

基于JAVA Socket编程的聊天程序

计算机93 李云广

1 实验原理

1. 使用TCP连接服务器和客户端，使用Java Socket实现客户端和服务器的信息传输。
2. Java把Socket作为一个普通的文件来看待，使用标准输入输出流来进行Socket的读取和写入。
3. 服务器开放一个5000端口等待客户端的连接，每有一个客户端连接上服务器，服务器都会新开一个线程来处理这个TCP连接。
4. 对于文件的处理Java也是使用的标准输入输出流，服务器仅仅用来转发收到的文件。

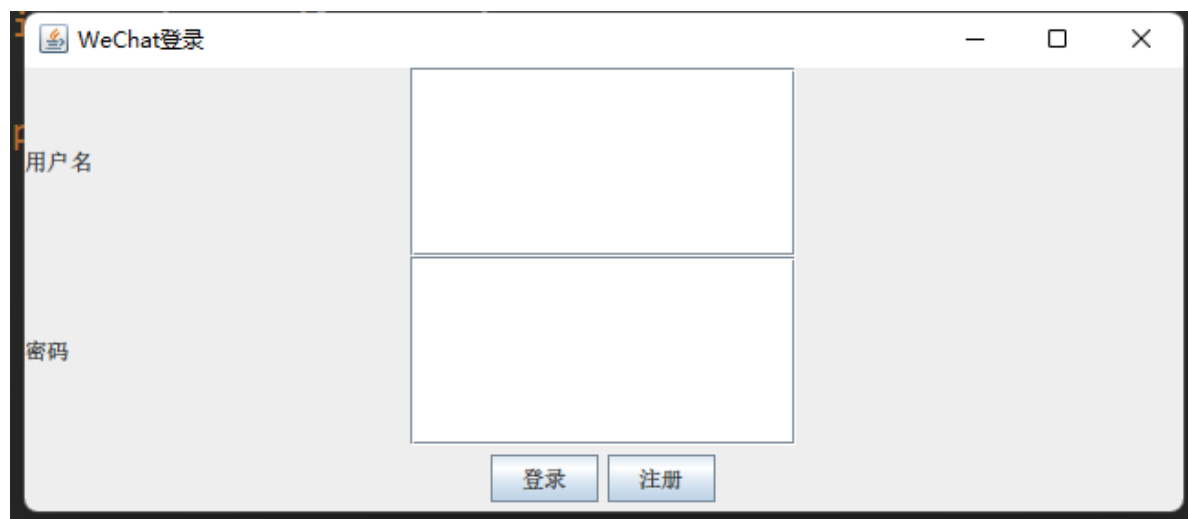
2 实验完成情况

1. 实现了客户端和服务器的聊天程序。
2. 实现了客户端的UI界面的设计。
3. 在客户端实现了登录和注册的功能。
4. 实现了客户端之间的文件传输。
5. 实现了客户端之间的群聊、私聊的功能。
6. 实现了一个在线的好友列表，在有新的好友上线的时候会更新好友列表。

