

福州大学

Java 程序设计实验报告

题 目_____局域网聊天程序_____

姓 名_____吴建东_____

学 号_____221900128_____

学 院_____软件学院_____

专 业_____软件工程_____

年 级_____2019 级_____

指导教师_____罗欢_____

2021 年 5 月 24 日

一、实验目的

- 1、熟练掌握基本网络编程技术。
- 2、掌握 Swing 图形用户界面编程。
- 3、掌握多线程编程的基本原理，能使用多线程设计服务器端程序。
- 4、培养独立查找资料，并解决问题的能力。

二、实验任务

设计并编程实现一个以 Socket 通信机制为基础的网络聊天程序。

服务器端:

多线程方式处理客户端，负责响应客户端的信息：用户登录、客户端请求处理。

客户端:

- (1)、负责用户界面初始化、为用户提供登录界面、将信息发送至服务器、接受服务器反馈，实时显示信息。
- (2)、负责实现用户界面的各种组件事件的处理。
- (3)、当用户单击好友图标时，将初始化新的聊天窗口，并通过 UDP 建立与其他用户的通信。当用户收到其他用户发来的数据报，将在本地初始化新的对话窗口，并开始通信。

三、开发工具与平台

开发工具：Eclipse

开发平台：JDK 1.8.0_131

四、设计思路

1、界面设计

根据实验要求分析，共需要三个窗口：

- (1) 客户端登录界面
- (2) 客户端用户中心(在线好友列表)窗口
- (3) 客户端聊天窗口(一对一)

由于服务器仅仅只是个程序，所以不需要构建窗口。

对于客户端登录界面，总共安排了六个组件：

- (1) 用户名标签文本框 (2) 密码标签文本框 (3) 用户名输入文本框
- (4) 密码输入文本框 (5) 退出按钮 (6) 登录按钮。

对于按下登录按钮时有以下几种情况：

- (1) 用户名输入框和密码输入框其中一个为空，此时将弹出窗口“用户名或密码不能为空!!”。
- (2) 当服务器未开启时，将弹出窗口“服务器未开启!”。
- (3) 当输入的用户名对应的用户已在线时，将弹出窗口“该用户已在线!”。

对于客户端在线好友列表界面，总共安排了两个组件：

- (1) 用户名标签文本框 (2) 在线用户列表 JTable

用户名标签文本框显示当前用户的用户名。

在线用户列表用于显示在线用户的用户名，点击某一个用户时，能够弹出对应的聊天窗口。

对于客户端聊天窗口，总共安排了四个组件：

- (1) 聊天内容显示文本框 (2) 聊天内容输入文本框
- (2) 清屏按钮 (4) 发送按钮。

聊天内容显示文本框用来显示双方所发送的聊天内容。

聊天内容输入文本框用于聊天内容的输入。

清屏按钮用于清空聊天内容显示文本框。

发送按钮用于发送消息（按下回车键也可发送消息）。

当输入文本框为空时，按下发送按钮，将会提示“发送内容不能为空!!”。

2、逻辑设计（课题中的难点的解决方案）

实验过程中遇到的三个难点是：

- （1）客户端如何实时更新在线用户列表。
- （2）UDP 聊天窗口如何获取好友的 IP 和端口。
- （3）收到其他用户发来的数据报如何开启对应的窗口。

客户端如何实时更新在线用户列表：

在线用户列表在两种情况下会发生变化：一是当用户通过登录窗口登录时，在线用户数增加。二是当用户点击右上角×号，退出程序时，在线用户数减少。

解决方案：在服务器端建立一个存放所有在线用户名的静态 ArrayList、建立一个用户名与 Socket 的 HashMap。在每个用户登录成功时，服务器将该用户信息记录到 ArrayList 和 HashMap 中，客户端开启一个新线程 ClientThread，这个线程不断接收由服务器传来的在线用户列表 ArrayList，同时，服务器通过 HashMap 中每一个用户的 Socket，将在线用户列表 ArrayList 以对象流的形式发送给每一个用户的客户端。当客户端接收到列表信息之后，更新在线列表 JTable 的信息。

当某个用户退出程序时，服务器程序会抛出 SocketException 异常，当捕捉到此异常时，服务器将该用户信息从 ArrayList 和 HashMap 中删除，同时将更新后的 ArrayList 发送给每一个用户。

UDP 聊天窗口如何获取好友的 IP 和端口：

解决方案：在服务器与客户端建立用户名与 InetSocketAddress 的 HashMap，在服务器更新在线列表的同时，用户名与 InetSocketAddress 的 HashMap 也同步更新，并与 ArrayList 一起发送给所有客户端，当客户端接收到服务器发来的 Map 时，同步更新自身的 Map。当点击用户名，打开聊天窗口时，通过用户名，在 Map 中找到对应的好友 IP 地址，与聊天窗口进行绑定。每次发送消息时，就向这个 IP 发送数据报。

