聊天室实验报告

小组成员：黄建武、骆铭涛、蔡桐钊、陈海涛
陈平永、何晋豪、洪创煌、童云钊

1. 实验简介

本次实验要求实现一个在线聊天室，实现在线用户之间信息交换的功能。需满足以下几项基本要求：

- 构建一个**服务器 S** 和若干个**客户机 C**；
- 每一台客户机 C 都可以与服务器 S 和其他客户机连接；
- 每一台客户机 C 都可以通过 S 获得其他客户机发来的信息，并显示哪一台客户机在线，使用的是什么 IP 等；
- S 可以支持**群聊功能**。每一台客户机 C 可以通过服务器 S 的群聊天室与其他客户机通信，即既可**接收**信息，也可**发送**信息；
- **P2P 聊天**：客户机 C 通过选择在线的用户并进行连接，通过 P2P 的模式收发信息。
- **拓展功能：文件传输**

2. 实验设计

2.1 服务器设计思路

一共有两台服务器，一台聊天服务器 ChatServer 和一台文件服务器 FileServer。

ChatServer

ChatServer 主要负责转发聊天内容和用户信息给所有用户，用于支持跟进用户信息功能和群聊功能。用户 A 连接服务器的时候，首先向服务器发送一个用户名，服务器收到之后搭建 socket 连接，然后记录用户 A 的 IP 和端口号，接着服务器将所有当前在线的用户的名字、IP 和端口信息发送给这位刚上线的用户 A，

最后服务器向其他在线的用户群发用户 A 的名字、IP 和端口号以提示新用户上线，这里 IP 和端口号的作用是方便客户端进行私聊。用户 A 要下线的时候，发送一个命令"[OFFLINE]"给服务器提示下线，服务器清除相关资源并关闭连接，同时将用户 A 要下线的消息转发给其他在线用户。在使用群聊功能时，某个客户端发起群聊消息给服务器，服务器会把该消息转发给其他在线用户，从而实现群聊的功能。私聊时，客户端向服务器发送私聊消息，由服务器转发给指定的接收用户。

下面是服务器与客户端的交互的过程以及发送数据的格式：

```
接下来我会用到的符号短语以及其表示的意义:
C: 客户端, C0, C1, ...,, Cn等后面加数字的C用于区别不同的客户端
S: 服务器
usr: 用户名
ip: IP地址
port: 端口号
msg: 消息
dest: 私聊消息的接收者
INFO: 当前在线用户信息标识符
ONLINE: 新用户上线标识符
OFFLINE: 用户下线标识符
GROUP: 群聊信息标识符
P2P: 私聊信息标识符
[OFFLINE]: 下线命令
[#]: 字段分隔符, 代码中用StringTokenizer切割之后即可获取相应信息

1. 用户上线, 建立连接
C0 ------------- usr ----------------> S              (客户端向服务器发送用户名建立连接)
S  --- INFO[#]usr[#]ip[#]port ------> C0             (服务器将在线用户的信息发给新上线用户C0, 可能有多条INFO)
S  --- ONLINE[#]usr[#]ip[#]port ----> C1, C2, ..., Cn (服务器将新用户上线的信息群发给其他在线用户)

2. 群聊
C0 ------- GROUP[#]msg -------------> S              (用户发送群聊消息到服务器)
S  --- GROUP[#]usr[#]msg -----------> C1, C2, ..., Cn (服务器将消息转发给其他在线用户, usr用于标记是哪个用户发出的消

3. 私聊
C0 --- P2P[#]msg[#]dest ----------> S  (用户发送群聊消息到服务器)
S  --- P2P[#]usr[#]msg -----------> Cj (服务器将消息转发给其他在线用户, usr用于标记是哪个用户发出的消息)

4. 下线, 断开连接
C0 ---------- [OFFLINE] ------------> S              (用户发送下线指令到服务器进行下线操作)
S  --- OFFLINE[#]usr[#]ip[#]port ---> C1, C2, ..., Cn (将下线信息转发给其他在线用户)
```

