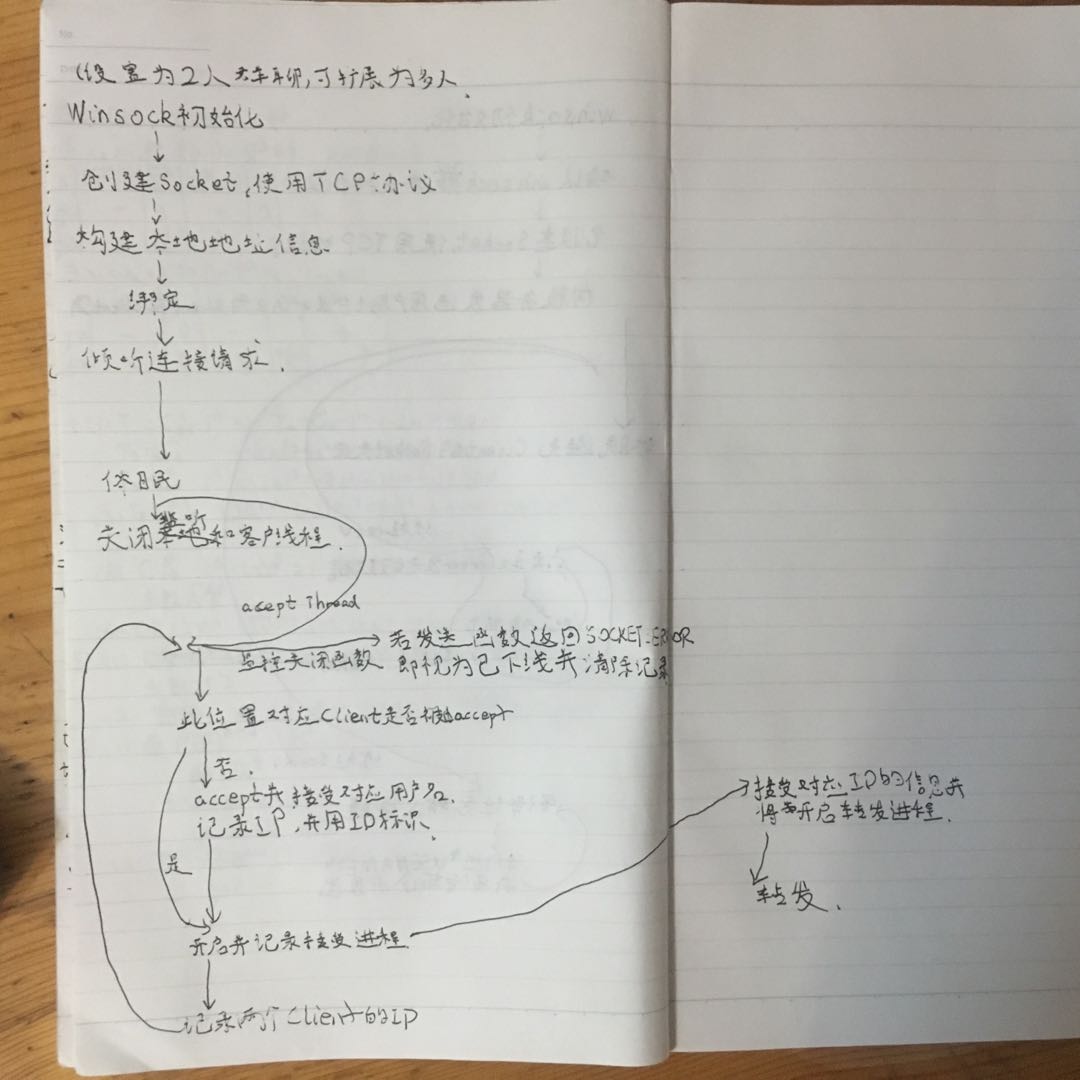
