

高等学校计算机专业教材

GAODENG XUEXIAO JISUANJI ZHUANYE JIAOCAI



计算机网络与安全 实用编程

● 袁津生 郭敏哲 编

GAODENG XUEXIAO JISUANJI ZHUANYE JIAOCAI
GAODENG XUEXIAO JISUANJI ZHUANYE JIAOCAI
GAODENG XUEXIAO JISUANJI ZHUANYE JIAOCAI



人民邮电出版社
POSTS & TELECOM PRESS

高等学校计算机专业教材

计算机网络与安全实用编程

袁津生 郭敏哲 编

人民邮电出版社

图书在版编目(CIP)数据

计算机网络与安全实用编程 / 袁津生, 郭敏哲编. —北京: 人民邮电出版社, 2005.5
高等学校计算机专业教材

ISBN 7-115-13377-8

I. 计... II. ①袁...②郭... III. 计算机网络—安全技术—高等学校—教材 IV. TP393.08

中国版本图书馆 CIP 数据核字 (2005) 第 027563 号

内 容 提 要

计算机网络及网络安全是高等学校计算机专业的核心骨干课程, 该课程除了系统讲述原理之外, 网络编程是不可缺少的一个重要部分。

全书共分为 7 章。主要介绍局域网聊天编程, 文件传输的概念和编程, 局域网广播的编程, 数据加密的编程, 局域网扫描的编程, 网络监听的编程, 以及网络编程的基本概念、原理等基础知识。

本书还附有光盘。提供了本书讲解的全部实例和源程序; 同时提供了两个大型综合编程实例。

本书可作为高等院校计算机专业及相关专业学生学习计算机网络和网络安全的课外参考书, 也可供从事计算机网络和网络安全工作的专业人员学习和编程参考。

高等学校计算机专业教材 计算机网络与安全实用编程

◆ 编 袁津生 郭敏哲

责任编辑 滑 玉

◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号

邮编 100061 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

读者热线 010-67170985

北京隆昌伟业印刷有限公司印刷

新华书店总店北京发行所经销

◆ 开本: 787×1092 1/16

印张: 14.75

字数: 360 千字

印数: 1—5 000 册

2005 年 5 月第 1 版

2005 年 5 月北京第 1 次印刷

ISBN 7-115-13377-8/TP · 4648

定价: 25.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

出版者的话

为了适应我国大学本科计算机专业教育发展对教材的需要,我社特邀请教育部所属的中国人民大学、中国地质大学、中国农业大学、北京科技大学、北京林业大学、北京语言文化大学(排名不分先后)6所高等学校的“计算机科学与技术系”系主任及资深教授组成专家组,规划、编写、审订了本套教材。

读者对象 本套教材的主要读者对象是普通高等院校计算机科学与技术专业的学生,兼顾信息、自动化及机电类专业学生学习计算机课程的需要。由于这些专业对学生的培养目标是:掌握计算机软件与硬件的基本理论和方法,能从事计算机应用、软件研制、技术开发和管理工作的高级技术人才,因此,本套教材的内容既注意计算机高级技术人才所应具有完整知识结构,又适当侧重主要专业知识。

教材特点 本套教材参考了美国 IEEE/CS 和 ACM 技术委员会 2001 年推荐的课程 CC2001(Computing Curricula 2001, 参见 <http://www.csab.org/~csab>, <http://cs.nju.edu.cn/~gchen/teaching/cc2001/overview-bok.html> 2001)及我国几十所高等院校计算机专业的 2001 年教学计划进行规划。教材内容在选择国际上先进的计算机理论、技术的同时,务求符合我国目前教学环境的实际情况,有一定深度,又有较高的实用性。

根据大学教学的特点,本套教材主要包括必修课程、选修课程和辅助课程三类教材。除了每本教材内容自成体系外,还考虑了它在整个教学计划中的安排顺序,适当增加了承上启下的内容。

写作风格 本套教材根据内容需要,沿着“这是什么、有什么用、怎样用、怎样用得更好”的思路编写教材,通过讲解具体知识,传授学习方法,使学生达到掌握理论和技术的目的;同时力求文笔流畅,言简意赅。

