

学生学号	0121610870910	实验课成绩	
------	---------------	-------	--

武汉理工大学

学 生 实 验 报 告 书

课 程 名 称 通信原理

开 课 学 院 计算机科学与技术学院

指 导 老 师 刘维

学 生 姓 名 冯钢果

专 业 班 级 软件 1604 班

2017 — 2018 学 年 第 二 学 期

实验课程名称： 通信原理

实验项目名称	采用 Winsock 在有线局域网上的点-点通信			实验成绩	
实验者	冯钢果	专业班级	软件 1604 班	组别	
同组者	唐标			实验日期	2018 年 5 月 16 日

第一部分：实验分析与设计（可加页）

一、实验内容描述

“采用 Winsock 在有线局域网上的点-点通信”：

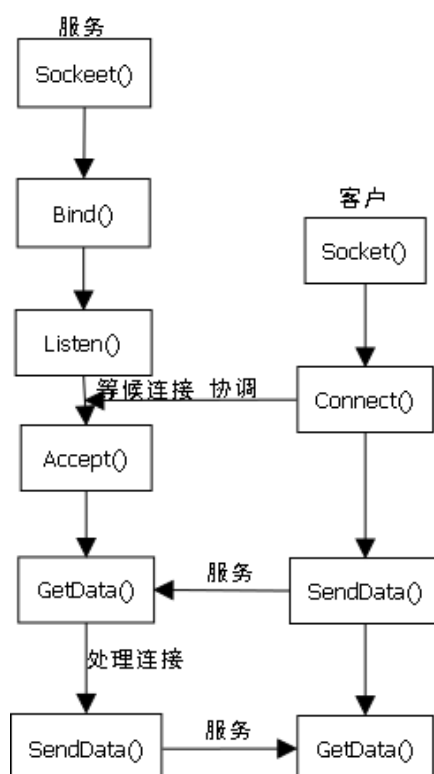
- (1) 熟悉 VB6.0 或其它程序开发工具的控制和界面设计，进而熟悉 Winsock 的有关控件及编程方法；
- (2) 在了解所用的两个工作站和服务器的 IP 地址后，采用 VB6.0 或其它程序开发工具的控制和 Winsock 控件编写并调试在有线局域网上的点-点通信程序。
- (3) 具有点对点通信功能，任意客户端之间能够发送信息。

二、实验基本原理与设计

WinSock 基本原理：

- (1) Winsock 控件对用户来说是不可见的，它提供了访问 TCP 和 UDP 网络服务的方便途径，通过设置 winsock 控件的属性，可以很方便的实现点到点或点到多点的互相通信。
- (2) 有线局域网上的点到点通信程序分为客户端通信和服务器通信两大部分：
 - a. 对于服务器通信程序来说，它必须先建立一个 socket 端口，将其绑定到一个指定的端口代号，然后被动的进行侦听，等待客户机的连接请求。一旦有客户机请求连接，马上重载 winsock 控件，利用新的 winsock 控件来接受请求，而原来的 winsock 控件仍在原端口上进行侦听；
 - b. 对于客户机通信程序来说，它首先要利用 winsock 控件建立一个 socket 端口，当得知服务器程序的 socket 端口（IP 地址和端口号）后，它就可以向服务器程序请求连接，如果连接成功，客户机和服务器姐可以进行双向的交换数据了。

Winsock 连接的点到点通信流程图如下：



//服务器端关键代码

```
sockServer = socket(AF_INET, SOCK_STREAM, 0);    //定义服务器端 socket
addrServer.sin_addr.S_un.S_addr = htonl(INADDR_ANY); //设置服务器端 socket

addrServer.sin_family = AF_INET;
addrServer.sin_port = htons(2018);
bind(sockServer, (SOCKADDR *)&addrServer, sizeof(SOCKADDR));
                                                                    //将服务器 socket 绑定在端口
listen(sockServer, 10);                                           //Listen 监听端口
printf("Server started\n");
len = sizeof(SOCKADDR);

while (true)
{
    //accept 会阻塞进程，直到有客户端连接上来为止
    sockClient = accept(sockServer, (SOCKADDR *)&addrClient, &len);

    //当客户端连接上来时，拼接字符串
    sprintf(sendBuf, " Welcome %s\n", inet_ntoa(addrClient.sin_addr));

    send(sockClient, sendBuf, strlen(sendBuf) + 1, 0);    //向客户端发送字符串
    recv(sockClient, recvBuf, 100, 0);                   //获取客户端返回的数据
    printf("%s\n", recvBuf);                             //打印客户端返回的数据

    closesocket(sockClient);                              //关闭 socket
}
```

