User Datagram Protocol Example

A UDP Server:

```
//
// UDPlistener.java based on code in Dietel and Dietel
// Server reads a datagram packet from a client and responds
// with a datagram packet.
//
import java.io.*;
import java.net.*;

public class UDPlistener {
private DatagramSocket mySocket;
private DatagramPacket receivePacket;
private DatagramPacket sendPacket;
public UDPlistener() {
System.out.println(" \nListening very carefully!\n\n");
try {
mySocket = new DatagramSocket( 4567);
}
catch( SocketException sE) { // error so bad server dies!
System.err.println( "Could not open a datagram socket");
sE.printStackTrace();
System.exit( 1);
}
} // constructor
//
// Wait for packets from client, display them and respond.
//
public void waitForPackets() {
do { // wait for 'END' from client!!
// receive a packet, display it and echo
try {
// set up a packet ready to receive the msg
byte data[] = new byte[ 100];
receivePacket = new DatagramPacket( data, data.length);
// now wait for the packet
mySocket.receive( receivePacket);
displayPacket();
sendPacketToClient( "I see you sent: ");
}
catch( IOException ioE) {
System.err.println( "Some receive error!");
System.err.println( ioE.toString());
}
} while( !(receivePacket.toString()).equals( "END"));
} // waitForPacket()
```

```
//
// Display the received packet
//
public void displayPacket() {

System.out.println( "Packet received: ");
System.out.println( "\tFrom host: " + receivePacket.getAddress());
System.out.println( "\tHost port: " + receivePacket.getPort());
System.out.println( "\tLength: " + receivePacket.getLength());
System.out.println( "\tContaining: ");
System.out.println( "\t\t" + new String( receivePacket.getData(), 0,
receivePacket.getLength()));
} // displayPacket()
//
// echo packet
//
private void sendPacketToClient( String msg) throws IOException {
// construct the packet to send
String sMsg = msg + " ";
sMsg += new String( receivePacket.getData(), 0, receivePacket.getLength());
// create the packet to send
sendPacket = new DatagramPacket( sMsg.getBytes(), sMsg.length(),
receivePacket.getAddress(), receivePacket.getPort());
// send the packet
mySocket.send( sendPacket);
System.out.println( "Packet sent back");
} // sendPacketToClient()
//
// execute the application
//
public static void main( String args[]) {
UDPlistener app = new UDPlistener();
app.waitForPackets();
} // main
} // class UDPlistener
```

A UDP Client:

```
//
// UDPclient.java based on code in Dietel and Dietel
// Sends a datagram packet to a server and
// waits for a reply.
//
import java.io.*;
import java.net.*;

import java.util.Scanner;
public class UDPclient {
private DatagramSocket mySocket;
private DatagramPacket sendPacket;
private DatagramPacket receivePacket;
public UDPclient() { // constructor
try {
mySocket = new DatagramSocket();
```

```
}
catch ( SocketException sockE) {
System.err.println( "Problem opening datagram socket!");
sockE.printStackTrace();
System.exit( 1);
}
} // constructor
//
// Send packets, wait for responses from server and display them.
//
public void sendPackets() {
Scanner kbd = new Scanner( System.in);
String message = null;
byte rdata[] = new byte[ 200]; // to hold response
do{ // wait for 'END'
try {
// set up datagram packet.
System.out.println( "\nEnter a message to send (END to disconnect): ");
// next line of input from keybd.
message = kbd.nextLine(); // what happens if we use next()?
if( !message.equals( "END")) { // if more messages to send
// set up the packet
byte sdata[] = message.getBytes();
sendPacket = new DatagramPacket( sdata, sdata.length,
InetAddress.getLocalHost(), 4567);
mySocket.send( sendPacket);
// wait for response!
receivePacket = new DatagramPacket( rdata, rdata.length);
mySocket.receive( receivePacket);
displayPacket();
} // if msgs to send
}
catch( IOException ioE) {
System.err.println( "Receive error!");
ioE.printStackTrace();
}
} while( !message.equals( "END"));
System.out.println( "Client finished");
mySocket.close();
} // sendPackets()
//
// display the received packet
//
public void displayPacket() {
System.out.println( "Packet received: ");
System.out.println( "\tFrom host: " + receivePacket.getAddress());
System.out.println( "\tHost port: " + receivePacket.getPort());
System.out.println( "\tLength: " + receivePacket.getLength());
System.out.println( "\tContaining: ");
System.out.println( "\t\t" + new String( receivePacket.getData(), 0,
receivePacket.getLength()));
} // displayPacket()

//
// start the application!
//
public static void main( String args[]) {
```

```
UDPclient app = new UDPclient();
app.sendPackets();
} // main
} // class UDPclient
```