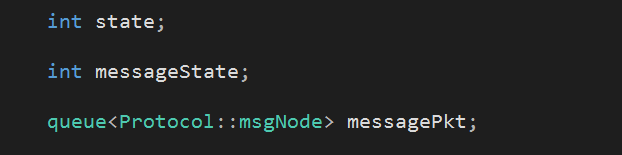
## 网络编程第四次作业---------select模型应用

**设计思路：**

客户端基本不变，主要将select模型应用于服务端程序，

使用select监听socket的变化，使用非阻塞socket，从而达到减少线程的使用，更改后使用两个线程，一个线程监听套接字的变化，并接受新连接，接收已建立连接发来的数据，然后将数据放到每个用户对用的session缓冲区内，session缓冲区内存放一个数据队列，队列节点数据放一个消息长度和只想消息内容的指针，另一个线程延续原来最基础的功能，处理包，对每个包处理相应的业务流程

在session中新增加以下几项：

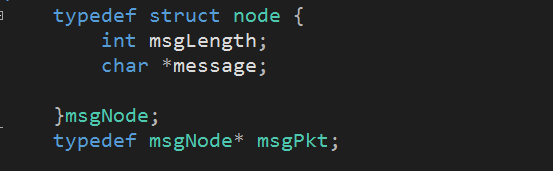


State表示当前session对应的用户是否已经登录

MessageState表示当前是否已经收到一个完整的数据包

messagePkt为一个数据队列，暂存接到的数据

msgNode节点设计如下，存放一个数据长度，存放一个指向消息数据的指针



**具体流程如下：**

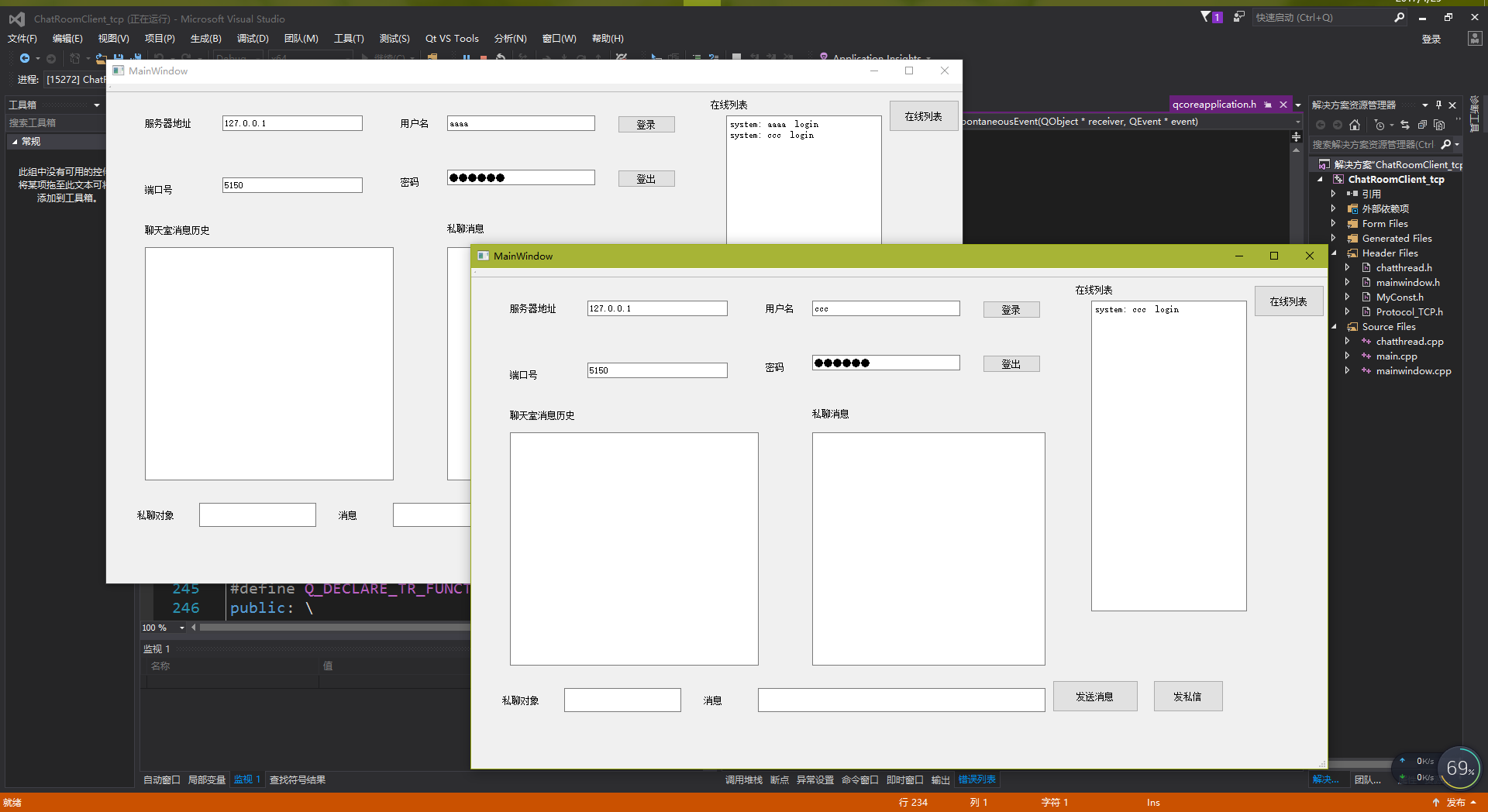
-->主线程创建服务端套接字，开始监听

-->开启另一个线程，在该线程中将服务端套接字描述符加入select集-->进行select操作 ，，， 同时主线程中循环扫描是否有用户的session接收到完整的消息数据包