以上使用 winsock 主要进行了服务器端的端口绑定、客户端的端口连接，这个过程是建立点对点通信的最重要过程。进行连接后服务器端就能与客户端发送信息，主要使用了 while(true) 实现实时监听，避免一次通信后断开连接，一次使用“死循环”来不错过任何消息侦听，同时在完成通信后要关闭 socket，避免异常抛出造成服务器端故障。另外，应注意程序异常的处理，如客户端与服务器连接失败、服务器向客户端发信失败、客户端连接后突然断开等异常，如果不进行异常处理很有可能程序因为操作不当就导致通信不能继续进行。因此，对异常的处理也是实验成功的关键。

第二部分：实验调试与结果分析（可加页）

一、调试过程

调试方法：输入代码并编译运行，如果报错检测语法错误或其他异常错误后更正，确保服务器端和客户端程序正确运行，再打开服务器程序，然后打开客户端程序输入数据，如在客户端输入“First message!”，然后点击“send”按钮，然后观察服务器端的显示结果。

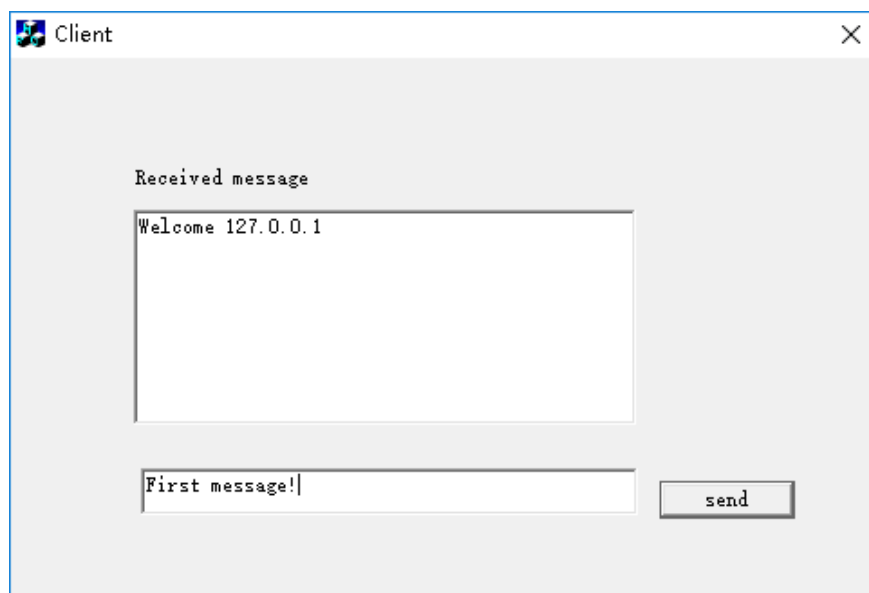
调试问题：

- (1) 刚开始由于对 winsock 在有线局域网的点对点通信理解不够透彻，对 winsock 控件使用不够熟练，导致实验过程中对函数（如 bind、GetData）的使用不熟悉；
- (2) 代码大致写完完后，由于没有对程序异常处理，导致程序操作不当经常引起小错误。

