



Date: 23/08/2024

Lab Practical #07:

Study Client-Server Socket programming - TCP & UDP

Practical Assignment #07:

1. Write a C/Java code for TCP Server-Client Socket Programming.
2. Write a C/Java code for UDP Server-Client Socket Programming.

1. For TCP Server-Client:

TCP Server Program:

```
import java.net.*;
import java.io.*;
import java.util.*;
public class Server{
    public static void main(String[] args){
        try{
            System.out.println("Waiting....");
            Scanner sc = new Scanner(System.in);
            ServerSocket serverSocket = new ServerSocket(8080);
            Socket socket = serverSocket.accept();
            System.out.println("Client connected.");
            BufferedReader reader = new BufferedReader(new
                InputStreamReader(socket.getInputStream()));
            PrintWriter writer = new
                PrintWriter(socket.getOutputStream(), true);
            System.out.println("Client : "+reader.readLine());
            System.out.print("server : ");
            writer.println(sc.nextLine());
        }catch(Exception e){
            System.out.println("ERROR : "+e.getMessage());
        }
    }
}
```



Date: 23/08/2024

```
}  
}
```

TCP Client Program:

```
import java.io.*;  
import java.net.*;  
public class Client {  
    public static void main(String[] args) {  
        try{  
            Socket socket = new Socket("10.20.54.131", 8080);  
            System.out.println("Connected to server.");  
            BufferedReader reader = new BufferedReader(new  
InputStreamReader(socket.getInputStream()));  
            PrintWriter writer = new PrintWriter(socket.getOutputStream(), true);  
            BufferedReader consoleReader = new BufferedReader(new  
InputStreamReader(System.in));  
            System.out.print("Client : ");  
            writer.println(consoleReader.readLine());  
            System.out.println("Server : " + reader.readLine());  
        } catch (IOException e) {  
            System.out.println("Client exception: " + e.getMessage());  
        }  
    }  
}
```

Date: 23/08/2024

2. For UDP Server-Client:

UDP Server Program:

```
import java.net.*;

public class UDPServer {

    public static void main(String[] args) {

        try {

            DatagramSocket serverSocket = new DatagramSocket(8080);

            byte[] receiveBuffer = new byte[1024];

            System.out.println("Server is waiting for a client on port 8080...");

            DatagramPacket receivePacket = new DatagramPacket(receiveBuffer,
receiveBuffer.length);

            serverSocket.receive(receivePacket);

            String clientMessage = new String(receivePacket.getData(), 0,
receivePacket.getLength());

            System.out.println("Client: " + clientMessage);

            String serverMessage = "Hello from Server";

            byte[] sendBuffer = serverMessage.getBytes();

            InetAddress clientAddress = receivePacket.getAddress();

            int clientPort = receivePacket.getPort();

            DatagramPacket sendPacket = new DatagramPacket(sendBuffer,
sendBuffer.length, clientAddress, clientPort);

            serverSocket.send(sendPacket);

            serverSocket.close();

        } catch (Exception e) {

            System.out.println("Server error: " + e.getMessage());

        }

    }

}
```

Date: 23/08/2024

UDP Client Program:

```
import java.net.*;

public class UDPClient {

    public static void main(String[] args) {

        try {

            DatagramSocket clientSocket = new DatagramSocket();

            InetAddress serverAddress = InetAddress.getByName("10.20.54.131"); //
            Replace with server's IP address

            byte[] sendBuffer = new byte[1024];

            byte[] receiveBuffer = new byte[1024];

            String clientMessage = "Hello from Client";

            sendBuffer = clientMessage.getBytes();

            DatagramPacket sendPacket = new DatagramPacket(sendBuffer,
            sendBuffer.length, serverAddress, 8080);

            clientSocket.send(sendPacket);

            DatagramPacket receivePacket = new DatagramPacket(receiveBuffer,
            receiveBuffer.length);

            clientSocket.receive(receivePacket);

            String serverMessage = new String(receivePacket.getData(), 0,
            receivePacket.getLength());

            System.out.println("Server: " + serverMessage);

            clientSocket.close();

        } catch (Exception e) {

            System.out.println("Client error: " + e.getMessage());

        }

    }

}
```