



南昌大学实验报告

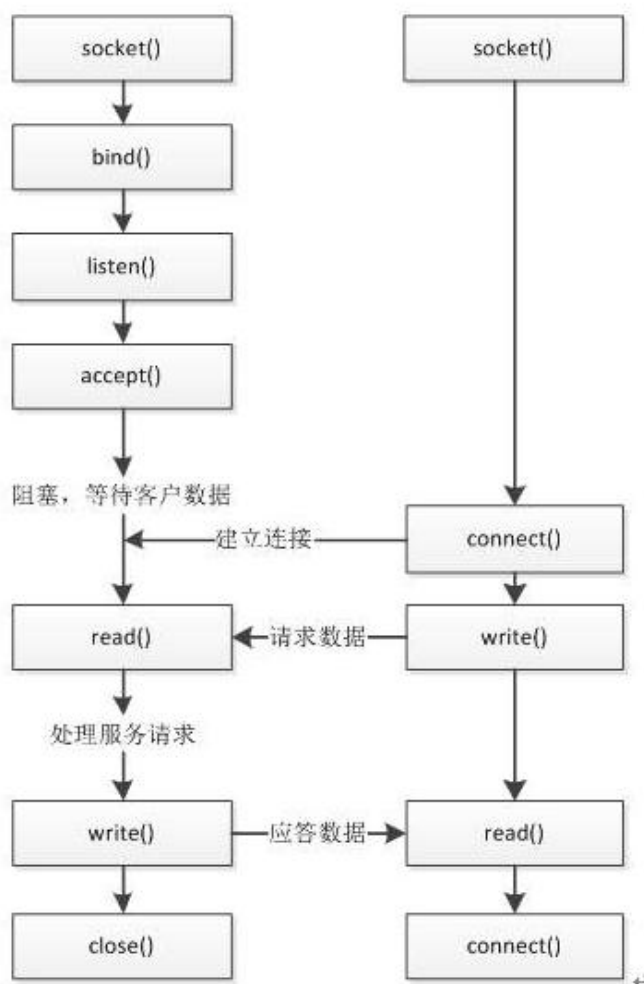
一、实验项目名称

掌握 socket 编程

二、实验目的

- 1) 掌握 TCP 的工作原理
- 2) 理解 socket 的概念和应用场合

三、实验基本原理



TCP socket 流程图

实验要求：

- 1、 实现一个发送数据，和接收数据的程序。
- 2、 语言为 c++或 java
- 3、 最好是图形界面

四、主要仪器设备及耗材

IntelliJ IDEA2020,JDK1.8

五、实验步骤 （源代码）

TcpClient.java

```
import javax.swing.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.BufferedWriter;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.net.Socket;

public class TcpClient extends JFrame implements ActionListener{

    JPanel panel; // 主面板

    JScrollPane p1; // 滚动条

    JTextArea sendmessage; // 发送信息框

    JButton send; // 发送按钮

    Socket socket; // 定义套接字对象

    String message; //
    BufferedWriter bufferedWriter;
```

```

public TcpClient() throws IOException{

    this.setTitle("客户端-client");

    this.setBounds(200,200,500,300);
    socket = new Socket("127.0.0.1",8080);

    bufferedWriter =new BufferedWriter(new
OutputStreamWriter(socket.getOutputStream()));

    // 发送信息框

    p1 = new JScrollPane();

    p1.setBounds(0,0,300,80);

    sendmessage = new JTextArea();

    sendmessage.setBounds(0,0,320,80);

    panel = new JPanel(null);

    p1.setViewPortView(sendmessage);

    panel.add(p1);

    send = new JButton("发送信息");

    send.setBounds(300,90,100,30);
    send.addActionListener(this);

    panel.add(send);
    this.setContentPane(panel);
    this.setDefaultCloseOperation(EXIT_ON_CLOSE);
    this.setVisible(true);
}
public void actionPerformed(ActionEvent e){
    if(e.getSource() == send){

        message = "客户端: ";

        message += sendmessage.getText(); // 把发送框的内容添加到
信息输入流里
        try{

            bufferedWriter.write(message); // 写进文件流中

            bufferedWriter.write("\n");
            bufferedWriter.flush();
        }catch (IOException ioException){

```

```

        IOException.printStackTrace();
    }finally {
        sendMessage.setText(null);
    }
}
}
public static void main(String[] args) throws IOException {
    TcpClient client = new TcpClient();
}
}

```

TcpServer.java

```

import javax.swing.*;
import java.awt.*;
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.ServerSocket;
import java.net.Socket;

public class TcpServer extends JFrame{
    JPanel jp;
    JScrollPane js;
    JTextArea chat;
    BufferedReader bufferedReader;

    String message; // 从缓冲区读取信息

    Socket socket; // 建立套接字对象

    ServerSocket serverSocket; // 服务器建立通信对象 ServerSocket

    public TcpServer() throws IOException {
        jp = new JPanel(null);
        js = new JScrollPane();
        js.setBounds(15,15,400,550);

        chat = new JTextArea("信息记录");
        chat.setBounds(0,0,400,500);
        chat.setFont(new Font("黑体",Font.PLAIN,18));
        chat.setEnabled(false);
    }
}