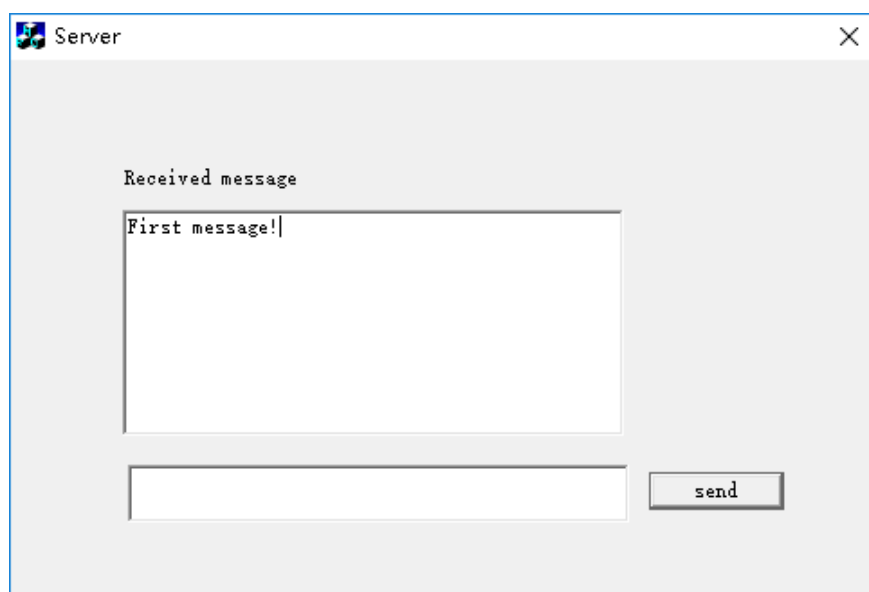
三、实验结果及分析

实验结果截图：

客户端运行截图：



服务端运行截图：



结果分析：先打开服务器端，然后打开客户端发现有提示“Welcome 127.0.0.1”，然后再使用客户端发信“First message!”，发现服务器端消息提示“First message!”，证明实验成功，实现了有线局域网上的点-点通信。

三、实验小结、建议及体会

通过本次实验我熟悉了 VB6.0 或其它程序开发工具的控制和界面设计以及 Winsock 的有关控件及编程方法，使我对网络编程有了更深的概念。同时让我发现自己知识结构的欠缺。虽然基本功能已经实现，但还是存在很多不稳定的问题尚待解决。

此次实验是对 Socket 的一个应用，让我更深的了解了 Socket 函数的作用及使用方法。实验过程中遇到了很多的问题，例如：程序只能通信一次。刚开始我以为建立的 socket 套接口只要建立一次并更新连接地址就可以多次使用。当初不知道问题存在的在于这，在程序加入很多错误报告代码，发现再客户端再次 connect 的时候返回负值。在网上查看很多相关代码之后，发现问题所在，把建立 socket 的代码写进客户端的循环体中，每次通信都重新建立 socket 套接口就可以了。

另外，最深的感触是，对于通信原理一定要多动手实验，可能在课本学习中，只能对知识了解框架，实验对于了解知识的原理有很大帮助。

实验课程名称： 通信原理

实验项目名称	采用 Winsock 在无线局域网上的点-点群发通信			实验成绩	
实验者	冯钢果	专业班级	软件 1604 班	组别	
同组者	唐标、郑佳浩			实验日期	2018 年 5 月 30 日

第一部分：实验分析与设计（可加页）

一、实验内容描述

“采用 Winsock 在无线局域网上的点-点群发通信”：

- (1) 修改实验 01 的界面设计，设计为多个接收地址的群发界面（因为群发是一对多的循环点-点通信），并修改相应的程序；
- (2) 在了解所用的 3 个工作站和服务器的 IP 地址后，采用 VB6.0 的控件和 Winsock 控件编写并调试在无线局域网上的点-点群发通信程序。

二、实验基本原理与设计

采用 Winsock 在无线局域网上的点-点群发通信的基本原理：

- (1) 点到群的通信就是客户端将消息发送至多个服务器端，相当于在客户端加入一个循环，依次给不同 ip 的服务器发送消息；
- (2) 在实验一的基础上， 修改服务器端的消息响应方法可以实现点到群的通信。实验一中点到点的通信原理是， 客户端将消息发送到服务器端， 然后服务器端反馈客户端相应的信息。在本实验中可以修改为客户端将消息发送至多个服务器端， 然后接受传送回来的信息。相当于在客户端加入一个循环，依次给不同 ip 的服务器端发送消息。

