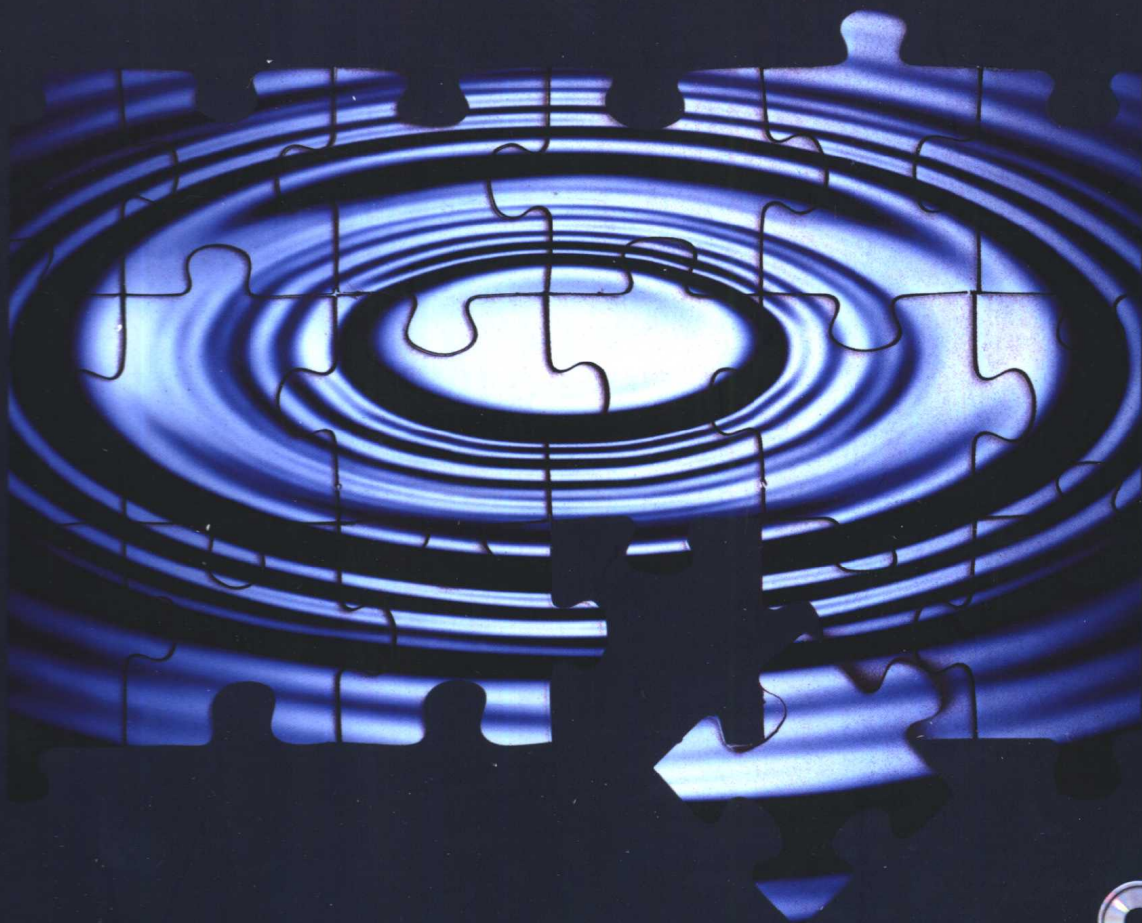




# Windows

# 网络编程

求是科技 黄超 编著



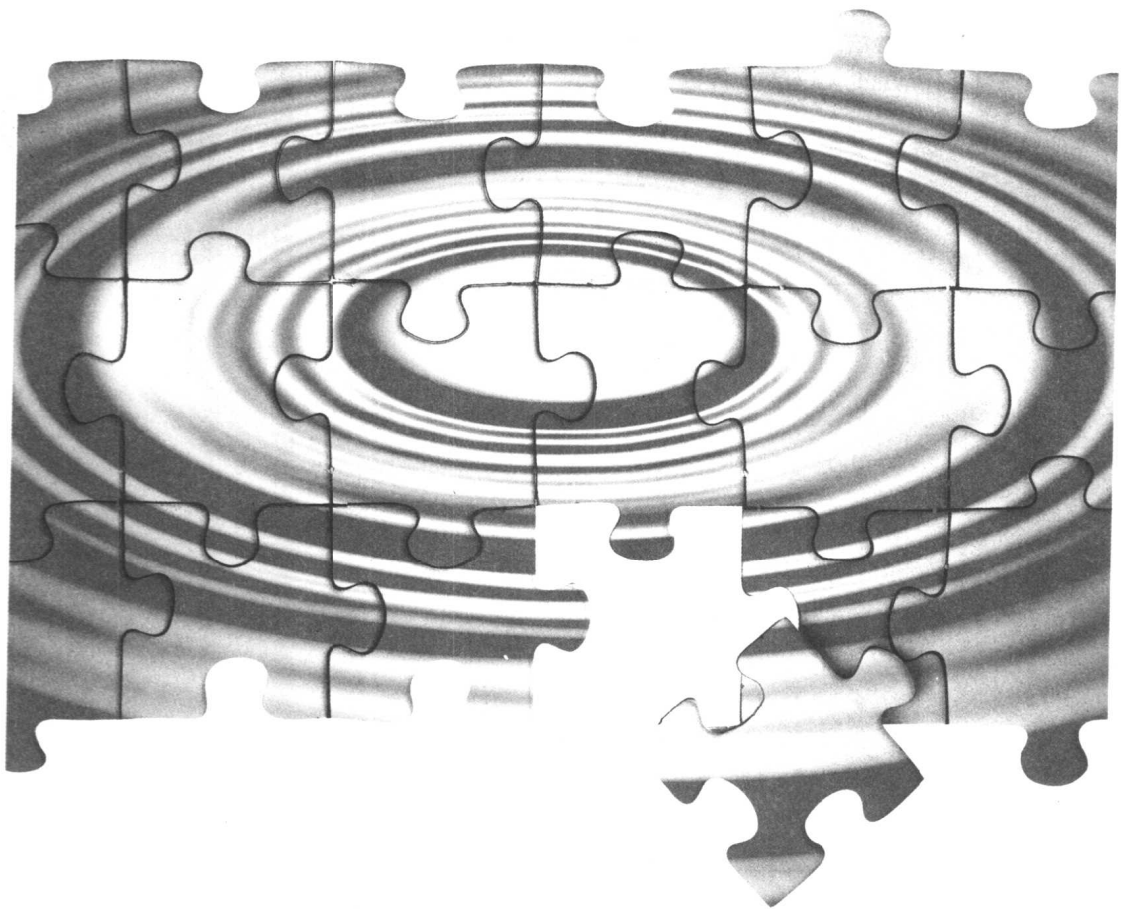
源代码光盘  
CD-ROM

人民邮电出版社  
POSTS & TELECOMMUNICATIONS PRESS

核  
心  
编  
程  
系  
列

# Windows 网络编程

求是科技 黄超 编著



人民邮电出版社

## 图书在版编目 (CIP) 数据

Windows 网络编程 / 黄超等编著. —北京: 人民邮电出版社, 2003.1

ISBN 7-115-10961-3

I. W... II. 黄... III. 窗口软件, Windows—程序设计 IV. TP316.7

中国版本图书馆 CIP 数据核字 (2002) 第 093537 号

## 内 容 提 要

Microsoft 公司的 Visual C++ 平台不仅具有很强的网络程序处理能力, 而且还能提供多种编程风格。与 Visual C++ 捆绑的 MFC 提供了大量封装良好的网络类, 如 Winsock、WinInet、ActiveX、DCOM 和 ATL 等, 提高了用户在利用 Visual C++ 进行网络开发时的效率。