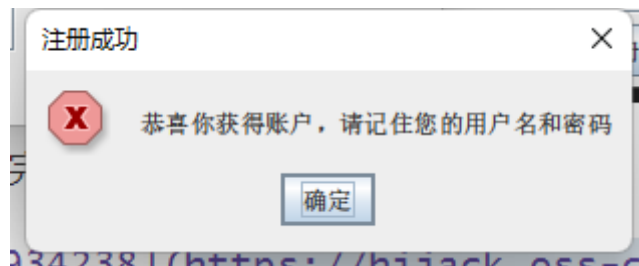
3 实验结果演示

3.1 登录和注册

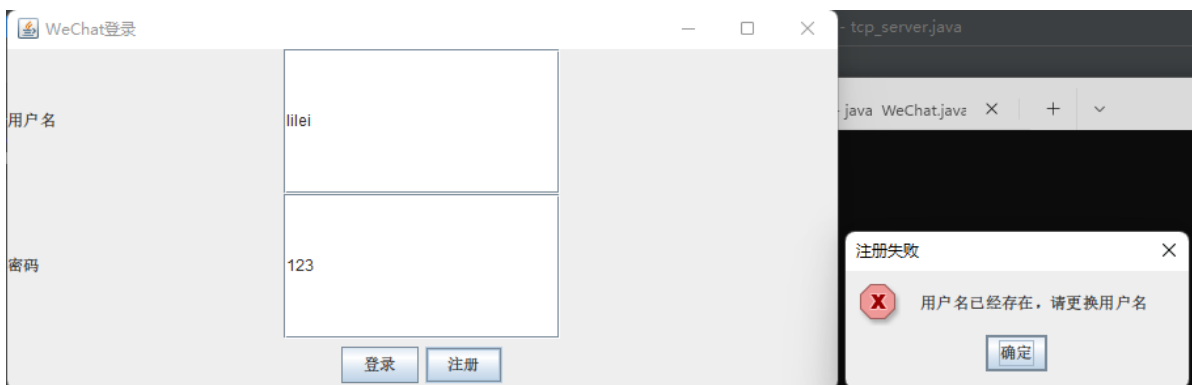
首先是运行程序后的第一个界面。



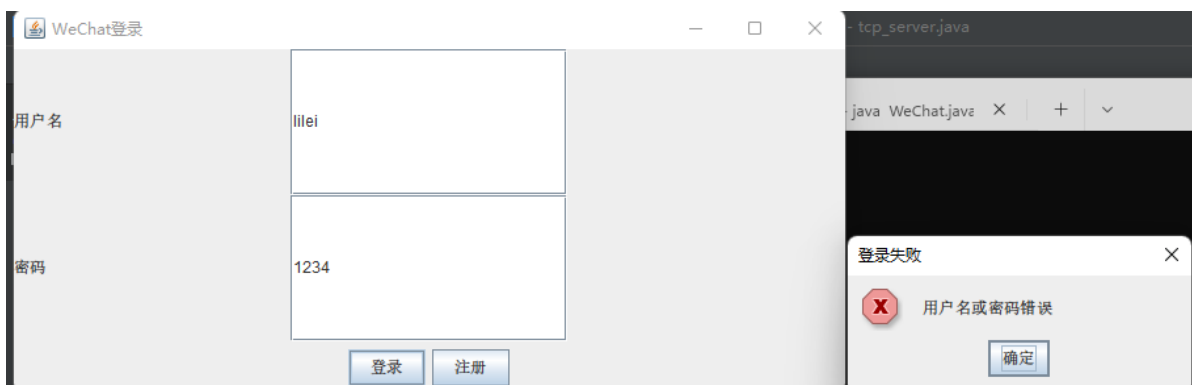
输入用户名密码点击注册即可完成注册。



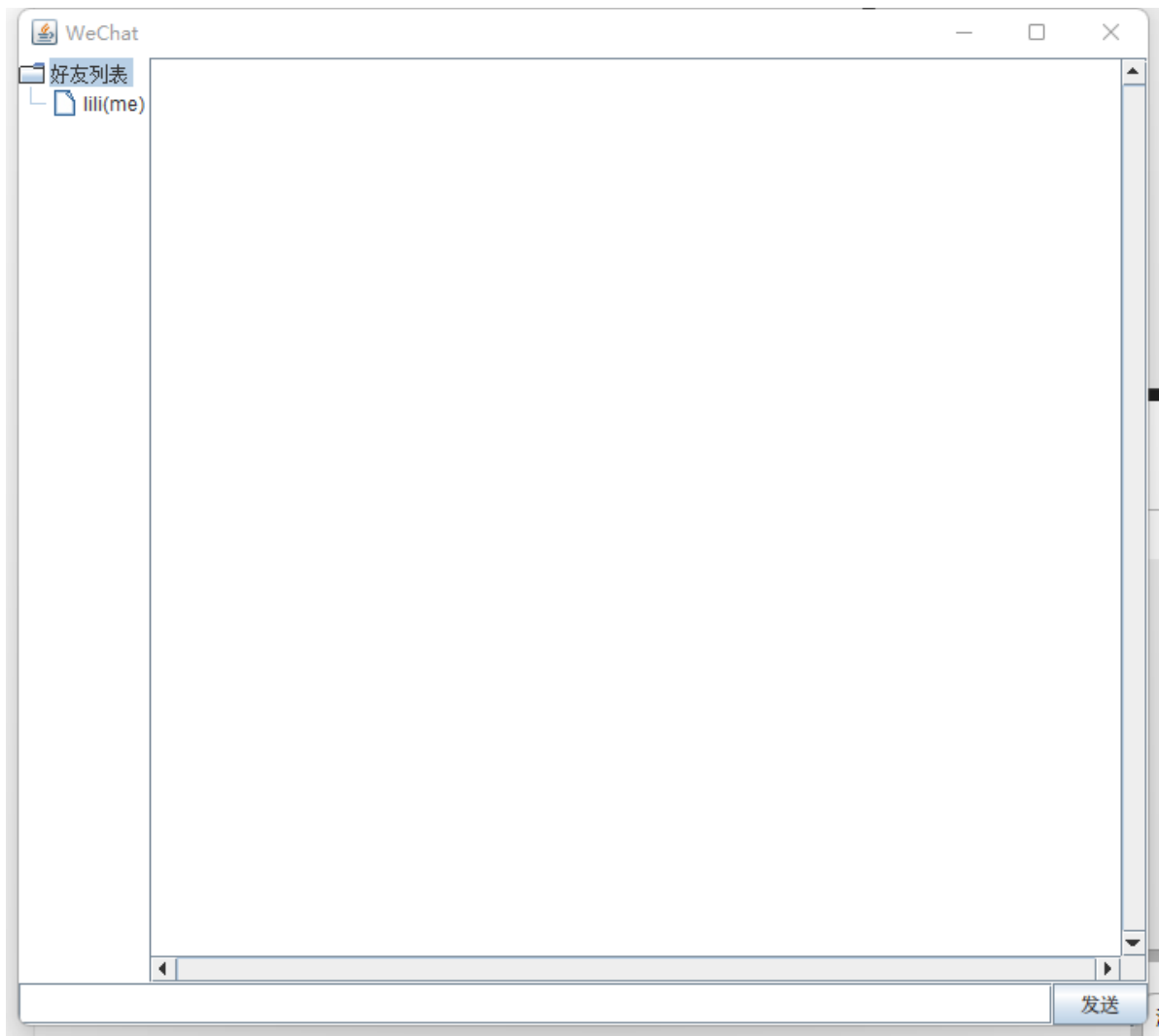
如果注册的用户名与服务器存储的用户名重复会返回错误：



注册之后用户名就存储在了服务器中，下一次就可以直接使用这个用户名进行登录，但是用户名或密码输入错误也会返回错误：



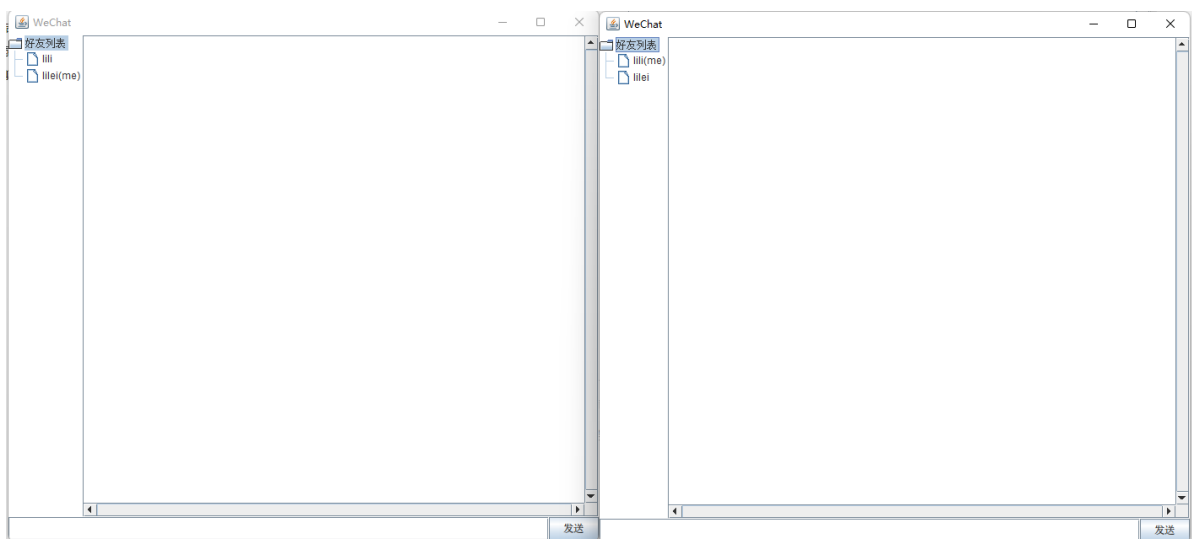
如果是正常登录会出现以下界面：



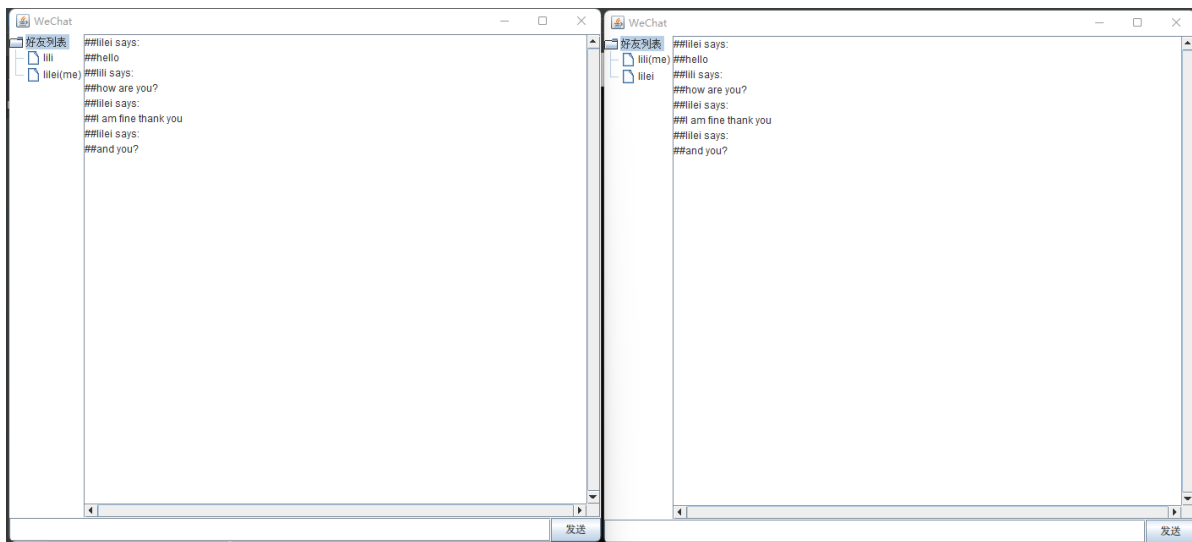
登录后的界面是这样的，有一个在线的好友列表，可以显示当前的在线好友。

3.2 群聊和私聊

分别注册李雷和丽丽两个账号，进行登录，显示结果如下：

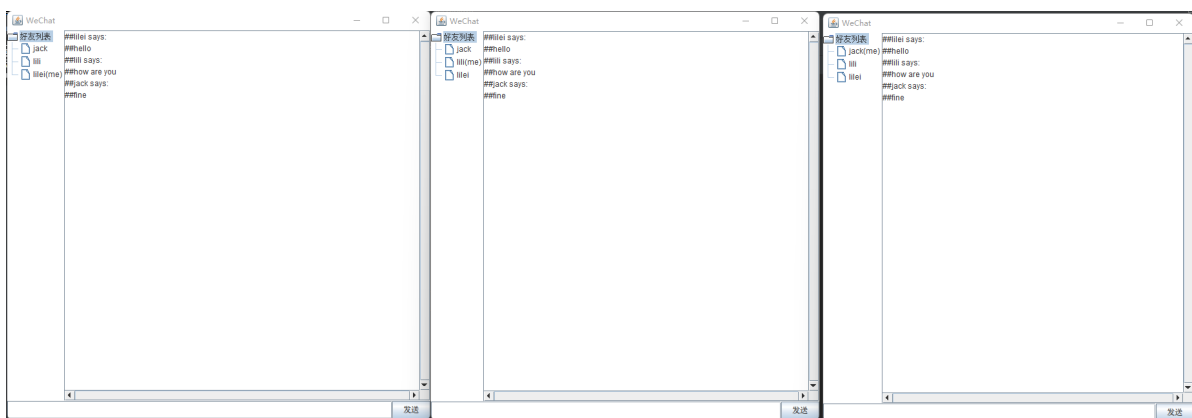


分别使用lilei和lili发送消息：



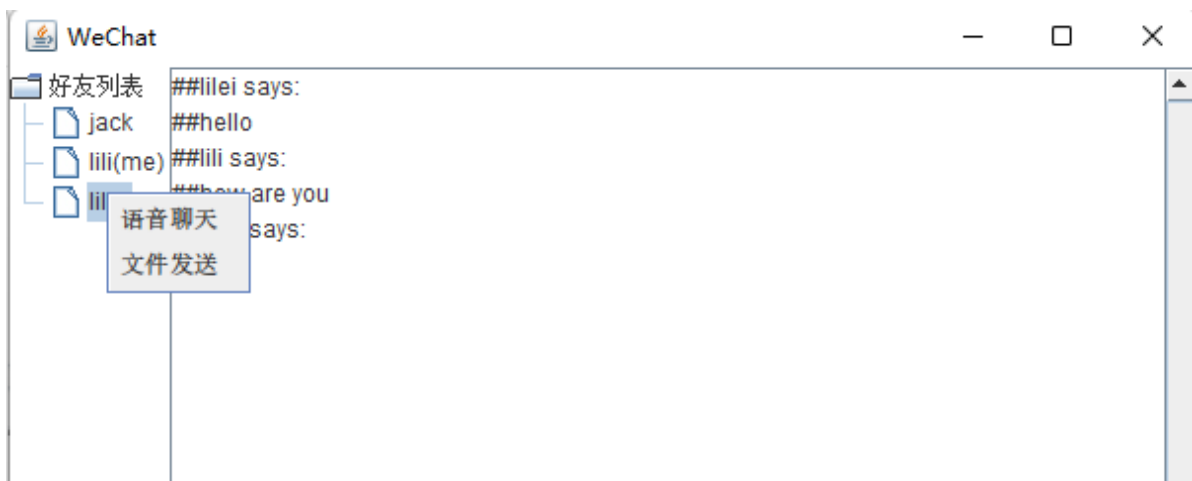
这样就实现了私聊的功能。

同时登录多个账户进行共同聊天，就实现了群聊的功能：



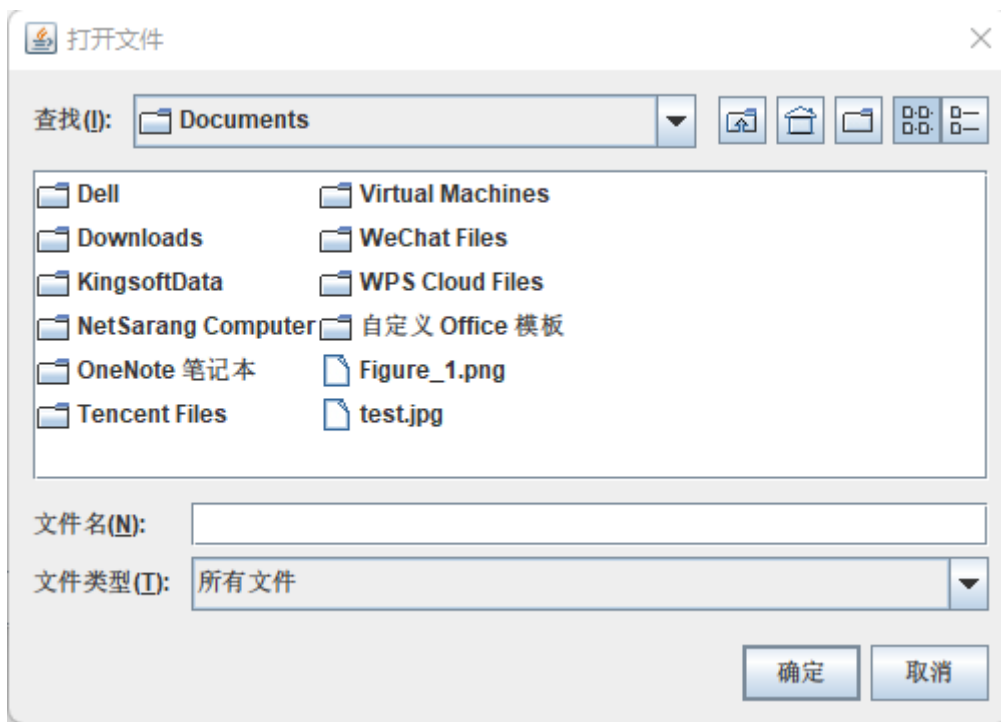
3.3 文件传输

在线好友列表的每一个好友都是可以进行双击的，双击效果如下：



由于时间原因，语音聊天我仅仅写出了UI框架但是没有实现其功能，所以这里仅仅演示文件发送：

点击文件发送会出现一个文件选择框：



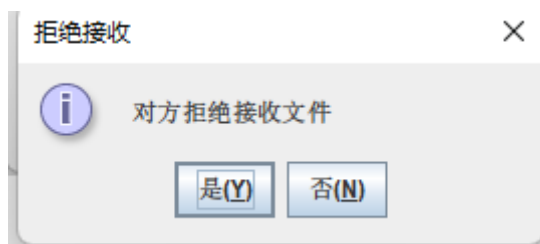
