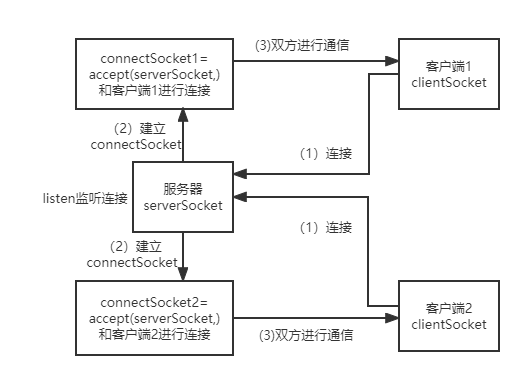
多人聊天室

1. 协议设计

本次实验设计了一个多人聊天室，首先介绍多人聊天室的协议设计：多人聊天室协议的持方是一个服务器端和多个客户端。每个客户端都与服务器端建立连接，客户端与客户端之间不进行连接。当某个客户端进入或者退出聊天室时，服务器端发送消息告知所有在聊天室的用户。客户发送带有时间标签的消息给服务器端后，服务器会处理该消息，并转播发送给聊天室中的所有客户端，以此达到聊天室的目的。

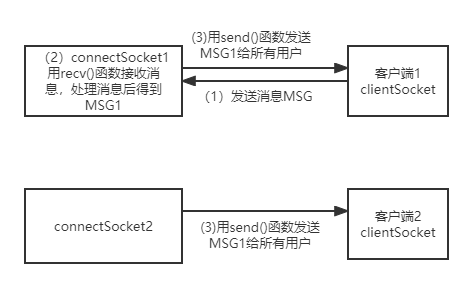
（一）服务器和客户端建立连接流程：

服务器的serverSocket通过listen()函数进行监听，客户端clientSocket发送连接请求，当listen()函数检测到有连接事件发生时，serverSocket用accept()函数建立对应的connectSocket，之后利用conectSocket和客户端进行消息的发送和接收，同时也用conectSocket检测客户端是否断开连接。



1. 客户端发送消息流程：

客户端发送消息给服务器后，服务器进行广播，让聊天室内所有用户看到。当客户端1发送消息给connectSocket1后，connectSocket1用recv()函数接收消息，经过消息处理后，让所有已经与客户端连接的connectSocket利用send()函数发送处理后的消息给对应客户端，完成服务器消息播报的功能。



（三）协议设计具体细节：

1、交互消息的类型：ASCII类型，发送和接收char类型的字符数组

2、消息的语法和语义

客户端发送消息的语法以及对应的语义：

1. 传递以字符串”name”开头的消息：客户端向服务器连接后报上用户的名称。
2. 传递以当前时间开头的消息： 客户端向服务器发送聊天内容。

服务器转播消息的语法以及对应的语义：

1. “[用户名]退出聊天室”：向所有在线的用户转播某位用户退出了聊天室，用户名是用户自拟的名称。
2. “欢迎[用户名]进入聊天室”：向所有用户转播某位用户进入了聊天室。
3. “[用户名]当前时间 消息内容”：接收到某位用户的以时间开头的聊天内容消息后，向所有用户以上面字符串的形式进行转播。

服务器接收的消息的语法以及对应的语义：

1. 以字符串”name”开头的消息：客户端向服务器连接后报上的用户的名称。
2. 以客户端发送时间开头的消息：客户端向服务器发送的带有时间标签的聊天内容。

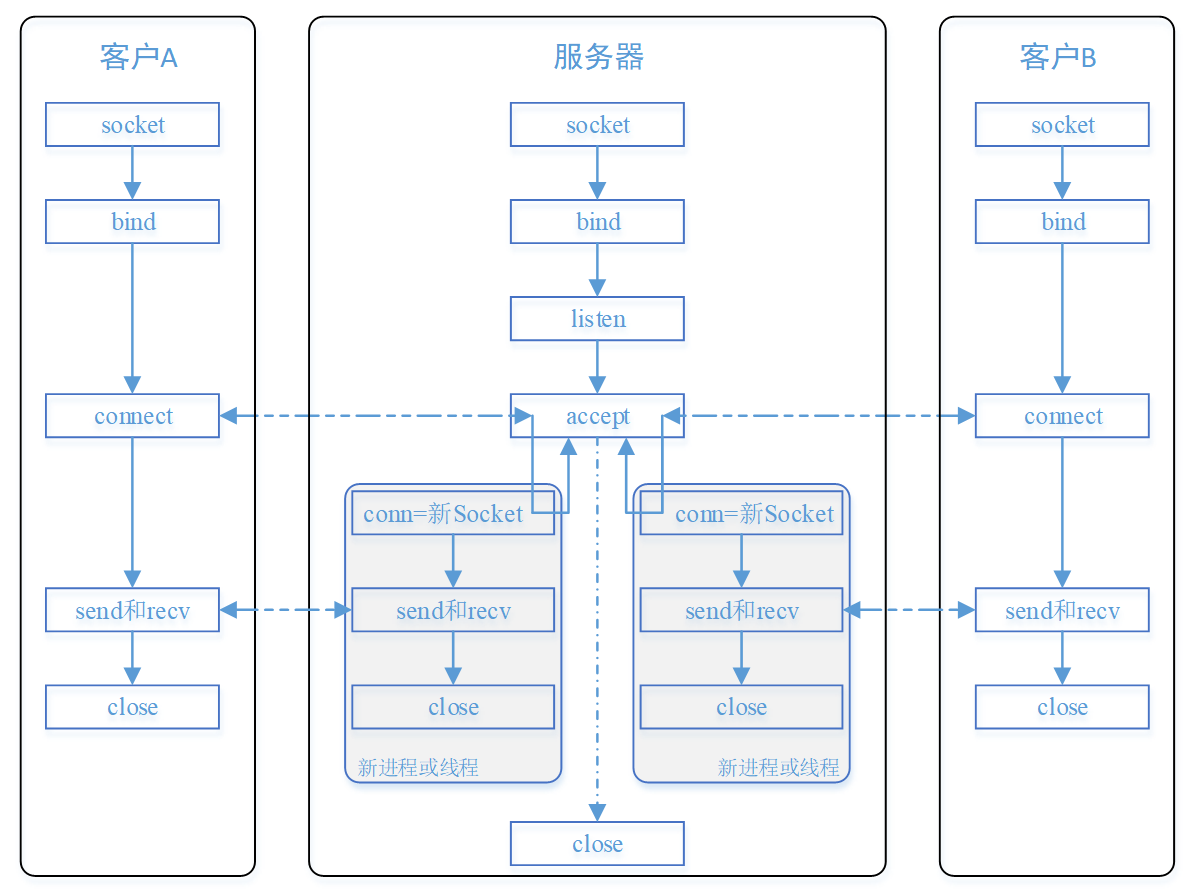
3、消息的具体处理方式：

客户端消息：设置两个线程分别进行发送消息和接收消息。对于发送的消息，即时处理，当检测到用户输入结束后，立即处理缓冲区字符串并发送消息；对于接收的消息，即时处理，以字符串的形式呈现给该客户端。

服务器消息：对每个连接进来的客户端都创建一个线程，使用多线程进行监听，为每个连接进聊天室的客户端分配一个线程。在每个线程中，对于客户端关闭socket的消息，服务器关闭与之连接的socket，并立即转播告知给所有用户；对于客户端发来的以”name”字符串开头的消息，将”name”之后的字符串设置为该客户端的用户名；对于客户端发来的带有时间标签的消息，立即处理，在每条消息前加上”[用户名]”字符串后，转播给所有用户。

1. 程序设计和模块功能展示

具体程序分为两个c++文件实现，分别是服务器和客户端，采用TCP协议的传输层设计，主要的服务器和客户端实现具体流程如下图所示：



1. 服务器

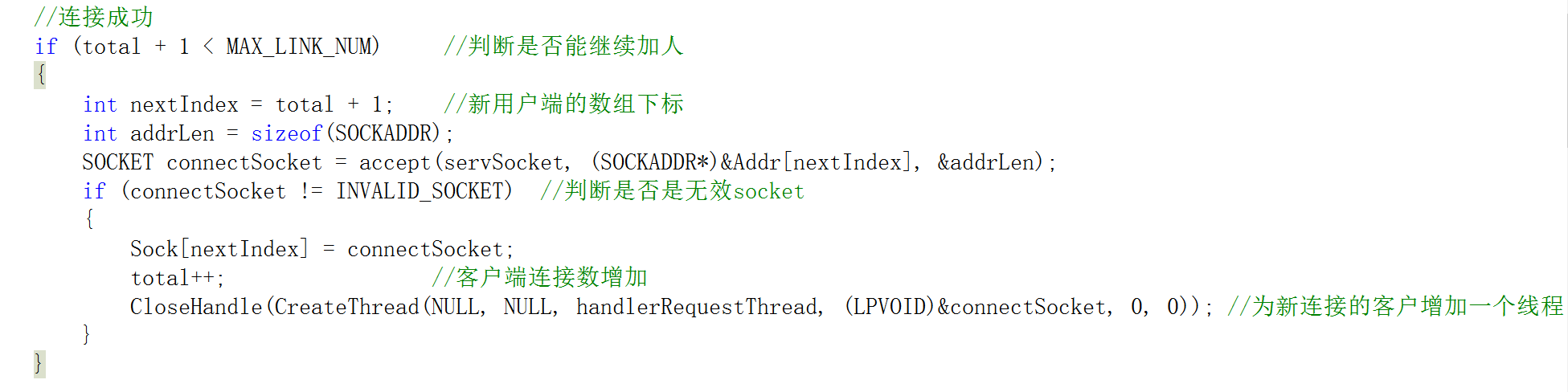
具体设计流程如下：

1. 说明使用Socket的版本信息。
2. 创建服务器Socket为servSocket。
3. 将服务器地址（IP和端口号等）封装在SOCKADDR\_IN类中，绑定服务器Socket和服务器地址。创建事件对象，绑定事件对象和服务器Socket。
4. 用listen()函数监听连接。使用while循环，确保可以连接多个客户端。当有连接事件发生时，利用accept()函数创建一个新的connectSocket，用来和客户端进行通信，同时调用线程函数，创建一个新的线程，传入参数connectSocket。
5. 在每个客户端对应的线程中，利用它们各自的connectSocket和客户端进行通信，并为每个connectSocket绑定一个事件，检测接收消息和断开连接。如果是消息事件，可以利用send()函数和recv()函数进行发送和接收消息。
6. 如果connectSocket绑定的事件响应为断开连接事件，则关闭对应的connectSocket和绑定的事件。

连接事件：

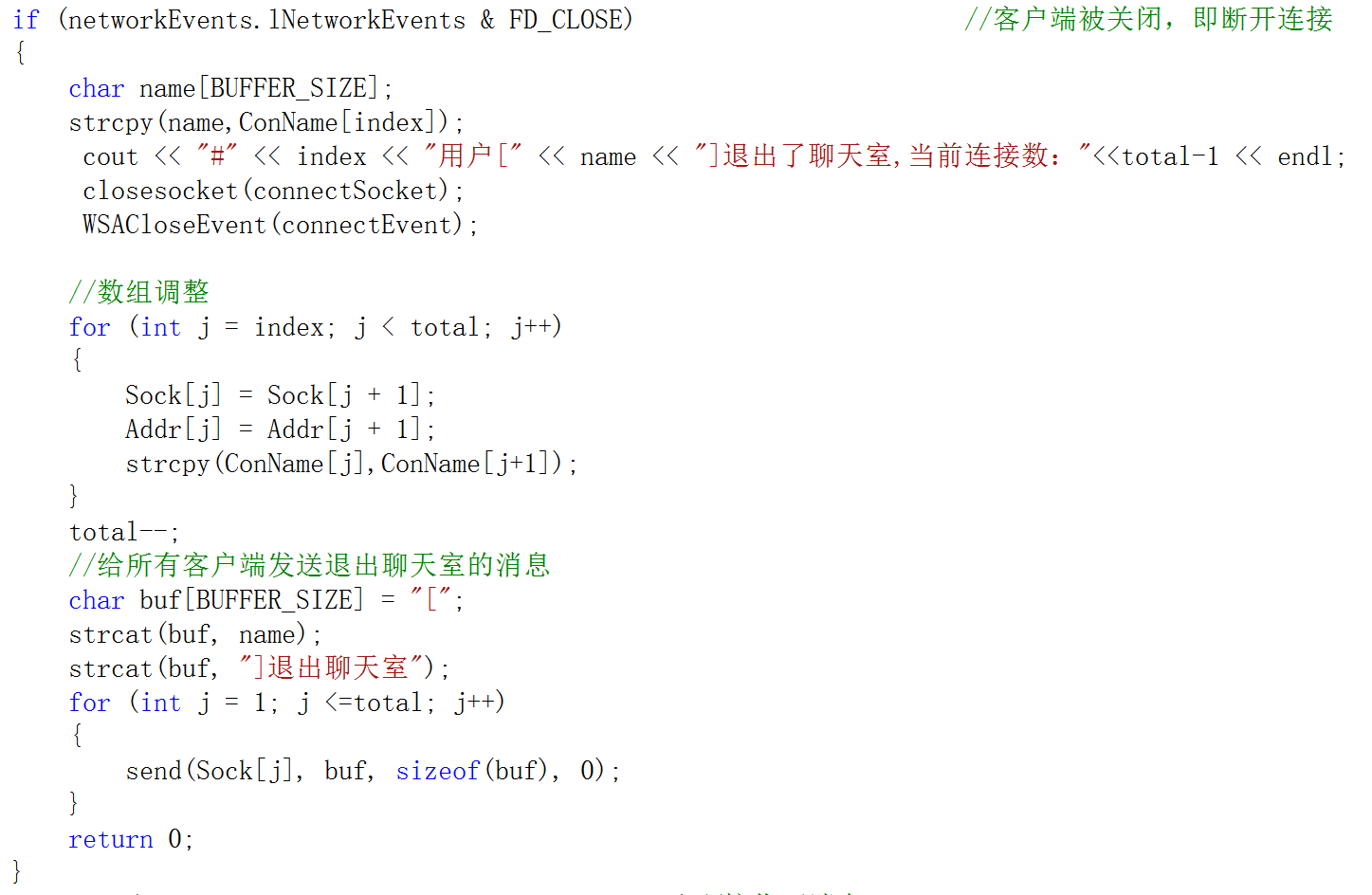
为了方便管理连接的客户端，维护一个Sock数组来存储客户端的Socket，维护一个Addr数组来存储连接进来客户端的地址类（SOCKADDR\_IN）对象，维护ConName二维字符数组来存储客户端的名称。同时，在服务器端维护一个int型total变量来标明目前连接服务器的客户端数量。