本书主要特色在于对技术细节做了深入的介绍, 探讨了大量网络专用函数以及 MFC 类, 并且仔细分析了编程中可能存在的问题和难点, 同时将这些知识应用于实际开发中, 利用这些理论知识指导综合实例的开发。通过对这些编程案例和编程技巧的学习, 可以让读者清楚地感受到笔者的开发意图, 并且能将这些经验同读者自己的开发实践结合起来。本书的另一个特点在于从传统的网络 API 开始讲述, 进而到高级的 Winsock API, 然后介绍如何利用 Visual C++ 强大的网络开发功能, 在更高级的层次上建立网络应用程序, 减少开发工作量。让读者对于 Windows 网络应用程序开发的发展过程有比较清楚的认识, 深入理解 Windows 网络服务的运行机制以及开发流程。

对于需要进行 Windows 网络编程的人员来说, 本书极具参考价值, 是一本不可多得的参考书。

核心编程系列

### Windows 网络编程

- 
- ◆ 编 著 求是科技 黄 超  
责任编辑 张立科
  - ◆ 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 010-67132692  
北京汉魂图文设计有限公司制作  
北京密云春雷印刷厂印刷  
新华书店总店北京发行所经销
  - ◆ 开本: 787×1092 1/16  
印张: 44  
字数: 1 364 千字 2003 年 1 月第 1 版  
印数: 1-5 000 册 2003 年 1 月北京第 1 次印刷

ISBN7-115-10961-3/TP·3280

定价: 72.00 元 (附光盘)

本书如有印装质量问题, 请与本社联系 电话: (010) 67129223

## 1. 关于 Windows 网络编程

最近几年，计算机网络的发展一日千里，Internet 已经步入千家万户，人们对于网络服务应用程序的要求也越来越高、越来越多。Microsoft 公司的 Visual C++ 平台不仅具有很强的网络程序处理能力，而且还能提供多种编程风格，与 Visual C++ 捆绑的 MFC 提供了大量封装良好的类，使用户在使用 Visual C++ 时有如虎添翼的感觉。它提供了丰富的网络编程技术，例如 Winsock、WinInet、ActiveX、DCOM 和 ATL 等，为用户开发网络应用程序提供全方位支持。

本书主要特色在于对技术细节做了非常深入的介绍，探讨了大量网络专用函数以及 MFC 类，并且仔细分析了编程中可能存在的问题和难点，同时进一步介绍了功能强大的综合开发实例。通过对这些编程案例和编程技巧的学习，可以让读者清楚地感受到笔者的开发意图，并且能将这些经验同读者自己的开发实践结合起来。本书的另一个特点在于从传统的网络 API 开始讲述，进而到比较高级的 Winsock API，然后介绍如何利用 Visual C++ 强大的网络开发功能，在更高级的层次上建立网络应用程序，减少开发工作量。让读者对于 Windows 网络应用程序开发的发展过程有比较清楚的认识，深入理解 Windows 网络服务的运行机制以及开发流程。

## 2. 本书的结构和内容

本书是一本专门讲解 Windows 网络编程的书，并以“由浅入深”、“相互贯穿”、“重点突出”、“文字叙述与典型代码实例相结合”为原则，向每一位 Windows 编程爱好者全面介绍 Windows 网络编程的基本思路、函数和类库支持以及在应用程序编写中的具体运用。

本书全书总共分为 12 章，首先介绍了计算机网络方面的基础知识，包括网络体系结构和局域网基本构造等内容。然后介绍了传统网络 API 开发，主要包括 NetBIOS、重定向器、邮槽和管道，接着深入介绍了网络协议 TCP/IP 以及 Visual C++ 对它的支持情况。然后介绍了局域网编程，接着介绍 Winsock API 和 MFC Winsock 开发、WinInet API 和 MFC WinInet 开发、UDP 协议开发，随后介绍了具体网络协议的实现及应用开发。最后介绍了几个实际应用的综合实例。

