1、 本项目是基于 Python 多线程 web 服务器的聊天室，主要功能为多人聊天、查看在线成员以及私聊某一成员。

2、 本项目的运行环境为 Windows 10，Python3.7.3，需要用到的 Python 库为：socket、threading

3、 服务端的代码思路为：

① 创建 socket 并绑定 ip 地址和端口号，初始化 socket 列表，定义用户与 IP 地址和 socket 的字典。

```
config = configparser.ConfigParser()
config.read('config.ini')
host = config.get('baseconf','host')
port = int(config.get('baseconf','port'))

s = socket.socket()
socket_list = []
client_List = {}
nickSocketList = {}
```

② 当有新的连接到来时，创建子进程进行处理，并且当用户发送的消息不是特殊命令时进行聊天室广播。

```
            if content =='#':
                s.send(str(client_List).encode('utf-8'))
                continue
            elif content == '&':
                s.send("1、输入#查询在线成员 2、其他功能正在开发中.....".encode('utf-8'))
                continue
            elif '@' in content:
                nick = content.split('@')[1]
                date = content.split('@')[0]
                nickSocketList[nick].send(date.encode('utf-8'))
                continue
            else:
                print(nickName + ' say: ' + content)
        # 将一个客户端发送过来的数据广播给其他客户端
        for client in socket_list:
            client.send((nickName + ' '+ 'say: ' + ' ' + content).encode('utf-8'))
```

③ 当有连接断开时显示用户离开

4、 客户端代码思路为:

① 创建套接字并连接服务器

② 向服务端发送消息

```
if flag ==0:
    nickName = input("请输入您的昵称：")
    nickName = '!'+nickName
    flag = 1
    s.send(nickName.encode('utf-8'))
    continue
else:
    line = input('')
    if line == 'exit':
        s.close()
        break
    # 主线程负责将用户输入的数据发送到socket中
    s.send(line.encode('utf-8'))
```

③ 创建子进程接受服务端返回的消息

```
def read_server(s):
    while True:
        # 子线程负责从服务端接受数据并打印
        content = s.recv(2048).decode('utf-8')
        print(content)

threading.Thread(target=read_server, args=(s,)).start()
```

5、 项目演示

① 在项目根目录命令行启动服务端

② 命令行分别启动三个以上的客户端

```
{'aaa': ('127.0.0.1', 4675)}
aaa ('127.0.0.1', 4675) Joined!
{'aaa': ('127.0.0.1', 4675), 'bbb': ('127.0.0.1', 4701)}
bbb ('127.0.0.1', 4701) Joined!
{'aaa': ('127.0.0.1', 4675), 'bbb': ('127.0.0.1', 4701), 'ccc': ('127.0.0.1', 4712)}
ccc ('127.0.0.1', 4712) Joined!
```

③ 发送消息，包括特殊命令

Aaa 向聊天室发送消息:

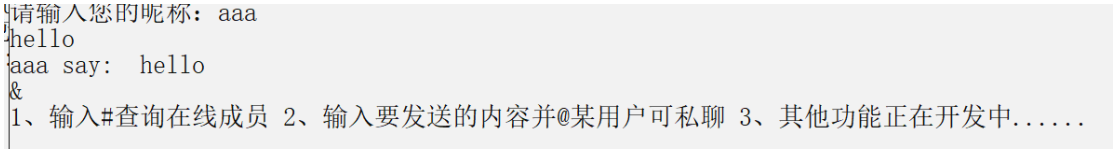
```
请输入您的昵称：aaa
hello
aaa say:  hello
```

Bbb 和 ccc 都接收到:

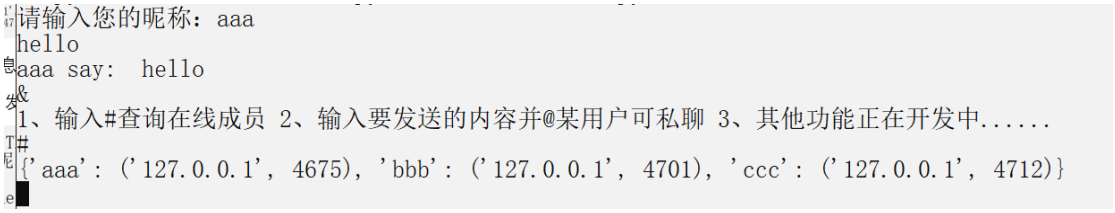
```
请输入您的昵称：bbb
aaa say:  hello
```

```
请输入您的昵称：ccc
aaa say:  hello
```

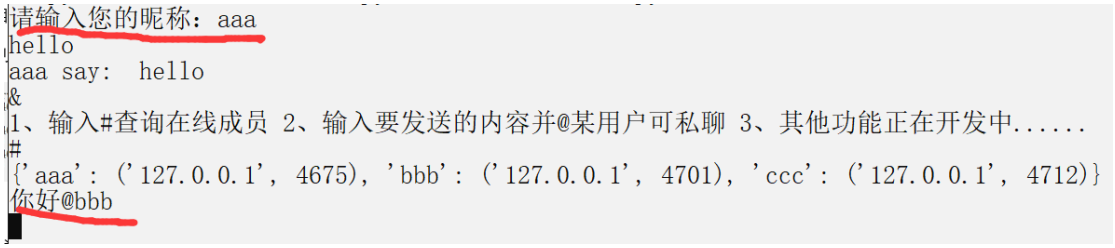
发送'&'查看提示:

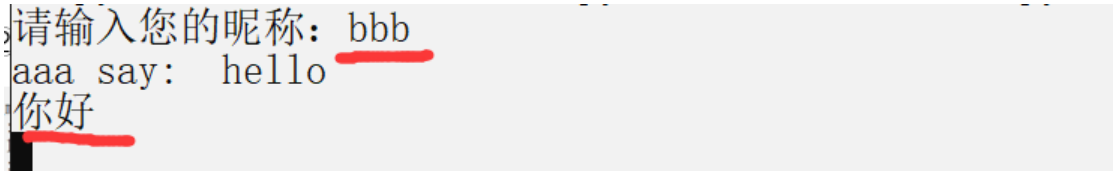


发送 # 查询在线成员:

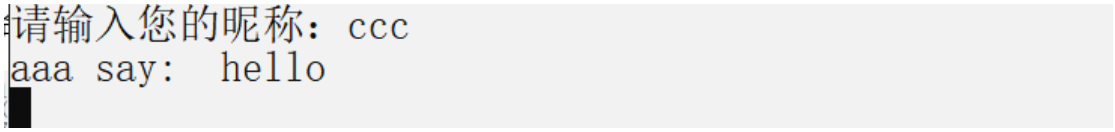


发送消息并@某人进行私聊:





而 ccc 并未接受到此消息:



其原理在于服务端解析发送的消息后将消息转发给了特定的用户而并没有进行广播。