_REUSEADDR socket option.
            //The initial setting of SO_REUSEADDR is disabled.
            server.setReuseAddress(true);
            // running infinite loop for getting client request
            while (true) {
                // socket object to receive incoming client requests
                Socket client = server.accept();
```

Multithreaded TCP server (2)

```
        // Displaying that new client is connected to server
        //getInetAddress () method either returns the
        //remote IP address to which the socket is connected
        //getHostAddress() method of InetAddress class
        //returns the IP address string in textual presentation.
        System.out.println("New client connected" + client.getInetAddress().getHostAddress());
        // create a new thread object
        ClientHandler1 clientSock = new ClientHandler1(client);
        // This thread will handle the client separately
        new Thread(clientSock).start();
    }
}
```

Multithreaded TCP server (3)

Step 2: Create ClientHandler.java for handling multiple clients.

```
import java.io.*;
import java.net.*;
public class ClientHandler1 implements Runnable
    {
        private final Socket clientSocket;
        public ClientHandler1(Socket socket)
        {
            this.clientSocket = socket;
        }
        public void run()
        {
            PrintWriter out = null;
            BufferedReader in = null;
            try {
```

Multithreaded TCP server (4)

```
    // get the outputstream of client
    out = new PrintWriter(clientSocket.getOutputStream(), true);
    // get the inputstream of client
    in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));
    String line;
    while ((line = in.readLine()) != null) {
        // writing the received message from client
        System.out.printf(" Sent from the client: %s\n",line);
        out.println(line);
        out.flush();
        if (line.trim().equals("BYE"))
        {
            System.out.println("Client socket close");
            break;
        }
    }
    clientSocket.close();
}
catch (IOException e) {
    e.printStackTrace();
}
```

Multithreaded TCP server (5)

```
        finally {
            try {
                if (out != null) {
                    out.close();
                }
                if (in != null) {
                    in.close();
                    clientSocket.close();
                }
            }
            catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

TCP client (1)

Step 3: Create Client.java

```
import java.io.*;
import java.net.*;
import java.util.*;
class Client {
    public static void main(String[] args)
    {
        // establish a connection by providing host and port number
        try (Socket socket = new Socket("localhost", 8009))
        {
            // writing to server
            PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
            // reading from server
            BufferedReader in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            // object of scanner class
            Scanner sc = new Scanner(System.in);
            String line = null;
```

TCP client (2)

```
        while (!"exit".equalsIgnoreCase(line))
            {
                // reading from user
                line = sc.nextLine();
                // sending the user input to server
                out.println(line);
                out.flush();
                // displaying server reply
                System.out.println("Server replied " + in.readLine());
            }
        // closing the scanner object
        sc.close();
    }
    catch (IOException e) {
        e.printStackTrace();
    }
  }
}
```

Execute Multithreaded Server and Clients

- Step 4: Run the server and connect multiple clients to it.

Command Prompt - java -jar Server.jar

```
D:\2020-21_sem2\CN\Practical\Assignme
nt\Ass8>java -jar Server.jar
New client connected127.0.0.1
 Sent from the client: Hi from client
 1
New client connected127.0.0.1
 Sent from the client: Hi from client
 2
 Sent from the client: BYE
Client socket close
```

Command Prompt - java -jar Client.jar localhost 8009

```
D:\2020-21_sem2\CN\Practical\Assignment\As
s8>java -jar Client.jar localhost 8009
Hi from client 1
Server replied Hi from client 1
```

Command Prompt - java -jar Client.jar localh...

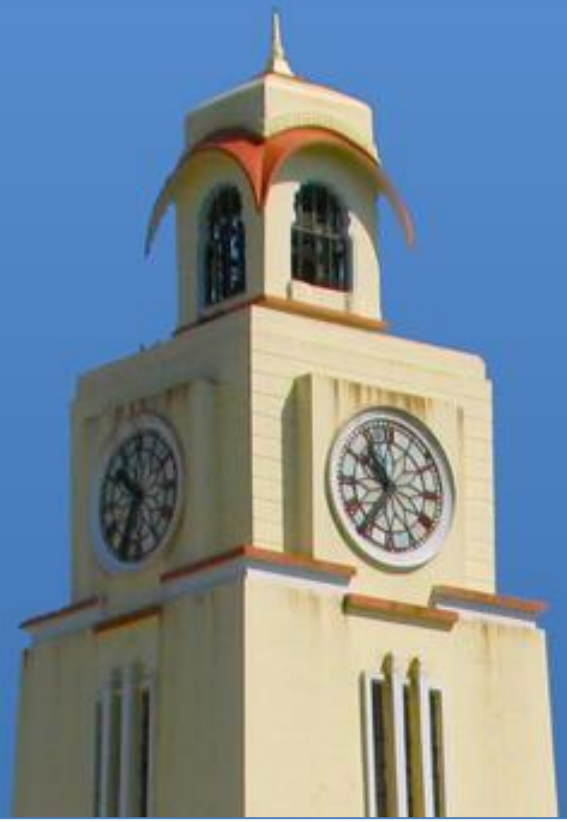
```
D:\2020-21_sem2\CN\Practical\Assignm
ent\Ass8>java -jar Client.jar localh
ost 8009
Hi from client 2
Server replied Hi from client 2
BYE
Server replied BYE
```

Practice Statement

- Develop a multi-threaded TCP server which perform following operations requested by clients and replying outcome of each operation to requested client,
  - Addition, Subtraction, Division and Multiplication.
  - Conversion of string send from client into capital letters.

Sources

- https://www.javatpoint.com/java-socket-setreuseaddress-method
- https://www.javatpoint.com/java-inetaddress-gethostaddress-method
- https://www.geeksforgeeks.org/introducing-threads-socket-programming-java/
- https://www.codejava.net/java-se/networking/javasocket-server-examples-tcp-ip
- https://www.javatpoint.com/socket-programming
- https://www.tutorialspoint.com/java/java_networking.htm#:~:text=Sockets%20provide%20the%20communication%20mechanism,its%20end%20of%20the%20communication.
- http://se.cs.depaul.edu/se450/lecture8-handout.pdf



innovate achieve lead

BITS Pilani
Dubai Campus

Thank You!