本书各章的内容具体是这样安排的：

“第 1 章 计算机网络体系结构”，本章主要介绍了网络与通信的基础知识，特别是关于 Internet 的基础知识，以便于进一步学习网络编程。

“第 2 章 NetBIOS 编程”，本章主要介绍了 NetBIOS 的接口功能，它的优点在于“与协议无关”，应用程序可通过 TCP/IP、NetBEUI 以及 SPX/IPX 运行。本章最后介绍了一个搜索指定网段内计算机的应用，通过该程序可以搜索指定网段内的计算机信息。

“第 3 章 重定向器、邮槽和管道”，本章介绍了 Windows 重定向器，解释了重定向器在网络上的基本通信方法，并且讨论了在应用程序使用重定向器时，由 Windows NT 和 Windows 2000 提供的一系列安全性。接着介绍了邮槽网络编程技术，它可以通过网络将一条消息广播给一台或多台计算机。随后介绍了命名管道网络编程技术，它为我们建立了一个简单的客户端/服务器数据通信体系，它直接利用了 Windows NT

及 Windows 2000 的安全机制，可确保数据进行可靠传输。

“第 4 章 网络协议 TCP/IP”，本章介绍了开发应用程序选择网络传输时应该知道的基本特性，同时介绍了枚举系统支持的网络协议、选择网络协议、IP 包监视和 TCP/IP 信息统计 4 个实例。这 4 个实例在应用开发的层次上用来理解 TCP/IP 的有关内容。比如 IP 包监视程序经过进一步修改，可以用于过滤局域网上的不良信息，同时监控局域网外的不友善的访问等。

“第 5 章 局域网编程”，本章主要讨论了局域网编程所涉及到的主要内容，包括局域网拓扑结构和体系结构、网络接口卡的硬件编址等内容，同时讲解了 3 个获取网卡信息的例子，比较了局域网编程的多种实现方法，比如使用封装好的 IPHlpApi 和 MIBII 库或调用传统的 Winsock API。

“第 6 章 Winsock 基础”，本章介绍了面向连接和无连接协议所需的一些 Winsock 函数。针对面向连接的协议介绍如何接受客户端连接、如何建立客户端与服务器的连接。另外还介绍了面向会话的数据收发操作。

“第 7 章 Winsock API 高级编程”，本章全面介绍了可用于 Winsock 的各种 I/O 模型。这些模型从简单的锁定 I/O，到高性能的完成端口 I/O，使不同的应用程序可根据自己的要求，对 Winsock I/O 的性能进行充分的调整。本章也提供了对 Winsock 一些常规用途的解释。同时还介绍了如何通过 Winsock，利用原始套接字来创建 ICMP 和 IGMP 应用程序。最后采用本章介绍的知识实现了枚举 TCP 和 UDP 协议和 Ping 实例这两个应用程序。

“第 8 章 MFC Winsock 高级编程”，本章介绍了利用 Winsock API 创建了支持同步工作方式的 Winsock 类，同时介绍了实现提供简单 HTTP 服务的程序的算法和实现过程。本章重点在于深入理解 Web 和 HTTP 协议的工作原理，并在此基础上实现 HTTP 服务器的基本功能。

“第 9 章 深入 UDP”，本章对多播技术进行了定义，并向大家展示了各种不同的多播模型，探讨了 IP 多播和 ATM 点到多点通信如何应用于这些模型，解释了 Winsock API 具体如何提供对多播通信的支持。同时介绍了 Winsock 1 和 Winsock 2 的方法。本章介绍了一个类似聊天的程序，但它使用的是多播 UDP 协议，和前面的基于 TCP/IP 的聊天程序完全不同，本章最后还介绍了一个语言全双工通信的程序。

“第 10 章 WinInet API 开发”，本章介绍了在没有必要知道基本协议（例如 FTP、HTTP 或 Gopher）的情况下，如何使用 WinInet API 把 Internet 功能加到应用程序中去。还介绍了 WinInet API 如何使用 HINTERNET 类型的具有继承特性的句柄来管理 Internet 对话、链接和文件，以及用于和这些句柄一起协调工作的函数。接着介绍 MFC 中 WinInet 相关类以及派生出来的特殊协议类，例如 CFtpConnection 和 CHttpFile。最后，向读者介绍几个具体的 MFC WinInet 编程的实例，在介绍实例的过程中专门用 Winsock 和 WinInet 开发了两个 FTP 客户端，读者可以从这两个实现同样功能的程序开发过程中，体会到这两种技术的区别以及各自的特点。

“第 11 章 网络协议实现及应用”，本章介绍了常用网络协议以及它们的运行机制，同时介绍了开发这些网络协议应用的方法。最后介绍了信报 API 的有关内容，以及利用 4 种不同的 MAPI 来处理各种不同的信报。还介绍了 MAPI 的结构体系，包括各种地址手册、信报存储以及传送服务程序等。所有这些内容组合在一起就构成了 Win32 的信报子系统，它们为用户的应用程序提供各种各样的接口服务。本章重点在于这些协议的运行机制上。

“第 12 章 综合实例”，本章介绍了管道高级通信、电子邮件检查程序、文件下载、网络版五子棋游戏 4 个实例，这 4 个实例综合应用了前面的章节中介绍到的网络编程知识，如命名管道、Winsock 类扩展、MFC 网络编程等，这些例子相对来说复杂一些，并且经过适当的修改可以应用于实际产品中。

本书为立志掌握 Windows 编程技术的人提供了一条有效的捷径，对于不同层次的 Windows 编程人员来说，本书都极具参考价值，是一本不可多得的参考书。

由于 Windows 网络编程所涉及的知识面极为广泛，而笔者的知识又很有限，所以尽管对本书中所涉及的内容一再推敲和仔细调试，仍有可能出现错误和纰漏，希望广大读者批评指正。

编者

# 目 录

第 1 章 计算机网络体系结构 .....	1
1.1 网络术语及其拓扑结构 .....	1
1.1.1 服务器、客户机和节点 .....	1
1.1.2 本地资源和远程资源 .....	1
1.1.3 网络操作系统 .....	2
1.1.4 网络协议 .....	2
1.1.5 网卡、桥和路由器 .....	2
1.1.6 Intranet .....	3
1.1.7 拓扑结构 .....	3
1.2 开放系统互连参考模型 .....	3
1.2.1 物理层 .....	4
1.2.2 数据链路层 .....	4
1.2.3 网络层 .....	5
1.2.4 传输层 .....	5
1.2.5 会话层 .....	5
1.2.6 表示层 .....	6
1.2.7 应用层 .....	6
1.2.8 OSI 模型综述 .....	6
1.3 TCP/IP 参考模型 .....	8
1.3.1 网络接口层 .....	8
1.3.2 网际层 .....	9
1.3.3 传输层 .....	9
1.3.4 应用层 .....	10
1.4 网络接口层及其相关协议 .....	10
1.4.1 面向字符的链路层协议和面向比特的链路层协议 .....	10
1.4.2 高级数据链路控制规程 HDLC .....	11
1.4.3 X.25 的链路层协议 LAPB .....	13
1.4.4 点到点协议 (PPP) .....	13
1.5 网际层及其相关协议 .....	14
1.5.1 IP .....	14
1.5.2 消息控制协议 .....	18
1.5.3 地址解析/反向地址解析协议 .....	20
1.6 传输层及其相关协议 .....	21