即可选择文件进行发送：

选择后我设置的接收端需要确定是否可以接收文件，如果不能接收文件就会在发送端返回错误。

接收端：（点击取消）



发送端：



如果接收端可以正常接收文件，则接收端点击确定，文件直接发送至接收端，在这里我默认发送到接收端的D盘根目录下：

	test.jpg	2022/5/10 13:27	JPG 文件	38 KB
--	----------	-----------------	--------	-------

可以看到文件正常传输了。

4 实验关键思路以及代码解释

我设计的Socket编程传输的每个包的格式基本是这样的：

```
1 | {ID:X,...}
```

通过开始一个ID号来标识这是一个什么包。

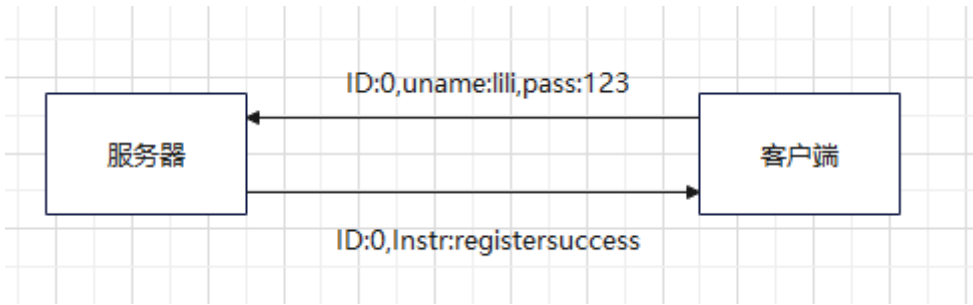
4.1 登录和注册

对于登录和注册，发送的消息是有一个专门的ID号的：

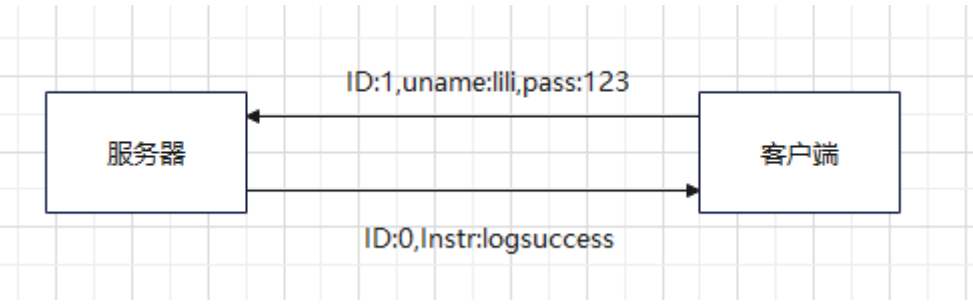
客户端发送的ID	类型
1	注册消息
2	登录消息

服务器发送的ID	类型
0	登录和注册的指令

注册：



登录：



抓包示例

0000	c0 a5 dd 56 0e d0 60 dd 8e 03 8c 98 08 00 45 00	...V...E.
0010	00 44 15 e0 40 00 80 06 00 00 c0 a8 01 64 27 6a	.D..@...d'j
0020	1d e0 e1 e9 13 88 cb 74 f6 59 eb 37 a0 6f 50 18t.Y.7.oP.
0030	02 05 07 8d 00 00 7b 49 44 3a 31 2c 55 6e 61 6d{I D:1,Unam
0040	65 3a 6c 69 6c 69 2c 50 61 73 73 3a 31 32 33 7d	e:lili,P ass:123}