教材每一章除基本知识外,还有本章要点、小结、思考与练习题。有些教材还附加教学大纲(包括教学重点、难点,讲授知识点的参考学时数),操作性较强的课程还配有实验教材(包括上机应具备的软硬件环境和实验内容及方法)。为了方便教师教学,本套教材提供了演示稿,可到人民邮电出版社网站(<http://www.ptpress.com.cn>)的“教材出版图书出版中心”→“教材附件”→“课件”下载。

本套教材的作者都具有多年教学经验,教材的草稿也都在各自的授课过程中多次使用。这套教材的出版,为计算机专业的师生提供了新的选择。我们希望这套以易教、易学,朴实、实用为特色的教材在培养信息化建设专业人才方面做出应有的贡献。

欢迎广大读者对本套教材的不足之处提出批评和建议。

编 者 的 话

计算机网络及网络安全是计算机及相关专业的核心专业课，其特点是涉及网络及安全概念非常多，协议比较抽象、难于理解。学生在学习这些课的过程中，不容易抓住重点内容，特别是编程的能力得不到提高。传统的教材都是以讲授基本原理为主，缺乏实践环节，学生在课堂学到的内容难于应用到实践。这样的学习对学生本身起不到一个积极的作用，如何提高学生的学习积极性和编程能力是每个教师应该考虑的问题。

作为一名合格的计算机专业的大学毕业生，应具有丰富的计算机软件和硬件方面的基础知识，熟悉计算机网络基本知识，熟悉网络安全的基本知识，尤其是编程方面要具有较强的能力，这样才能提高自己的学习能力和编程水平。我们觉得在大学阶段如果不提高编程能力，那么将来在工作中就难以发挥和体现自己的能力，特别是在毕业求职的过程中就失去了许多机会和优势。因此提高编程能力对自己今后的工作具有很大的帮助。编写本书的目的主要是提高学生的编程能力和基础知识和基本概念的理解能力。

全书分为7章。第1章介绍了局域网聊天编程，加强学生对网络基本知识的运用能力；第2章介绍了文件传输的概念和编程，加强对数据通信知识的理解；第3章介绍了局域网广播的编程，加强对TCP/IP协议的理解；第4章介绍了数据加密的编程，加强学生对加密算法的理解；第5章介绍了局域网扫描的编程，加强对扫描的原理及概念的深入理解；第6章介绍了网络监听的编程，加强对网络监听概念的理解和编程的技巧；第7章介绍了网络编程的基本概念、原理等基础知识。

由于篇幅限制，将书中讲解的全部实例、源程序和两个大型综合编程实例的讲解及程序放在书中的光盘中，供读者参考。

由于作者的时间与水平有限，书中难免出现一些缺点和错误，殷切希望读者批评指正。

编者

2005年2月

目 录

第1章 局域网聊天	1
1.1 利用 VB 的 Winsock 控件实现局域网聊天	1
1.1.1 VB 的 Winsock 控件简介	1
1.1.2 程序实现原理	4
1.1.3 程序实现步骤	4
1.2 利用 Visual C++ 的 CSocket 类实现局域网聊天	8
1.2.1 程序实现原理	8
1.2.2 程序实现步骤	8
第2章 文件传输	27
2.1 简单文件传输程序	27
2.1.1 简单文件传输程序实现原理	27
2.1.2 简单文件传输程序实现步骤	28
2.2 大型文件传输程序 TransferDemo	33
2.2.1 大型文件传输程序实现原理	33
2.2.2 大型文件传输程序实现步骤	34
第3章 局域网广播	62
3.1 局域网广播简介	62
3.2 利用 net send 服务实现广播	64
3.2.1 程序实现原理	64
3.2.2 程序实现步骤	68
3.3 使用 NetAPI 函数来实现广播	70
3.3.1 Visual Basic 中如何调用 API 函数	70
3.3.2 NetMessageBufferSend 函数介绍	72
3.3.3 程序实现原理	73
3.3.4 程序实现步骤	74
3.4 采用 C/S 结构的连接型广播	75
3.4.1 程序实现原理	76
3.4.2 程序实现步骤	78
第4章 数据加密	89
4.1 古典加密方法	89
4.1.1 古典加密方法简介	89
4.1.2 古典加密程序实例	90

4.2	数据加密标准 (DES)	98
4.2.1	DES 简介	98
4.2.2	DES 实例——扩展置换和 S 盒置换演示程序	100
4.2.3	利用 VC++ 实现 DES 加密程序 DESEXAMPLE	124
4.3	RSA 加密算法	141
4.3.1	公钥密码学与 RSA 加密算法	142
4.3.2	利用 VB 实现 RSA 加密算法	144
第 5 章	局域网扫描	153
5.1	局域网扫描简介	153
5.2	扫描器实例——主机扫描	155
5.2.1	程序实现原理	155
5.2.2	程序实现步骤	158
5.3	扫描器实例——端口扫描	166
5.3.1	程序实现原理	166
5.3.2	程序实现步骤	169
第 6 章	局域网监听	179
6.1	网络监听简介	179
6.2	网络监听实例——命令行模式	181
6.2.1	程序实现原理	181
6.2.2	源代码分析	182
6.3	网络监听实例——VC++ 实现网络监听	193
6.3.1	程序实现原理	193
6.3.2	程序实现步骤	194
第 7 章	网络应用编程	211
7.1	基本概念	211
7.1.1	接口 API	211
7.1.2	网络应用 Socket 编程	212
7.2	基本接口函数的应用	216
7.2.1	服务绑定	216
7.2.2	本地地址绑定	218
7.2.3	建立接口连接	221
7.2.4	接口被动转换	223
7.2.5	从被动接口的完成队列中接收一个连接请求	225
7.2.6	基本接口 I/O 函数	226
7.2.7	关闭接口通道与撤销接口	228
参考文献		230

第 1 章 局域网聊天

聊天程序是 Socket 网络编程的基本应用形式之一，其本质是数据在通信双方的发送和接收。本章将通过两个实例分别详细介绍利用 Visual Basic (VB) 的 Winsock 控件和利用 Visual C++ 的 CSocket 实现局域网聊天程序的原理和过程。

1.1 利用 VB 的 Winsock 控件实现局域网聊天

本实例利用 VB 的 Winsock 控件来编写基于 UDP 的局域网聊天程序，下面我们先来简要了解一下 Winsock 控件的概念和使用方法。

1.1.1 VB 的 Winsock 控件简介

Visual Basic 6.0 为 Windows 环境下的网络开发提供了强大的工具，Winsock 控件就是其中之一。本程序就是利用 Winsock 控件来实现聊天程序，它使得程序的实现变得非常简单。

Winsock 控件是从 Visual Basic 5.0 开始新增加的控件，它建立在 TCP、UDP 的基础上，完成与远程计算机的通信，它已经预先在内部封装了所有的技术细节，并提供了访问 TCP 和 UDP 网络服务的方便途径。只需通过设置控件的属性并调用其方法就可轻易实现与远程计算机的连接，并且还可以双向交换数据，而这一切都不需了解 TCP/UDP 的细节或调用低级的 Winsock API。即使对 TCP/IP 不太熟悉的用户，使用该控件也可以在十几分钟内创建一个简单的客户机/服务器程序。

Winsock 控件有两个重要的属性，即 Protocol 和 State。Protocol 设定使用的协议是 TCP 还是 UDP：取值 `sckTCPProtocol` 表示 TCP，取值 `sckUDPProtocol` 则表示 UDP，Winsock 控件的缺省设置是 `sckTCPProtocol`；State 属性反映的是当前 TCP/IP 的连接状态，取值如表 1-1 所示。

如果用户对网络协议及计算机之间通信的原理有所了解的话，理论上讲用户可以利用 Winsock 控件编写任何 Internet 程序。下面介绍如何使用 Winsock 控件以及如何通过它来实现几乎所有的 Internet 协议，例如 HTTP (WWW)、FTP、NNTP (News)、SMTP 和 POP3 (Mail) 等协议的基本原理。

让我们先来看看两台计算机通信的模型，一台是客户机，另一台是服务器，所有必需的软件被装在客户端计算机上。客户端计算机可以收发电子邮件信息，浏览 WWW 网页，参加新闻组讨论以及通过 FTP 从服务器上下载文件。这些软件共存于一台计算机中并相互影响。另一种软件被安装在服务器上，这类软件向远程客户端计算机提供服务。服务器可同时响应多个客户计算机的服务请求。换句话说，服务器能在同一时间承担几个服务器的功能，如 FTP 服务器，WWW 服务器和邮件服务器。

表1-1 State属性

常 数	数值	描 述
sckClosed	0	缺省值, 关闭
sckOpen	1	打开
sckListening	2	侦听
sckConnectionPending	3	连接挂起
sckResolvingHost	4	识别主机
sckHostResolved	5	已识别主机
sckConnecting	6	正在连接
sckConnected	7	已连接
sckClosing	8	同级人员正在关闭连接
sckError	9	错误

由于 Socket 技术, 使得同时稳定运行几个 Internet 应用程序成为可能。Socket 在 MS Windows 中的应用被称为 Windows Socket 或干脆叫 Winsock。Socket 是一个程序设备, 它允许用户通过 TCP/IP 端口为某个网络应用程序接收或发送数据。程序可按需要产生足够数量的 Socket, 但是一个 Socket 必须对应一个 TCP/IP 端口。客户端计算机产生一个 Socket 并随机分配给它一个 TCP/IP 端口, 该 Socket 就利用这个端口来进行工作。但在服务器端, 却不是这样做的, 作为一条规则, 服务器端的应用程序是用预先定义好的 TCP/IP 端口来工作, 比如说: FTP 服务器的端口号为 21, WWW 服务器的端口号为 80。表 1-2 列出了常用协议的默认端口号。

表1-2 常用协议的默认端口号

协 议	端口号	描 述
SMTP	25	Simple Mail Transfer Protocol
POP3	110	Post Office Protocol
NNTP	119	Network News Transfer Protocol
FTP	21	File Transfer Protocol
HTTP	80	Hyper Text Transfer Protocol

客户端计算机和服务器是通过其中一个网络协议来开始网络进程的。客户机首先产生一个 Socket, 并利用它来同服务器建立连接。Socket 取得服务器的地址和要进行连接的端口号。对 Winsock 控件来讲, 必须用它的 Connect 方法。下面这个代码表示与一个 FTP 服务器进行连接:

```
Winsock1.Connect "ftp.microsoft.com", 21
```

该方法包括两个参数: 第一个是远程服务器的名称或 IP 地址, 第二个是端口号, 它规定

了用户想从远程服务器那里获得哪种服务。在这里，21 号端口是 FTP 服务器正在侦听的端口号。所谓侦听就是说服务器应用程序等待来自 21 号端口的客户机的服务请求。如果连接成功，Winsock 控件会产生一个 Connect 事件。

接下来的事件根据用户使用的网络协议不同而不同：

- 服务器等待用户的命令（HTTP）；
- 根据所使用的协议，服务器发送所要求的数据并断开连接；
- 服务器向客户端计算机发送欢迎信息及代码（FTP、SMTP、POP3、NNTP），表示服务器现在等待接收你的命令；

- 服务器向客户端发送不能维持现有连接的消息及代码（FTP、SMTP、POP3、NNTP）。

在连接期间，服务器会向客户端计算机发送消息，而我们要做的就是接收并处理这些消息。在 Winsock 控件中，主要通过其 DataArrival 事件来进行。在 DataArrival 事件中添加如下代码：

```
Dim strData As String
```

```
Winsock1.GetData strData
```

上面的最后一行代码的作用是从缓冲中读取服务器传来的数据并储存在 strData 变量中，然后就可以根据需要处理信息了。

与服务器建立连接并从服务器处获得第一条消息后，用户就可以通过向服务器发送命令及等待其响应的形式同服务器进行通话了。为了向服务器发送数据，用户需要用到 SendData 方法，例如在 FTP 中传送用户名和密码的命令时，使用的代码为：

```
Winsock1.SendData "USER anonymous" & vbCrLf
```

所有的命令、规则及命令顺序都可在 RFC（request for comments）文件中找到。表 1-3 给出了 RFC 文件的网址。

表1-3 协议及对应的RFC文件的网址

协 议	RFC	网 址
SMTP	821	http://www.faqs.org/rfcs/rfc821.html
POP3	1725	http://www.faqs.org/rfcs/rfc1725.html
NNTP	977	http://www.faqs.org/rfcs/rfc977.html
FTP	959	http://www.faqs.org/rfcs/rfc959.html
HTTP	2068	http://www.faqs.org/rfcs/rfc2068.html

每个协议都有自己的一套命令及回应。尽管如此，在开发程序时仍需遵循一些基本的规则：命令是大小写敏感的；一些命令需要参数，在命令与参数之间要留空格；命令必须以 vbCrLf 结尾。