FileServer

FileServer 主要为整个应用提供传输文件服务器，包括群发文件和私发文件。客户端想要发送文件的时候，将文件发送给服务器并附加相关信息(群发/私发)，服务器根据客户端的需求将文件转发给其他在线用户。下面是文件服务器和客户端的交互过程：

```
接下来我会用到的符号短语以及其表示的意义：
C：客户端，C0，C1，...，Cn等后面加数字的C用于区别不同的客户端
FS：文件服务器
usr：用户名
dest：私发文件的接收者
fname：文件名
flength：文件长度
file：文件内容
GROUP：群发文件标识符
P2P：私发文件标识符
[OFFLINE]：下线命令
[#]：字段分隔符，代码中用StringTokenizer切割之后即可获取相应信息
```

```
1. 客户端与文件服务器建立连接
C0 --- usr ---> FS

2. 群发文件
C0 --- GROUP[#]fname ----------> FS              (客户端发送需求信息给服务器)
C0 --- flength ----------------> FS              (客户端发送文件长度给服务器)
C0 --- file -------------------> FS              (发送文件内容给服务器)
FS --- GROUP[#]fname[#]usr ---> C1, C2, ..., Cn (服务器转发基本信息给其他用户，usr标识发送者)
FS --- flength ----------------> C1, C2, ..., Cn (服务器发送文件长度给其他用户)
FS --- file -------------------> C1, C2, ..., Cn (服务器发送文件内容给其他用户)

3. 私发文件
C0 --- P2P[#]fname[#]dest -----> FS (客户端发送需求信息给服务器)
C0 --- flength ----------------> FS (客户端发送文件长度给服务器)
C0 --- file -------------------> FS (发送文件内容给服务器)
FS --- P2P[#]fname[#]usr ------> Cj (服务器转发基本信息给其他用户，usr标识发送者)
FS --- flength ----------------> Cj (服务器发送文件长度给其他用户)
FS --- file -------------------> Cj (服务器发送文件内容给其他用户)

4. 用户下线时候断开与文件服务器的连接
C0 --- [OFFLINE] --------------> FS
```

2.2 客户端设计思路

群聊和私聊

与服务器交互，直接按照前面所述的格式向服务器发送数据，并且接受来自服务器的数据即可。发送数据时，将相关的数据按照相应的格式封装好再进行发送，接收数据时，将来自服务器的数据按照"[#]"进行切割，提取相关字段的数据之后呈递给 GUI 层使用。

文件传输

客户端不管是私聊还是群聊的文件都直接发送给服务器，再由服务器完成相应的转发。通信过程在前面已经有所阐述，这里不再重复。

与 GUI 层的交互

客户端和 GUI 分别是两个线程，GUI 负责用户界面的渲染和相关操作的响应，客户端则是负责跟服务器进行数据交互或者私聊的时候跟其他客户端进行交互，并且将交互的数据呈递给 GUI 使用。下面简单描述 GUI 与服务器的交互过程。

```
我将用"--->"来表示方法调用
    GUI                         Client
  点击连接按钮 ---------------> connect()  (连接服务器)
  点击退出按钮 ---------------> disconnect()  (下线操作)
  获取在线用户 ---------------> getOnlineUsers()  (获取当前在线用户的用户名、IP、端口号)
  群发消息     ---------------> sendGroupMsg(msg)  (发送消息msg给所有人)
  私聊消息     ---------------> sendMsg(msg, usr)  (私发消息msg给usr)
  切换用户     ---------------> getChatRecords(usr)  (切换为跟别人聊天，调出对应的聊天记录用于更新消息界面)
  发送文件     ---------------> sendFile(isGroup, filename, usr)  (将文件发送给所有人/某人，isGroup判断是否群发)
  更新界面     <--------------- 收到新消息（有人上线、下线、收到消息、收到文件的时候通知GUI进行信息更新)
```