0050	0d 0a	..

[Length: 40]

0000	60 dd 8e 03 8c 98 c0 a5 dd 56 0e d0 08 00 45 00	~.....V....E.
0010	00 50 6c e8 40 00 34 06 d2 69 27 6a 1d e0 c0 a8	.P1.@.4. .i'j....
0020	01 64 13 88 e1 e4 7d 83 42 48 ca 16 b8 3d 50 18	.d....}. BH...=P.
0030	00 e5 7c ed 00 00 7b 49 44 3a 30 2c 49 6e 73 74{I D:0,Inst
0040	72 3a 72 65 67 69 73 74 65 72 73 75 63 63 65 73	r:regist ersucces
0050	73 2c 55 6e 61 6d 65 3a 6c 69 6c 69 7d 0a	s,Unicode: lili}.

```

00 c0 a5 dd 56 0e d0 60 dd 8e 03 8c 98 08 00 45 00 ...V...` . ....E.
00 00 44 15 e4 40 00 80 06 00 00 c0 a8 01 64 27 6a .D...@... .....d'j
00 1d e0 e1 e9 13 88 cb 74 f6 75 eb 37 a0 97 50 18 .....t .u.7..P.
00 02 05 07 8d 00 00 7b 49 44 3a 32 2c 55 6e 61 6d .....{I D:2,Unam
00 65 3a 6c 69 6c 69 2c 50 61 73 73 3a 31 32 33 7d e:lili,P ass:123}
00 0d 0a ..

