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： 唐韬**

**学号： 1305010206**

**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：开启一个线程来接收信息，同时在主线程中等待发送信息

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())。

// TTScoket.cpp : 定义控制台应用程序的入口点。

//

#include "stdafx.h"

using namespace std;

SOCKET serConn;

DWORD WINAPI funproc(LPVOID lpparentet)

{

while(1){

Sleep(100);

char receiveBuf[100];//接收

memset(receiveBuf, 0, 100);

if(recv(serConn,receiveBuf,100,0)!=-1){

printf("客户端：%s\n",receiveBuf);

}

}

return 0;

}

int main()

{

//创建套接字

WORD myVersionRequest;

WSADATA wsaData;

myVersionRequest=MAKEWORD(1,1);

int err;

err=WSAStartup(myVersionRequest,&wsaData);

if (!err)

{

printf("服务端就绪\n");

}

else

{

//进一步绑定套接字

printf("Scoket未打开!");

return -1;

}

SOCKET serSocket=socket(AF\_INET,SOCK\_STREAM,0);//创建了可识别套接字

//需要绑定的参数

SOCKADDR\_IN addr;

addr.sin\_family=AF\_INET;

addr.sin\_addr.S\_un.S\_addr=htonl(INADDR\_ANY);//ip地址

addr.sin\_port=htons(6000);//绑定端口

bind(serSocket,(SOCKADDR\*)&addr,sizeof(SOCKADDR));//绑定完成

listen(serSocket,5);//其中第二个参数代表能够接收的最多的连接数

//////////////////////////////////////////////////////////////////////////

//开始进行监听

//////////////////////////////////////////////////////////////////////////

SOCKADDR\_IN clientsocket;

int len=sizeof(SOCKADDR);

serConn=accept(serSocket,(SOCKADDR\*)&clientsocket,&len);//如果这里不是accept而是conection的话。。就会不断的监听

CreateThread(NULL,0,funproc,NULL,0,NULL);

while (1)

{

char sendBuf[100];

gets(sendBuf);

//sprintf(sendBuf,"welcome %s to bejing",inet\_ntoa(clientsocket.sin\_addr));//找对对应的IP并且将这行字打印到那里

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

}

closesocket(serConn);//关闭

WSACleanup();//释放资源的操作

}

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：

// TTCilent.cpp : 定义控制台应用程序的入口点。

//

#include "stdafx.h"

SOCKET clientSocket;

DWORD WINAPI funproc(LPVOID lpparentet)

{

while(1){

Sleep(100);

char receiveBuf[100];

memset(receiveBuf, 0, 100);

if(recv(clientSocket,receiveBuf,100,0)!=-1){

printf("服务端：%s\n",receiveBuf);

}

}

return 0;

}

int main()

{

int err;

WORD versionRequired;

WSADATA wsaData;

versionRequired=MAKEWORD(1,1);

err=WSAStartup(versionRequired,&wsaData);//协议库的版本信息

if (!err)

{

printf("客户端就绪!\n");

}

else

{

printf("客户端的嵌套字打开失败!\n");

return -1;//结束

}

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

SOCKADDR\_IN clientsock\_in;

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

clientsock\_in.sin\_family=AF\_INET;

clientsock\_in.sin\_port=htons(6000);

//bind(clientSocket,(SOCKADDR\*)&clientsock\_in,strlen(SOCKADDR));//注意第三个参数

//listen(clientSocket,5);

connect(clientSocket,(SOCKADDR\*)&clientsock\_in,sizeof(SOCKADDR));//开始连接

CreateThread(NULL,0,funproc,NULL,0,NULL);

while(1){

char str[100];

gets(str);

int sum = strlen(str)+1;

send(clientSocket,str,strlen(str)+1,0);

}

closesocket(clientSocket);

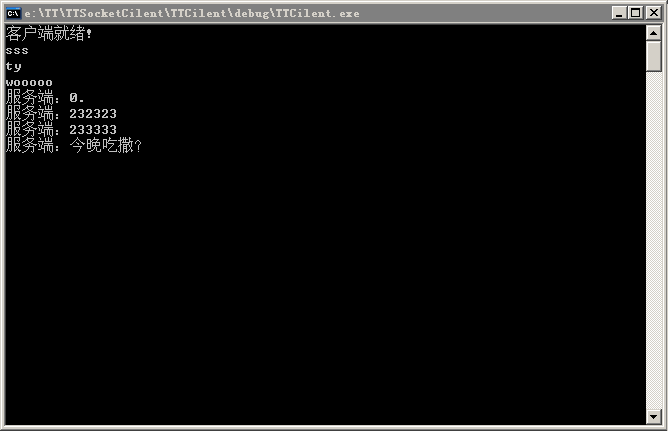
WSACleanup();

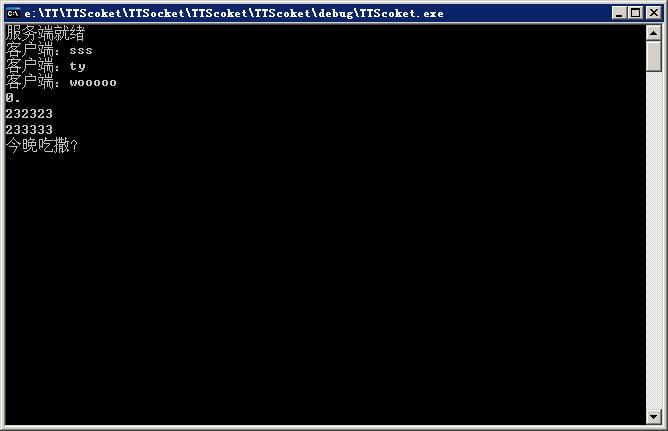
}

以上就是所有的代码部分，通过添加进程以及循环来循环收发信息，核心部分就是Socket的通信。

1. **实验结果及结果分析**

实验图：





1. **心得体会**

以前从来没写过c++的socket，java和C#的倒是写过很多，后面两种语言抽象程度更高一些，所以用起来更加的方便一些，但是c++的自由程度要高一些，与之相对应的代价就是编程的时候要麻烦一些，收获良多。