**Socket 简单通信事例 （C#版）**

Socket 在网络通信上是很常用的类型，在本文中只做事例，并不解析原理，如果想了解原理自行到搜索引擎中查找。

MSDN 文档： http://msdn.microsoft.com/zh-cn/library/vstudio/system.net.sockets.socket.aspx

在Socket 通信是由服务端和客户端形成，可以有多个服务端或者多个客户端组成。

本文为了让一些初学者了解Socket通讯基础用法而写的。

在Socket通讯中主要使用的类有 IPEndPoint 和 Socket 类，它们是在System.Net 和 System.Net.Sockets命名空间里面。

服务端使用方法：

第一步：创建预设端口如使用本机的9050端口用来通讯

IPEndPoint ipep = new IPEndPoint(IPAddress.Any, 9050);//本机预使用的IP和端口

第二步：实例Socket容器，这里使用的是IPv4协议的TCP通讯

Socket \_Socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

第三步：绑定监听端口

\_Socket.Bind(ipep);//绑定

设置监听列队长度

\_Socket.Listen(int.MaxValue);//监听列队长度

服务端启动完成

第四步：监听新客户连接，调用 \_Socket.Accept(); 方法等待获取到一个用户连接返回该连接通讯的容器 Socket 的引用。可以再重复调用该方法获取更多的用户连接

Socket client = \_Socket.Accept();//当有可用的客户端连接尝试时执行，并返回一个新的socket,用于与客户端之间的通信

获取用户通讯地址可以通过client.RemoteEndPoint 来获取

IPEndPoint clientip = (IPEndPoint)client.RemoteEndPoint;//获取用户IP地址信息和通讯端口

string ip = clientip.Address.ToString();int port = clientip.Port;

第五步：发送信息给已连接的用户

client.Send(data);

第六步：等待接收已连接用户的信息，返回该信息的长度，参数要求一个byte[] 容器这里事例定义了长度为1024

byte[] data = new byte[1024];

int recv = client.Receive(data);

如果返回长度0就是说明用户断开连接

第七步：断开和关闭服务端

如果有客户端连接首先把所有的客户端连接断开关闭方法client.Close();

最后才关闭服务端连接 \_Socket.Close();

客户端使用方法：

第一步：创建连接端口信息

IPEndPoint ie = new IPEndPoint(IPAddress.Parse(ipadd), port);//服务器的IP和端口

第二步：创建连接容器 Socket 的实例

Socket newclient = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

第三部：使用容器连接到服务器

newclient.Connect(ie);

因为这里是客户端所以不用监听端口

第四步：接收发送信息到服务器

接收信息方法 int recv = newclient.Receive(data);

发送信息方法 client.Send(data);

该方法和服务端一样的使用方式

第五步：客户端断开和关闭连接

newclient.Shutdown(SocketShutdown.Both);

newclient.Close();

事例下载地址

文件说明

服务端 SocketServerApp.rar

客户端 SocketClient.rar

包含两套事例

第一套是 控制台模式的一个客户端通讯事例，相关项目是服务端DemoServer，客户端SocketClient

第二套是 WPF界面的模式支持多个客户端通讯，使用Socket 结合多线程的事例

服务端 SocketServerApp

客户端 SocketClientApp