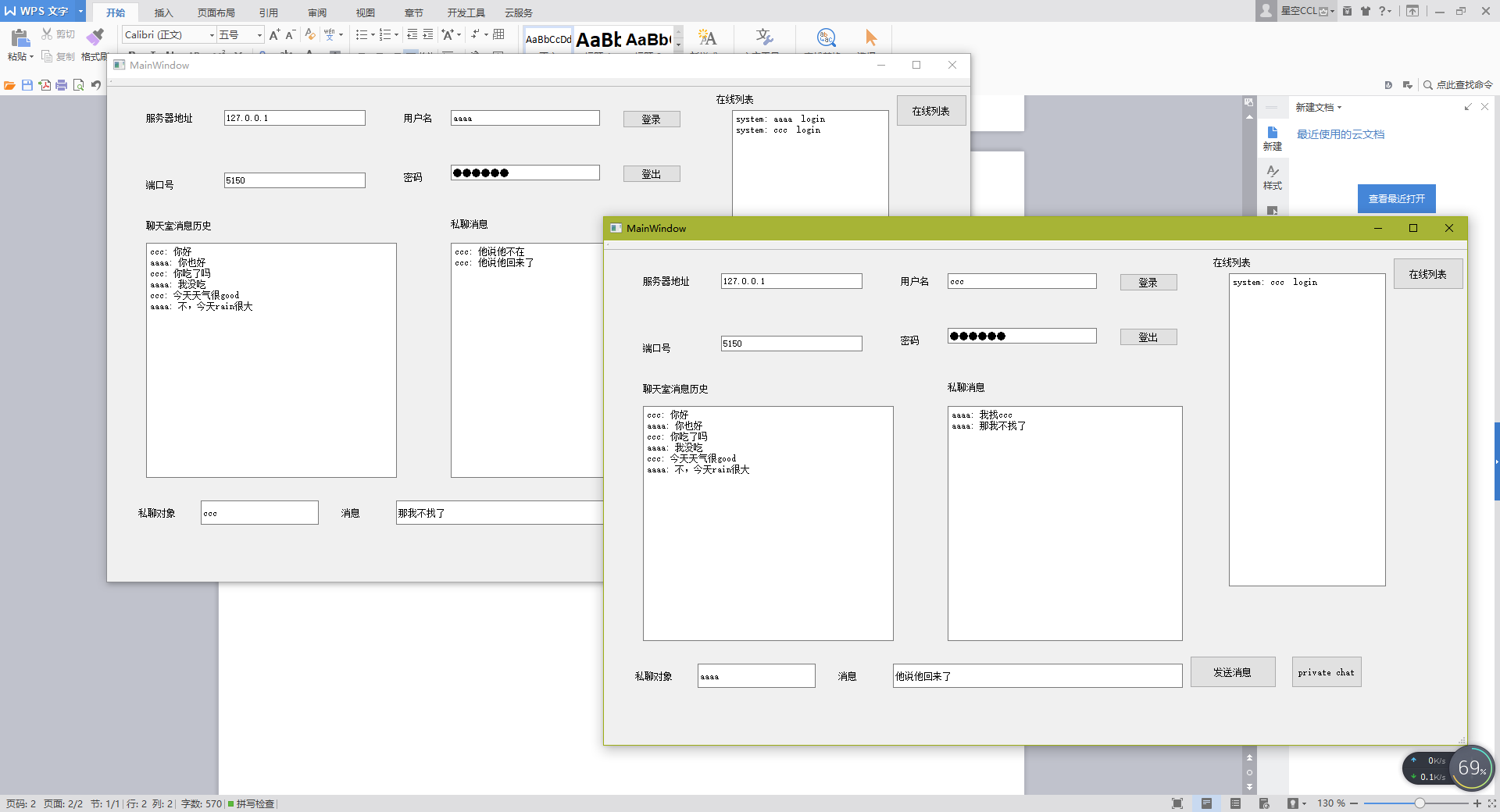
-->监听到文件描述符发生变化后，如果有新连接，将新的文件描述符加入select集中，并加一个未登录状态的session，当接收到该session对应的文件描述符发生变化后接受登录消息，向改session补充完整信息，并将接收到的完整数据信息包放入对应用户的session中，循环执行该操作