利用while不停地循环来检测是否有连接事件的发生。当有连接事件发生时，先判断客户端数量是否达到最大值，如果小于最大值，服务器利用accept()函数，传入服务器的servSocket、接收连接的地址类对象和地址类（SOCKADDR）长度作为参数，来创建一个connectSocket。并将connectSocket存入Sock数组，地址也存入Addr数组，为该客户端创建一个线程函数，来接收该客户端发送的消息，并且检测该客户端是否断连。下图展示了连接模块的代码：



断开连接：

在为客户端创建的线程中，实现了判断该客户端是否断连，以及接收客户端发送的消息的功能。如果发生客户端被关闭的事件，在服务器形成字符串提示，在服务器这边关闭对应的connectSocket，同时关闭绑定的connectEvent事件，对三个维护的数组进行调整，当前客户端数量减1，并用for循环利用Sock数组，向目前在线的所有客户端播报该用户退出的消息。下图展示了断开连接模块的代码：



接收和播报消息：

接收消息功能同样在线程函数中实现。当判断接收消息的字符数大于0后，进入两个分支。如果消息以”name”字符串开头，表明有新的客户端进入并报上自己的名称，将该用户的名称存入ConName数组，并向所有用户播报该用户进入聊天室。否则，对客户端传来的消息前加上”[用户名]”字符串后，播报给所有用户。接收和播报消息模块的代码如下：



1. 客户端

客户端的总体设计分为主程序线程和一个新创建的线程：在主线程中利用一个while循环不停的接受用户键盘的输入，并发送消息给服务器；在线程函数中利用一个while循环，监听从服务器传来的消息。

客户端的具体流程也大致分为：说明Socket版本信息、创建客户端cliSocket、封装服务器和客户端地址（但cliSocket不用绑定地址）、利用connect()函数传入服务器地址来和服务器建立连接、之后在连个线程中进行发送消息给服务器和接收服务器的消息、输入”886”退出聊天室或直接关闭窗口强制退出。

发送消息模块

在和服务器连接成功以后，进入while循环。根据提示先输入”name用户名”字符串并用send()函数发送该消息给服务器，表示名称。如果输入聊天消息，在消息前加上当前时间的标签，以字符串的形式发送给服务器。如果输入”886”直接退出循环，之后关闭cliSocket，并用WSACleanup()函数清理释放Socket库占用的资源。



接收消息模块

在接收消息的线程中，进入一个while循环来不停的接收来自服务器的消息。判断一下接收的字符数量，大于0说明接收消息正常，在命令行直接显示消息；小于0则说明可能遇到异常，和服务器断开连接。