三、主要仪器设备及耗材

1. 一台 PC 机；
2. 软件 Microsoft Visual Studio 2010；
3. winsock 开发工具；
4. 草稿纸、笔、教材《Visual Basic6.0 中文版实用培训教程》。

关键代码示例：

//服务器端关键代码

```

char sendBuf[100];           // 发送至客户端的字符串
char recvBuf[100];           // 接受客户端放回的字符串
SOCKE sTockServer;           // 服务器端 Socket
SOCKADDR_ IaNddrServer;       // 服务器端地址
SOCKE sTockClient;           // 客户端 Socket
SOCKADDR_ IaNddrClient;       // 客户端地址
WSADATA wAsaData;
WOR wDVersinRequested;
wVersinRequested = MAKEWO(R2,D 2) ;// 配置 Windows Socket 版本
err = WSASStartup(wVersinRequested, &wsaData); // 初始化 Windows Socket

```

```

if (err != 0)                                // 启动错误，程序结束
{
    return ;
}
if ( LOBYTE(wsaData.wVersion) != 2 || HIBYTE(wsaData.wVersion) != 2)
{
    // 启动错误

    WSACleanup();
    return ;
}
sockServer = socket( AF_INET, SOCK_STREAM, 0);    // 设置服务器端 socket
addrServer.sin_addr.S_un.S_addr = htonl( INADDR_ANY); // 本机 IP
addrServer.sin_family = AF_INET;
addrServer.sin_port = htons(2018);

// 将服务器 socket 绑定在本地端口
bind(sockServer, ( SOCKADDR *)&addrServer, sizeof ( SOCKADDR));

listen(sockServer, 5);                        // Listen 监听端口
printf( "Server started!\n" );
len = sizeof ( SOCKADDR);

//accept 会阻塞进程，直到有客户端连接上来为止
sockClient = accept(sockServer, ( SOCKADDR *)&addrClient, &len);
// 当客户端连接上来时，拼接字符串
sprintf(sendBuf, "Welcome ip:%s\n", inet_ntoa(addrClient.sin_addr));

send(sockClient, sendBuf, strlen(sendBuf) + 1, 0);    // 向客户端发送字符串
recv(sockClient, recvBuf, 100, 0);                  // 获取客户端返回的数据
CString text;
text = recvBuf;
SetDlgItemText ( IDC_EDIT1, text);
closesocket(sockClient);                            // 关闭 socket

```

//客户端关键代码

```

char * ip1 = "192.168.1.110" ;
char * ip2 = "192.168.1.111" ;
CString text;
GetDlgItemText (IDC_EDIT1, text);
int nLength = text.GetLength();
int nBytes = WideCharToMultiByte( CP_ACP, 0, text, nLength, NULL, 0, NULL, NULL);
char * message = new char [nBytes + 1];
memset(message, 0, nLength + 1);
WideCharToMultiByte( CP_OEMCP, 0, text, nLength, message, nBytes, NULL, NULL);
message[nBytes] = 0;