来自服务器的回应以字符和数字组成。附加的文本取决于服务器所用的软件，但是数字代码是常量。用户要处理的就是这些数字，这些数字由三位数组成，通常服务器的回应以这三个数字开头，因此用 LEFT 函数就可方便地截取这三个数字。例如：

```
strResponseCode = Left$(strServerResponse, 3)
```

关于断开连接，通常用 QUIT 命令来断开同服务器的连接。发送了 QUIT 命令后，服务器会传回最后一条消息并关闭连接。对于 Winsock 而言，会产生一个 Close 事件。

最后需要注意的一点是错误处理。用户需要注意两类错误。一类是应用的协议的错误，如果使用了不正确的命令就会产生这个错误，从服务器返回的消息代码中可得知此类错误。另一类错误是 Winsows Socket 错误，对于这类错误，Winsock 控件会产生一个 Error 事件，可以使用 On Error 对错误进行处理。

1.1.2 程序实现原理

以上就是 Visual Basic 6.0 中利用 Winsock 控件实现网络协议程序的一般原理。现在对将要实现的聊天程序进行以下设计及相应的 Winsock 应用。

本聊天程序基于 UDP，在 Winsock 控件中设置 protocol 属性为 sockUDPProtocol。为了方便，将服务器和客户端集成到一起，因此若要实现双向通话就需要两边都向对方发送连接请求，建立连接后才可进行聊天。使用 Winsock 的 SendData 函数向对方发送数据，然后添加 Winsock 的 DataArrival 事件，在其中使用 GetData 函数得到对方发送过来的数据，进行处理。程序中没有设计发送按钮，使用的是输入框的 KeyPress 事件，即使用回车键进行信息发送。Winsock 控件的使用简化了程序的实现，下面介绍具体的实现步骤。

1.1.3 程序实现步骤

程序实现步骤如下：

- (1) 打开 Visual Basic 6.0，创建一个 Standard Exe 的工程，并添加图 1-1 所示的控件。

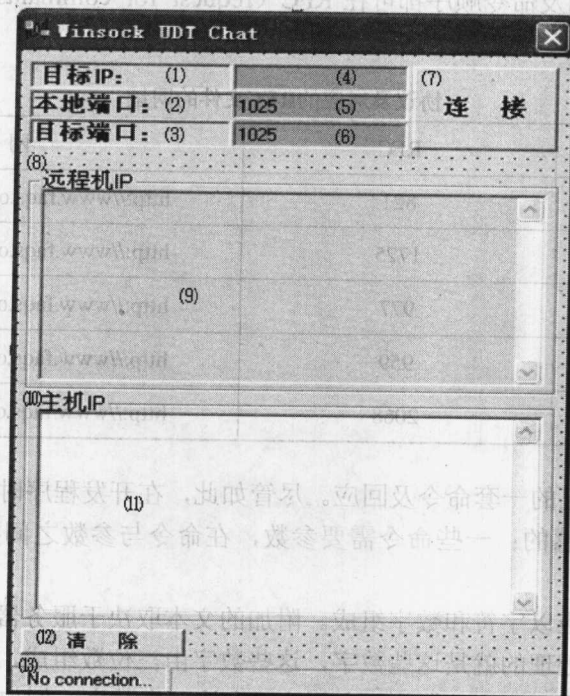


图 1-1 控件添加窗口

界面各控件属性如表 1-4 所示。

表1-4 控件属性表

序号	控件类型	Name	Caption	Text
①	Label	Label1	目标 IP	
②	Label	Label2	本机端口	
③	Label	Label3	目标端口	
④	TextBox	txtRemoteIP		
⑤	TextBox	txtLocalPort		1025
⑥	TextBox	txtRemotePort		1025
⑦	CommandButton	cmdConnect	连接	
⑧	Frame	Frame1	对方 IP	
⑨	TextBox	Text1		
⑩	Frame	Frame2	本机 IP	
⑪	TextBox	Text2		
⑫	CommandButton	cmdClear	清除	
⑬	StatusBar	StatusBar1		

最后将 Winsock 控件添加到界面中，在其属性中设置 Name 为 “winsock1”，protocol 为 “1-sckUDPProtocol”，即使用 UDP。