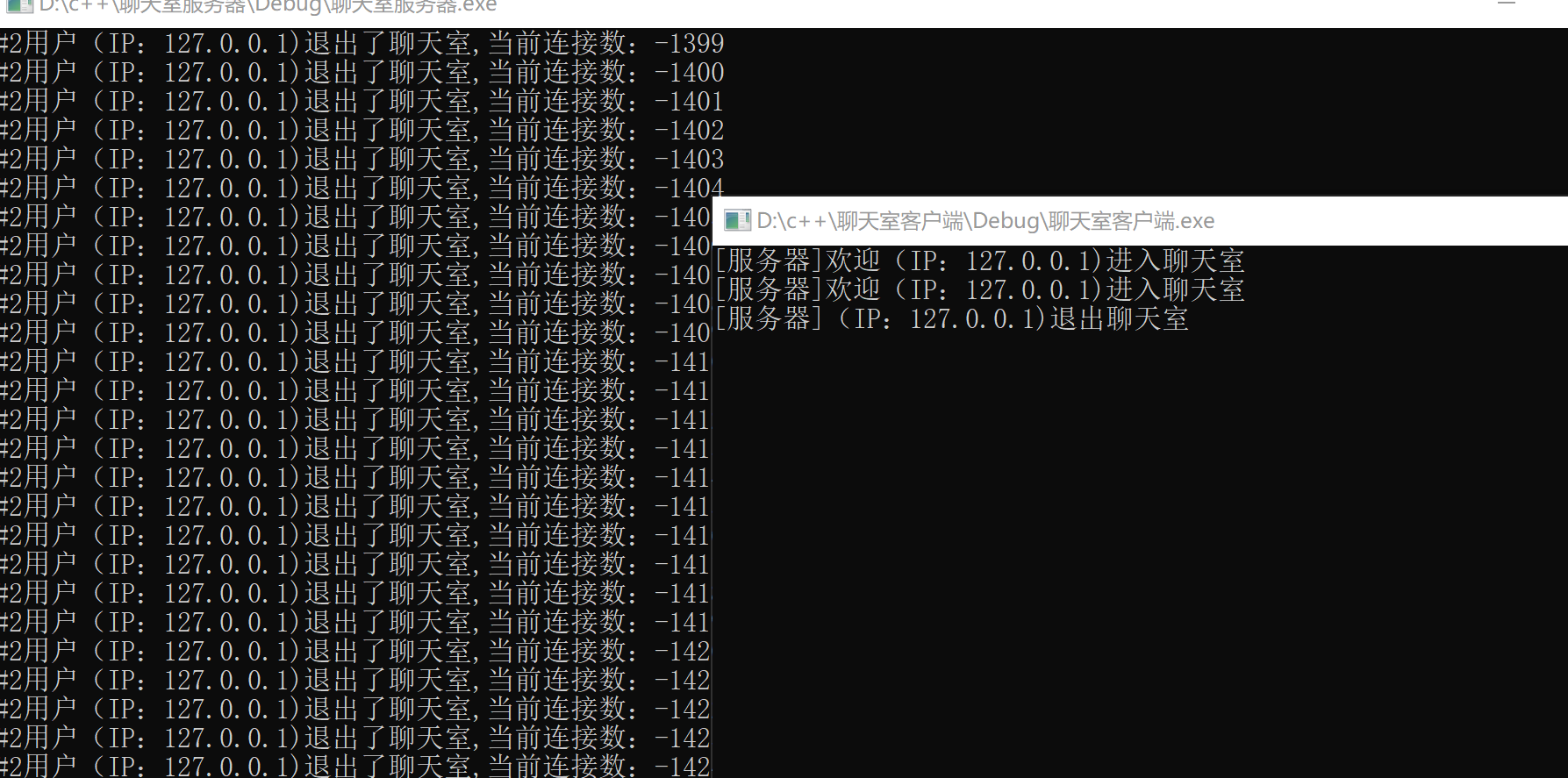


三、遇到的问题以及解决方法

1、在聊天室服务器中，直接使用accept接收连接，有时会发生连接错误。

解决办法：给socket绑定事件，并使用WSANETWORKEVENTS来进行进行发生事件的判断，如果是连接事件，再进一步判断是否连接成功。

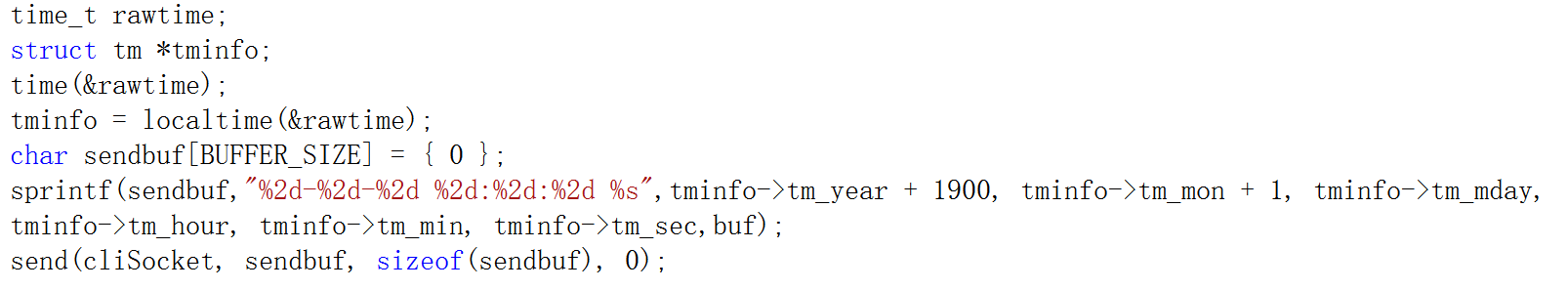
1. 如果有用户退出聊天室，服务器端会一直循环处理断开连接事件。



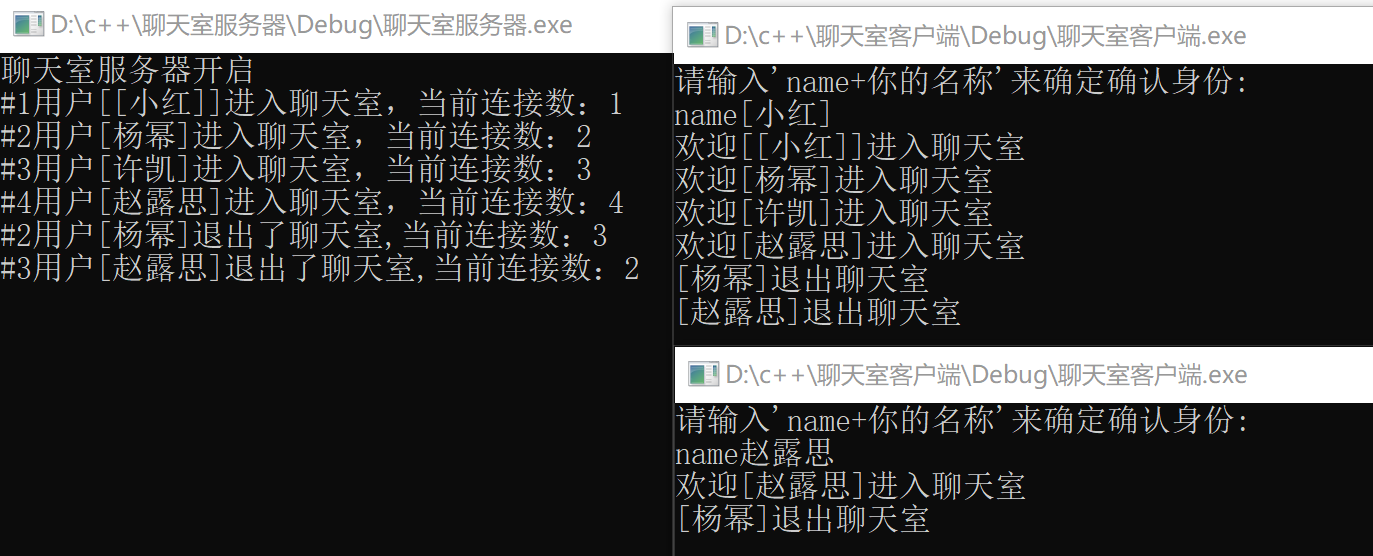