```



```

int err;
char recvBuf[100];
SOCKET sTockClient; // 客户端 Scket
SOCKADDR_ IaNddrServer; // 服务端地址
WSADATA wAsaData;
WORD wDVersionRequested;
wVersionRequested = MAKEWORD(2, 2);
err = WSAStartup(wVersionRequested, &wAsaData);
if (err != 0)
{
    return ;
}
if ( LOBYTE(wAsaData.wVersion) != 2 || HIBYTE(wAsaData.wVersion) != 2)
{
    WSACleanup();
    return ;
}
sockClient = socket( AF_INET, SOCK_STREAM, 0); // 新建客户端 socket

// 定义要连接的服务端地址
addrServer.sin_addr.S_un.S_addr = inet_addr(ip1); //目标 IP (175.0.170.28 是本地地址)
addrServer.sin_family = AF_INET; // 协议类型是 INET
addrServer.sin_port = htons(2018);

// 让 sockClient 连接到服务端
connect(sockClient, ( SOCKADDR *)&addrServer, sizeof ( SOCKADDR));

send(sockClient, message, strlen(message) + 1, 0); // 发送数据到服务端
recv(sockClient, recvBuf, 100, 0); // 从服务端获取数据
text = recvBuf;
SetDlgItemText ( IDC_EDIT2, text);
closesocket(sockClient); // 关闭 socket
WSACleanup();
addrServer.sin_addr.S_un.S_addr = inet_addr(ip2); //目标 IP
addrServer.sin_family = AF_INET; // 协议类型是 INET
addrServer.sin_port = htons(2018);

// 让 sockClient 连接到服务端
connect(sockClient, ( SOCKADDR *)&addrServer, sizeof ( SOCKADDR));
send(sockClient, message, strlen(message) + 1, 0); // 发送数据到服务端
recv(sockClient, recvBuf, 100, 0); //从服务端获取数据

text = recvBuf;
SetDlgItemText ( IDC_EDIT2, text);
closesocket(sockClient); // 关闭 socket
WSACleanup();

```

第二部分：实验调试与结果分析（可加页）

一、调试过程

调试方法：输入代码并编译运行，如果报错检测语法错误或其他异常错误后更正，确保服务器端和客户端程序正确运行，再打开服务器程序，然后打开两个客户端程序，在服务器端输入数据“Hello everyone!”，然后点击“send”按钮，然后观察客户端的显示结果。

调试问题：

- (1) 代码大致写完后，由于没有对程序异常处理，导致程序操作不当经常引起小错误。
- (2) 服务器向两个客户端发信有一个未收到来信，检查原因是由于前一个的 socket 未关闭导致错误。

二、实验结果及分析

运行结果截图：

服务器端截图：

客户端 A 截图：



客户端 B 截图：

客户端 C 截图：



结果分析：当客户端打开时都会收到“Welcome (ip 地址)”的提醒，然后服务器向客户端发信“Hello everyone!”，接着在客户端都受到来信“Hello everyone”，说明实验成功实现了无线局域网上的点-点群发通信。

调试问题：当打开一个服务器端和三个客户端时，有可能出现其中一个不能实现局域网通信，检查原因发现是由于该客户端缺少与服务器端连接代码，然后加上关键代码运行排除问题；还有在处理异常时，由于服务器端与多个客户端连接，因此对于异常处理比较敏感，如果不进行异常处理，只要有一个客户端异常下线，就导致服务器端异常终端，导致其他机器与服务器的通信过程中断，可是多次使用异常捕获，但对于一些位置错误也要预判，如客户机输入一个不能在其他机器显示的字符，因此这些都要仔细考虑。

三、实验小结、建议及体会

通过在实验一的基础上继续开发，我实现了多个接收地址的群发界面，更加深入了解了 winsock 通信机制，以及利用 mfc 编程实现的方法。了解了群发是一对多的循环点-点通信，熟悉了通信的技术以及实现方法。当初不知道问题存在的在于这，在程序加入很多错误报告代码，发现再客户端再次 connect 的时候返回负值。在网上查看很多相关代码之后，发现问题所在，把建立 socket 的代码写进客户端的循环体中，每次通信都重新建立 socket 套接口就可以了。

在进行在无线局域网上的点-点群发通信的过程，与队友的配合也十分重要，比如开启服务器端要先于客户端开启时间，如果时间颠倒就导致客户端不能接到群发通信时就因为异常而报错下线。另外，体会最深的是通信原理的实验环节对于理论知识理解帮助很大，因此要想学好通信原理和其他工科课程实验要认真对待，多动手可以帮助理解，同时实验过程中，注意与他人的交流，注意解决问题的方法和解决问题的思路，更多的耐心才能学到更多的知识。