Assignment 6

Title:- To develop client-server applications based on TCP/UDP java Sockets.

Code:

UDPServer.java:

package javaapplication7;

import java.net.\*;

import java.io.\*;

import java.util.\*;

/\*\*

\*

\* @author Lenovo

\*/

public class UDPServer {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

DatagramSocket aSocket = null;

try {

aSocket = new DatagramSocket(6678);

while (true) {

byte[] buffer = new byte[1000];

Scanner in = new Scanner(System.in);

DatagramPacket request = new DatagramPacket(buffer, buffer.length);

aSocket.receive(request);

System.out.println("Client Send : " + new String(request.getData()));

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

String msg = in.nextLine();

byte[] m = msg.getBytes();

DatagramPacket reply = new DatagramPacket(m, m.length, request.getAddress(), request.getPort());

aSocket.send(reply);

}

} catch (SocketException e) {

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

} catch (IOException e) {

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

} finally {

if (aSocket != null) {

aSocket.close();

}

//in.close();

}

}

}

UDPClient:

package javaapplication7;

import java.net.\*;

import java.io.\*;

import java.util.\*;

/\*\*

\*

\* @author Lenovo

\*/

public class UDPClient {

public static void main(String[] args) {

// args give message contents and server hostname

String m1;

DatagramSocket aSocket = null;

try {

do {

Scanner in = new Scanner(System.in);

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

String msg = in.nextLine();

byte[] m = msg.getBytes();

aSocket = new DatagramSocket();

InetAddress aHost = InetAddress.getByName("127.0.0.1");

int serverPort = 6678;

DatagramPacket request = new DatagramPacket(m, m.length, aHost, serverPort);

aSocket.send(request);

byte[] buffer = new byte[1000];

DatagramPacket reply = new DatagramPacket(buffer, buffer.length);

aSocket.receive(reply);

m1 = new String(reply.getData());

System.out.println("Server Send : " + new String(reply.getData()));

} while (true);

} catch (SocketException e) {

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

} catch (IOException e) {

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

} finally {

if (aSocket != null)

aSocket.close();

}

}

}

Output:



