

# Visual Basic网络高级编程

作者：汪晓平 和钟军

人民邮电出版社

Visual Basic网络高级编程

作者：汪晓平 和钟军

人民邮电出版社

ISBN : 7-115-09171-4/TP312BA

出版日期：2001年

定价：60.00元

## 内容提要

本书介绍 Visual Basic 开发高级网络应用程序的方法和技巧。本书主要针对目前流行的 FTP、HTTP、E-mail、NNTP、Telnet 等 Internet 协议进行详细地讲解，并结合大量的实例使读者能够深入地了解各种网络应用程序的开发技巧。

本书技术内容较深，实用性强，适合于中高级 Visual Basic 程序员阅读、参考。

# 前 言

Internet 有强大的通信功能,如文件传送、远程登录、E-mail、Internet Phone 和 Internet Fax 等,同时,它又是一个大型信息资源库。其所含信息不仅包罗万象,而且日新月异。尤其是 WWW (World Wide Web) 的出现使全球信息连成一体,并使千家万户可以随时共享这一人类伟大的资源。对于任何一位软件开发爱好者来说,学习开发 Internet 应用程序已经是一件刻不容缓的事情了。

软件离不开编程,而 Visual Basic (以下简称 VB) 是编程爱好者首选的编程语言,因为它简单易懂,只要写少量的代码,就可以实现许多功能。不仅是对于专业程序员,而且对于那些希望能够尽快看到自己的编程成果的业余程序员来说,VB 无疑是个很好的编程工具。同时程序员可以通过 VB 快速开发程序,迅速增加对新知识和内容的体验。

对于 Internet 编程而言,VB 无疑是一个能够快速开发网络应用程序的语言。

本书主要讲述使用 VB 进行网络高级开发的技巧和方法。与市面上许多网络开发的书不一样,本书的特点体现在以下几个方面:

1. 突出网络高级编程,重点讲解网络编程中的使用技巧及难点;
2. 针对比较流行的协议,如 Telnet、FTP、HTTP、E-mail 等进行详细的理论讲解;
3. 每一章都有丰富的实例讲解,使读者从实例深入了解各个协议的内容。

下面是本书的章节概述。

## 第 1 章:网络编程概述

本章主要介绍的是有关 TCP/IP、Winsock 编程的基础知识,让读者能对后面的程序更容易理解。

## 第 2 章:E-mail 协议及高级编程

E-mail 作为日益流行的信息传递方式,是 Internet 最常用的应用服务之一。本章首先通过 SMTP 或 POP3 了解收发 E-mail 的工作原理,结合程序讲解如何实现收发 E-mail 信件的功能,接着详细分析 E-mail 的内容结构及格式。可能最令你激动的是在这一章中详细地讲解了发送附件的 E-mail 格式以及各种编码解码的算法和相应的源程序,同时对 E-mail 的乱码进行了分析。最后讲解设计 E-mail 程序的另一种方法——功能强大的 MAPI。对于每种理论 (SMTP 发送 E-mail、POP3 接收 E-mail、用于 E-mail 的编码解码、E-mail 附件的发送、MAPI) 都有相应的实例源程序。

## 第 3 章:Telnet 高级编程

Telnet 的一个常用的功能就是登录 BBS,所以熟悉 BBS 的读者要注意这一章了。本章不但详细讲解了 Telnet 的基本理论,而且给出了 BBS 客户端的源程序,这方面的内容在以前的介绍网络开发的图书中很少提及。

## 第 4 章:HTTP 协议及高级开发

HTTP 对于读者来说一定不陌生,但是对于断点续传文件、代理服务器下载文件、Web

服务器、下载整个网站、Windows 中的 HTTP API 等内容，相信许多读者都还很陌生。这一章中不但有 HTTP 的理论内容，而且有全面的实例介绍。在本章中提供了功能类似于网络蚂蚁的源程序、WWW 服务器源程序、下载整个站点的源程序等。同时向读者揭示了断点续传及代理服务器的工作原理。

#### 第 5 章：FTP 高级编程

FTP 无疑是在局域网内使用的最多的用来下载文件的服务，本章讲述了从使用 VB 提供的 Internet Transfer 控件，到 FTP API 函数，甚至是使用 Winsock 进行较底层的通信应用开发的方法。FTP 的理论以及网络编程的思路及方式，一切都在本章中得到了最充分的体现。其中最重要的就是从底层协议来进行通信应用开发的方法，从中读者可以充分理解 FTP 的工作模式。