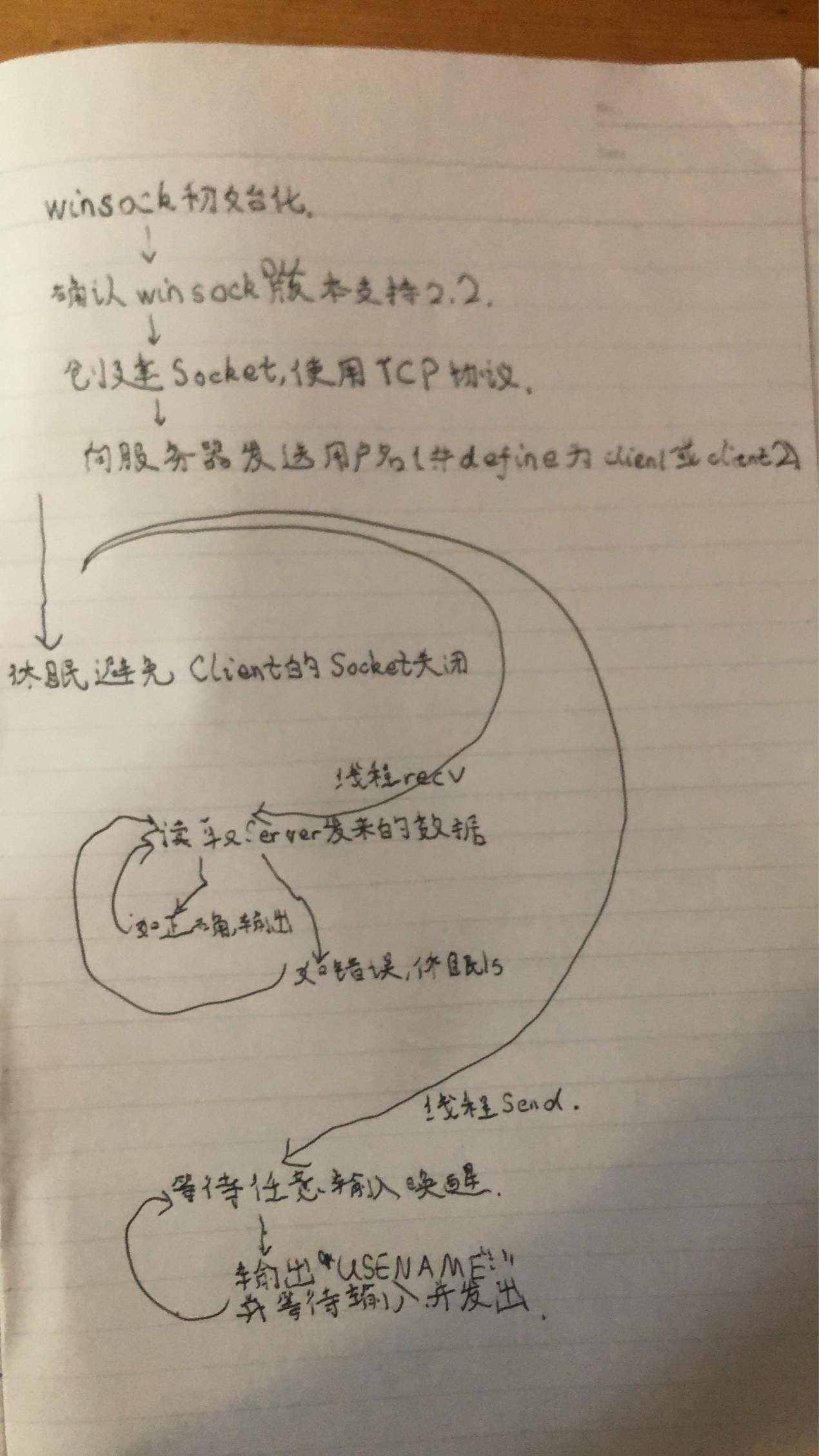
程序流程图