(2) 使用 OptionExplicit 声明 Boolean 变量 IgnoreText。在 VB 中用户可以不声明而直接使用变量，但容易产生错误。如果使用 Option Explicit 关键字强制声明变量，变量就必须先声明再引用。Option Explicit 关键字放在窗体或模块的通用声明 (GeneralDeclarations) 处。也可以通过设置将 Option Explicit 自动加到每个模块中。声明 IgnoreText 变量的代码：

```
Option Explicit
```

```
Private IgnoreText As Boolean
```

(3) 主框架 Load 事件响应函数，用于在程序主界面装载的时候使用 Winsock1 控件获得本机 IP，并赋给 txtRemoteIP 控件显示。

```
Private Sub Form_Load()
```

```
Show
```

```
txtRemoteIP = Winsock1.LocalIP
```

```
End Sub
```

(4) “连接”按钮的 Click() 事件响应函数。使用 Winsock 控件连接对方，并获取对方计算机信息。在此函数中使用 On Error 进行连接错误处理。

```
Private Sub cmdConnect_Click()
```

```
On Error GoTo ErrHandler
```

```
//在遇到错误时转到 ErrHandler 进行处理
```

```

With Winsock1                                //使用 Winsock 控件获得连接对方的信息
.RemoteHost = Trim(txtRemoteIP)              '得到远程机 IP
.RemotePort = Trim(txtRemotePort)            '得到远程机端口
If .LocalPort = Empty Then
    .LocalPort = Trim(txtLocalPort)
    .Bind .LocalPort
End If
End With

txtLocalPort.Locked = True                    //确定用户不能改变本地机端口
//显示连接状态
StatusBar1.Panels(1).Text = " Connected to " & Winsock1.RemoteHost & " "
//启动本方信息输入框 Frame 和对方信息显示框 Frame
Frame1.Enabled = True
Frame2.Enabled = True
Frame2.Caption = "本机 IP 为: " + Winsock1.LocalIP
Frame1.Caption = "对方 IP 为: " + Winsock1.RemoteHost
Text2.SetFocus
Exit Sub
ErrorHandler: //错误处理
MsgBox "Winsock 不能和过程机建立连接", vbCritical //弹出对话框显示错误
End Sub

```

(5) 添加“Text2”控件的 KeyPress 事件响应函数。用于在本方信息输入框 Frame 中按下回车键的时候, 使用 Winsock 的 SendData 将框中的信息发送给对方。

```

Private Sub Text2_KeyPress(KeyAscii As Integer)
Static Last_Line_Feed As Long
Dim New_Line As String          '输入的语句
If Trim(Text2) = vbNullString Then Last_Line_Feed = 0
If KeyAscii = 13 Then           '输入回车
    New_Line = Mid(Text2, Last_Line_Feed + 1)    '获得新的输入语句
    Last_Line_Feed = Text2.SelStart
    Winsock1.SendData New_Line
    StatusBar1.Panels(2).Text = " 发送 " & (LenB(New_Line) / 2) & " 字节 "
End If
End Sub

```

(6) 编写 Winsock 控件的 DataArrival 函数。在此函数中使用 GetData 函数接收对方传送过来的信息, 将其显示在对方信息显示框, 并在状态栏控件中显示接收的字节数。

```

Private Sub Winsock1_DataArrival(ByVal bytesTotal As Long)
Dim New_Text As String          '远程机的输入到达本地机
Winsock1.GetData New_Text       '得到远程机的输入语句

```

```
Text1.SelText = New_Text '显示
```

```
Text1.SelText = ""
```

```
StatusBar1.Panels(2).Text = " 收到 " & bytesTotal & " 字节 "
```

```
End Sub
```

(7) “清除”按钮 Click()响应函数。将对方信息显示框和本方信息输入框都清空。

```
Private Sub cmdClear_Click()
```

```
Text1 = ""
```

```
With Text2
```

```
.Text = "" '清除
```

```
.SetFocus '获得焦点
```

```
End With
```

```
End Sub
```

(8) 主框架的 KeyDown 事件响应函数

```
Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
```

```
If KeyCode = vbKeyF1 Then
```

```
ChDir App.Path
```

```
Shell "notepad.exe readme.txt", vbNormalFocus
```

```
End If
```

```
End Sub
```

