运城学院

Java语言程序设计
实验报告

实验四

实验题目：基于TCP协议的Socket编程

指导老师：贺玉珍

专业班级：计算机科学与技术系网络工程方向1002班

姓　　名：任华瑞（2010100241）

2013年 5月10日

实验类型 <u>验证性</u>　　　　　　　　　　实验室_<u>软件实验室三</u>

一、实验目的：

1. 掌握 Java 中的有关网络编程的相关类。
2. 练习使用这些类进行基于 TCP 协议的 Socket 网络编程。

二、实验内容：

分别编写一个客户端程序和一个服务端程序，两个程序能够实现通讯。

三、实验代码：

1.客户端界面代码**(ClientFrame.java)**：

```
    import java.awt.*;
    import java.net.*;
    import java.awt.event.*;
public class ClientFrame {
    private Frame f;
    private TextField tf;
    private Button but;
    private TextArea ta;

    ClientFrame() {
        init();

    }

    public TextArea getTextArea() {
        return this.ta;
    }

    public void init() {
        f = new Frame("客户端");
        f.setBounds(300, 100, 400, 300);
        f.setLayout(new FlowLayout());
        tf = new TextField(40);

        but = new Button("发送");

        ta = new TextArea(25, 50);
        f.add(tf);
        f.add(but);
        f.add(ta);
        myEvent();
        f.setVisible(true);
    }

    private void myEvent() {

        f.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                System.exit(0);
            }
        });

        but.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                //ta.setText(tf.getText());

                try {
                    DatagramSocket ds = new DatagramSocket();

                    byte[] buf = tf.getText().getBytes();
                    DatagramPacket dp = new DatagramPacket(buf, buf.length,
                            InetAddress.getByName("127.0.0.1"), 10001);
                    ds.send(dp);
                    ta.append("客户端说::"+tf.getText()+"\r\n");
                    tf.setText("");


                } catch (Exception ex) {
                    ex.toString();
                } finally {
                    // ds.close();
                }

            }
        });

    }

}
```

2.服务端界面代码**(ServerFrame.java)**:

```
    import java.net.DatagramPacket;
    import java.net.DatagramSocket;
    import java.net.SocketException;
public class ServerFrame {
    private Frame f;
    private TextField tf;
    private Button but;
private TextArea ta;
private DatagramSocket ds1;

ServerFrame() {
    init();
}

public TextArea getTextArea() {
    return this.ta;
}

public TextField getTextField() {
    return this.tf;
}

public void init() {
    f = new Frame("服务端");
    f.setBounds(300, 100, 400, 300);
    f.setLayout(new FlowLayout());
    tf = new TextField(40);

    but = new Button("发送");

    ta = new TextArea(25, 50);
    f.add(tf);
    f.add(but);
    f.add(ta);
    myEvent();
    f.setVisible(true);
}

private void myEvent() {

    f.addWindowListener(new WindowAdapter() {
        public void windowClosing(WindowEvent e) {
            System.exit(0);
        }
    });

    but.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            try {
                ds1 = new DatagramSocket();
```

```
                    byte[] buf1 = tf.getText().getBytes();
                    DatagramPacket dp1 = new DatagramPacket(buf1, buf1.length,
                            InetAddress.getByName("127.0.0.1"), 10002);
                    ds1.send(dp1);
                    ta.append("服务端说::"+tf.getText()+"\r\n");
                    tf.setText("");

                } catch (Exception ex) {
                    ex.toString();
                }

            }
        });
    }
}
```