2.3 GUI 设计思路

页面布局

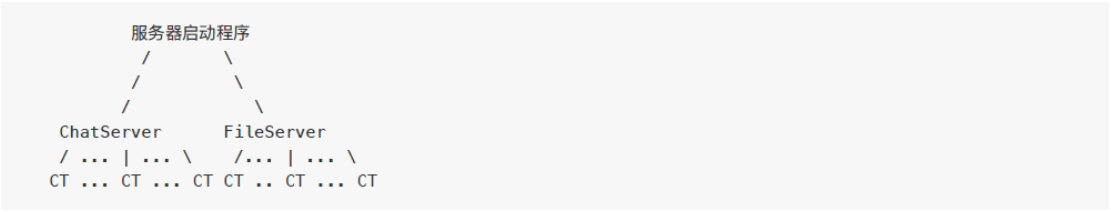
布局参考微信客户端，页面如下：

交互设计

- 用户打开程序后，填写服务器 IP 地址、端口、用户名，点击“连接”按钮就可以进入聊天。当用户想要退出聊天并关闭程序时，点击“退出”按钮即可。
- 窗口左侧为在线用户列表（默认第一个是群聊，即多人聊天），点击其中的一项可以与对方聊天（点击群聊进入多人聊天）。
- 当不是你当前聊天的用户发送消息给你时，在线用户列表中，发送消息的那个用户的名字后面将会出现“（新消息）”，以提示他有消息发送给你。
- 窗口右侧的上面为聊天消息列表，显示用户与对方的聊天记录。
- 窗口右侧的下面为发送消息窗口，用户可以填写消息，点击发送按钮消息将发送给对方，并提示对方收到消息。
- 假如用户需要发送文件，点击发送文件并选择需要发送的文件，对方将收到接收到文件的提示，发送的文件默认保存在当前目录下。

2.4 程序结构

服务端



服务端采取多线程模式，首先由启动程序开启两个服务器线程，每个服务器进程里面分别监听相应的端口，当收到客户端发起的连接时候，就新开辟一个新的线程 CT(ClientThread)去服务相应的客户端。

客户端

```
    Client
   /      \
  /        \
 G          F
```

客户端这边有 2 个线程，G 负责群聊，F 负责文件，G 和 F 都是简单跟服务器交互即可完成相关任务。

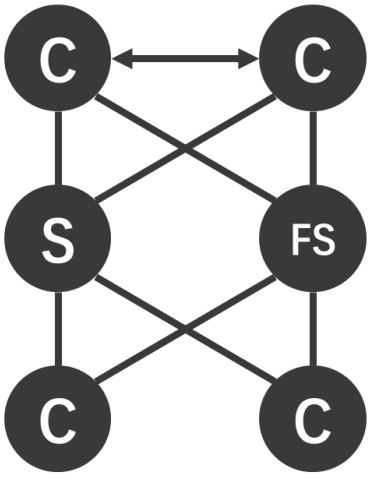
用户程序

```
    用户程序
    /    \
  GUI    Client
```

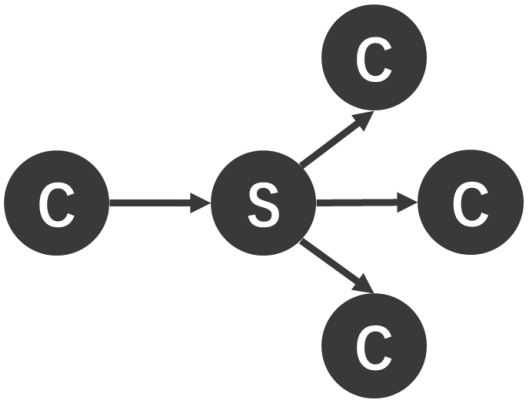
用户程序启动之后会创建两个线程，GUI 线程负责用户页面，Client 负责跟服务器交互以及跟其他私聊的客户端交互并向 GUI 呈递数据。