```

服务器代码

- 注册

```

1  if(message_id == 1){
2      System.out.println("register message");
3      //注册
4      String uname = getUname(message);
5      String password = getPassword(message);
6      //找找看用户名是不是已经存在
7      int flg = name_list.indexOf(uname);
8      if(flg == -1){
9          name_list.add(uname);
10         password_list.add(password);
11         //回复一个注册成功
12         registersuccess(uname);
13     }
14 }
15 else{
16     //注册失败
17     registerfailed(uname);
18 }
19
20 }

```

```

1  //注册成功 向套接字发送 ID:0 Instr:registersuccess Uname: uname 告诉这个用户注册成功
   了
2  public void registersuccess(String Uname){
3      String message = "{ID:0,Instr:registersuccess,Uname:" + Uname + "}";
4      tellEveryone(message);
5  }
6  //注册失败 向套接字发送 ID:0 Instr:registerfailed Uname: uname 告诉这个用户注册成功
   了
7  public void registerfailed(String Uname){
8      String message = "{ID:0,Instr:registerfailed,Uname:" + Uname + "}";
9      tellEveryone(message);
10 }

```

- 登录

```

1  else if(message_id == 2){
2      System.out.println("log_in message");
3      //登录
4      String uname = getUname(message);
5      String password = getPassword(message);
6      int index = name_list.indexOf(uname);

```

```

7      if(index == -1){
8          //用户名错误 登录失败
9          loginfailed(uname);
10     }
11     else if(password_list.get(index).equals(password)){
12         //登录成功
13         loginsuccess(uname);
14     }
15     else{
16         //密码错误 登录失败
17         loginfailed(uname);
18     }
19 }

```

```

1  //登录成功 向套接字发送 ID:0 Instr:logsuccess Uname: uname 告诉这个用户注册成功了
2  public void loginsuccess(String Uname){
3      user_online.add(Uname);
4      String message = "{ID:0,Instr:logsuccess,Uname:" + Uname + "}";
5      tellEveryone(message);
6      //发送 更新好友列表的信息
7      int num = clientOutputStreams.size();
8      String friends_message = "{ID:1," + "Num:" + Integer.toString(num) +
9      ",";
10     for(int i = 0;i < user_online.size();i++){
11         friends_message = friends_message + user_online.get(i) + ",";
12     }
13     friends_message = friends_message + "}";
14     try{
15         Thread.sleep(100);
16     }catch (Exception ex){
17         ex.printStackTrace();
18     }
19     tellEveryone(friends_message);
20 }
21 //登录失败 向套接字发送 ID:0 Instr:logfailed Uname: uname 告诉这个用户注册成功了
22 public void loginfailed(String Uname){
23     String message = "{ID:0,Instr:logfailed,Uname:" + Uname + "}";
24     tellEveryone(message);
25 }

```

客户端代码:

```

1  int ID = get_ID(message);
2  if(ID == 0){
3      //ID为0的消息是有关登录的指令
4      String instr = get_Instr(message);
5
6      if(instr.equals("logsuccess")){
7          //成功登录
8          //关闭登录界面
9          //显示主界面
10         System.out.println("log_success!!!");
11         my_uname = login_m.log_success();
12         break;
13     }
14     else if(instr.equals("logfailed")){

```



```

15         //如果不成功登录，显示用户名或密码错误
16         login_m.log_failed();
17     }
18     else if(instr.equals("registersuccess")){
19         //如果成功注册，显示成功注册
20         login_m.register_success();
21     }
22     else if(instr.equals("registerfailed")){
23         //如果不成功注册，显示用户名或密码已注册
24         login_m.register_failed();
25     }
26 }

```

```

1 //登录成功
2 public String log_success(){
3     System.out.println("log_success");
4     //关闭登录界面
5     frame.dispose();
6     return my_uname;
7 }
8 //登录失败，用户名或密码错误
9 public void log_failed(){
10     JOptionPane.showMessageDialog(null, "用户名或密码错误", "登录失败",
11     JOptionPane.ERROR_MESSAGE);
12     System.out.println("log_failed");
13 }
14 //注册成功
15 public void register_success(){
16     JOptionPane.showMessageDialog(null, "恭喜你获得账户，请记住您的用户名和密码",
17     "注册成功", JOptionPane.YES_NO_OPTION);
18     System.out.println("register_success");
19 }
20 //注册失败
21 public void register_failed(){
22     JOptionPane.showMessageDialog(null, "用户名已经存在，请更换用户名", "注册失
23     败", JOptionPane.ERROR_MESSAGE);
24     System.out.println("register_failed");
25 }
26 }

```

4.2 群聊私聊

本次实验实现的群聊非常简单，直接使用服务器对于接收到的所有聊天消息拷贝一份发送给每个用户：

```

1 public void tellEveryone(String message) {
2     //使用迭代器遍历ArrayList
3     Iterator it = clientOutputStreams.iterator();
4     //迭代将clientOutputStream的东西全部输出
5     while (it.hasNext()) {
6         try {
7             client_output co = (client_output) it.next();
8             PrintWriter writer = co.writer;
9             //PrintWriter writer = (PrintWriter) it.next();
10            writer.println(message);
11            //刷新这个流 将缓冲区中的数据强行发送到目的地
12            writer.flush();

```

```
13         } catch (Exception ex) { ex.printStackTrace(); }
14     }
15 }
```

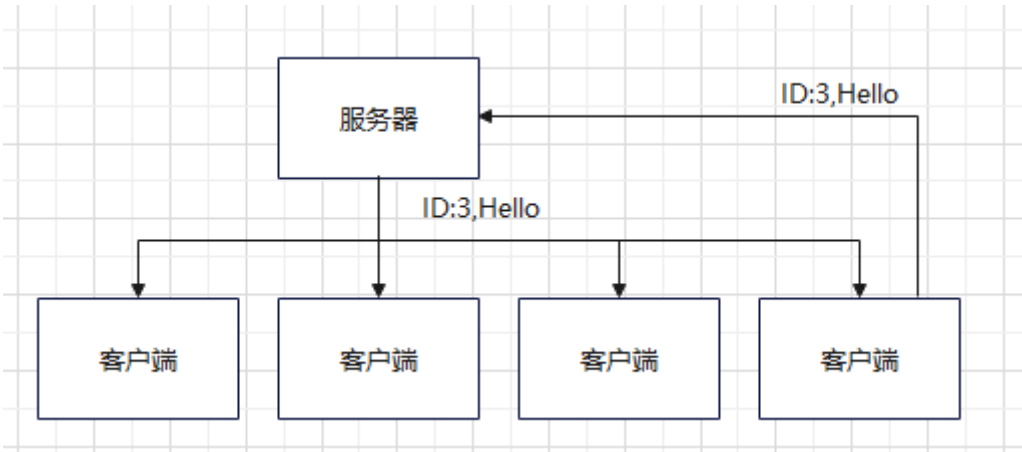
当然也有发送给某一个用户消息的方法：

```
1 public void tellSomeone(String message,String uname){
2     int x = user_online.indexOf(uname);
3     if(x == -1){
4         System.out.println("no user");
5         System.out.println(uname);
6         return;
7     }
8     System.out.println(x);
9     System.out.println("tellSomeone:");
10    try{
11        client_output co = (client_output) clientOutputStreams.get(x);
12        PrintWriter writer = co.writer;
13        writer.println(message);
14        writer.flush();
15    }catch (Exception ex){ex.printStackTrace();}
16
17 }
```