3. 服务端代码（**UdpRece2.java**）：

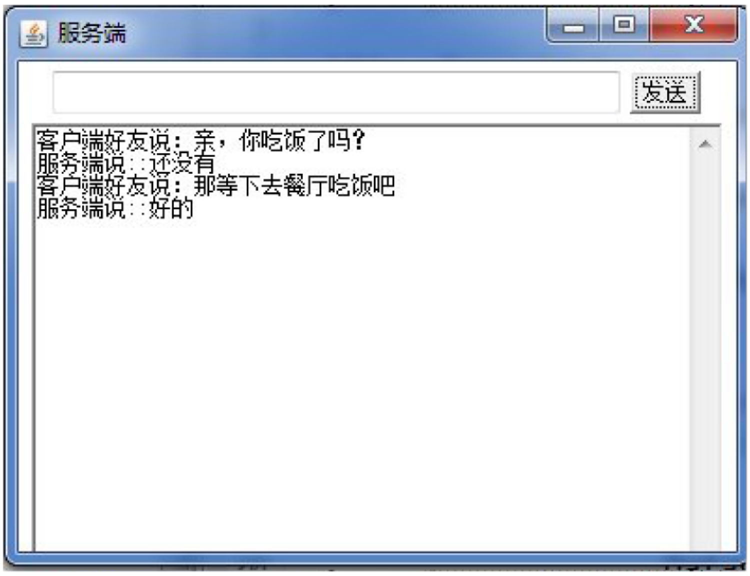
```
import java.net.*;
public class UdpRece2 {
    public static void main(String[] args) throws Exception {
        final ServerFrame frame = new ServerFrame();
        new Thread(new Runnable() {

            @Override
            public void run() {
                // TODO Auto-generated method stub
                try {
                    DatagramSocket ds = new DatagramSocket(10001);
                    while (true) {

                        byte[] buf = new byte[1024];
                        DatagramPacket dp = new DatagramPacket(buf,
buf.length);

                        ds.receive(dp);

                        String ip = dp.getAddress().getHostAddress();
                        String data = new String(dp.getData(), 0, dp
                                .getLength());
                        frame.getTextArea().append("客户端好友说：
"+data+"\r\n");
                    }
                } catch (Exception e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }

            }
        }).start();

    }
}
```

4.客户端代码(**UdpSend2.java):**

```
import java.net.*;
import java.net.SocketException;public class UdpSend2 {
    public static void main(String[] args) throws Exception {

        final ClientFrame frame = new ClientFrame();

        new Thread(new Runnable() {

            @Override
            public void run() {
                // TODO Auto-generated method stub

                try {
                    DatagramSocket ds1 = new DatagramSocket(10002);
                    while (true) {

                        byte[] buf1 = new byte[1024];
                        DatagramPacket dp1 = new DatagramPacket(buf1,
                                buf1.length);

                        ds1.receive(dp1);

                        String ip = dp1.getAddress().getHostAddress();
                        String data = new String(dp1.getData(), 0,
                                dp1.getLength());
                        frame.getTextArea().append("服务端好友说：
"+data+"\r\n");
                    }
                } catch (Exception e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }

            }
        }).start();
    }
}
```

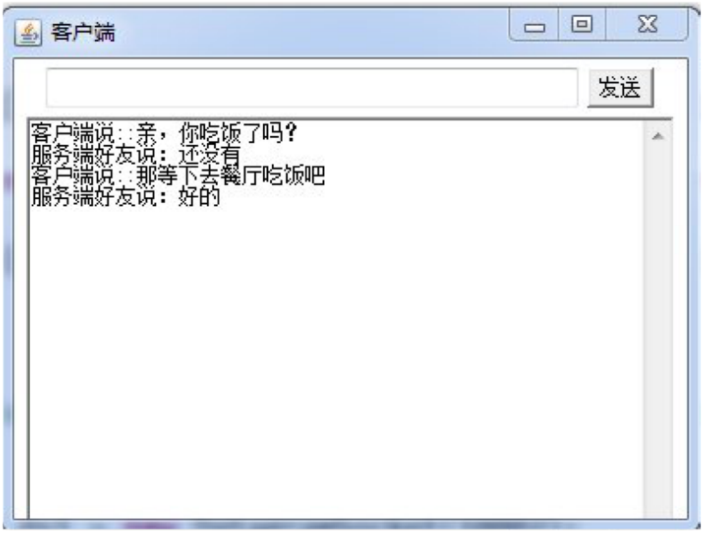
四、运行结果：

1、运行 **UdSend2.java**，出现的结果为：



```
客户端好友说：亲，你吃饭了吗？
服务端说::还没有
客户端好友说：那等下去餐厅吃饭吧
服务端说::好的
```

2、运行 **UdpRece2.java**，出现的结果为



```
客户端说::亲，你吃饭了吗？
服务端好友说：还没有
客户端说::那等下去餐厅吃饭吧
服务端好友说：好的
```

五、实验总结：

在这次实验中，我对 Socket 编程有了进一步的了解和掌握了。但是在实验中，我发现自己在有些地方还是不太熟悉，希望自己在平时多加的练习和掌握。