2.5 基本模型图

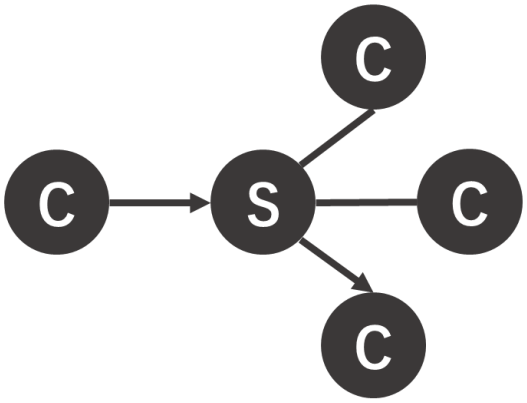
整个网络的拓扑图：



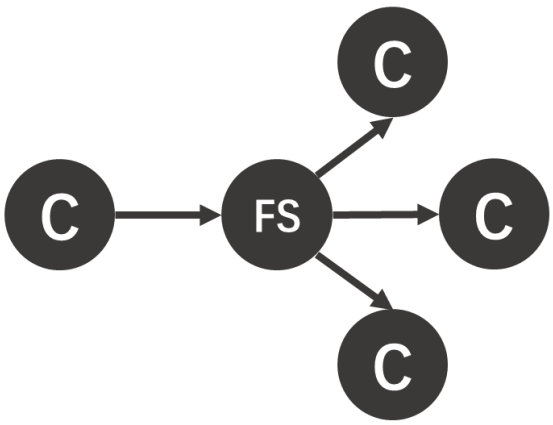
各功能的通信模型图：



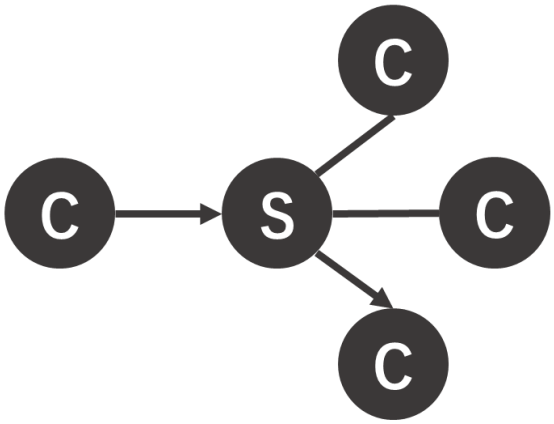
群聊



私聊



群发文件



私发文件

3. 部署启动

进入到项目的主目录之后，按照以下步骤进行操作即可完成项目的启动和运行。

编译所有文件

```
cd server
javac *.java
cd ..
cd client
javac *.java -classpath .:beautyeye_lnf.jar
cd ..
javac *.java
```

启动服务器

```
java ServerMain 8080
```

启动客户端

```
java -classpath .:client/beautyeye_lnf.jar ClientMain
```

4. 实验结果

实验结果请见视频。

5. 项目管理记录

10.20-10.23：实验设计与框架选择

10.24-10.26：客户端基本功能开发、服务器端开发、GUI 开发

10.27-10.28：客户端文件传输开发、服务器端开发

10.29：客户端代码与 GUI 代码整合

10.30：客户端与服务器端联合调试

10.31：完成实验报告、PPT、视频 demo

6. 团队分工

团队成员	学号	工作任务	小组自评
黄建武	14331098	客户端基本功能开发	95
陈平永	14331029	客户端基本功能开发	95
蔡桐钊	14331010	客户端 GUI 开发	95
骆铭涛	14331206	客户端 GUI 开发	95
陈海涛	14331016	客户端文件传输开发	95
何晋豪	14331084	客户端文件传输开发	95
洪创煌	14331085	服务器端开发	95
童云钊	14331250	实验报告、PPT	95