1.6.1 面向连接的 TCP.....	21
1.6.2 无连接 UDP.....	22
1.7 应用层及其相关协议.....	22
1.8 Intranet 网络系统.....	23
1.8.1 Intranet 网络组成.....	23
1.8.2 Intranet 硬件结构.....	24
1.8.3 Intranet 软件结构.....	25
1.9 小结.....	26
<b>第 2 章 NetBIOS 编程.....</b>	<b>27</b>
2.1 Microsoft NetBIOS.....	27
2.1.1 LANA 编号.....	28
2.1.2 NetBIOS 名字.....	28
2.1.3 NetBIOS 特性.....	31
2.2 NetBIOS 基础.....	31
2.3 NetBIOS 例程.....	33
2.3.1 异步回调模型.....	39
2.3.2 异步事件模型.....	43
2.3.3 NetBIOS 会话客户端.....	47
2.4 其他 NetBIOS 命令.....	50
2.4.1 适配器状态.....	51
2.4.2 查找名字.....	52
2.4.3 对应传送协议同 LANA 编号.....	53
2.5 搜索指定网段内计算机.....	53
2.6 小结.....	58
<b>第 3 章 重定向器、邮槽和管道.....</b>	<b>59</b>
3.1 重定向器.....	59
3.1.1 命名规范.....	59
3.1.2 网络提供者.....	61
3.1.3 重定向器简介.....	61
3.1.4 服务器消息块.....	62
3.1.5 安全问题.....	62
3.1.6 网络安全.....	64
3.1.7 实例.....	64
3.2 邮槽.....	65
3.2.1 邮槽简介.....	66
3.2.2 基本客户端/服务器.....	68
3.2.3 其他邮槽 API.....	74
3.3 管道.....	74



3.3.1 匿名管道.....	75
3.3.2 命名管道.....	77
3.3.3 客户端与服务器基础.....	83
3.3.4 其他 API.....	97
3.3.5 命名管道通信.....	99
3.4 小结.....	103
<b>第4章 网络协议 TCP/IP.....</b>	<b>105</b>
4.1 协议特征.....	105
4.1.1 面向消息.....	105
4.1.2 面向连接和无连接.....	106
4.1.3 可靠性和次序性.....	106
4.1.4 从容关闭.....	107
4.1.5 广播数据.....	107
4.1.6 多播数据.....	107
4.1.7 服务质量.....	107
4.1.8 部分消息.....	108
4.1.9 路由选择的考虑.....	108
4.1.10 其他特征.....	108
4.2 支持的协议.....	108
4.2.1 支持的 Win32 网络协议.....	109
4.2.2 Windows CE 网络协议.....	110
4.3 网际协议 (IP).....	110
4.3.1 IP 主要特征.....	110
4.3.2 IP 数据报格式.....	111
4.3.3 IP 服务定义和原语.....	116
4.3.4 IPv6.....	118
4.4 传输层协议 TCP 和 UDP.....	123
4.4.1 传输层连接和端口地址.....	124
4.4.2 用户数据报协议 (UDP).....	127
4.4.3 传输控制协议 TCP.....	129
4.4.4 定址.....	137
4.4.5 创建套接字.....	139
4.4.6 名字解析.....	139
4.4.7 端口号.....	140
4.5 TCP/IP 安全性分析.....	141
4.5.1 TCP/IP 整体构架安全分析.....	141
4.5.2 安全性和提高安全性方法.....	142
4.6 Winsock 2 协议信息.....	145
4.7 具体平台的问题.....	147

4.8	综合实例.....	148
4.8.1	枚举系统支持网络协议.....	148
4.8.2	选择网络协议.....	156
4.8.3	TCP/IP 信息统计.....	160
4.8.4	IP 包监视.....	171
4.9	小结.....	178
<b>第 5 章</b>	<b>局域网编程.....</b>	<b>179</b>
5.1	局域网概述.....	179
5.1.1	局域网简史.....	179
5.1.2	局域网特点.....	180
5.1.3	局域网组成.....	180
5.2	网络接口卡与硬件编址.....	180
5.2.1	网卡基本结构.....	180
5.2.2	网卡参数.....	181
5.2.3	硬件编址与包过滤.....	182
5.2.4	硬件编址方式.....	183
5.2.5	广播与组播.....	184
5.2.6	帧格式.....	184
5.2.7	隐式帧网络.....	185
5.3	局域网拓扑结构.....	186
5.3.1	星型拓扑结构.....	186
5.3.2	环形拓扑结构.....	186
5.3.3	总线拓扑结构.....	187
5.3.4	树型结构.....	188
5.3.5	点对点连接.....	188
5.3.6	网状结构.....	188
5.4	局域网体系结构.....	189
5.4.1	IEEE 802 局域网参考模型.....	189
5.4.2	IEEE 802 局域网标准.....	191
5.4.3	以太网技术.....	192
5.4.4	令牌环网.....	196
5.5	综合实例.....	198
5.5.1	获取网卡信息 1.....	198
5.5.2	获取网卡信息 2.....	201
5.5.3	获取网卡信息 3.....	203
5.6	小结.....	211
<b>第 6 章</b>	<b>Winsock 基础.....</b>	<b>212</b>
6.1	套接字.....	212

