课程设计报告二：聊天室系统

1. 课程设计要求与目的

目的：编写一个小型Java聊天室系统，掌握Java网络通信、多线程、IO文件操作等高级应用编程技能。

要求：以课本第15章 Java网络通信例15.3、15.4的源代码为基础，编写一个小型Java聊天室系统。

完成如下功能：

1、多客户端模式下，实现客户与客户的单独通信，要求信息通过服务器中转

2、端到端的通信，实现并行通信模式（不再是你说一句，我说一句，一端的信息发送不受另一端的影响）

3、实现端到端的文件传输

4、添加图形界面（选做）

1. 系统设计

服务器端：分为两个class文件，一个是MultiTalkServer类，还有一个是ServerThread类。

MultiTalkServer实现的功能有：监听客户请求，根据得到的Socket对象和客户计数创建服务线程，并启动。将serverSocket.accept()返回的socket作为参数，传入到ServerThread中。

ServerThread实现的功能有：判断是否下线，判断IP地址是否正确，判断目标客户是否在线，判断是否传输文件，获取目标的socket套接字。

1. 系统实现

服务器端：



MultiTalkServer：

public class MultiTalkServer{

静态成员变量，记录当前客户的个数

**public** **static** **void** main(String args[]) {

初始化

新建服务器套接字

**while**(listening){//监听

监听到客户请求，根据得到的Socket对象和客户计数创建服务线程，并启动它

}

ServerThread：

public class ServerThread exte**nds** Thread{

数据成员： 保存与本线程相关的Socket对象/保存本进程的客户计数

新建哈希表，用于保存用户的Ip与socket的键值对

String IP用于保存IP地址的模板

构造方法

初始化socket变量

初始化clientnum变量

将IP和socket放入哈希表中

splitMessage判断用户是否退出

当发送空消息或发送“quit”时则退出

用于发送文件 SendDocument

用于创建输入输出流并收发文件

线程主体run()

//获取哈希表的大小并输出;

//创建输入输出流

**while**(**true**) {//监听

①判断是否要下线

如果下线则发送quit

并将下线用户套接字移除

②判断输入的IP是否为正确的IP地址

如果IP地址不能和模板相匹配，则输出Wrong IP

③判断目标客户是否在线

如果不在线则输出：The Client you call is not online!

④判断是否传输文件

获取目标的Socket套接字

调用模块进行文件发送

关闭Socket输出流

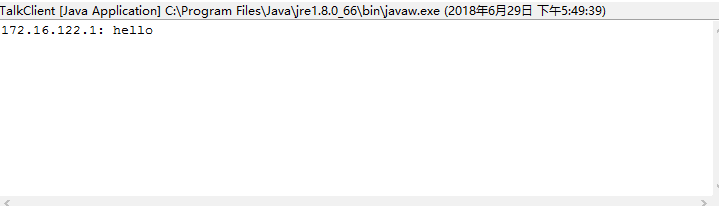
关闭Socket输入流

关闭Socket

}

1. 系统测试
2. 收发消息基本功能测试

在用户输入对象的IP 发送的消息后，对方可以收到



1. 收发文件进本功能测试

在用户输入对象的IP +消息+文件路径后，对方可以收到

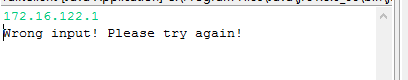
Y%~905U%6_U3N`Z8RYB5W)H

35)[F_SZ_(0~Z`$}N0SNG$3

1. 异常测试

（1）发消息

①格式错误



②格式正确

A、IP地址错误



B、目标客户不在线



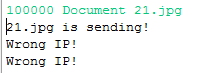
2、发文件

①格式错误

会当成消息发送出去

②格式正确

A、IP地址错误



B、目标客户不在线



C、找不到文件



D、文件重名（文件覆盖）

会成功发送，但是会将源文件覆盖。

1. 课程设计总结

在本次课程设计中，学会了Java网络编程。实现了端到端通信，实现并行通信模式（不再是你说一句，我说一句，一端的信息发送不受另一端的影响），实现端到端的文件传输。学会了收发消息、收发文件的相关操作，输入输出流的包装，知道了做项目的基本流程。理解了Java要按接口编程的核心思想。