实现聊天需要一定的格式：

客户端发送的ID	类型
3	普通消息

服务器发送的ID	类型
3	普通消息



抓包分析

```
00 c0 a5 dd 56 0e d0 60 dd 8e 03 8c 98 08 00 45 00 ...V...`...E.
00 00 3b 15 fb 40 00 80 06 00 00 c0 a8 01 64 27 6a -;..@... ..d'j
00 1d e0 e1 f8 13 88 b1 44 23 b6 45 e6 4c 5b 50 18 .....D #·E·L[P·
00 02 05 07 84 00 00 7b 49 44 3a 33 23 23 6a 61 63 .....{I D:3##jac
00 6b 20 73 61 79 73 3a 0d 0a k says:..
```

```
00 c0 a5 dd 56 0e d0 60 dd 8e 03 8c 98 08 00 45 00 ...V...`...E.
00 00 36 15 fc 40 00 80 06 00 00 c0 a8 01 64 27 6a -6..@... ..d'j
00 1d e0 e1 f8 13 88 b1 44 23 c9 45 e6 4c 6d 50 18 .....D #·E·LmP·
00 02 05 07 7f 00 00 7b 49 44 3a 33 23 23 68 65 6c .....{I D:3##hel
00 6c 6f 0d 0a lo..
```

服务器代码

```
1 else if(message_id == 3){
2     //正常消息 发送到所有用户
3     System.out.println("normal message!!");
4     System.out.println(message);
5     String msg = message.substring(5);
6     msg = "{ID:3" + msg;
7     tellEveryone(msg);
8 }
```

客户端代码

```
1 else if(ID == 3){
2     //更新聊天框
3     textArea1.append(message.substring(5) + "\n");
4 }
```