收到其他用户发来的数据报如何开启对应的窗口：

解决方案：在每个用户登录成功时，客户端开启一个新线程 MessageThread，这个线程不断接收由其他用户发来的数据报，发送的数据报包含两个实体，一个是用户信息实体，一个是聊天信息实体。当客户端接收到数据报，如果聊天窗口未打开，则开启新的聊天窗口，通过用户信息实体得到好友的用户名，在客户端 Map 中得到好友的 IP 地址，将好友 IP 地址与聊天窗口进行绑定。如果聊天窗口已经打开，仅需要将聊天内容显示在窗口上即可。

五、实验总结

1、登录界面



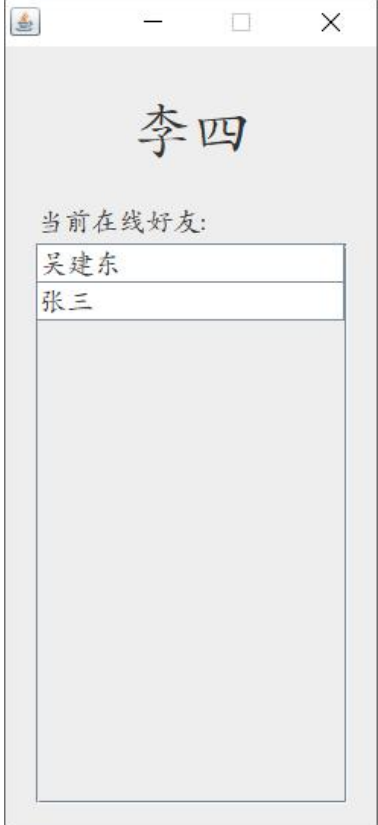
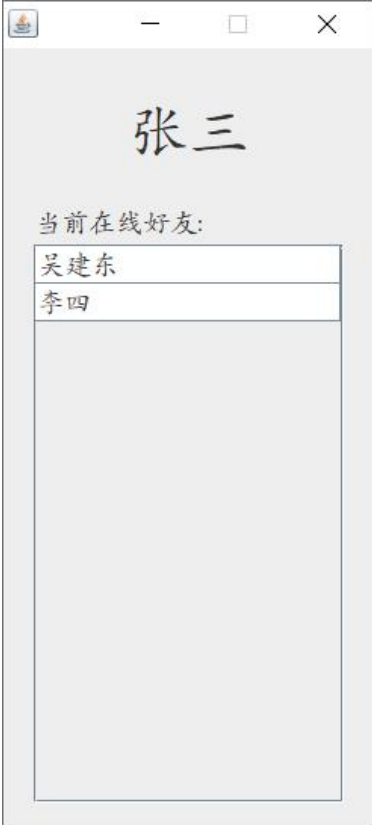
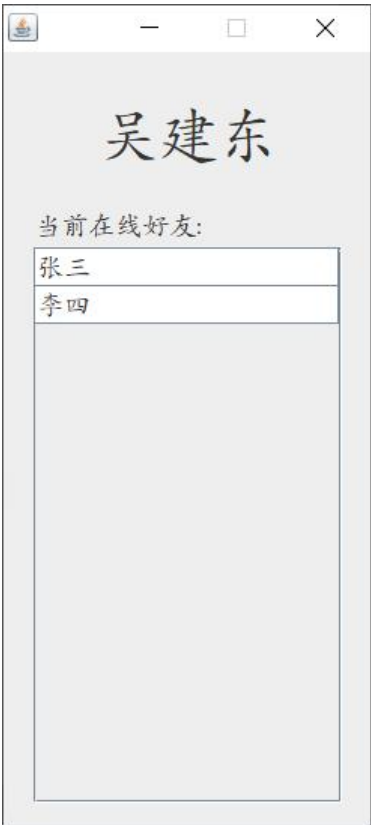
异常处理:



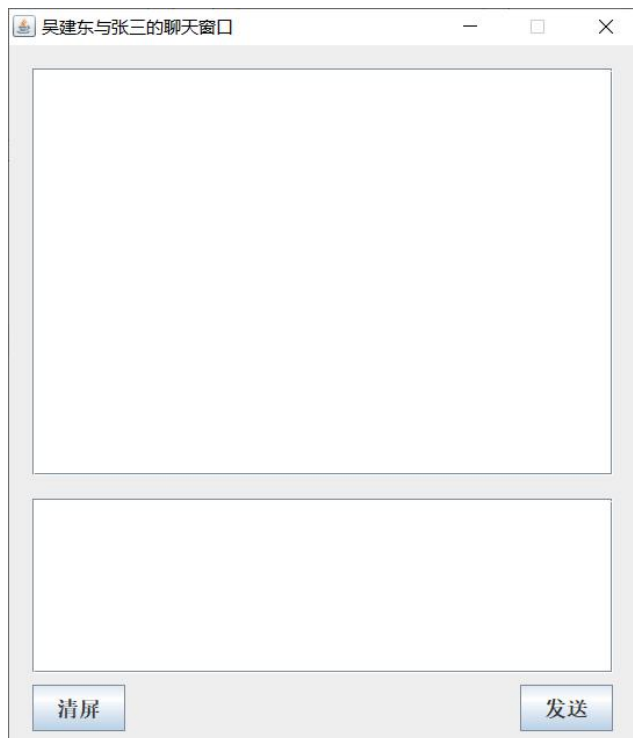
2、用户界面



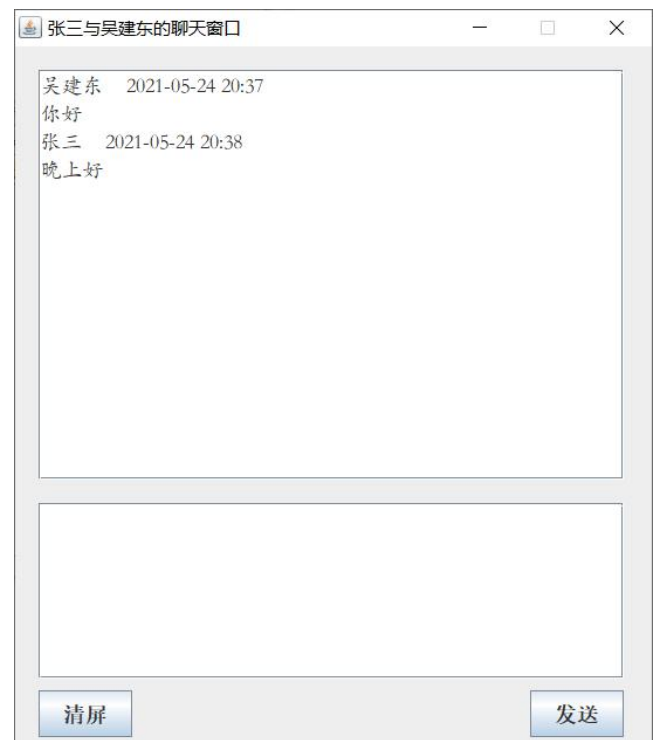
有多个用户登录时：



2、聊天窗口 （用户）与（好友）的聊天窗口

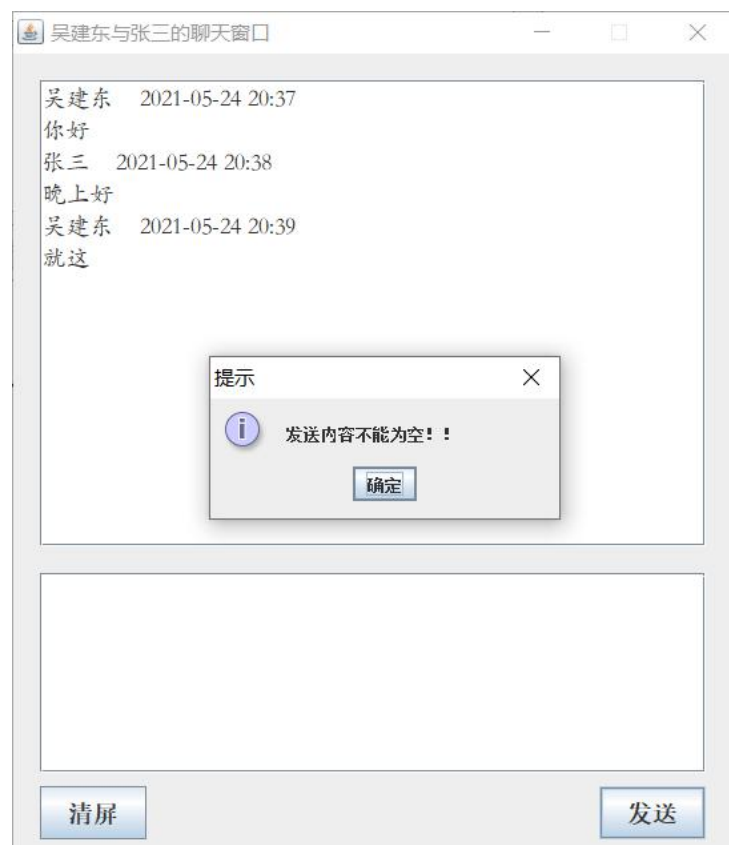


聊天过程:





当输入为空的消息提示:



总结：

通过本次实验，能够将网络编程，多线程，GUI 等相关内容得到实际应用。

整个实验总体耗时 7 天，主要时间花在查找相关资料上，对于多线程和 UI 设计部分难度不是太大，耗时也相对较短。最难的主要是在网络编程部分，首先是在建立 TCP 连接的过程中踩了许多坑，由于对 Socket 的使用和流的应用不熟悉，导致在编程过程中非常痛苦，报了错，却无从下手解决，只能通过百度来解决，又因为网络信息鱼龙混杂，难以得到真正的错误原因，最终了解了相关的资料后进行调试，才将 BUG 一一解决。本次实验从另一方面来说也是锻炼我个人的一个心态，加强了查找资料的能力。