#### 第 6 章：NNTP 协议及高级开发

本章新闻组的理论和实例将会使你掌握 NNTP 协议和阅读新闻组客户端程序的编写。

#### 第 7 章：CGI 高级开发

编写 CGI 可以使用许多语言，VB 并不是最好的，但对于一个 VB 程序员来说，没有必要为了编写 CGI 程序而学习另外的语言。留言簿、通过浏览器的网页上传文件，VB 编写的 CGI 程序也能很好地实现这些功能。此外从 CGI 的原理上结合实例解释为什么一些 VB 编写的 CGI 程序不能用于 Microsoft 的 IIS 服务器，而只能用于 WebSite 服务器。本章并不只是简单地解释如何通过程序来实现 CGI，而是从 Web 服务器的工作原理来深入了解 CGI 的原理，而这些理论很少在其他书上提到，并且这些理论也适用于其他各种开发语言。

#### 第 8 章：ASP 高级编程

本章介绍了 ASP 的理论知识，并结合功能十分强大的网上购物系统的实例讲解了 ASP 的运用，最重要的是为读者提供了许多 ASP 的高级使用技巧。

本书的实例都是经过作者精心调试的。但是由于作者水平有限，书中错误之处在所难免，欢迎读者朋友批评指正。

作 者  
2001 年 3 月

# 目 录

第 1 章 网络编程概述 .....	1
1.1 网络的基本应用 .....	1
1.2 Winsock 规范 .....	2
1.3 TCP/IP 模型 .....	3
1.4 小结 .....	5
第 2 章 E-mail 协议及高级编程 .....	6
2.1 SMTP、POP3 与 E-mail .....	6
2.2 SMTP 及发送电子邮件 .....	7
2.2.1 SMTP 的模型描述 .....	7
2.2.2 SMTP 的会话过程 .....	7
2.3 发送无附件 E-mail 程序 .....	18
2.3.1 建立工程项目 .....	18
2.3.2 代码分析 .....	18
2.4 POP3 与接收电子邮件 .....	23
2.4.1 POP3 的模型描述 .....	23
2.4.2 POP3 的会话过程 .....	23
2.5 接收 E-mail 的程序 .....	33
2.5.1 建立工程项目 .....	33
2.5.2 代码分析 .....	33
2.6 信件结构详述 .....	47
2.6.1 RFC822 信件的格式和内容 .....	47
2.6.2 构造符合 RFC822 的信件 .....	56
2.6.3 RFC822 信件的语法分析 .....	57
2.7 MIME 编码解码与发送附件 .....	60
2.7.1 RFC822 的局限 .....	60
2.7.2 Uuencode 编码与解码 .....	60
2.7.3 MIME 及其编码 .....	66
2.7.4 构造 MIME 信件 .....	90
2.7.5 MIME 信件的语法分析 .....	93
2.8 E-mail 客户端高级编程 .....	94
2.9 E-mail 乱码 .....	99
2.9.1 乱码的常见形式及形成原因 .....	99

2.9.2	避免乱码的方法	99
2.10	MAPI 概述	100
2.10.1	Windows 的 MAPI 介绍	100
2.10.2	在 VB 中使用 MAPI	100
2.11	MAPI 高级编程	107
2.11.1	建立工程项目	107
2.11.2	代码分析	107
第3章 Telnet 高级编程		123
3.1	Telnet 简介	123
3.2	使用 Windows 的 Telnet 程序登录远程服务器	124
3.3	深入 Telnet 协议	125
3.3.1	NVT ASCII 字符集	125
3.3.2	Telnet 命令	125
3.3.3	协商选项	127
3.3.4	子协商选项	128
3.3.5	Telnet 操作方式	129
3.4	BBS 客户端高级开发	129
3.4.1	建立工程项目	130
3.4.2	关键代码分析	130
第4章 HTTP 协议及高级开发		162
4.1	HTTP 协议介绍	162
4.1.1	HTTP 背景	162
4.1.2	HTTP 的内容	165
4.1.3	消息(Message)	166
4.1.4	请求(Request)	167
4.1.5	响应(Response)	171
4.1.6	访问认证	175
4.1.7	URL 编码	178
4.1.8	HTTP 协议的应用	179
4.2	断点续传下载高级开发	179
4.2.1	建立工程项目	179
4.2.2	代码分析	181
4.3	网页服务器高级开发	211
4.3.1	Web Server 的一些理论	211
4.3.2	建立工程项目	212
4.3.3	代码分析	214
4.4	Windows 的 HTTP API	241
4.5	网站下载程序高级开发	241

4.5.1	实例介绍	241
4.5.2	WinInet HTTP API 实现文件下载的使用方法	242
4.5.3	代码分析	244
4.6	HTTP API 高级开发	269
4.6.1	实例介绍	269
4.6.2	WinInet HTTP API 实现断点续传的使用方法	269
4.6.3	关键代码分析	273
第 5 章 FTP ( 文件传输协议 ) 高级编程		295
5.1	FTP 简介	295
5.2	安装设置 FTP 服务器	296
5.3	使用 Windows 内置 FTP 程序	302
5.4	深入 FTP 协议	305
5.4.1	FTP 命令大全	305
5.4.2	FTP 工作模式	325
5.5	Internet Transfer 控件实现 FTP 程序	326
5.5.1	建立工程项目	326
5.5.2	关键代码分析	326
5.6	Winsock 开发高级 FTP 客户端程序	341
5.6.1	建立工程项目	341
5.6.2	关键代码分析	343
5.7	API 开发高级 FTP 客户端程序	387
5.7.1	建立工程项目	388
5.7.2	关键代码分析	389
5.8	三种 FTP 客户端程序开发方法的比较	408
第 6 章 NNTP 协议及高级开发		410
6.1	概述	410
6.1.1	NNTP 协议	410
6.1.2	新闻组的结构	410
6.1.3	使用 Windows 的 Outlook 阅读新闻	412
6.2	阅读新闻组的流程	414
6.3	NNTP 命令	420
6.4	NNTP 响应码	438
6.5	实例：新闻组阅读软件 NewsReader	439
6.5.1	建立工程项目	439
6.5.2	分析程序代码	440
第 7 章 CGI 高级开发		458
7.1	CGI 简介	458

7.1.1	WWW 服务的两种形式 .....	458
7.1.2	CGI 的工作原理 .....	458
7.1.3	CGI 选用的编程语言 .....	459
7.1.4	CGI 的类型 .....	459
7.1.5	CGI 的运行环境 .....	460
7.2	编写第一个 VB-CGI .....	461
7.2.1	所需的前提知识 .....	461
7.2.2	建立调试环境 .....	461
7.2.3	第一个 VB-CGI .....	462
7.3	Win CGI 文件上载及管理 .....	463
7.3.1	从模块 CGI.BAS 了解 CGI 的本质 .....	463
7.3.2	深入 CGI 高级编程 .....	480
7.3.3	建立工程项目 .....	482
7.3.4	分析程序代码 .....	483
7.4	标准 CGI 留言簿 .....	491
7.4.1	几个重要的函数 .....	491
7.4.2	运行环境 .....	493
7.4.3	建立工程项目 .....	494
7.4.4	分析程序代码 .....	494
第 8 章 ASP 高级编程 .....		504
8.1	ASP 简介 .....	504
8.1.1	ASP 的特点 .....	505
8.1.2	ASP 的运行环境 .....	506
8.2	ASP 高级使用技巧 .....	506
8.3	“网上购物”系统的高级开发 .....	540
8.3.1	系统设计 .....	540
8.3.2	建立工程项目 .....	544

# 第 1 章 网络编程概述

## 1.1 网络的基本应用

计算机的诞生无疑给人类的学习方式带来了巨大的变化，让人类从繁琐的工作中解脱出来。而网络的诞生却给人类的生活带来了根本的变化，许多人们根本没有想到过的事情现在就很容易地发生在身边。目前网络的应用有许多方面，而对于广大的编程人员来说，应用最为广泛的就是以下几种协议的开发：

- (1) HTTP 协议
- (2) FTP 协议
- (3) Telnet 协议
- (4) E-mail(SMTP 和 POP3)协议
- (5) NNTP 协议