Server:



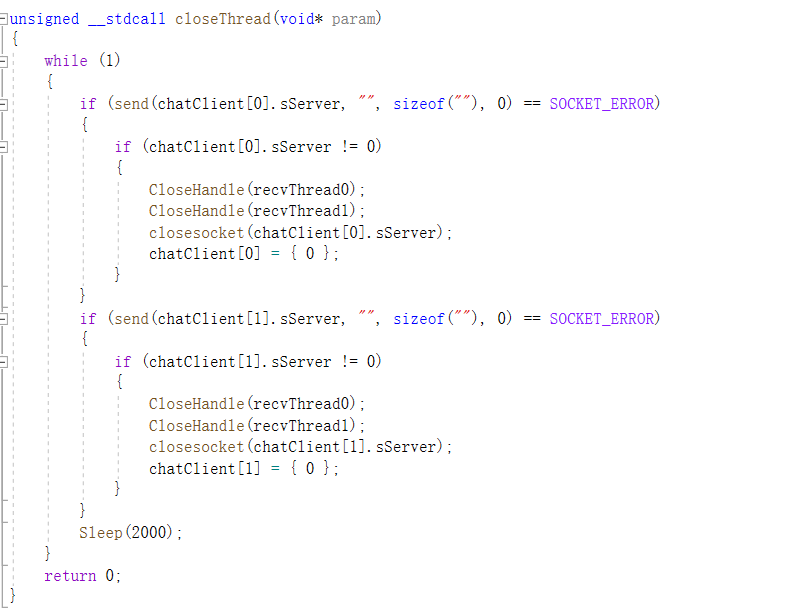
Client:



关键函数说明：

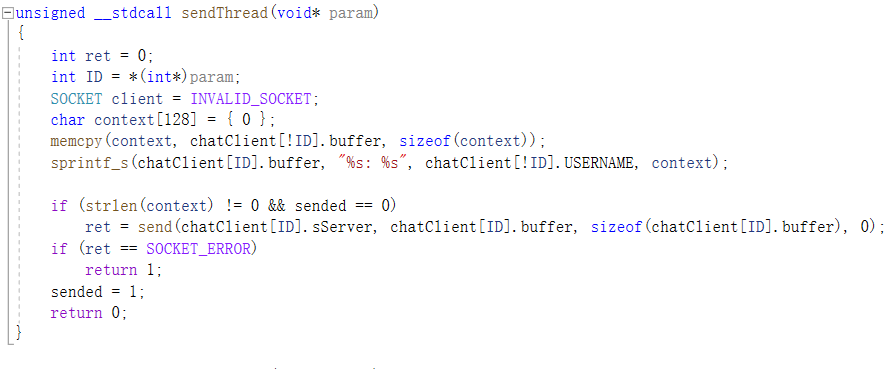
Server：

通过send返回值判断socket是否有效，如果无效就关闭会话的线程并关闭socket

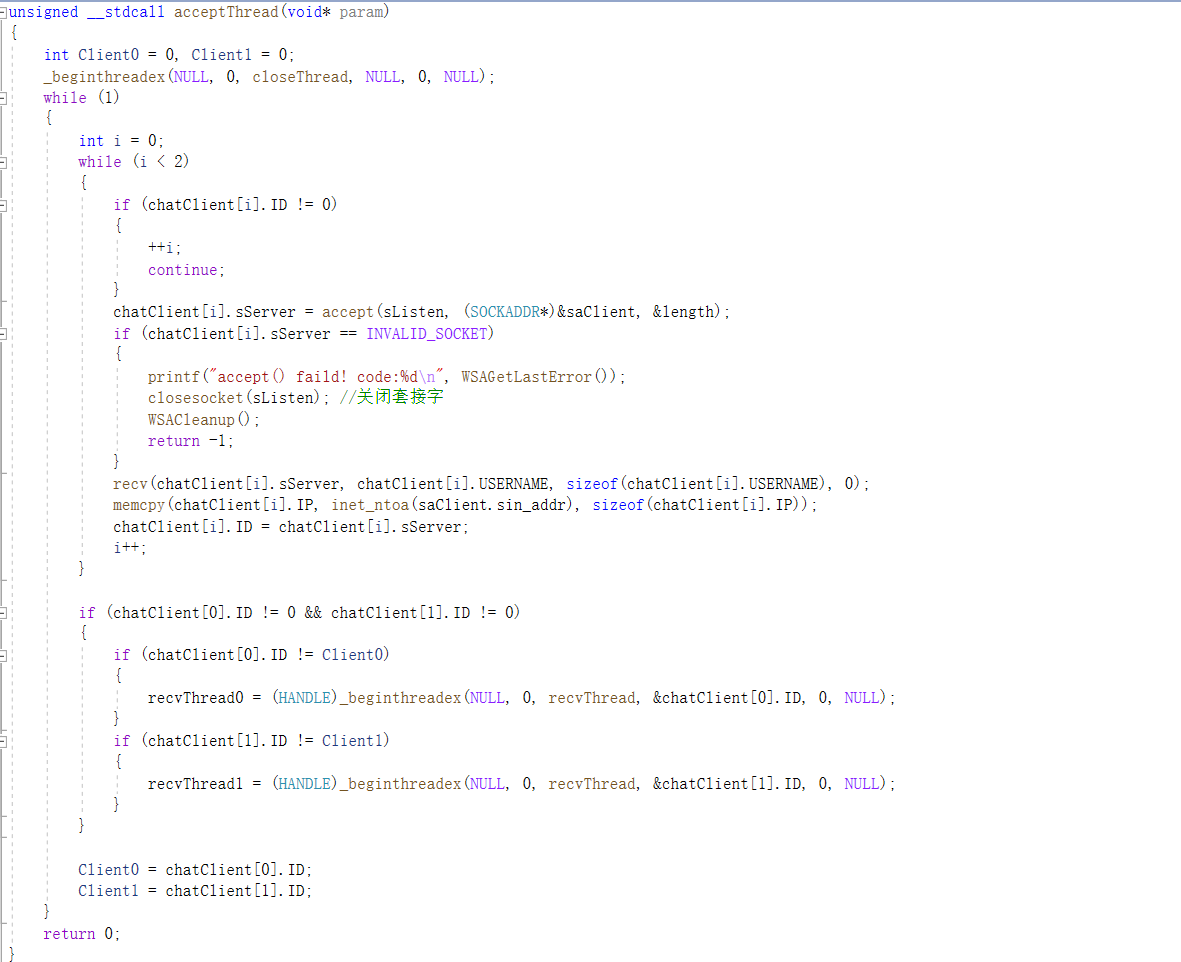


接受进程，if语句判断此接受进程对应的发送socket，ID表示接收的client。接受以后sended置0表示此时发送的还没有被转发。



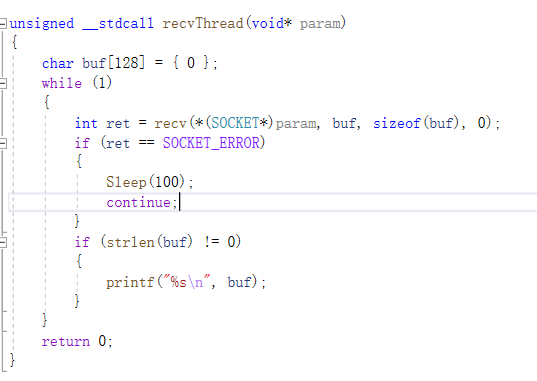


发送进程，仅当sended为0时候转发，发送以后将sended置1，表示已经转发

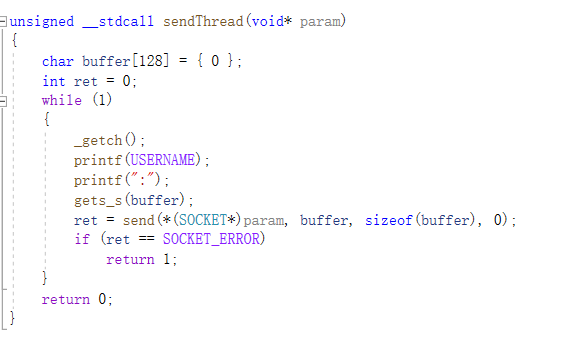


如果还有一个结构体没有赋值（chatClient[0].ID != 0）就尝试该结构体是否可以接收client的请求并且创建客户对应的socket，记录名字、ip地址，用id去标识一个client，如果两个client都准备好则开始接受转发信息。Client0、client1记录上一个循环id信息，避免接受转发进程多次开启。

