/\*

\*Target：简易的聊天程序，客户端和服务器之间互相发送消息，直到客户端断开连接

\*Author：张磊

\*Date：.10.31

\*/

#include <cstdlib>

#include <iostream>

#include <conio.h>

#include <stdio.h>

#include <winsock2.h>

#pragma comment(lib,"ws2\_32.lib")

bool InitSocket();

using namespace std;

int main(int argc, char \*argv[])

{

struct sockaddr ser\_addr,client\_addr;

int len,off;

bool over;

char msg1[255];

char msg[255];

SOCKET welcomesock ;

if( !InitSocket() ) return 0; //初始化Window Sockets DLL(ws2\_32.dll)

welcomesock = socket( AF\_INET,SOCK\_STREAM,0 ); //创建流Socket：即传输层使用TCP

if( welcomesock==INVALID\_SOCKET ){ //不能创建，返回

printf("不能创建Socket!");

getch();

WSACleanup( ); //卸载Window Sockets DLL

return 0;

}

//设置服务器IP地址和端口号

((sockaddr\_in\*)&ser\_addr)->sin\_family = AF\_INET; //AF\_INET:使用Internet 协议

((sockaddr\_in\*)&ser\_addr)->sin\_port = htons(3000); //服务器端口号

((sockaddr\_in\*)&ser\_addr)->sin\_addr.s\_addr = inet\_addr("127.0.0.1");//主机地址

//把套接字与地址绑定

bind(welcomesock,&ser\_addr,sizeof(ser\_addr));

//监听网络连接

listen(welcomesock,5); //监听连接：--允许等待队列的长度

printf("等待客户连接!\n");

//struct sockaddr client\_addr;

len = sizeof(client\_addr);

//接受网络连接，生成新的套接字sersock标识这一连接

SOCKET connectsock = accept( welcomesock,&client\_addr,&len );

//从等待队列中检取客户，如队列空则进程挂起

//如不空，则接受并生成新Socket以表示此连接，而原Socket继续等待新客户

if(connectsock==INVALID\_SOCKET){

DWORD err = WSAGetLastError();

char txt[100];

sprintf(txt,"error when accept!---errno:%d",err);

printf(txt);

getch();

WSACleanup( ); //卸载Window Sockets DLL

return 0;

}

printf("有用户在线！\n");

while(true)

{

printf("等待对方发送消息：");

len = recv ( connectsock,msg,200,0 ); //接收用户信息

printf("\n对方发来信息:%s",msg);

printf("\n输入发送消息：");

gets(msg1);

send( connectsock,msg1,strlen(msg)+1,0 ); //发送输入消息

// char \* msg1;

// msg1=strupr(msg);

//

// send ( connectsock,msg1,strlen(msg1)+1,0 ); //发出输入信息

//

// printf("\n\n已返回处理结果!\n");

////关闭套接字

// closesocket(connectsock);

//printf("\n已关闭与该客户的连接!\n");

//printf("\n等待下一客户连接!\n");

}

closesocket(welcomesock);

printf("\n对方用户已下线!\n");

WSACleanup( ); //卸载Window Sockets DLL

return 0;

}

bool InitSocket()

{

WORD wVersionRequested;

WSADATA wsaData;

int err;

wVersionRequested = MAKEWORD( 2, 0 );

//初始化Windows Sockets DLL,

err = WSAStartup( wVersionRequested, &wsaData );//加载Winsock DLL到内存

if ( err != 0 ) {

printf("没有Windows Socket动态库!\n");

getch();

return false;

}

if ( LOBYTE( wsaData.wVersion ) != 2 || HIBYTE( wsaData.wVersion ) != 0 ) {

printf("需要Windows Socket 2!\n");

getch();

WSACleanup( ); //非winsock 2.0时，卸载Window Sockets DLL

return false;

}

return true;

}

#include <cstdlib>

#include <iostream>

#include <conio.h>

#include <stdio.h>

#include <string.h>

#include <winsock2.h> //Windows Sockets函数定义

#include <stdlib.h>

#pragma comment(lib,"ws2\_32.lib") //自己添加

bool InitSocket(); //初始化Windows Sockets动态连接库

using namespace std;

int main(int argc, char \*argv[])

{

int type;

SOCKET clientsock; //套接字

struct sockaddr addr; //地址

int len,portno;

int err;

char msg[255];

bool over;

int i;

if( !InitSocket() ) return 0; //初始化Window Sockets DLL

type = SOCK\_STREAM; //面向连接的流式Socket类型

clientsock = socket( AF\_INET,type,0 ); //创建支持Internet协议的流式Socket

if( clientsock==INVALID\_SOCKET ){ //不能创建，返回

printf("不能创建Socket!");

getch();

WSACleanup( ); //卸载Window Sockets DLL

return 0;

}

//客户端地址设定

((sockaddr\_in\*)&addr)->sin\_family = AF\_INET; //AF\_INET:使用Internet 协议

((sockaddr\_in\*)&addr)->sin\_port = 0; //客户端端口号:0--系统分配

((sockaddr\_in\*)&addr)->sin\_addr.s\_addr = 0; //主机地址：--本机

bind(clientsock,&addr,sizeof(addr)); //把地址与Socket绑定

//获取服务器地址

((sockaddr\_in\*)&addr)->sin\_family = AF\_INET; //AF\_INET:使用Internet 协议

((sockaddr\_in\*)&addr)->sin\_addr.s\_addr = inet\_addr("127.0.0.1"); //服务器地址

((sockaddr\_in\*)&addr)->sin\_port = htons( 3000 ); //服务器端口号；

//htons()：把主机表示的短整数转化成网络字节顺序

len = sizeof(addr);

err = connect( clientsock,(sockaddr\*)&addr,len );

//在使用面向连接协议时(TCP)，必须与服务器连接成功后，才可通信

//在无连接协议(UDP)中，可以直接向对方发数据，而无需连接

if( err==SOCKET\_ERROR ){

printf("连接失败!");

getch();

WSACleanup( ); //卸载Window Sockets DLL

return 0;

}

printf("成功连接到服务器!\n输入发送内容：\n");

while(1)

{

gets(msg);

send( clientsock,msg,strlen(msg)+1,0 ); //发送输入信息

printf("\n消息已发送等待回应");

len = recv ( clientsock,msg,200,0 );//接受信息

msg[len]=0;

printf("\n\n收到信息:");

printf(msg); //显示接受信息

printf("\n\n输入发送内容:");

}

closesocket(clientsock); //关闭套接字

WSACleanup( ); //卸载Window Sockets DLL

printf("\n\n已关闭连接！");

getch();

getch();

return 0;

}

bool InitSocket()

{

WORD wVersionRequested;

WSADATA wsaData;

int err;

wVersionRequested = MAKEWORD( 2, 0 ); //Winsock2.0

//初始化Windows Sockets DLL,

err = WSAStartup( wVersionRequested, &wsaData ); //加载Winsock DLL到内存

if ( err != 0 ) {

printf("没有Windows Socket动态库!\n");

getch();

return false;

}

if ( LOBYTE( wsaData.wVersion ) != 2 || HIBYTE( wsaData.wVersion ) != 0 ) {

printf("需要Windows Socket 2!\n");

getch();

WSACleanup( ); //非Winsock2.0时，卸载Window Sockets DLL

return false;

}

return true;

}