```

```

js.setViewportView(chat);
jp.add(js);

this.setTitle("服务器端");

this.setContentPane(jp);
this.setSize(500, 650);
this.setLocationRelativeTo(null);
this.setDefaultCloseOperation(EXIT_ON_CLOSE);
this.setVisible(true);
serverSocket = new ServerSocket(8080);

socket = serverSocket.accept(); // 等待客户端连接

bufferedReader = new BufferedReader(new
InputStreamReader(socket.getInputStream()));
while((message = bufferedReader.readLine()) != null) {
    // 从字符输入流中读取文本。如果不空就把数据追加到信息 message
    中, 然后显示在 chat 信息框中

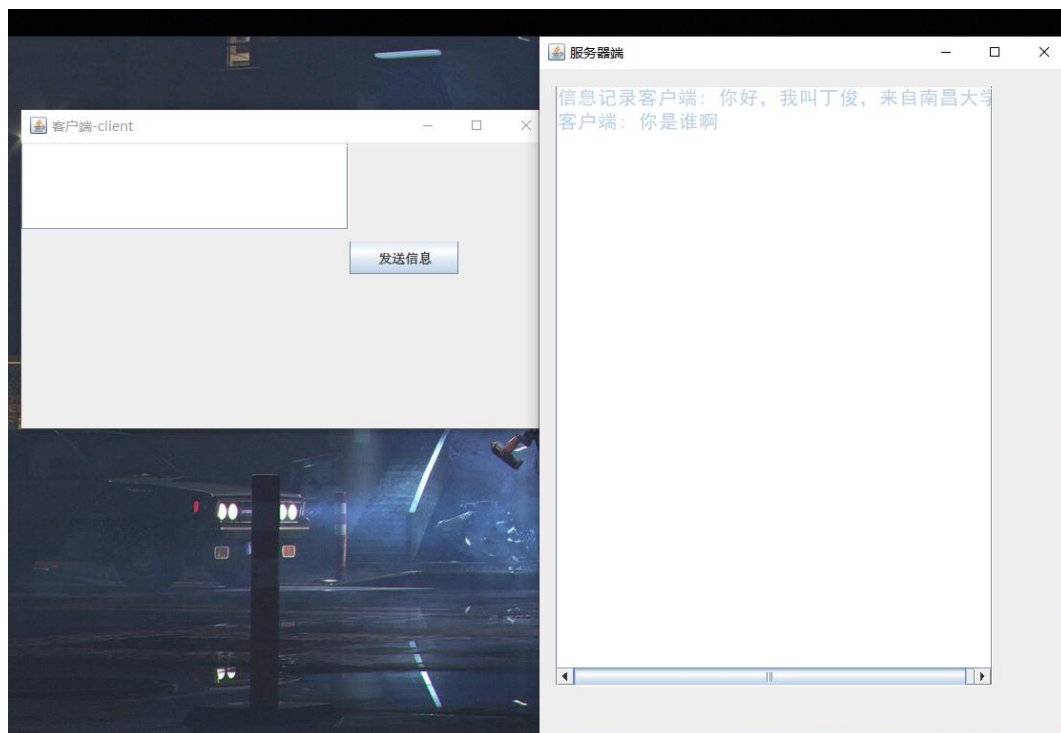
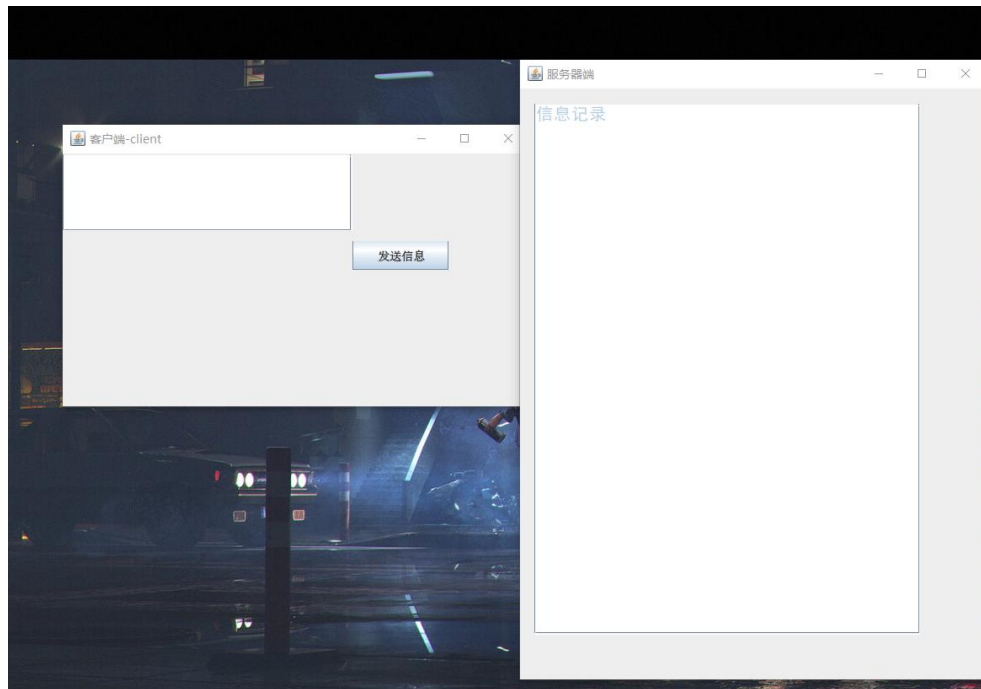
    chat.append(message+"\n"); // 向聊天框中追加信息内容
}
}

public static void main(String[] args) throws IOException {
    TcpServer server = new TcpServer();
}
}

```

部分解释：TcpClient 类向缓冲流中写入数据，给发送按钮添加监听，将发送框中的内容写进流中；TcpServer 类则从输入流中读取文本并缓冲字符，添加到聊天框中。

运行结果：



七、思考讨论题或体会或对改进实验的建议

该实验还可增加双方的通信功能，并实现通信连接的开关控制，并能实现全双工通信。

八、参考资料

- [1] Java2 实用教程(第五版)，耿祥义.
- [2] 计算机网络第五版，谢希仁.