# 实验报告netchat

### 原理

netchat利用Socket编程，服务端监听6666端口，为连入的用户创建用户线程peopleThread。

当聊天为公共聊天时：

if(s.startsWith("公共聊天内容:"))

{

String message=s.substring(s.indexOf(":")+1);

Enumeration enum1=peopleList.elements();

while(enum1.hasMoreElements())

{

((Server\_thread)enum1.nextElement()).out.writeUTF("公共聊天内容:"+message);

}

}

服务器枚举peoplelist，获取所有的与客户端通信的线程，把消息发送出去。

当聊天内容为私聊时：

else if(s.startsWith("私人聊天内容:"))

{

String 悄悄话=s.substring(s.indexOf(":")+1,s.indexOf("#"));

String toPeople=s.substring(s.indexOf("#")+1);

Server\_thread toThread=(Server\_thread)peopleList.get(toPeople);

if(toThread!=null)

{

toThread.out.writeUTF("私人聊天内容:"+悄悄话);

}

else

out.writeUTF("私人聊天内容:"+toPeople+"已经离线");

}

}

服务器根据用户ID找到相应线程并转发消息。

### 改进的地方

1. 在界面可以观察到程序实现了基本的公开聊天和私人聊天功能。发现一个可以改进的地方，在用户A对B发起私聊后，用户A发送的消息出现在用户B的私聊窗口，但是没有显示在用户A自己的窗口，这会造成用户前后文理解、记忆的不便，因此我在程序中添加一段代码，一旦用户发送信息成功，相应的信息也打印在自己的私聊框内。

**if**(toThread!=**null**)

{

toThread.out.writeUTF("私人聊天内容:"+悄悄话);

out.writeUTF("私人聊天内容:"+悄悄话);

}

B.消息添加时间戳，服务器时间为准。

SimpleDateFormat df = **new** SimpleDateFormat("(yyyy-MM-dd HH:mm:ss)");//设置日期格式

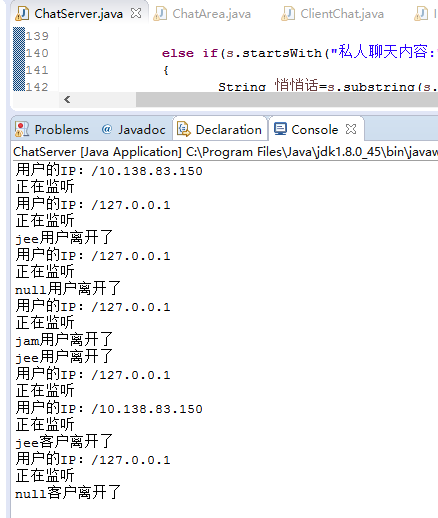
toThread.out.writeUTF("私人聊天内容:"+悄悄话+df.format(**new** Date()));

out.writeUTF("私人聊天内容:"+悄悄话+df.format(**new** Date()));

((Server\_thread)enum1.nextElement()).out.writeUTF("公共聊天内容:"+message+df.format(**new** Date()));

下面是实现过程叙述和截图。

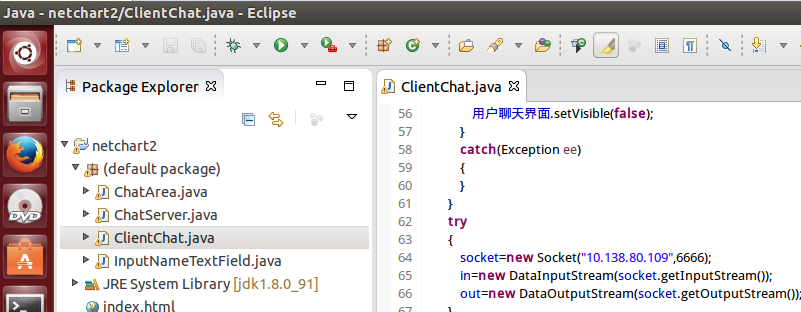
本次实验分为服务端和客户端，服务段运行于主机上，下图为服务端运行截图：



客户端A，运行于ubuntu虚拟机上，在ClientChat.java里修改了host为主机的地址

socket=**new** Socket(**this**.getCodeBase().getHost(),6666);

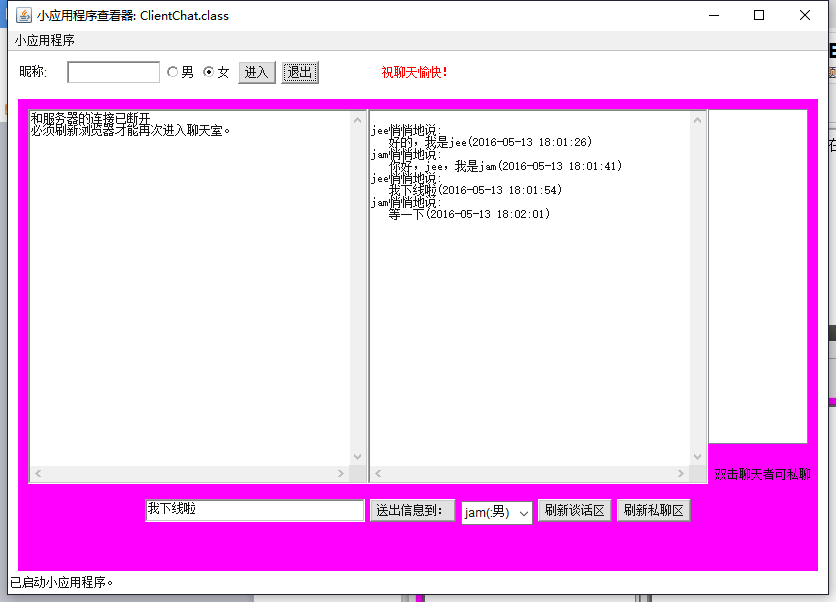
socket=**new** Socket("10.138.80.109",6666);



下图为客户端A运行截图（增加了私聊自反馈和时间戳）



下图为客户端B运行截图



### 3.收获

熟悉了socket编程，

对网络在应用层的实现有了更深刻的了解，

对网络聊天应用架构原理有了了解。

已上传至github https://github.com/CsterKuroi/netchat