解决办法：在处理断开连接的函数执行完毕后，直接返回，结束该线程。

1. 针对需要添加时间标签，解决办法如下：使用time(time\_t \*)和localtime(time\_t \*函数)，以及用到结构体tm，可以看下表找到tm对应的成员变量。然后添加相应的事件信息到消息中即可。

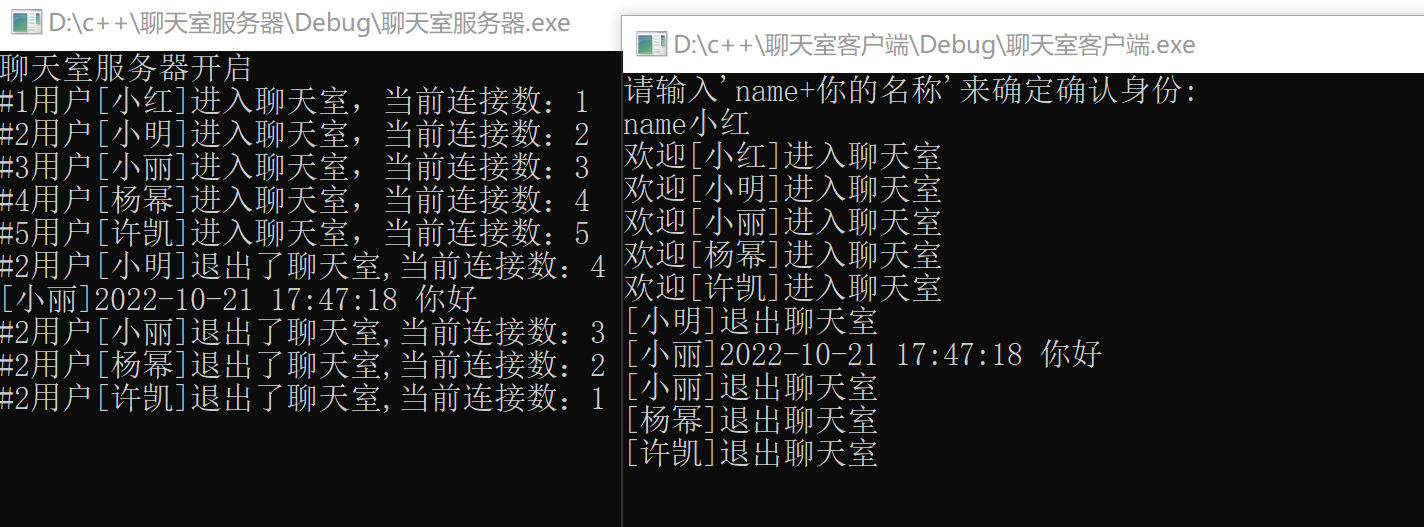




1. 在用户退出聊天室时，由于各线程之间时并行的互不干扰，且在线程中用户的序号是固定不变的的，但实际数组调整后被删除之后的用户序号也应该改变，因此会造成数组维护时发生错误，表现出来的是在服务器显示退出时名字错误。



