Server实现具细：

1.server.cpp和server.h目的就是建立服务器的。其中server.h中定义了一些

**属性变量**，包括窗口句柄，定义的端口号，以及套接字类，以及sockADDR\_in（调用bind函数时需要这个参数，包括地址族（TCP-IP，IPv4），端口号，以及INADDR\_ANY（意为不管数据是从哪个网卡过来的，只要是绑定的端口号过来的数据，都可以接收））

**成员函数**：只有一个成员函数buildserver（即建立服务器，会执行初始化socket，调用bind函数，调用listen函数，此时服务器已经就绪）

而server.cpp实际上就是对成员函数的具体实现。

2.chatserver.h和chatserverDlg.cpp的目的实际上就是与客户端在进行交互了。其中chatserver.h中定义了一些：

**属性变量**：包括一个server类，一个用户socket缓冲队列client\_socket，一个用户ip地址缓冲队列client\_addr,已经连接到服务器端的用户个数client\_num，消息具细recvDetails

**成员函数**：这里一定是先调用server类中的buildserver函数先建立服务器的。然后才有之后的send/recv循环。

对于send/recv循环，我使用的是winsock库函数中的WSAAsyncSelect函数（在一个 socket 上接收以 windows 消息为基础的网络事件。提供读写数据的异步通知功能，当有相应的网络事件时，进入相应的case并处理。）server类可能收到的网络事件有accept、read，send，close。

其中accpet会接受到用户的socket类，并存到缓冲队列中，以及sockaddr\_in，并存到缓冲队列中。

其中read/send会先找到该用户的套接字，并将接收到的信息放到缓存区buf中（即read）。最后再遍历socket缓冲队列，将buf存储的信息转发到各用户端（即send）。

其中close用于更新socket缓冲队列和sockaddr\_in缓冲队列，以及更新界面上的人数等。

而chatserverDlg.cpp实际上就是对成员函数的具体实现。

Client实现具细：

1.和server类似，client.h和client.cpp目的是建立客户端，其中client.h中定义了一些

**属性变量**：包括窗口句柄，服务器的端口号，以及套接字类，以及sockADDR\_in（需要填写地址族，服务器的端口号以及服务器ip地址）

**成员函数**：只有一个成员函数Initclient（即建立客户端，会执行初始化socket，调用connect函数，此时客户端已经准备连接）

而client.cpp实际上就是对成员函数的具体实现。

2. chatclient.h和chatclientDlg.cpp的目的实际上就是与服务器端在进行交互了。其中chatclient.h中定义了一些：

**属性变量**：用户名name，服务器端口号port，服务器ip地址，消息内容具细recvDetails，发送内容具细sendDetails

调用相应的send函数即可发送信息。

而接受信息的话，也是采用WSAAsyncSelect函数，client类可能收到的网络事件有connect、read/recv，更新相应的recvDetails即可。

大概就这么多啦。。。。。。

服务器是被动的，客户端是主动的