# 实验B.1-B.3：套接字编程实验

网络编程是通过使用[套接字](http://baike.baidu.com/view/538713.htm" \t "_blank)来达到[进程间通信](http://baike.baidu.com/view/1492468.htm" \t "_blank)目的的编程，Socket编程是网络编程的主流工具，Socket API是实现进程间通信的一种编程设施，也是一种为进程间提供底层抽象的机制，提供了访问下层通信协议的大量系统调用和相应的数据结构。本实验利用Socket API编写网络通信程序，具体实验要求及内容如下。

【实验目的和要求】

1. 掌握C++、JAVA或Python等集成开发环境编写网络程序的方法；
2. 掌握客户/服务器（C/S）应用的工作方式；
3. 学习网络中进程之间通信的原理和实现方法；
4. 要求本机既是客户端又是服务器端；

【实验内容】（任选一个）

1. 编写一个基于socket的简易聊天程序。所编写的程序应具有如下功能：
2. 具有点对点通信功能，任意客户端之间能够发送消息；
3. 具有群组通信功能，客户端能够向组内成员同时发送消息，其他组成员不能收到；
4. 具有广播功能，客户端能够向所有其他成员广播消息；
5. 邮件代理端软件的设计与开发：基本要求可以完成邮件编辑，与标准邮件服务器的连接，发送并且下载邮件，进一步要求邮件管理，邮件的加密解密传送。

设计提示—学习邮件客户端和服务器之间通信协议SMTP和POP3协议的规范，选择一特定的邮件服务器（自己安装一个或选择一公共邮件服务器）建立邮件账户，编程实现与服务器的通信建立、利用SMTP协议完成邮件发送，利用POP3完成邮件接收。

1. 编写一个端口扫描器（类似nmap的简易版本）。给定目的ip地址，可以扫描目的IP地址在哪个端口上可以接受tcp连接和udp连接。

【编程语言和环境】

1. 编程语言C/C++/C#/Java等均可；
2. 编程环境Windows（MS Visual系列，VC/VB/VS.Net；）和Linux（编辑器vi+编译器GCC）均可；

【实验报告】

所交实验报告内容包括：

1. 实验目的与要求；
2. 实验内容与实现原理；
3. 实验具体设计实现及结果（含流程图及关键代码说明）；
4. 实验设备与实验环境；
5. 实验总结

【实验主要功能实现说明】

以下为针对三个实验内容实现方法的简要说明，示例所用语言为C#。

1. 点对点通信功能

实现网络点对点通讯程序的关键步骤就是实现信息在网络中的发送和接收。数据接收使用的是Socket，数据发送使用的是NetworkStream。

1.1利用Socket来接收信息

TcpListener tlListen1 = new TcpListener ( 8889 ) ;  
    //侦听端口号  
    tlListen1.Start ( ) ;  
    Socket skSocket = tlListen1.AcceptSocket ( ) ;  
    //接受远程计算机的连接请求，并获得用以接收数据的Socket实例  
    EndPoint tempRemoteEP = skSocket.RemoteEndPoint  ;  
    //获得远程计算机对应的网络远程终结点  
    while (  true )  
    {  
         Byte [] byStream = new Byte[80] ;  
         //定义从远程计算机接收到数据存放的数据缓冲区  
         int i = skSocket.ReceiveFrom  ( byStream   , ref tempRemoteEP  ) ;  
         //接收数据，并存放到定义的缓冲区中  
         string sMessage = System.Text.Encoding.UTF8.GetString ( byStream  ) ;  
         //以指定的编码，从缓冲区中解析出内容  
         MessageBox.Show ( sMessage ) ;  
         //显示传送来的数据  
     }

1.2利用NetworkStream来传送信息

TcpClient tcpc = new TcpClient (  "10.138.198.213"  , 8888  ) ;  
     //对IP地址为“10.138.198.213”的计算机的8888端口提出连接申请  
    NetworkStream tcpStream = tcpc.GetStream ( ) ;  
    //如果连接申请建立，则获得用以传送数据的数据流

string sMsg = "您好，见到您很高兴" ;  
    StreamWriter reqStreamW = new StreamWriter ( tcpStream  ) ;  
    //以特定的编码往向数据流中写入数据 ,默认为UTF8编码  
    reqStreamW.Write ( sMsg  ) ;  
    //将字符串写入数据流中  
    reqStreamW.Flush ( ) ;  
    //清理当前编写器的所有缓冲区，并使所有缓冲数据写入基础流

1. 群组通信功能

组播编程需要UDP，有两个类支持组播网络编程Socket和UdpClient。一台计算机要加入某一个组，然后接收发往这个组的信息。Socket类要调用SetSocketOption函数加入和离开某一个组。UdpClient类有直接的加入和离开某个组的成员函数可以调用。而向某个组发信息，则没有什么特殊的，只需把发送数据的目的地址设为组播地址就可以了。

发送端：

            Socket s = new Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp);  
            IPEndPoint iep = new IPEndPoint(IPAddress.Parse("224.0.0.1"), 3000);  
            EndPoint ep = (EndPoint)iep;

      byte[] b = Encoding.ASCII.GetBytes("just a test!");  
            s.SendTo(b, ep);  
            s.Close();

接收端：

              Socket s = new Socket(AddressFamily.InterNetwork, SocketType.Dgram, ProtocolType.Udp);  
              IPEndPoint iep = new IPEndPoint(IPAddress.Any, 3000);  
              EndPoint ep=(EndPoint)iep;  
              s.Bind(iep);  
              s.SetSocketOption(SocketOptionLevel.IP,SocketOptionName.AddMembership,new MulticastOption(IPAddress.Parse("224.0.0.1")));  
              byte[]b=new byte[1024];  
              s.ReceiveFrom(b,ref ep);

              string test;  
              test = System.Text.Encoding.ASCII.GetString(b);

             Console.WriteLine(test);

        s.Close();  
              Console.ReadKey();

1. 广播功能

此功能和组播功能实现类似，只要在发送端获得子网中IP广播地址发送即可。

// 广播模式(自动获得子网中的IP广播地址)

broadcastIpEndPoint = new IPEndPoint(IPAddress.Broadcast, 3000);