

**计算机网络实验报告**

实 验 题 目 Socket编程实现客户端与服务器通信

姓 名 俞 家 宝

专 业 软件工程

班 级 2021级软件工程2班

学 号 2021117338

西北大学信息学院

**实验18 Socket编程实现客户端与服务器通信**

1. **实验目的**
2. 学习Socket编程基础知识；
3. 理解并学会应用socket API，比如，学习基于C的Socket编程相关函数和数据类型；（语言不限，代码可以是任意语言）
4. 熟练掌握TCP的Client/Server模式的通信原理。
5. 学会建立主机与服务器之间的连接；
6. 学会在主机和服务器之间收发消息；

**二、实验环境**

Windows Java

**三、实验流程**

1、分别模仿客户端与服务器端；

2、使用socket建立连接；

3、在服务器与客户端进行信息交流；

1. 以流程图形式展现程序运行过程；

**四、实验结果展示**

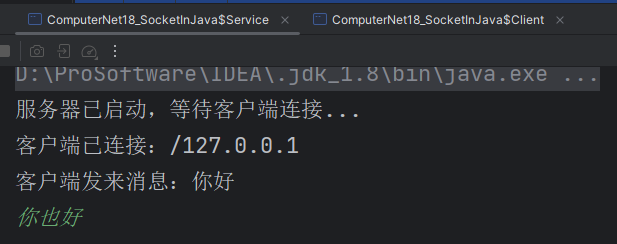
1、代码以及通信过程截图如下（使用Java实现）

*/\*\*  
 \* @author 俞家宝  
 \* @version 1.0  
 \* @className ComputerNet18\_SocketInJava  
 \* @dataTime 2023/10/23 15:46  
 \*/*import java.io.\*;  
import java.net.\*;  
public class ComputerNet18\_SocketInJava {  
*/\*\*  
\* TCP：服务端  
\*/* static class Service {  
 public static void main(String[] args) {  
 try {  
 ServerSocket serverSocket = new ServerSocket(8080); // 创建服务器Socket，监听端口8080  
 System.*out*.println("服务器已启动，等待客户端连接...");  
 while (true) {  
 Socket clientSocket = serverSocket.accept(); // 等待客户端连接  
 System.*out*.println("客户端已连接：" + clientSocket.getInetAddress());  
 // 从客户端获取输入流  
 BufferedReader in = new BufferedReader(new InputStreamReader(clientSocket.getInputStream()));  
 // 获取客户端的输出流  
 PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);  
 // 从键盘获取输入  
 BufferedReader consoleInput = new BufferedReader(new InputStreamReader(System.*in*));  
 String message, toClient;  
 while ((message = in.readLine()) != null) {  
 System.*out*.println("客户端发来消息：" + message);  
 while ((toClient = consoleInput.readLine()) != null) {  
 out.println("服务器：" + toClient); // 将消息原样返回给客户端  
 }  
 }  
 clientSocket.close();  
 }  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
*/\*\*  
 \* TCP：客户端  
\*/* static class Client {  
 public static void main(String[] args) {  
 try {  
 Socket clientSocket = new Socket("localhost", 8080); // 连接到服务器，指定服务器的主机名和端口号  
 System.*out*.println("已连接到服务器");  
 // 获取客户端的输出流  
 PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);  
 // 从键盘获取输入  
 BufferedReader consoleInput = new BufferedReader(new InputStreamReader(System.*in*));  
 String message;  
 while ((message = consoleInput.readLine()) != null) {  
 out.println(message); // 发送消息给服务器  
 System.*out*.println(new BufferedReader(new InputStreamReader(clientSocket.getInputStream())).readLine());  
 }  
 clientSocket.close();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}



运行结果：

服务器端：



客户端：

文本

描述已自动生成

2、程序流程图如下

采用TCP，其流程图示例