6.2	Socket 编程模型与 Winsock 规范 .....	214
6.2.1	Socket 编程模型演化 .....	215
6.2.2	Winsock 套接字主要特点 .....	216
6.3	初始化 Winsock .....	217
6.4	建立 Windows 套接字 .....	218
6.5	错误检查和控制 .....	220
6.6	面向连接的协议 .....	221
6.6.1	服务器 API 函数 .....	221
6.6.2	客户端 API 函数 .....	225
6.6.3	数据传输 .....	228
6.6.4	流协议 .....	231
6.6.5	中断连接 .....	233
6.6.6	综合分析 .....	233
6.6.7	有连接通信示例 .....	241
6.7	无连接协议 .....	243
6.7.1	bind .....	243
6.7.2	创建服务器套接字 .....	246
6.7.3	接收端 .....	246
6.7.4	发送端 .....	247
6.7.5	基于消息的协议 .....	248
6.7.6	释放套接字资源 .....	249
6.7.7	综合分析 .....	249
6.7.8	无连接通信 .....	255
6.8	其他 API 函数 .....	258
6.9	小结 .....	260
<b>第 7 章</b>	<b>Winsock API 高级编程 .....</b>	<b>261</b>
7.1	Winsock 输入/输出 .....	261
7.1.1	套接字模式 .....	261
7.1.2	I/O 模型 .....	264
7.2	套接字选项 .....	302
7.2.1	SOL_SOCKET 选项 .....	303
7.2.2	IPPROTO_IP 选项 .....	308
7.2.3	IPPROTO_TCP 选项 .....	311
7.2.4	NSPROTO_IPX 选项 .....	311
7.3	套接字输出/输出控制 .....	314
7.3.1	标准 I/O 控制 .....	315
7.3.2	其他 I/O 控制 .....	315
7.3.3	安全套接字层的 I/O 控制 .....	320
7.4	原始套接字与底层传输协议 .....	321

7.4.1	创建原始套接字 .....	322
7.4.2	ICMP 的实现.....	323
7.4.3	Internet 组管理协议 .....	336
7.4.4	IP_HDRINCL 的使用 .....	338
7.5	综合实例.....	345
7.5.1	枚举 TCP 和 UDP 连接状态 .....	345
7.5.2	Ping 实例 .....	351
7.6	小结.....	357
<b>第 8 章</b>	<b>MFC Winsock 高级编程 .....</b>	<b>359</b>
8.1	Web 基础知识 .....	359
8.1.1	客户端 .....	359
8.1.2	服务器 .....	360
8.1.3	HTTP .....	362
8.2	HTTP 服务器设计 .....	363
8.2.1	同步操作 .....	364
8.2.2	错误异常处理 .....	364
8.2.3	发送/接收超时处理 .....	364
8.3	创建自己的 Winsock 类 .....	365
8.3.1	CSocketAddress 辅助类.....	365
8.3.2	CMyBlockSocketException 类 .....	367
8.3.3	CMyBlockSocket 类.....	368
8.3.4	CMyHttpBlockSocket 类.....	373
8.4	HTTP 服务器示例 .....	375
8.5	小结.....	386
<b>第 9 章</b>	<b>深入 UDP.....</b>	<b>387</b>
9.1	局域网广播.....	387
9.1.1	广播通信的优缺点 .....	387
9.1.2	广播通信实现 .....	387
9.2	多播通信.....	389
9.2.1	多播的含义 .....	389
9.2.2	IP 多播 .....	391
9.2.3	多播与 Winsock .....	393
9.2.4	IP 多播实例 .....	407
9.3	多媒体通信与常规服务质量 .....	412
9.3.1	背景知识 .....	412
9.3.2	QOS 和 Winsock .....	416
9.3.3	QOS 中止 .....	421
9.3.4	QOS 编程 .....	429