当然还有其他的许多协议，以上列举的只是目前比较常见的一些协议。其实在实际的开发中，完全可以根据需要，自定义开发各种不同的协议以满足自己的需要。在本书中所涉及到的网络编程，多数是网络的高级应用，而不涉及到网络的底层编程。网络程序的开发是包括许多方面的，在后面读者会知道，ISO 把网络结构划分为 7 层，每一层都可以进行程序开发。通常说的网络编程也就是最上层即应用层程序的开发。

上面列举了 5 种开发协议，从这 5 种开发协议可以引申出许多的网络开发。其中最为复杂的就是 HTTP 协议的程序开发，它涉及了很多方面，当然它的功能也是最强大的。目前，在网络上，最流行的也莫过于 WWW 应用，而它正是基于 HTTP 协议的。通过 HTTP 协议，可以进行许多方面的编程，比如 WWW 服务器的开发、网络浏览器的开发、代理服务器开发、CGI 的开发、ASP 开发(PHP、JSP 等)、网络搜索引擎开发等等。而这些都是作为一个优秀的程序员所必须掌握的。

本书的目的是要让一个普通的程序员成为一个网络编程的高手。本书基本上是从上面列举的 5 个方面来进行讲述的，并不是仅仅停留在应用上或者只是简单地用一些代码来说明，而是从理论和工作的底层协议来进行说明的，让读者能够明白每一个协议或者应用的来龙去脉。

书后面的各章主要是利用 Visual Basic(以下简称 VB)来开发这些应用，但是作为一个程序员，仅仅知道这些应用是不够的，还应该对稍微底层的知识了解一点，这也就是我们通常所说的 TCP/IP 协议。事实上，上面所有列举的所有协议的开发都是建立在该协议基础上的，但是在开发中，并不需要理会 TCP/IP 协议是怎样工作的。因为在程序开发中，使用的是系统已经给我们建立好的 SOCKET 对象来实现的。为了能让读者了解 TCP/IP 和 SOCKET 对象的有关知识，接下来介绍一些 TCP/IP 的基本理论。

## 1.2 Winsock 规范

我们目前绝大部分的应用层程序都是利用 SOCKET 来进行开发的，在 Windows 操作系统中，又称为 Winsock 开发。对于众多的底层网络协议，Winsock 是访问它们的首选接口。注意在每个 Win32 平台上，Winsock 都以不同的形式存在着。Winsock 是网络编程的接口，而不是协议，这一点读者要搞清楚。常用应用层的协议就是基于 Winsock 接口而实现的。

Windows Sockets 规范以 U.C. Berkeley 大学 BSD UNIX 中流行的 Socket 接口为范例定义了一套 Microsoft Windows 下网络编程接口。它不仅包含了人们所熟悉的 Berkeley Socket 风格的库函数，也包含了一组针对 Windows 的扩展库函数，以使程序员能充分地利用 Windows 消息驱动机制进行编程。在 Win32 环境中，Winsock 接口最终成为一个真正与协议无关的接口，尤其是在 Winsock 2 发布之后。

Windows Sockets 规范本意在于提供给应用程序开发者一套简单的 API，并让各家网络软件供应商共同遵守。此外，在一个特定 Windows 版本的基础上，Windows Sockets 也定义了一个二进制接口(ABI)，以此来保证应用 Windows Sockets API 的应用程序能够在任何网络软件供应商的符合 Windows Sockets 协议的实现上工作。因此这份规范定义了应用程序开发者能够使用，并且网络软件供应商能够实现的一套库函数调用和相关语义。

遵守这套 Windows Sockets 规范的网络软件，我们称之为 Windows Sockets 兼容，而 Windows Sockets 兼容实现的提供者，我们称之为 Windows Sockets 提供者。一个网络软件供应商必须百分之百地实现 Windows Sockets 规范才能做到 Windows Sockets 兼容。

任何能够与 Windows Sockets 兼容实现协同工作的应用程序就被认为是具有 Windows Sockets 接口，我们称这种应用程序为 Windows Sockets 应用程序。

Windows Sockets 规范定义并记录了如何使用 API 与 Internet 协议族(IPS，通常我们指的是 TCP/IP)连接，尤其要指出的是所有的 Windows Sockets 实现都支持流套接口和数据报套接口。

应用程序调用 Windows Sockets 的 API 实现相互之间的通信。Windows Sockets 又利用下层的网络通信协议功能和操作系统调用实现实际的通信工作。

