**Lab Practical #06:**

Study Client-Server Socket programming - TCP & UDP

**Practical Assignment #06:**

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

**TCP Server Program:**

import java.io.\*;

import java.net.\*;

public class TCPServer {

public static void main(String[] args) throws IOException {

ServerSocket serverSocket = new ServerSocket(6789);

System.out.println("TCP Server started. Waiting for client...");

Socket clientSocket = serverSocket.accept();

System.out.println("Client connected: " + clientSocket.getInetAddress());

BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);

String inputLine;

while ((inputLine = in.readLine()) != null) {

System.out.println("Received from client: " + inputLine);

out.println("Server echoes: " + inputLine);

}

in.close();

out.close();

clientSocket.close();

serverSocket.close();

}}

**TCP Client Program:**

import java.io.\*;

import java.net.\*;

public class TCPClient {

public static void main(String[] args) throws IOException {

Socket clientSocket = new Socket("localhost", 6789);

BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));

PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);

BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in));

String userInput;

while ((userInput = stdIn.readLine()) != null) {

out.println(userInput);

System.out.println("Server response: " + in.readLine());

}

out.close();

in.close();

stdIn.close();

clientSocket.close();

}

}

1. **For UDP Server-Client:**

**UDP Server Program:**

import java.io.\*;

import java.net.\*;

public class UDPServer {

public static void main(String[] args) throws IOException {

DatagramSocket serverSocket = new DatagramSocket(9876);

byte[] receiveData = new byte[1024];

System.out.println("UDP Server started. Waiting for datagrams...");

while (true) {

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

serverSocket.receive(receivePacket);

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

System.out.println("Received from client: " + sentence);

InetAddress IPAddress = receivePacket.getAddress();

int port = receivePacket.getPort();

String capitalizedSentence = sentence.toUpperCase();

byte[] sendData = capitalizedSentence.getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, port);

serverSocket.send(sendPacket);

}

}

}

**UDP Client Program:**

import java.io.\*;

import java.net.\*;

public class UDPClient {

public static void main(String[] args) throws IOException {

BufferedReader inFromUser = new BufferedReader(new InputStreamReader(System.in));

DatagramSocket clientSocket = new DatagramSocket();

InetAddress IPAddress = InetAddress.getByName("localhost");

byte[] sendData;

byte[] receiveData = new byte[1024];

System.out.print("Enter message: ");

String sentence = inFromUser.readLine();

sendData = sentence.getBytes();

DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, 9876);

clientSocket.send(sendPacket);

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

clientSocket.receive(receivePacket);

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

System.out.println("FROM SERVER: " + modifiedSentence);

clientSocket.close();

}

}