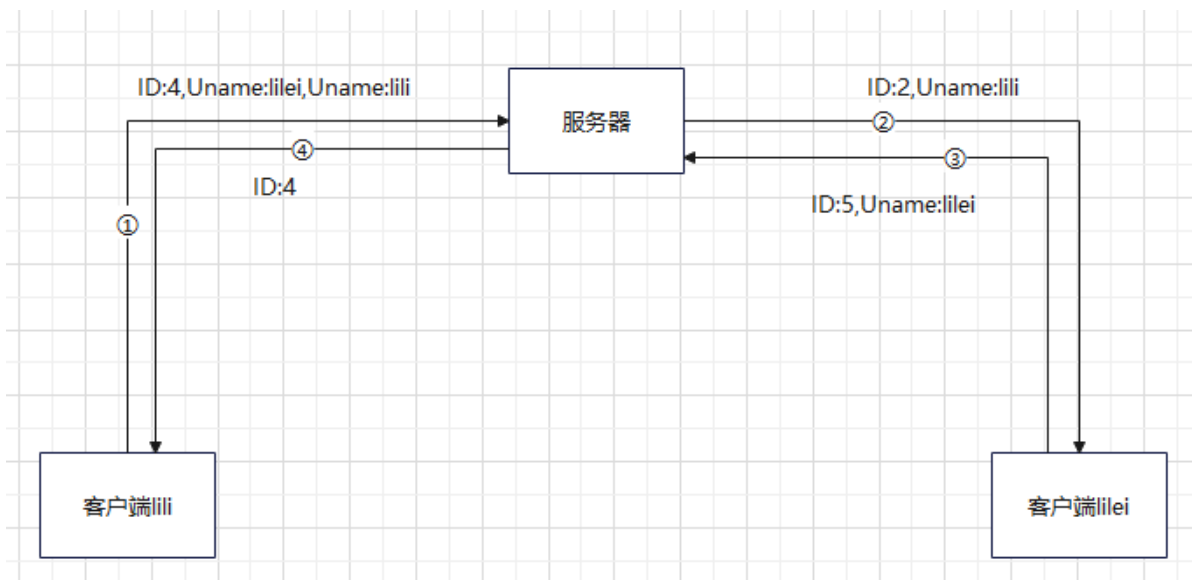
4.3 文件发送

本实验实现的文件发送非常复杂，需要两端类似握手的操作：

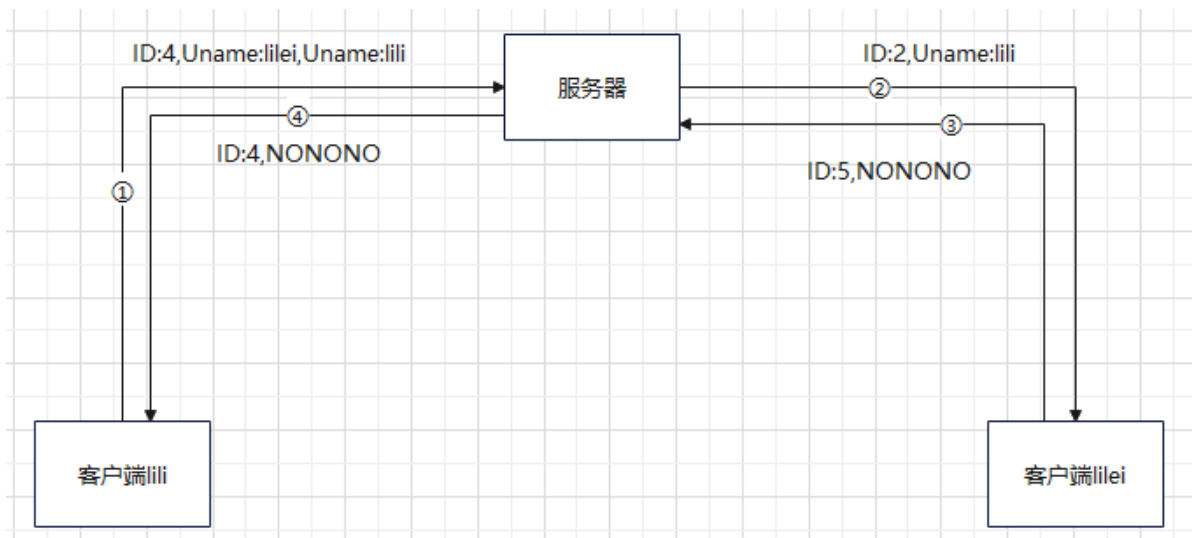
客户端发送的ID	类型
4	请求发送文件
5	可以接收文件

服务器发送的ID	类型
2	询问是否接收文件
4	允许发送文件

- 正常发送文件 lili -> lilei



- 拒接收文件



抓包分析

- jack想给lilei发送test.jpg

000	c0 a5 dd 56 0e d0 60 dd 8e 03 8c 98 08 00 45 00	...V...E.
010	00 59 16 03 40 00 80 06 00 00 c0 a8 01 64 27 6a	·Y·@·...·d'j
020	1d e0 e1 f8 13 88 b1 44 23 d7 45 e6 4c 7a 50 18D #·E·LzP·
030	02 04 07 a2 00 00 7b 49 44 3a 34 2c 55 6e 61 6d{I D:4,Unam
040	65 3a 6c 69 6c 65 69 2c 55 6e 61 6d 65 30 3a 6a	e:lilei, Uname0:j
050	61 63 6b 2c 46 69 6c 65 4e 61 6d 65 3a 74 65 73	ack,File Name:tes
060	74 2e 6a 70 67 0d 0a	t.jpg..

- 服务器询问lilei

000	60 dd 8e 03 8c 98 c0 a5 dd 56 0e d0 08 00 45 00	·.....·V....E·
010	00 4a 6c f1 40 00 34 06 d2 66 27 6a 1d e0 c0 a8	·Jl·@·4··f'j....
020	01 64 13 88 e1 e4 7d 83 43 34 ca 16 b8 3d 50 18	·d....}·C4...=P·
030	00 e5 40 4a 00 00 7b 49 44 3a 32 2c 55 6e 61 6d	·@J·{I D:2,Unam
040	65 6a 61 63 6b 2c 46 69 6c 65 4e 61 6d 65 3a 74	ejack,Fi leName:t
050	65 73 74 2e 6a 70 67 0a	est.jpg·

- lilei回复服务器 可以接收

```

00 c0 a5 dd 56 0e d0 60 dd 8e 03 8c 98 08 00 45 00 ...V...`.....E.
00 00 3b 16 06 40 00 80 06 00 00 c0 a8 01 64 27 6a .;..@... ..d'j
00 1d e0 e1 e4 13 88 ca 16 b8 3d 7d 83 43 56 50 18 .....=}.CVP.
00 02 04 07 84 00 00 7b 49 44 3a 35 2c 55 6e 61 6d .....{I D:5,Unam
00 65 3a 6c 69 6c 65 69 0d 0a .....e:lilei. .

```

- 服务器告诉jack可以发送

```

00 60 dd 8e 03 8c 98 c0 a5 dd 56 0e d0 08 00 45 00 ^.....V....E.
00 00 2f 7d 3e 40 00 34 06 c2 34 27 6a 1d e0 c0 a8 ./}>@.4. 4'j....
00 01 64 13 88 e1 f8 45 e6 4c 7a b1 44 24 08 50 18 .d....E. Lz.D$.P.
00 00 e5 4c 5b 00 00 7b 49 44 3a 34 7d 0a ...L[...{I D:4}.

```

服务器代码

```

1 else if(message_id == 4){
2     System.out.println(message);
3     System.out.println("want to send file");
4     String uname_receiver = getUname(message);
5     uname_send = getSender(message);
6     String file_name = getFileName(message);
7     //询问接收方 是否接收
8     accept_or_not(uname_receiver, file_name);
9 }

```

```

1 public void accept_or_not(String uname,String fileName){
2     String message = "{ID:2," + "Uname" + uname_send + ",FileName:" +
3     fileName;
4     tellSomeone(message,uname);
5 }

```

```

1 else if(message_id == 5){
2
3     if(message.substring(message.length() - 2).equals("NO")){
4         //对方拒接 要求停止发送
5         System.out.println("对方拒绝接收文件，请停止发送");
6         String message_n = "{ID:4,NONONO";
7         tellSomeone(message_n,uname_send);
8     }
9     else{
10        //对方同意接收
11        System.out.println("开始传输文件");
12        String message_y = "{ID:4}";
13        String uname = getUname(message);
14        tellSomeone(message_y,uname_send);
15    }
16
17 }

```

```

1 public void tellSomeone(String message,String uname){
2     int x = user_online.indexOf(uname);
3     if(x == -1){
4         System.out.println("no user");
5     }
6 }

```

```

5         System.out.println(uname);
6         return;
7     }
8     System.out.println(x);
9     System.out.println("tellSomeone:");
10    try{
11        client_output co = (client_output) clientOutputStreams.get(x);
12        PrintWriter writer = co.writer;
13        writer.println(message);
14        writer.flush();
15    }catch (Exception ex){ex.printStackTrace();}
16
17 }

```

客户端代码

- 接收端

```

1  else if(ID == 2){
2      //询问接收文件
3      String uname_sender = getUname(message);
4      fileName_rev = getFileName(message);
5      //弹出确认窗口
6      int rst = JOptionPane.showConfirmDialog(null, uname_sender + "想要给你传输文件" + fileName_rev, "是否接收", JOptionPane.OK_CANCEL_OPTION,
JOptionPane.INFORMATION_MESSAGE);
7      if(rst == JOptionPane.OK_OPTION){
8          //点击确定 ---- 正常接收文件
9          String message_yes = "{ID:5,uname:" + my_uname;
10         System.out.println("yes");
11         writer.println(message_yes);
12         writer.flush();
13         getfile();
14     }
15     else{
16         //点击取消 ---- 拒绝接收文件
17         String message_no = "{ID:5,NONONO";
18         System.out.println("no");
19         writer.println(message_no);
20         writer.flush();
21     }
22 }

```

```

1  public void getfile(){
2      try{
3          //直接存到D盘根目录
4          FileOutputStream fos = new FileOutputStream("D://" + fileName_rev);
5          byte[] buffer = new byte[10];
6          int len;
7          while ((len = is.read(buffer)) == 10)
8          {
9              fos.write(buffer);
10             fos.flush();
11         }
12         fos.write(buffer);
13         fos.flush();

```

```

14         fos.close();
15         System.out.println("文件接收完成");
16     }catch (Exception ex){
17         ex.printStackTrace();
18     }
19 }

```

- 发送端

```

1  else if(ID == 4){
2      if(message.substring(message.length() - 2).equals("NO")){
3          //对方拒绝接收文件
4          System.out.println("对方不接受的呀");
5          JOptionPane.showConfirmDialog(null, "对方拒绝接收文件", "拒绝接收",
JOptionPane.OK_OPTION, JOptionPane.INFORMATION_MESSAGE);
6      }
7      else{
8          //可以发送文件了
9          System.out.println("OK to send");
10         writer.println("{ID:6}");
11         writer.flush();
12         send_file();
13     }
14 }