建立 Winsock 2 规范的主要目的是提供一个与协议无关的传送接口。在本书中，基本上都是利用 Winsock 对象来实现各种应用程序开发的，而并不需要知道数据在网络中是如何传输的。关于该对象，在 MSDN 中有详细的介绍，因此本书中也不准备介绍该控件或者对象的使用。

下面介绍一下什么是套接字。

实际上通信的基础是套接字。一个套接字是通信的一端，在这一端上可以找到与其对应的一个名字。一个正在被使用的套接字都有它的类型和与其相关的进程。套接字存在于通信域中。通信域是为了处理一般的线程通过套接字通信而引进的一种抽象概念。套接字通常和同一个域中的套接字交换数据(数据交换也可能穿越域的界限，但这时一定要执行某种解释程序)。Windows Sockets 规范支持单一的通信域，即 Internet 域。各种进程使用这个域互相之间用 Internet 协议族来进行通信(Windows Sockets 1.1 以上的版本支持其他的域，例如 Windows

Sockets 2)。

套接字可以根据通信性质分类；这种性质对于用户是可见的。应用程序一般仅在同一类的套接字间通信。不过只要底层的通信协议允许，不同类型的套接字间也照样可以通信。

用户目前可以使用两种套接字，即流套接字和数据报套接字。流套接字提供了双向的、有序的、无重复并且无记录边界的数据流服务。数据报套接字支持双向的数据流，但并不保证是可靠、有序、无重复的。也就是说，一个从数据报套接字接收信息的进程有可能发现信息重复了或者与发出时的顺序不同。数据报套接字的一个重要特点是它保留了记录边界。对于这一特点，数据报套接字采用了与现在许多包交换网络(例如以太网)非常类似的模型。

## 1.3 TCP/IP 模型

TCP/IP 协议其实是协议的集合。其名称代表它的两个基本协议，TCP(Transmission Control Protocol, 传输控制协议)和 IP(Internet Protocol, 网际协议)。虽然从名称上来说只有两个协议，其实它包含了 4 层协议。

### 1. TCP/IP 协议的分层

TCP/IP 通常被认为是一个 4 层协议系统，由以下 4 层组成：

- (1) 应用层
- (2) 传输层
- (3) 网络层
- (4) 链路层

注意这里只有 4 层，它同 ISO 规定 7 层协议 OSI(开放系统互连)是有所区别的，稍后再介绍 OSI 系统。上面提到的每一层都有不同的功能。

- 链路层

有时候也称为数据链路层或者网络接口层，通常包括操作系统中的设备驱动程序和计算机中对应的网络接口卡。它们一起处理电缆(或者其他任何传输媒介)的物理接口细节。

- 网络层

有时候也被称为互联网层，处理分组在网络中的活动，例如分组的选录。在 TCP/IP 协议族中，网络层包括 IP 协议(网际协议)、ICMP 协议(网际控制报文协议)，以及 IGMP 协议(Internet 组管理协议)。

- 传输层

传输层主要为两台主机上的应用程序提供端到端的通信。在 TCP/IP 协议族中，有两个互不相同的传输协议：TCP(传输控制协议)和 UDP(用户数据报协议)。TCP 为两台主机提供可靠的数据通信。它所做的工作包括把应用程序交给它的数据分成合适的小块交给下面的网络层，确认接收到的分组，设置发送最后确定分组的超时时钟等。由于传输层提供了高可靠性的端到端的通信，因此应用层可以忽略所有的细节。而另一方面，UDP 则为应用层提供一种非常简单的服务。它只是把称作数据报的分组从一台主机发送到另外一台主机，但并不保证该数据报能到达另一端。任何必需的可靠性必须由应用层来提供。

- 应用层

应用层负责处理特定的程序细节，在本书中的所有例程全部是在应用层进行的。

## 2. OSI 模型

OSI 网络模型主要是由 ISO(国际标准化组织)规定的一套协议。OSI 为 Open System Interconnection 的简写。实际上该模型是源自 TCP/IP 协议，但比 TCP/IP 模型多了 3 层，一共有 7 层协议。具体分层如下：