代码编写完成可以编译运行，可使用本机进行测试，运行界面如图 1-2 所示。

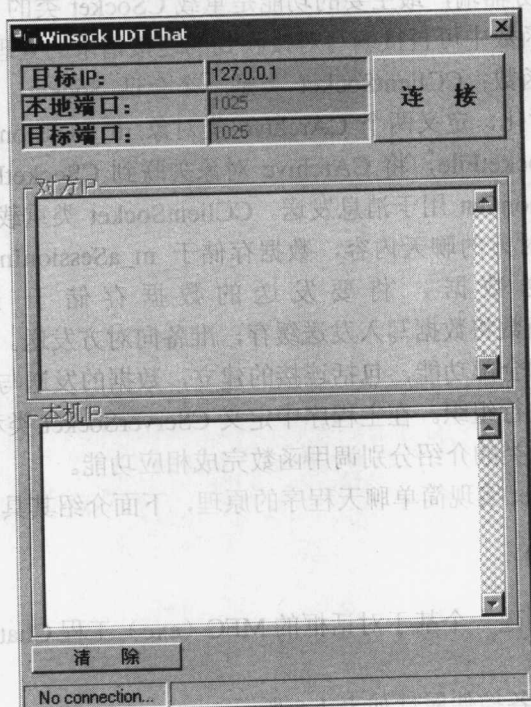


图 1-2 运行界面

1.2 利用 Visual C++ 的 CSocket 类实现局域网聊天

本实例实现在 Visual C++6.0 中利用 CSocket 类编写简单的聊天程序。

1.2.1 程序实现原理

CSocket 继承于 CAsyncSocket, 是 Windows Socket API 的高层抽象 CSocket 通常和 CSocketFile 以及 CArchive 类混合使用, 这两个类主要负责数据的发送和接收。

CSocket 提供了阻塞式的访问方式, 这一点对于 CArchive 的同步操作来说是必需的。类成员的阻塞函数, 例如 Receive()、Send()、ReceiveFrom()、SendTo()和 Accept()不会像 Winsock 中的函数一样返回 WSAEWOULDBLOCK 的错误。取而代之的是, 这些函数会自己等待直到操作完成。如果在这些阻塞函数正在等待的时候, 应用程序调用了 CancelBlockingCall 函数, 那么这些阻塞函数将会完成等待并返回一个 WSAEINTR 的错误。

要使用 CSocket 对象, 首先调用构造函数, 然后调用 Create 函数创建一个 Socket 句柄。Create 函数缺省创建一个流 Socket, 如果没有使用 CArchive 类, 那么还可以创建一个数据报 Socket。服务器端调用 Accept, 客户端调用 Connect, 然后创建一个 CSocketFile 去关联 CSocket。下一步可以创建一个 CArchive 对象关联 CSocketFile, 用来发送和接收数据。

下面以本程序为例介绍利用 CSocket 类来编写聊天程序的原理。

首先定义继承于 CSocket 类的两个子类 CServerSocket 和 CClientSocket, 其中 CServerSocket 类用于服务器端, 最主要的功能是重载 CSocket 类的 OnAccept 函数来接收客户端的连接请求, 在此函数中可直接编写对接收到的连接请求的处理代码, 也可以调用别的类中对连接请求的处理函数; CClientSocket 类是一个会话 Socket, 在此类中结合 CArchive 和 CSocketFile 进行串行化, 定义两个 CArchive 类对象 m_aSessionIn 和 m_aSessionOut 及 CSocketFile 对象 m_sfSocketFile, 将 CArchive 对象关联到 CSocketFile 对象, m_aSessionIn 用于消息接收, m_aSessionOut 用于消息发送。CClientSocket 类重载 OnReceive 函数用于接收对方发送的数据, 即对方的聊天内容, 数据存储于 m_aSessionIn, 然后使用自定义函数 SendMsg() 向对方发送数据, 将要发送的数据存储于 m_aSessionOut, 使用 m_aSessionOut->Flash()函数将数据写入发送缓存, 准备向对方发送。这两个 CSocket 子类已经包含进行聊天所需要的全部功能, 包括连接的建立, 数据的发送与接收。

最后是在主程序中进行组织。在主程序中定义 CServerSocket 类和 CClientSocket 类的对象, 然后根据以上对两个类的介绍分别调用函数完成相应功能。

