

# 基于 python 的多人 聊天室

学号：1210122634  
姓名：白墨琛

## 一、任务目的

本项目是为了实现一个简单的 socket 多人聊天室程序。采用多线程设计，支持不同用户名，不同聊天室。整个软件设计简单、易用。

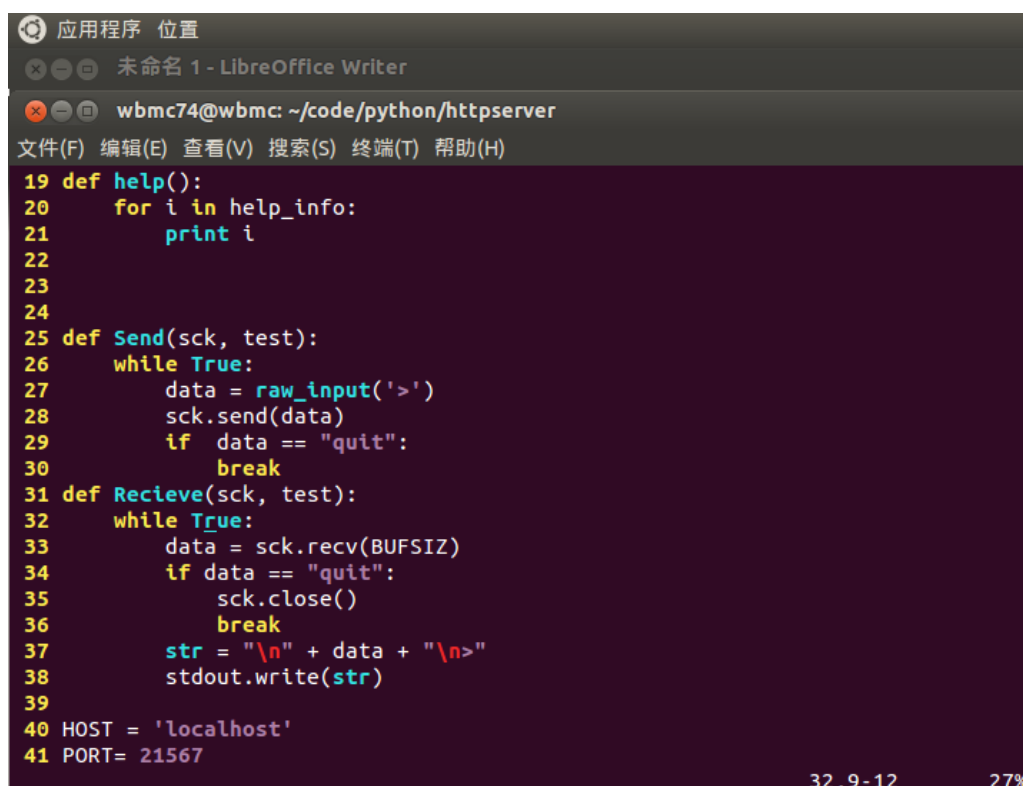
## 二、设计要求

本项目采用 python 语言。因为 python 语言简单灵活，类库丰富，较易实现。且本人教熟悉 python 语言，所以采用 python 编程实现。设计要求实现多人全双工通信聊天室。

## 三、软件模块

软件采用 c/s 架构，由客户端 cc.py 和服务端 cs.py 两部分组成。

cc.py 的主要函数有 send() 用来发送数据包，receive() 接收数据包，help() 用来说明软件的使用法。端口地址和主机地址可在代码中直接修改，默认为本机地址 localhost，21567 端口。cc.py 连接 cs.py 的时候会指定聊天室与登陆用户名。



The image shows a terminal window with a dark background and light-colored text. The window title is "应用程序 位置" (Application Location). The terminal shows the following Python code:

```
19 def help():
20     for i in help_info:
21         print i
22
23
24
25 def Send(sck, test):
26     while True:
27         data = raw_input('>')
28         sck.send(data)
29         if data == "quit":
30             break
31 def Recieve(sck, test):
32     while True:
33         data = sck.recv(BUFSIZ)
34         if data == "quit":
35             sck.close()
36             break
37         str = "\n" + data + "\n>"
38         stdout.write(str)
39
40 HOST = 'localhost'
41 PORT= 21567
```

The terminal also shows the window title "wbmc74@wbmc: ~/code/python/httpserver" and the menu bar "文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)". The bottom right corner of the terminal shows "32.9-12" and "27%".

cs.py 是服务端，支持多线程与并发，可同时与多个 cc.py 保持通信。主要函数有 Deal() 用来处理多用户登陆与退出聊天室。以及采用 threading 类库实现多线程。

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
34 clients = {"":{}}
35
36 while True:
37     print 'waiting for connection...'
38     chatCliSock, addr = chatSerSock.accept()
39     print "...connected romt: ", addr
40     data = chatCliSock.recv(BUFSIZE)
41     username, room = split(data)
42     print username
43     if not clients.has_key(room):
44         clients[room] = {}
45     if clients[room].has_key(username):
46         chatCliSock.send("reuse")
47         chatCliSock.close()
48     else:
49         chatCliSock.send("success")
50         clients[room][username] = chatCliSock
51         t = threading.Thread(target=Deal, args=(chatCliSock, username, room)
52     )
53         t.start()
54 chatSerSock.close()
```

37,34-37 底端

## 四、具体实现

实现的操作系统为 linux 的 ubuntu12.10 版本。采用 python 2.7 版本。涉及主要类库为: socket threading os sys 等。

socket 的 python 中的实现是创建一个 socket 类的对象，类似于 c 语言中的 socket 描述符。而创建多线程也是创建一个 Thread 类的对象。整个程序代码量估计不到 c 语言的二分之一，程序较为简洁，这也是采用 python 编程的主要原因。

## 五、未来计划 & 心得体会

未来计划加入图形界面，拟采用 QT 实现，还想加入密码验证登陆机制。相信后续版本还能有更多收获。

通过这次大作业，让我对网络程序设计有了一定的了解，也让我对python 这门强大的语言有了更深的体会。相信在未来能有更多进步。张老师本学期教授的这门网络编程课，让我进步很大，非常感谢！