- (1) 应用层(使用网络应用程序)
- (2) 表示层(应用程序的标准化数据表示)
- (3) 会话层(管理应用程序之间的会话)
- (4) 传输层(提供端对端的错误检测和校正)
- (5) 网络层(为上一层管理网络之间的连接)
- (6) 数据链路层(提供跨越物理链路的可靠数据传输)
- (7) 物理层(定义网络媒体的物理特性)

其中应用层、传输层、网络层以及数据链路层同 TCP/IP 协议的定义相同。物理层上有电缆、电线和其他的通过数据传递的电子附件。物理层的任务是要使线路正确地设计与实现。数据传送时应该没有噪音和干扰，而信号应该很强。会话层被用在 LAN(局域网)中并负责一些事务，如当访问网络时进行管理，使得没有两个用户在同一时间试图访问它；向会话用户提供面向应用程序的数据流等。表示层决定了由应用程序传输数据的格式。例如，ASCII 是传输数据最常用的格式，任何数据加密，例如 PGP 都是在表示层进行的，数据压缩和解压也是在这一层进行的。

## 3. 网际地址

互联网上的每一个接口必须有唯一的 Internet 地址也就是 IP 地址。一个 IP 地址通常由 4 个用点号隔开的 8 个二进制位的数字(从 0 到 255)组成的，地址总长位 32bit。目前 IP 地址共分为五类即 A、B、C、D、E。其中具体的分布如下：

- A 类：从 0.0.0.0 到 127.255.255.255
- B 类：从 128.0.0.0 到 191.255.255.255
- C 类：从 192.0.0.0 到 223.255.255.255
- D 类：从 224.0.0.0 到 239.255.255.255
- E 类：从 240.0.0.0 到 247.255.255.255

注意，多接口的主机具有多个 IP 地址，其中每个接口都对应一个 IP 地址。由于互联网上的每一个接口必须有一个唯一的 IP 地址，因此必须要由一个管理机构为进入互联网的网络分配 IP 地址。这个管理机构就是互联网络信息中心，称作 InterNIC。

通常在记忆 IP 地址的时候比较困难，因此在实际的应用中，会经常使用到域名。在进行网络程序开发的时候，必须要给出远程主机的地址，一般可以使用 IP 地址或者使用域名。

这里要注意，域名通常要经过 DNS(域名服务器)的解析，所以比直接使用 IP 地址会慢一些。因为网络不理解域名，只能理解 IP 地址。将域名解析为 IP 地址的方法通常有 2 种。古老的方法是将所有的域名与 IP 地址存放在一个名为 HOSTS 的文本文件中。在 Unix 中，这个表格可从一个共享目录中访问。在 Windows 95 中，这个文件位于网络上的每一台机器的

“ Windows ” 目录中。在 Windows NT 中，它驻留在 “ WINNT\System32\Drivers\etc ” 目录下。如果需要，可以由用户来编辑这个文件。

这个方法的缺点是非常明显的。因为如果每个人都使用一个 HOSTS 文件作为域名解析，就不得不把自己的 IP 地址和名称给别人，而其他人都必须更新，因此该系统只能在局域网内实现。目前实现名称解析的方法是使用 DNS(域名服务)系统。DNS 是一个分布式的数据库，它包含了所有注册过的 Internet 主机的 IP 地址。域名也是由 InterNIC 来管理的。

#### 4. 端口

在 TCP/IP 连接过程中，都是通过采用 16 位端口号来识别的。因为 IP 地址只是标志了一台机器在网络中的位置。而 IP 端口是对应了一个主机上运行的应用程序。服务器一般都是通过知名端口号来识别应用的。例如对于每个 TCP/IP 实现来说，FTP 服务器的 TCP/IP 端口号是 21、Telnet 的端口号是 23、TFTP(简单文件传送协议)服务器的 UDP 端口号是 69。任何 TCP/IP 实现所能提供的服务都用知名的 1 ~ 1023 之间的端口号。这些端口号由 Internet 号分配机构 (IANA)来管理。

实际上，客户端对它所使用的端口号并不关心，只需要保证该端口号在本机上是唯一的就可以了。注意，在同一个服务器上只能同时启动一个端口服务。客户端口号又称作临时端口号(即存在时间很短)。这是因为它通常只是在用户运行该客户程序时才存在。而服务器只要没有停止服务，就一直运行并侦听连接。