解决办法：在while循环定义名称数组，保证用户名称在线程中保持不变。每次while循环，都由姓名判断并更新一下序号，来保证序号的正确性，数组维护也正确。结果如下：



四、程序使用方法以及结果展示

使用方法：在相同或不同的电脑中，先打开聊天室服务器.exe文件，之后再打开聊天室客户端.exe文件，其中限制客户端的数量不超过5个。

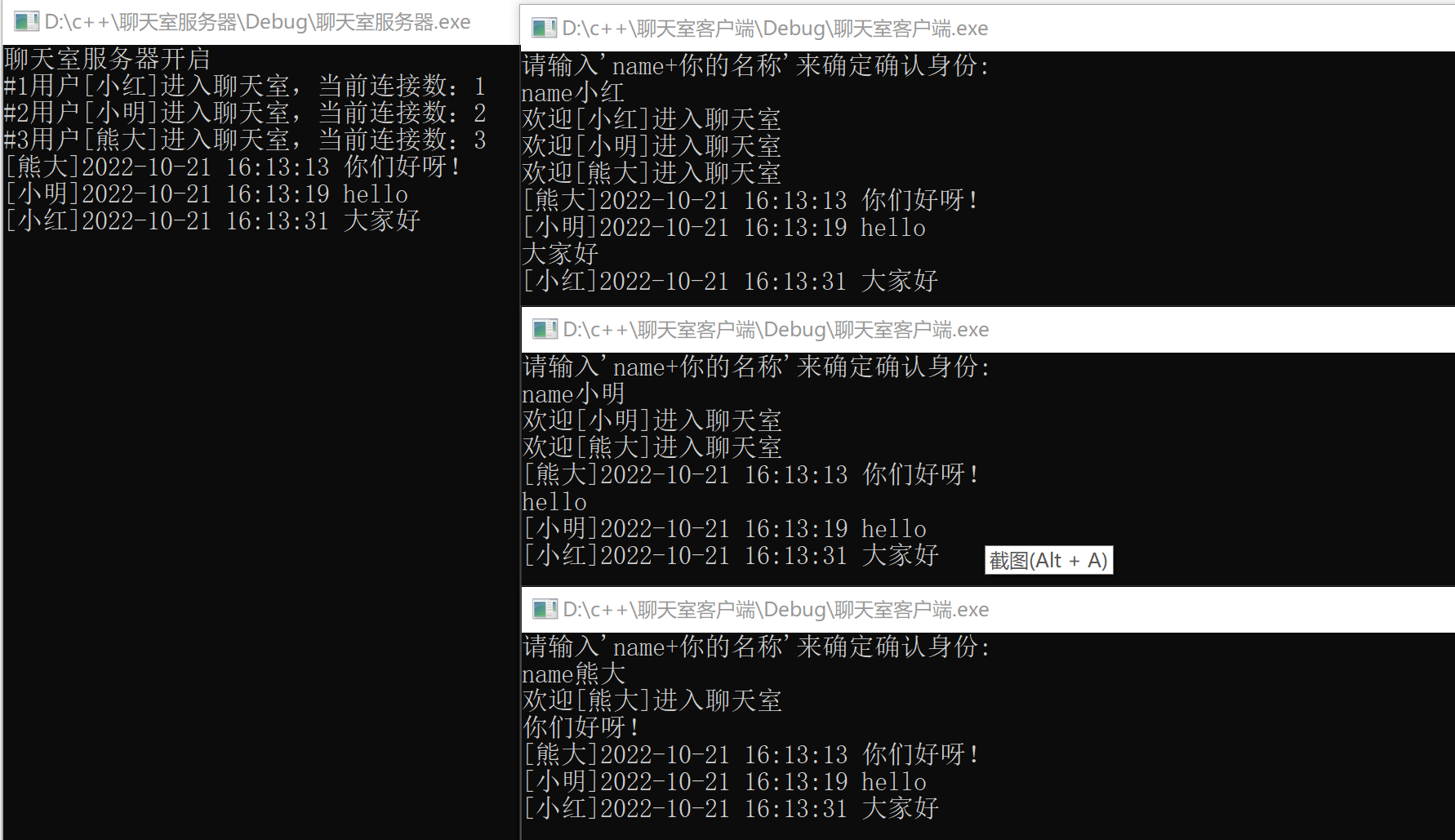
在同学的电脑上测试时有些打不开，出现了缺少MSVCP100D.dll文件的提示，将该文件下载并放进电脑系统磁盘中即可。

