第五章：网络编程（下）

1.Socket 多客户端通信实现

1.客户端程序代码：

客户端代码与第四章中客户端代码相同，不再累述。

2.服务器端程序代码：

```
//MultiTalkeServer.java 文件
import java.io.*;
import java.net.*;

public class MultiTalkServer {
	static int clientnum = 0;//统计客户端数量

	public static void main(String[] args) throws IOException {
		// TODO Auto-generated method stub
		ServerSocket server = null;//声明一个 serversocket 对象
		boolean listening = true;
		try {
				server = new ServerSocket(4700);
			}catch(IOException e) {
				System.out.println("could not listen on port:4700" );
			}
			Socket socket = null;
			try {
				socket = server.accept();
			}catch(Exception e) {
				System.out.println("Error."+e);
				System.exit(-1);
			}
			while(listening)
			{
				new ServerThread(serverSocket.accept(),
clientnum).start();
				clientnum++;
			}
			serverSocket.close();
	}
}
//ServerThread.java 文件代码
import java.io.*;
import java.net.*;

public class ServerThread extends Thread {
        Socket socket = null;
        int clientnum;
        public ServerThread(Socket socket, int num){
                this.socket = socket;
                clientnum = num+1;
        }
        public void run(){
                try{
                        //下面开始通信。
                        String line;
                        BufferedReader is = new BufferedReader
(new InputStreamReader(socket.getInputStream()));
                        PrintWriter os =new PrintWriter(socket.getOutputStream());
                        BufferedReader sin = new BufferedReader
(new InputStreamReader(System.in));
                        System.out.println("Client:" + clientnum + ":" + is.readLine());
                        line =sin.readLine();
                        while(!line.equals("bye"))
                        {
                                os.println(line);
                                os.flush();

                                //下面是打印客户端和服务器端发送的信息
                                System.out.println("Server:"+ line);
                                System.out.println("Client:"+ clientnum
+":" + is.readLine());
                                line = sin.readLine();
                        }
                        os.close();
                        is.close();
                        socket.close();
                }catch(Exception e) {\
                        System.out.println("Error:"+e);
                }
        }

}
```

2.数据报通信

数据包通信采用的是 UDP(User Datagram Protocol)，它是非面向连接的，提供不可靠的

数据包式的数据传说的协议，类似于从邮局发送信件的过程。Java 中的 DatagramPacket/DatagramSocket/MulticastSocket 等累使用 UDP 协议进行网络通信。

TCP(Transport Control Protocol)是面向对接的能够提供可靠的流式数据传输的协议。Java 中的 ULR/URLConnection/socket/serverSocket 等类使用 TCP 协议进行网络通信。

1.TCP 和 UDP 的区别

（1）TCP 有建立时间，UDP 没有；

（2）UDP 传输限制包在 64k 以内；

（3）TCP 的应用有 Telnet、FTP 等，UDP 的应用有 ping 等。

2.Java 中进行数据报通信

所用的类有：

①DatagramSocket()//只允许数据报发往一个目的地址

②DatagramSocket(int port)

③DatagramPacket(byte ibuf[], int ilength)//接受

④DatagramPacket(byte ibuf[], int ilength， InetAdress iaddr, int iport)；//发送

收数据报：

```
DatagramPacket packet = new DatagramPacket(buf, 256);
socket.receive(packet);
```

发数据报：

```
DatagramPacket packet = new DatagramPacket(buf, buf.length,address,port);
socket.send(packet);
```

例：客户方程序 QuoteClient.java

代码：

```
import java.io.*;
import java.net.*;
import java.util.*;
public class QuoteClient {
	public static void main(String[] args) throws IOException{
		if(args.length!=1){//如果没有参数的执行内容
			System.out.println("Usage:java QuoteClient<hostname");
			return
		}
		DatagramPacket packet = new DatagramPacket();
		//send request
		byte[] buf = new byte[256];
		InetAddress address = InetAddress.getByName(args[0]);
		DatagramPacket packet = new
 DatagramPacket(buf, buf.length,address,4445);
		socket.send(packet);
		//get response
		  packet = new DatagramPacket(buf, buf.length);
		  socket.receive(packet);
		  //display response
		  String received = new String(packet.getData());
		  System.out.println("Quote of the Moment:" +received);
```

```
            socket.close();
        }
}
```