大多数的 TCP/IP 实现给临时端口分配 1024 ~ 5000 之间的端口号。大于 5000 的端口号是为其他服务预留的。

## 1.4 小 结

本章并没有介绍具体的编程细节，而是介绍了一些编程的基本概念。本书主要是在 TCP/IP 协议的基础上进行 Winsock 编程，所以着重介绍了与 TCP/IP 相关的一些知识。比如地址、端口、Winsock 等基础知识。这样对于理解后面的程序会有一定的帮助。由于本书并没有讲解本书中使用的 Internet Transfer 控件和 Winsock 控件，读者可以参照 MSDN 中的帮助来学习。

## 第 2 章 E - mail 协议及高级编程

### 2.1 SMTP、POP3 与 E-mail

电子邮件是当今 Internet 使用率最高的服务之一，上网的主要目的是获取信息和交流，电子邮件作为一个快速交流的媒介，发挥着很大的作用。

电子邮件是什么？它又是如何操作的？在最高的层次它很简单，使用 Outlook 或 Foxmail 等收发电子邮件的软件可以使我们轻易地发送一封邮件和获取信箱的邮件。我们完全可以不用知道软件是怎么发送和取得 E-mail，只要知道怎么使用就行了。在最低的层次，它又很复杂，涉及到许多协议的实现。作为一个程序员，要编制收发 E-mail 的程序，必须对 E-mail 的工作过程了解得很清楚才能实现。

为什么要从 SMTP 和 POP3 开始了解 E-mail 呢？对于想编写 E-mail 的程序员来说，通常主要是对怎么收发电子邮件感兴趣。事实上，网络上传送 E-mail 所涉及的程序很多。打个比方，网络上的 E-mail 传送有点像我们日常生活中发送的普通信件一样，我们首先把信件交给邮局，在信封上告诉邮局你的信件是要寄到哪里去的，再由邮局传递（寄到比较远的地方，如外省，可能要经过好几个邮局传递），最后投递到收信人的信箱中，收信人只要凭着他的信箱钥匙就可以拿到信件。同样的一封 E-mail 从用户的手中发出，首先要有一个程序，把这封 E-mail 发到 SMTP 服务器（发送邮件服务器）上，再由发送邮件服务器负责传递邮件到达目的信箱。然后再需要一个程序接收信箱的邮件，当然也不是只要随便有一个程序就可以收的，还需要帐号（就像收信人的姓名）和口令（信箱钥匙）。以此看来很明显发送邮件至少需要 4 个处理不同任务，有不同功能的程序：发送邮件到服务器的程序，我们把它叫做发送邮件客户端程序；负责在网络上传送邮件到信箱的程序，我们称之为 SMTP 服务器程序；接收并储存邮件并给用户提取的服务器程序，我们称之为 POP3 服务器程序；从 POP3 服务器上收取邮件的程序，我们称之为接收邮件客户端程序。（当然有些功能可以合并在同一个程序里，如发送邮件和接收邮件）。

通常我们进行 E-mail 的程序编程工作集中在收发电子邮件的客户端程序上。怎么样与 SMTP 服务器进行对话发送邮件？怎么样与 POP3 服务器进行对话将在信箱的邮件内容接收下来？我们都知道，对话要有彼此知道的一套协议标准，这样才能理解对方的意思，Internet 上的通信都有彼此的一套协议，可能读者或多或少都听说过诸如 TCP/IP，HTTP，FTP 的协议名称。收发电子邮件也是有一套标准的会话协议的，其中与发送有关的协议叫 SMTP（Simple Mail Transfer Protocol 简单邮件发送协议），这个协议规定了与 SMTP 服务器进行对话的一系列命令及过程标准。与接收有关的协议叫 POP3 协议（Post Office Protocol 邮局协议）该协议规定了与 POP3 服务器进行对话的一系列命令及过程标准。因此有必要从这两个协议开始了解收发 E-mail 的原理及过程。

## 2.2 SMTP 及发送电子邮件

### 2.2.1 SMTP 的模型描述

SMTP 模型采用的也是现在流行的 C/S(客户机/服务器)模式,用户直接使用的是用于编写和发送的客户端软件,而通常的 SMTP 服务器运行在远程站点上。客户机/服务器之间的通信是通过 TCP/IP 协议进行的。