结果展示：

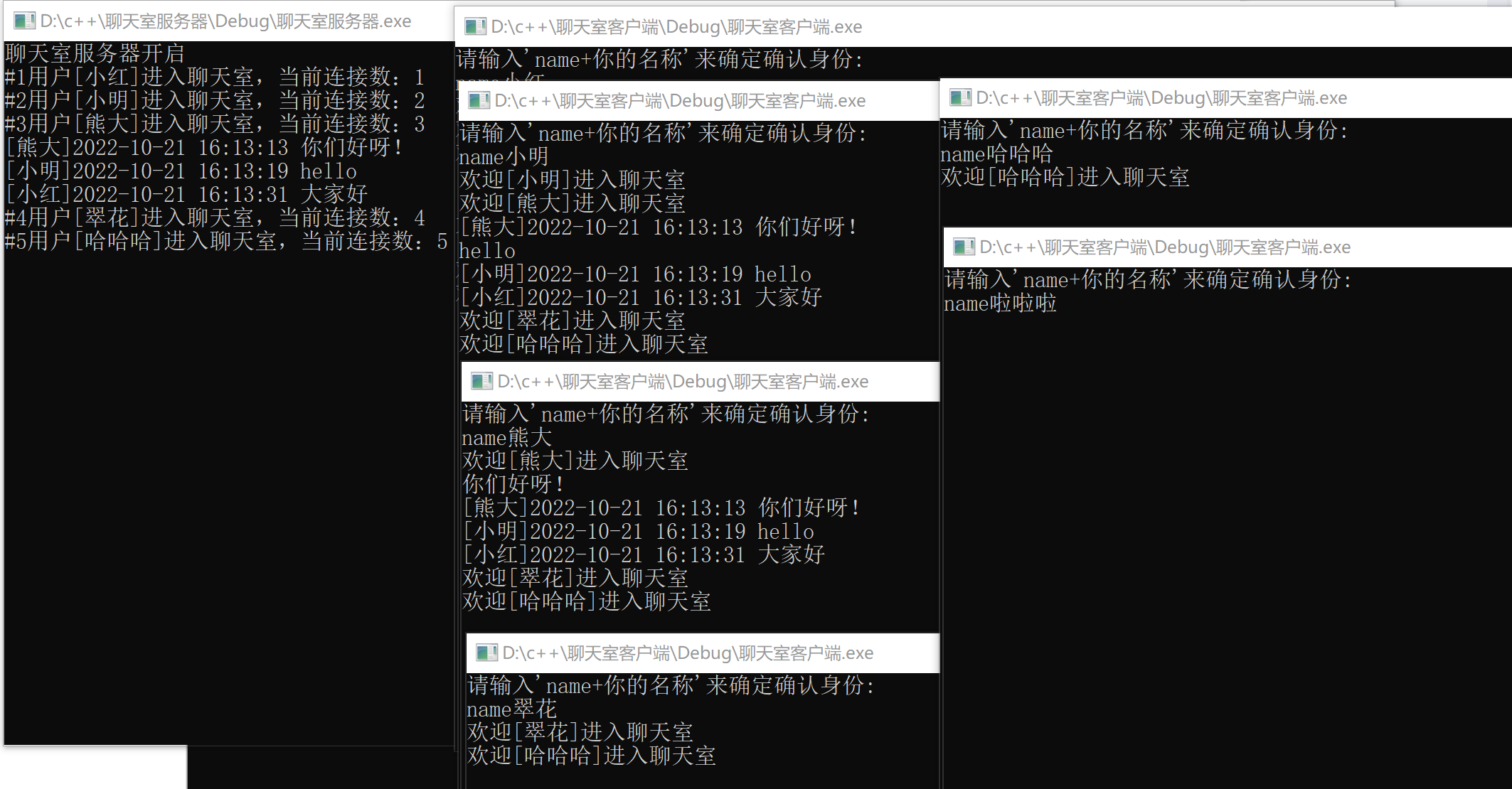
测试连接，连接正常进行。



接收发送消息，消息都能正常发送和接收，同时也都标有时间标签。



测试最大连接数，我们定义最大连接数为5，当打开第6个客户端不能和服务器连接，发消息服务器也接收不到。



断开连接测试，在遇到的问题和解决方法这章第4个问题描述，修改bug后，可以正常退出。