以上就是利用 CSocket 实现简单聊天程序的原理, 下面介绍其具体的实现步骤。

1.2.2 程序实现步骤

(1) 利用 VC++6.0 创建一个基于对话框的 MFC (exe) 工程 ChatDemo, 并添加如图 1-3 所示的控件。

在添加控件窗口中, 各控件属性如表 1-5 所示。

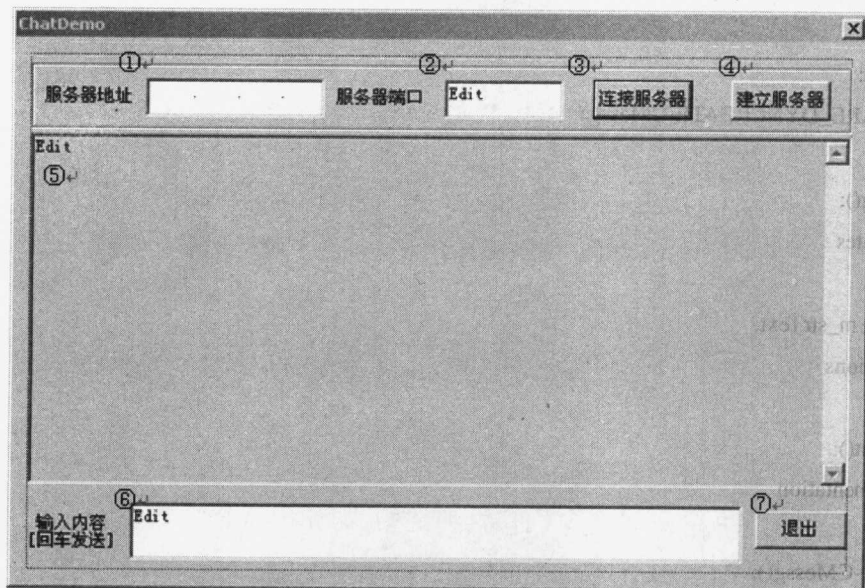


图 1-3 添加控件窗口

表1-5

各控件属性

序号	控件类型	控件 ID	变 量
①	IP Address	IDC_IPADDRESS	
②	Edit Box	IDC_PORT	UINT m_uPort
③	Button	IDC_CONNECTSERVER	
④	Button	IDC_SETSERVER	
⑤	Edit Box	IDC_SHOWTEXT	CString m_sShowString
⑥	Edit Box	IDC_INPUTTEXT	CString m_sInputString
⑦	Button	IDOK	

设置控件⑤的属性中的 Style 属性页,选中“MultiLine”、“Vertical Scroll”、“Auto VScroll”,“Border”和“Read-only”这 5 个属性,取消选择其他属性;设置控件⑥的属性中的 Style 属性页,在原有属性的基础上选中“Want return”。

(2) 自定义 CMessage 类,在 VC 编程环境的 ClassView 视图中,右键单击“ChatDemo Classes”,选择“New Class”菜单项,弹出 New Class 对话框,在“Class Type”中选择“Generic Class”,“Name”为 CMessage,“Derived from”为“CObject”,“As”为“public”,单击 OK 完成添加。CMessage 类用于封装消息,其主要目的是重载 Serialize()函数对 CArchive 对象进行串行化,CArchive 对象存储着接收到的和将要发送的数据,因此,CMessage 类也就是对聊天内容进行封装,而串行化便于接受和发送。CMessage 类代码如下:

```

/////////////////////////////////Message.h/////////////////////////////////
class CMessage : public CObject

```



```
{
protected:
    DECLARE_DYNCREATE(CMessg)
public:
    CMessg();
// Attributes
public:
    CString m_strText;
// Operations
public:
    void Init();
// Implementation
public:
    virtual ~CMessg();
    virtual void Serialize(CArchive& ar);    // overridden for document i/o
#ifdef _DEBUG
    virtual void AssertValid() const;
    virtual void Dump(CDumpContext& dc) const;
#endif
};
#endif

////////////////////Message.cpp////////////////////////////////////

#include "stdafx.h"
#include "CMessg.h"

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

// CMsg
CMessg::CMessg()
{
    Init();
}

CMessg::~CMessg()
{
}

// CMsg Operations
void CMessg::Init()
{
    m_strText = _T("");
}
```