计算机网络

**课程设计报告**

题目： 网络聊天程序的设计与实现

**姓名： 朱政雄**

**学号： 1305010211**

**班级： 13计算机二班**

**指导老师： 姜磊**

**湖南科技大学计算机科学与工程学院**

**2016年 1 月**

1. **实验题目**

网络聊天程序的设计与实现

1. **实验内容**

1、了解TCP/IP协议与WinSock网络编程接口的关系

2、了解TCP/IP协议

3、熟悉WinSock编程简单流程

4、掌握VC中socket编程步骤

5、完成基本网络聊天程序

1. **实验步骤**

1、阅读课程设计指导书，了解TCP/IP协议与WinSock网络编程接口的关系

2、阅读课程设计指导书，了解TCP/IP协议

3、通过课程设计指导书，熟悉WinSock编程简单流程

4、通过课程设计指导书，熟悉掌握VC中socket编程步骤

5、通过课程设计指导书，编写出基本的客户端与服务端Winsock的基本程序

6、根据基本程序及实验目的的需求来修改基本程序，实现实验网络聊天程序

1. **调试过程**

基本Winsock编程的步骤及实现

服务器端编程的步骤：

1：加载套接字库，创建套接字(WSAStartup()/socket())；

2：绑定套接字到一个IP地址和一个端口上(bind())；

3：将套接字设置为监听模式等待连接请求(listen())；

4：请求到来后，接受连接请求，返回一个新的对应于此次连接的套接字(accept())；

5：用返回的套接字和客户端进行通信(send()/recv())；

6：返回，等待另一连接请求；

7：关闭套接字，关闭加载的套接字库(closesocket()/WSACleanup())。

#include <stdio.h>

#include <Winsock2.h>

void main()

{

WORD wVersionRequested;

WSADATA wsaData;

int err;

wVersionRequested = MAKEWORD( 1, 1 );

err = WSAStartup( wVersionRequested, &wsaData );

if ( err != 0 ) {

return;

}

if ( LOBYTE( wsaData.wVersion ) != 1 || HIBYTE( wsaData.wVersion ) != 1 ) {

WSACleanup( );

return;

}

SOCKET sockSrv=socket(AF\_INET,SOCK\_STREAM,0);

SOCKADDR\_IN addrSrv;

addrSrv.sin\_addr.S\_un.S\_addr=htonl(INADDR\_ANY);

addrSrv.sin\_family=AF\_INET;

addrSrv.sin\_port=htons(6000);

bind(sockSrv,(SOCKADDR\*)&addrSrv,sizeof(SOCKADDR));

listen(sockSrv,5);

SOCKADDR\_IN addrClient;

int len=sizeof(SOCKADDR);

while(1){

SOCKET sockConn=accept(sockSrv,(SOCKADDR\*)&addrClient,&len);

char sendBuf[50];

sprintf(sendBuf,"Welcome %s to here!",inet\_ntoa(addrClient.sin\_addr));

send(sockConn,sendBuf,strlen(sendBuf)+1,0);

char recvBuf[50];

recv(sockConn,recvBuf,50,0);

printf("%s\n",recvBuf);

closesocket(sockConn);

}

}

客户端编程的步骤：

1：加载套接字库，创建套接字(WSAStartup()/socket())；

2：向服务器发出连接请求(connect())；

3：和服务器端进行通信(send()/recv())；

4：关闭套接字，关闭加载的套接字库(closesocket()/WSACleanup())。

客户端的代码如下：

#include <stdio.h>

#include <Winsock2.h>

void main()

{

WORD wVersionRequested;

WSADATA wsaData;

int err;

wVersionRequested = MAKEWORD( 1, 1 );

err = WSAStartup( wVersionRequested, &wsaData );

if ( err != 0 ) {

return;

}

if ( LOBYTE( wsaData.wVersion ) != 1 || HIBYTE( wsaData.wVersion ) != 1 ) {

WSACleanup( );

return;

}

SOCKET sockClient=socket(AF\_INET,SOCK\_STREAM,0);

SOCKADDR\_IN addrSrv;

addrSrv.sin\_addr.S\_un.S\_addr=inet\_addr("127.0.0.1");

addrSrv.sin\_family=AF\_INET;

addrSrv.sin\_port=htons(6000);

connect(sockClient,(SOCKADDR\*)&addrSrv,sizeof(SOCKADDR));

send(sockClient,"hello",strlen("hello")+1,0);

char recvBuf[50];

recv(sockClient,recvBuf,50,0);

printf("%s\n",recvBuf);

closesocket(sockClient);

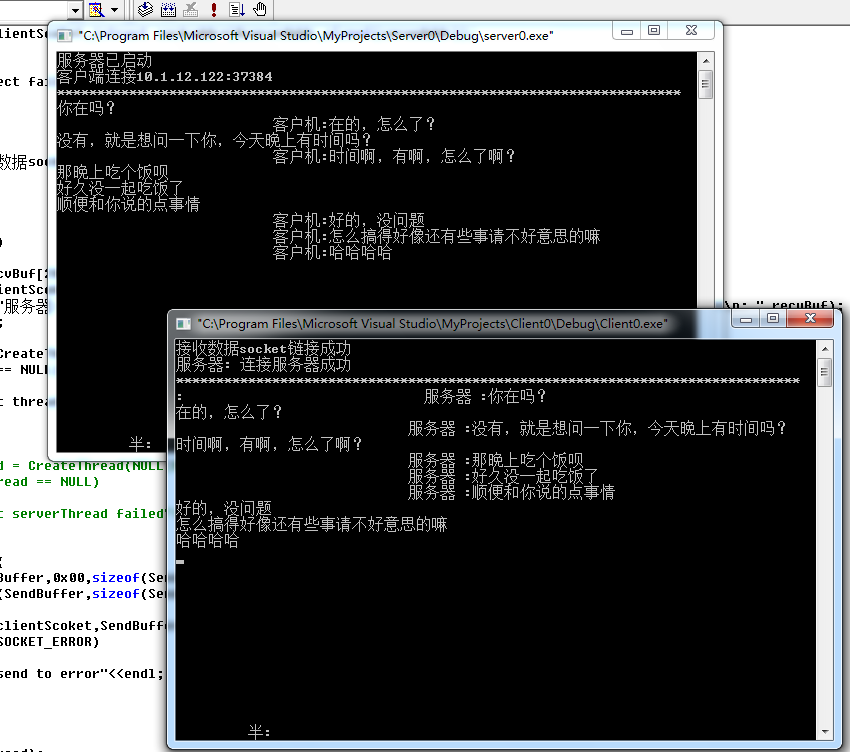
WSACleanup();

}

实现上面的基本程序以后，客户端与服务端之间能够实现一次信息发送与接收，对于我们的聊天程序来说是不够的，只能一句的聊天程序怎么看也是很没有用的东西，所以我们就要对上面的基本程序进行修改。

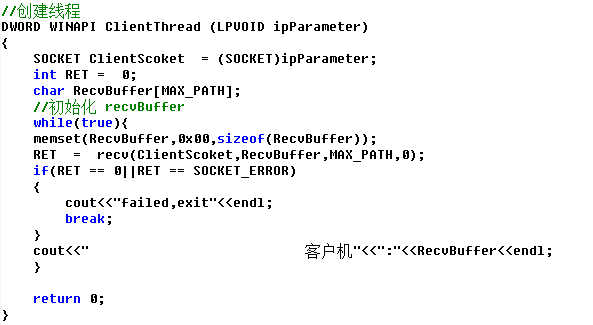
1. **实验结果及结果分析**

经过修改后得到了我们实验想要的程序，程序通过改变以后，去掉了之前的毛病，不仅仅可以实现互相发送多次，而且任何一方都能够多次发送，而不受对方影响，不需要对方回应才能进行下一句；而且对话的发起方可以是服务器和客户端的任意一端，不受次序影响。实验结果效果图如下：



程序的主要修改如下：

1. 对服务器增加一个接收线程，是服务端随时可以接收，并且通过main函数主线程定义while循环实现随时发送：





2、根据上面的程序，同理，我们也在客户端实现一个接收信息的线程，以及在客户端main函数中实现循环发送信息，使得客户端循环接收信息与发送信息不互相影响。具体代码与上面差不多就不在重述。

1. **心得体会**

通过这个实验，我学习到了Winsock编程有关VC下socket的一些基础知识，能够实现基本的服务器与客户端的信息，同时巩固了自己VC编程的线程的有关知识。总的来说，解决这个实验问题，让自己对VC编程有了一个更加深刻的认识与体会，以后也会更加努力的去学习这方面的知识，了解底层的一些实现。有关windows网络编程方面的基本网络与编程知识。