前面介绍过,使用 SMTP 服务器并不是邮件的目的地,它只是邮件的中间传递机构。发送邮件的客户端软件不用了解如何把邮件发送到目的信箱的服务器上,只要告诉具有传递机制的 SMTP 服务器一些必要的信息,接下来怎么投递邮件就是 SMTP 服务器的事情了。

### 2.2.2 SMTP 的会话过程

一开始就讲 SMTP 的命令和标准是没多大意思的,读者可能会被这大批命令弄得满头雾水。还是先让我们看一段比较简单的 SMTP 对话吧。

```
(connect to the SMTP server.....)
220 BigFox ESMTP service ready (MDaemon v2.1 rB b1 32-T)
HELLO zhongjun
250 ZHONG Hello worldcomputers.com, pleased to meet you
MAIL FROM:zhong@10.10.10.10
250 <from: zhong@10.10.10.10>, Sender accepted
RCPT TO: zhjuna@263.net
250 < zhjuna@263.net>, Recipient ok
(发送邮件内容)
354 Enter mail, end with <CRLF>.<CRLF>
(邮件发送完毕, 发送 crlf & . & crlf)
250 Ok, message saved
QUIT
221 See ya in cyberspace
```

在上面的对话过程中,黑体字部分是发送邮件的客户端软件发送的内容,其他部分是 SMTP 服务器的应答内容。从以上的这段简单的会话内容中我们可以明显地看出以下的一些特点:

- (1) 会话的过程全部是发送文本来完成的,过程为交互式的请求应答模式;
- (2) 命令是文本形式的命令;
- (3) 每次会话服务器总是返回一定的响应码,表示客户端的请求是否被正确地回答;
- (4) 会话过程有一定的顺序。

到这里可能读者急于想知道如何连接 SMTP 服务器、如何与服务器对话、对话的命令和含义是什么、如何从服务器的响应码中知道服务器的意思、等等。下面就这些问题进行详细

的阐述。

### 1. 连接服务器

E-mail 的通信过程是基于 TCP/IP 协议的，所以在 Windows 中采取 Winsock 就可以达到和服务器进行通信的目的。在 Visual Basic 中可以使用 Winsock 控件很容易的实现。在这里有必要对 Winsock 的通信过程进行简单的介绍。

利用 Winsock 控件进行 TCP 通信(对话)通常包含以下的几个步骤：

- (1) 设置服务器在指定 IP 端口监听客户端的连接请求
- (2) 告诉客户端连接到特定 IP 地址和端口的服务器
- (3) 服务器通过接受连接响应连接请求
- (4) 客户端和服务器向对方发送数据
- (5) 客户端或服务器读取缓冲区中的数据
- (6) 必要时重复 4 和 5
- (7) 关闭客户端和服务器之间的连接

对于我们这个要与 SMTP 服务器进行通信的客户端程序，所要做的事情就是设置 Winsock 连接的 IP 地址或域名，指定端口号。SMTP 的端口号一般为 25。在 VB 中形式如下：

```
Winsock.Protocol = sckTCPProtocol
Winsock.RemoteHost = "10.10.10.10"
Winsock.RemotePort = 25
Winsock.Connect
```

这样如果指定的地址运行着 SMTP 服务器程序，那么就可以与服务器进行通信了。

读者可以在 VB 中试试看，指定一个知道的发送邮件服务器地址，然后连接上去，看看接收到的从服务器返回的响应文字，如果是类似：

```
220 BigFox ESMTP service ready (MDaemon v2.1 rB b1 32-T)
```

的文字，那就对了，说明你已经成功地实现了第一步，接着往下看。

### 2. 发送命令及接收响应信息

命令是利用 Winsock 控件在连接上服务器之后发送的，如：

```
Winsock.SendData "HELO ZHONG" & vbCrLf
```

其中 HELO 是编写的程序(客户端程序)与 SMTP 服务器对话的命令，ZHONG 是 HELO 命令参数，在每个命令的结尾加上回车换行符(VB 中使用常数变量 vbCrLf 表示)。

如果服务器接收客户机的命令，会给客户机返回一个响应码和消息行，告诉请求的命令是否被准确执行。在 VB 编写的程序中，利用 Winsock 控件与服务器进行通信，客户机接收到服务器的数据会触发该控件的 Wsock\_DataArrival 事件。可以在该事件中取得服务器返回