Client：



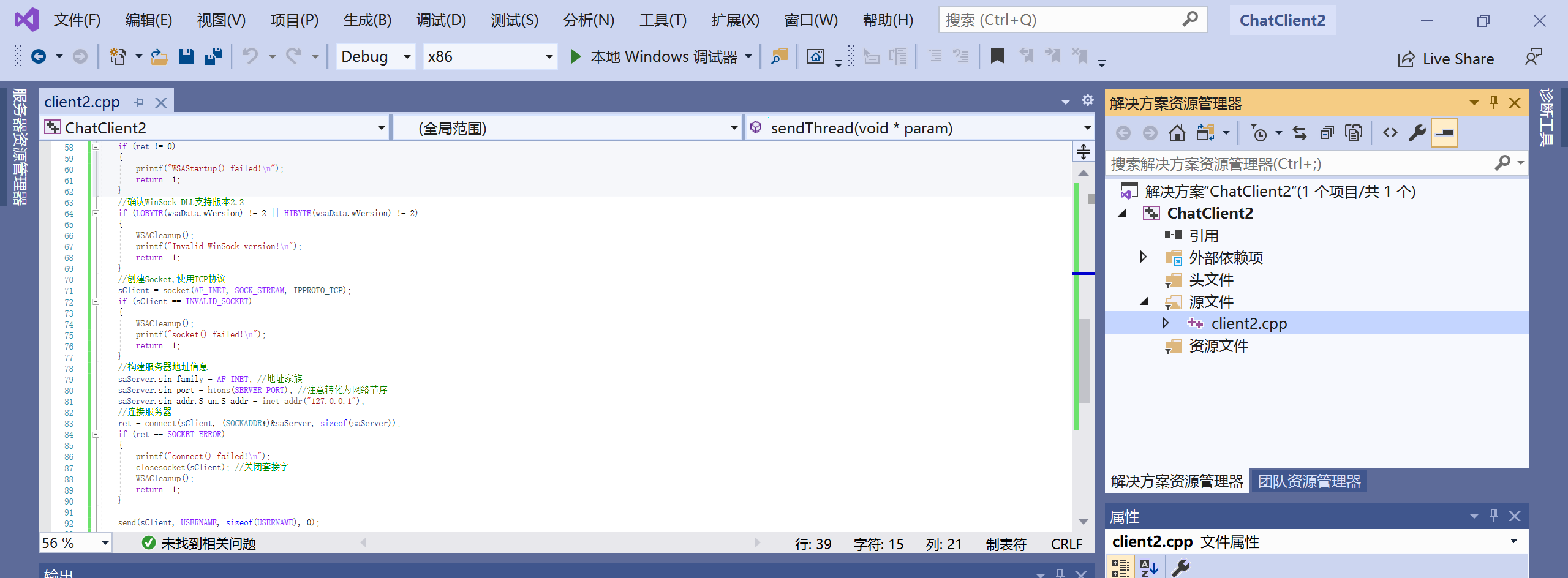
接受输入的函数，如果输入出现问题就休眠1s，如果没有问题就输出



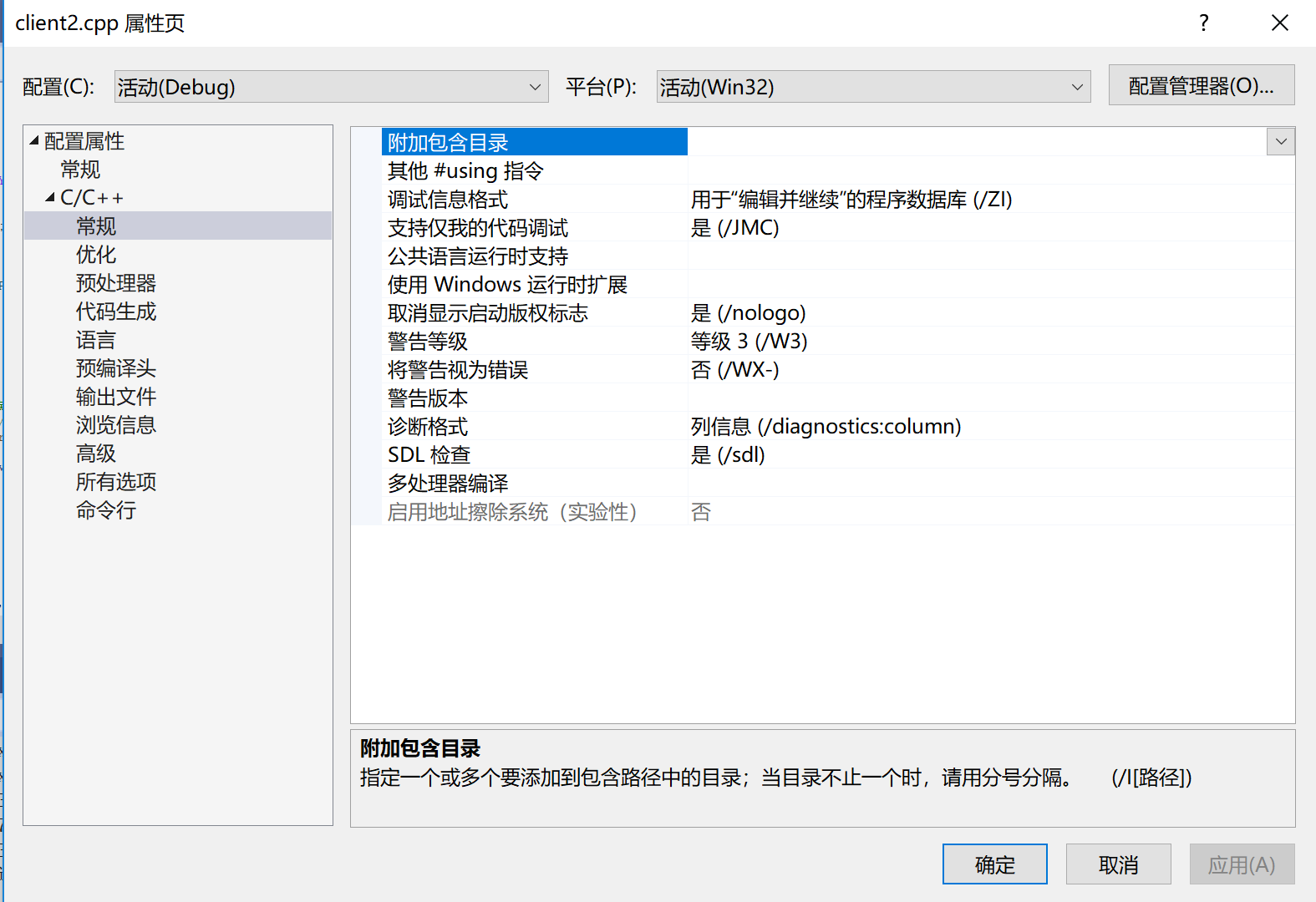
Send函数之中，\_getch是用来唤醒出入的（否则可能出现username后面是别人的username和信息的情况）

从terminal中接收到输入之后就发出

操作流程：



由于我未能找到vs2019产生的exe文件，就直接将控制台提交了，编译前右键源文件如client2.cpp，选择属性。



常规选项之中把sdl检查置为否。

需先编译运行（点击本地windows调试器服务器）

然后编译运行两个client，在client每次输入之前，需先点击任意键，出现“client1：”或者“client1：”之后便可以输入，输入之后按回车，便可以看到对方client接受到消息。

（因为对进程不熟悉，如果关闭了client再开启，将无法正常发送，此时需将所有端口关闭重新按照上面顺序编译）

运行结果：