-->主线程中发现有完整数据包之后，使用processPkt函数处理对应消息包，处理完对应消息后，消息删除，消息状态设为无消息，，然后循环执行该操作

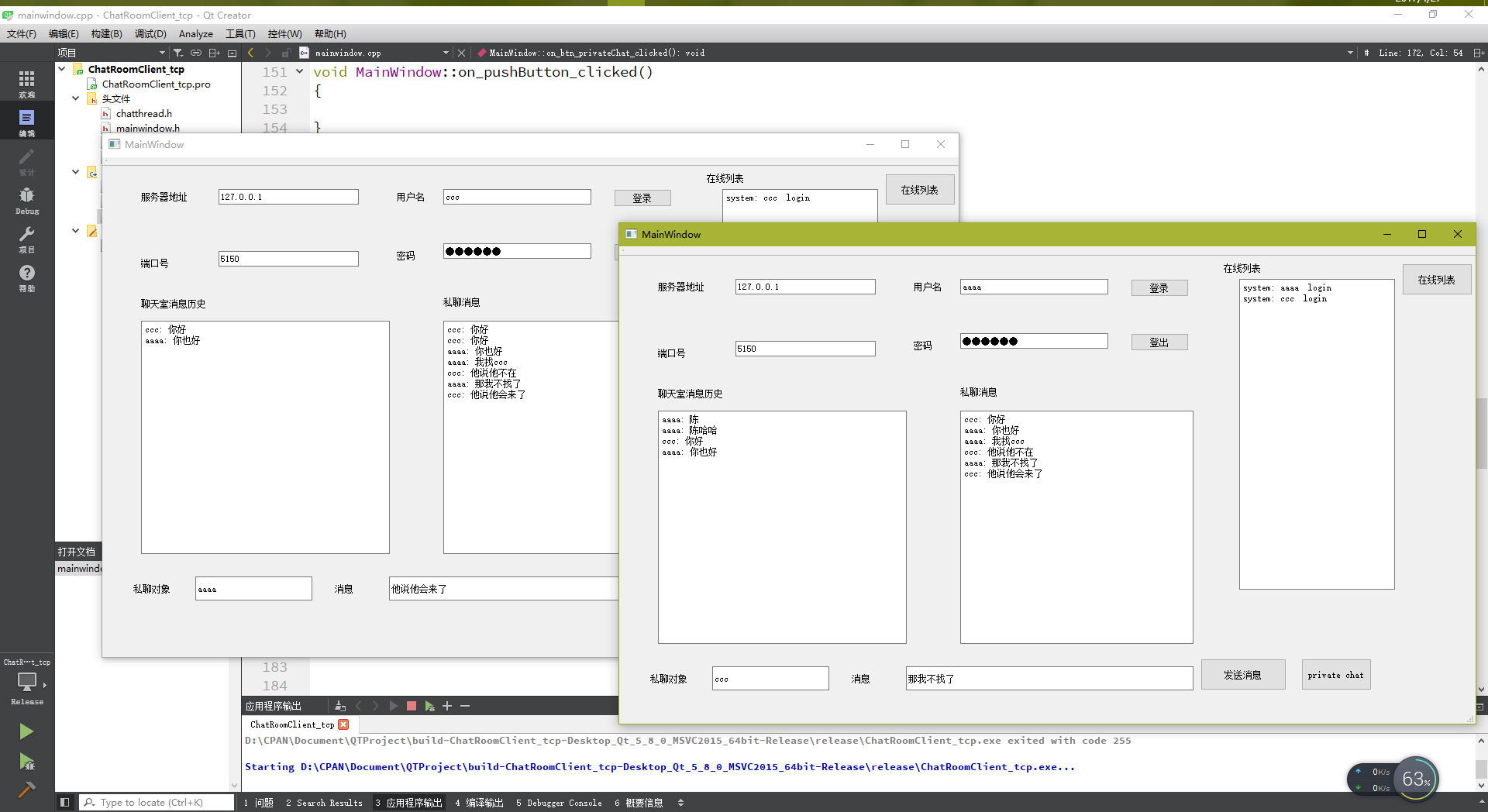
**运行图如下:**



Ccc和aaaa登录了



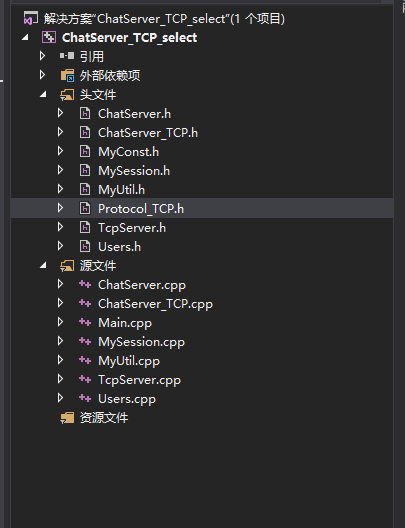
Ccc和aaaa在公共聊天室聊天，并和对方进行了私聊



Ccc和aaaa在公共聊天室聊天，并和对方进行了私聊

项目结构一撇

服务端：



客户端：

