实现一个聊天系统

想要实现的功能：可以实现多人聊天，私聊，群聊，多线程

服务器端：

1. 加载socket库

//加载socket库

WSADATA wsadata;

if (WSAStartup(MAKEWORD(2, 2), &wsadata)!=0)//成功返回0

{

cout << "socket初始化失败" << endl;

return 0;

}

1. 创建socket

//创建一个socket，并将该socket绑定到一个特定的传输层

sockSer = socket(AF\_INET, SOCK\_STREAM, 0);//地址类型(ipv4)，服务类型(流式套接字)

1. 初始化地址

addrSer.sin\_addr.s\_addr = inet\_addr("192.168.47.1");

addrSer.sin\_family = AF\_INET;

addrSer.sin\_port = htons(6666);

1. 绑定bind

//绑定bind

if (bind(sockSer, (sockaddr\*)&addrSer, sizeof(sockaddr)) == -1)//失败返回-1，成功返回0

cout << "bind error" << endl;

[(102条消息) socket通信 bind() 函数详解\_行孤、的博客-CSDN博客\_socket.bind](https://blog.csdn.net/qq_43403759/article/details/114727885)

1. 监听端口

listen(sockSer, 5);//5是内核为此套接口排队的最大连接个数

1. 接收连接请求

sockConn[i] = accept(sockSer, (sockaddr\*)&addrCli, &len);

[(102条消息) socket编程之accept()函数\_明潮的博客-CSDN博客\_accept socket](https://blog.csdn.net/u010144805/article/details/78276659)

1. 创建线程

DWORD WINAPI handlerRequest(LPVOID lparam);

HANDLE hThread, hThread2;

DWORD dwThreadId1, dwThreadId2;

hThread = ::CreateThread(NULL, NULL, handlerRequest, LPVOID(i), 0, &dwThreadId1);

[(102条消息) C++使用CreateThread函数创建线程\_Estella1024的博客-CSDN博客\_用createthread函数创建线程](https://blog.csdn.net/qq_41924554/article/details/118335070?spm=1001.2101.3001.6650.1&utm_medium=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-118335070-blog-82623531.pc_relevant_multi_platform_whitelistv3&depth_1-utm_source=distribute.pc_relevant.none-task-blog-2%7Edefault%7ECTRLIST%7ERate-1-118335070-blog-82623531.pc_relevant_multi_platform_whitelistv3&utm_relevant_index=2)

1. 线程函数

DWORD WINAPI handlerRequest(LPVOID lparam)

{

SOCKET sockTemp = sockConn[(int)lparam];

char recvBuf[BUF\_SIZE] = {};

recv(sockTemp, recvBuf, 1024, 0);

if (recvBuf[0])

{

cout << recvBuf << endl;

MSG\_FORM m = str2msg(recvBuf);

int id = m.recvID - 48;

send(sockConn[id], recvBuf, 1024, 0);

if (!strcmp(m.msg, "quit"))

{

cout << "client " << (int)lparam << " quit" << endl;

conNum--;

if (!conNum)

return 0;

}

}

return 0;

}

1. 关闭socket和socket库

closesocket(sockSer);

WSACleanup();

客户端：

1. 加载socket库
2. 创建socket
3. 初始化地址

addrCli.sin\_addr.s\_addr = inet\_addr("192.168.47.1");

addrCli.sin\_family = AF\_INET;

addrCli.sin\_port = htons(2333);

addrSer.sin\_addr.s\_addr = inet\_addr("192.168.47.1");

addrSer.sin\_family = AF\_INET;

addrSer.sin\_port = htons(6666);

1. connect

sockCli = connect(sockclient, (sockaddr\*)&addrSer, sizeof(sockaddr));//数据发送的套接字，目的地址

[(102条消息) 网络编程socket之connect函数\_普通网友的博客-CSDN博客\_socket connect](https://blog.csdn.net/m0_67393619/article/details/124487791)

1. 接收和发送线程函数

发送线程

DWORD WINAPI handlerRequest1(LPVOID lparam)

{

memset(sendBuf, 0, BUF\_SIZE);

char buffer[BUF\_SIZE] = {};

SOCKET sockclient = (SOCKET)(LPVOID)lparam;

cin.getline(buffer, 1024, '\n');

sendBuf[0] = ID + 48;

strcat(sendBuf, buffer);

send(sockclient, sendBuf, 1024, 0);

MSG\_FORM m = str2msg(sendBuf);

if (!strcmp(m.msg, "quit") || !strcmp(buffer, "quit"))

cond = 1;

return 0;

}

接收线程

DWORD WINAPI handlerRequest2(LPVOID lparam)

{

memset(recvBuf, 0, BUF\_SIZE);

SOCKET sockclient = (SOCKET)(LPVOID)lparam;

recv(sockclient, recvBuf, 1024, 0);

if (recvBuf[0])

{

MSG\_FORM m = str2msg(recvBuf);

cout << m.sendID << ": " << m.msg << endl;

}

return 0;

}