例：服务器方程序 QuoteServer.java

代码：

```
public class QuoteServer{
        public static void main(String[] args) throws java.io.IOException{
                new QuoteServerThread().start();
        }
}
```

例：服务器方程序 QuoteServerThread.java

代码：

```
import java.io.*;
import java.net.*;
import java.util.*;
public class QuoteServerThread extends Thread {
        protected DatagramSocket socket = null;
        protected BufferedReader in =null;
        protected boolean moreQuotes = true;
        public QuoteServerThread() throws IOException{
                this("QuoteServerThread");
        }
        public QuoteServerThread(String name) throws IOException{
                super(name);
                socket    = new DatagramSocket(4445);
                try{
                        in = new BufferedReader(new FileReader("one-liners.txt"));
                }catch(FileNotFoundException e){
                        System.err.println
("Counldnot open quote file.Serving time instead.");
                }
        }
        public void run(){
                while(moreQuotes){
                        try{
                                byte[] buf =new byte[256];
                                DatagramPacket packet =
new DatagramPacket(buf,buf.length);
                                socket.receive(packet);
                                String dString = null;
                                if(in==null)
                                        dString = new Date().toString();
                                else dString = getNextQuote();
                                buf =dString.getBytes();
```

```
                    //send the response to the client at "address"
                    //and "port"
                    InetAddress address = packet.getAddress();
                    int port = packet.getPort();
                    packet =
new DatagramPacket(buf,buf.length,address,port);
                    socket.send(packet);
                }catch(IOException e){
                    e.printStackTrace();
                    moreQuotes=false;
                }
            }
            socket.close();
        }
        protected String getNextQuote(){
            String returnValue = null;
            try{
                if((returnValue=in.readLine())==null){
                    in.close();
                    moreQuotes=false;
                    returnValue = "No more quotes.Goodbye.";
                }catch(IOException e){
                    returnValue="IOException occurred in server";
                }
                return returnValue;
            }
        }
}
```

3.使用数据报进行广播通信

MulticastSocket 讲数据报以广播方式发送到该端口的所有客户。

例：客户方程序 MulticastClient.java

代码：

```
import java.io.*;
import java.net.*;
import java.util.*;
public class MulticastClient {
        public static void main(String[] args) throws IOException{
            MulticastSocket socket = new MulticastSocket(4446);
            InetAddress address = InetAddress.getName("230.0.0.1");
            socket.joinGroup(address);
            DatagramPacket packet;
            //get a few quotes
            for(int i = 0;i<5;i++){
                  byte[] buf = new byte[256];
                  packet = new DatagramPacket(buf,buf.length);
                  socket.receive(packet);
                  String received = new String(packet.getData());
                  System.out.println
("Quote of the Moment:" +received);
            }
            socket.leaveGroup(address);
            socket.close();
      }
}
```

例：客户方程序 MulticastServer.java

代码：

```
public class MulticastServer{
      public static void main(String args[])
  throws java.io.IOException{
            new MulticastServerThread().start();
      }
}
```

例：服务器方程序 QuoteServerThread.java

代码：

```
import java.io.*;
import java.net.*;
import java.util.*;
public class MulticasterverThread extends Thread {
      private long FIVE_SECOND=5000;
      public MulticastServerThread() throws IOException{
            super("MulticastServerThread");
      }
      public void run(){
            while(moreQuotes){
                  try{
                        byte[] buf = new byte[256];
                        //construct quote
                        String dString = null;
                        if(in==null)
                              dString = new Date().toString();
                        else dString = getNextQuote();
                        buf =dString.getBytes();
                        //send it
                        InetAddress group =
InetAddress.getByName("230.0.0.1");
                        DatagramPacket pacet = new
DatagramPacket(buf,buf.length,group,4446);
                        socket.send(packet);
                        //sleep for a while
                        try{
                              sleep((long)(Math.random()*FIVE_SECOND));
                        }catch(InterruptedException e){}
                  }catch(IOException e){
                        e.printStackTrace();
                        moreQuotes=false;
                  }
            }
            socket.close();
      }
}
```