```

```

1  void send_file(){
2      System.out.println(file.getPath());
3      //发送文件
4      try{
5          FileInputStream fis = new FileInputStream(file);
6          BufferedInputStream bis = new BufferedInputStream(fis);
7          byte[] b = new byte[10];
8          int i = -1;
9          while((i = bis.read(b)) == 10){
10             System.out.println("len = ");
11             System.out.println(i);
12             os.write(b,0,i);
13             os.flush();
14         }
15         os.write(b);
16         os.flush();
17         writer.write(" ");
18         writer.flush();
19         System.out.println("文件发送完毕");
20         //关闭流
21         bis.close();
22     }catch (Exception ex){
23         ex.printStackTrace();
24     }
25 }

```

这里接收输入文件都是使用的java标准输入输出流，使用字节流进行传输的，同时在这里规定的每个传输单位为10个字节，可以有效防止传输过大导致失败。

4.4 好友列表动态更新

本次实验除了进行规定的几个要求之外，设置了好友列表动态更新的功能，每上线一个好友，好友列表都会更新一下，这样就实现了进行私聊文件传输的功能。

服务器发送的ID	类型
1	更新好友列表

每次有新的用户登录，服务器都会发送一个ID为1的包，用于更新好友列表，客户端处理这个消息，动态更新自己的列表并显示出来。

抓包分析

0000	60 dd 8e 03 8c 98 c0 a5 dd 56 0e d0 08 00 45 00	~.....-V....E.
0010	00 3c 6c e7 40 00 34 06 d2 7e 27 6a 1d e0 c0 a8	·<1·@·4· ·~'j·...
0020	01 64 13 88 e1 e4 7d 83 42 34 ca 16 b8 3d 50 18	·d·...}· B4·...=P·
0030	00 e5 d3 72 00 00 7b 49 44 3a 31 2c 4e 75 6d 3a	...r·...{I D:1,Num:
0040	31 2c 6c 69 6c 65 69 2c 7d 0a	1,lilei, }·

0000	60 dd 8e 03 8c 98 c0 a5 dd 56 0e d0 08 00 45 00	~.....-V....E.
0010	00 46 7d 3a 40 00 34 06 c2 21 27 6a 1d e0 c0 a8	·F}:@·4· ·!'j·...
0020	01 64 13 88 e1 f8 45 e6 4c 3d b1 44 23 b6 50 18	·d·...E· L=D#·P·
0030	00 e5 da 75 00 00 7b 49 44 3a 31 2c 4e 75 6d 3a	...u·...{I D:1,Num:
0040	33 2c 6c 69 6c 65 69 2c 6c 69 6c 69 2c 6a 61 63	3,lilei, lili,jac
0050	6b 2c 7d 0a	k,}·

服务器代码

```
1 //登录成功 向套接字发送 ID:0 Instr:logsuccess Uname: uname 告诉这个用户注册成功了
2 public void loginsuccess(String Uname){
3     user_online.add(Uname);
4     String message = "{ID:0,Instr:logsuccess,Uname:" + Uname + "}";
5     tellEveryone(message);
6     int num = clientOutputStreams.size();
7     //在这里获取了存储的好友列表并发送给所有客户端
8     String friends_message = "{ID:1," + "Num:" + Integer.toString(num) +
9     ",";
10    for(int i = 0;i < user_online.size();i++){
11        friends_message = friends_message + user_online.get(i) + ",";
12    }
13    friends_message = friends_message + "}";
14    try{
15        Thread.sleep(100);
16    }catch (Exception ex){
17        ex.printStackTrace();
18    }
19    tellEveryone(friends_message);
20 }
```



```
1  else if(ID == 1){
2      //更新好友列表
3      update_onlinelist(message);
4      System.out.println(online_users);
5      top.removeAllChildren();
6      for(int i = 0;i < online_num;i++){
7          DefaultMutableTreeNode node;
8          if(online_users.get(i).equals(my_uname)){
9              //这是用户自己
10             node = new DefaultMutableTreeNode(online_users.get(i) + "(me)");
11             if(node.isNodeChild(top))continue;
12             model.insertNodeInto(node,top,0);
13         }
14         else{
15             //这是其他用户
16             node = new DefaultMutableTreeNode(online_users.get(i));
17             if(node.isNodeChild(top))continue;
18             model.insertNodeInto(node,top,0);
19         }
20     }
21 }
```

好友列表这里是使用的java swing组件中的一个动态树JTree结构，可以实现动态更新的效果。

5 实验总结

5.1 学到的知识

1. 从头学习的Java语言，Java语言把面向对象的特征发挥的淋漓尽致，每个class间都可以相互调用，每个函数均定义在类之中，本次实验我在服务器端仅写了一个类，而在客户端一共写了3个类，分别是登录、主界面、聊天程序，三个类的相互联系非常复杂，在摸索中才能勉强运行起来。
2. Java对于Socket的处理比较好，由于是使用标准输入输出流进行输入输出的，因此有许多封装好的方法或类，例如：
 - `Socket.getInputStream()` 可以获得标准的输入流，`InputStream` 超类，我们一般可以使用这个类的子类。
 - `InputStreamReader` 是字节流转化为字符流的桥梁，也就是将字节流转化为字符流进行处理。
 - `BufferedReader` 以缓冲的方式读取文本，提供了许多实用的方法。
 - `PrintWriter` 将标准输出流 `OutputStream`（字节流）转化为字符流进行输出，也封装了许多实用的方法。
3. Java图形化设计也比较有挑战性，有许多实验中的难点：
 - 使用 `JPopupMenu` 实现好友按钮双击出现菜单。
 - 使用 `JFileChooser` 实现选择文件的框。
 - 使用 `idea` 的 `Swing UI Designer` 自动生成 `UI` 的代码大大提高工作效率。
 - 使用 `JTree` 结构实现一个动态的好友列表，并且每个好友可以作为一个组件进行双击。

5.2 实验中的不足

实验中的不足之处非常多：

1. 刚开始自己写代码并没有考虑到自顶向下设计，没有从UI先写出框架，于是刚开始写底层的代码很多都是无用功，实际在UI中根本用不上，所以说一定要先写UI，或者先形成一个大体框架，然后再慢慢向框架中补充内容，形成整个项目。
2. 对于文件传输，我在顶层进行设计时就已经设计的是服务器对于每个客户端仅仅开一个线程进行工作，于是在我实际补充文件传输的代码的时候，发现如果文件传输可以新建一个线程的话会好写很多，但是当时的代码已经不允许这样大幅度的修改了，所以说进行顶层设计的时候要考虑到这些小问题，最终我实现了单个线程的文件传输，这样的话在进行大文件的传输的时候是不允许消息的发送的。
3. 写了语音聊天的框架，但是没有时间去加上这些内容。