---

9.3.5 语音全双工通信 .....	434
9.4 小结 .....	450
<b>第 10 章 WinInet API 开发 .....</b>	<b>451</b>
10.1 WinInet 与 Winsock .....	451
10.2 使用 WinInet API .....	452
10.2.1 句柄 .....	452
10.2.2 错误处理 .....	452
10.2.3 缓冲区参数 .....	452
10.2.4 异步 I/O .....	452
10.3 Internet 函数 .....	453
10.3.1 常用 Internet 函数 .....	453
10.3.2 FTP 客户端函数 .....	459
10.3.3 HTTP 客户端函数 .....	461
10.3.4 Gopher 客户端函数 .....	465
10.4 WinInet API 编程 .....	465
10.4.1 HTTP 客户编程 .....	466
10.4.2 Cookies 编程 .....	468
10.4.3 FTP 客户编程步骤 .....	469
10.5 MFC WinInet 类 .....	472
10.5.1 WinInet 类编程模型 .....	473
10.5.2 其他常用函数 .....	478
10.6 深入 MFC WinInet 类 .....	482
10.6.1 CInternetSession 类 .....	483
10.6.2 连接类 .....	483
10.6.3 文件类 .....	484
10.6.4 CInternetException 类 .....	485
10.6.5 MFC WinInet 类的关系 .....	485
10.6.6 使用 CInternetSession .....	486
10.6.7 FTP 服务器处理 .....	489
10.6.8 HTTP 服务器处理 .....	492
10.6.9 Gopher 服务器处理 .....	495
10.6.10 实现 Internet 查询 .....	497
10.7 综合实例 .....	505
10.7.1 多线程 HTTP 服务器 .....	505
10.7.2 FTP 客户端开发 .....	529
10.8 小结 .....	559
<b>第 11 章 网络协议实现及应用 .....</b>	<b>560</b>
11.1 实现 HTTP .....	560

---

11.1.1	HTTP.....	560
11.1.2	实现 HTTP 客户端.....	568
11.2	实现 Telnet 协议.....	572
11.2.1	Telnet 协议.....	573
11.2.2	创建 Telnet 客户端.....	576
11.3	实现 FTP.....	587
11.3.1	FTP.....	588
11.3.2	FTP 应用程序.....	593
11.4	代理服务器.....	602
11.4.1	Socket 5 协议.....	602
11.4.2	HTTP 代理服务器.....	604
11.4.3	支持 Socket 5 代理.....	615
11.5	信报 API.....	622
11.5.1	MAPI 结构.....	622
11.5.2	在 MFC 中支持 MAPI.....	624
11.5.3	通用信报调用.....	626
11.5.4	简单 MAPI.....	633
11.5.5	扩展 MAPI 和 OLE 信报库.....	633
11.6	小结.....	633
<b>第 12 章</b>	<b>综合实例.....</b>	<b>634</b>
12.1	管道高级通信.....	634
12.2	电子邮件检查程序.....	647
12.3	文件下载.....	661
12.4	网络版五子棋游戏.....	675
12.5	小结.....	687

# 第 1 章 计算机网络体系结构

Internet 是一个跨越全球的计算机网络，通过 Internet 可以在全球范围内将信息快速、有效和方便地传递。目前，Internet 已经拥有数千万的用户，应用范围从商业、教育到个人，影响极为广泛。网络发展如此之快，必然增强对网络应用软件的需求。目前网络开发软件很多，例如 Visual Studio 6.0、Visual Studio.Net 等开发平台。本章主要介绍计算机网络方面的基础知识，包括网络体系结构和局域网基本构造等内容。

## 1.1 网络术语及其拓扑结构

本节首先介绍几个经常使用的网络术语，便于后续章节对网络技术的阐述。

### 1.1.1 服务器、客户机和节点

服务器 (Server) 为用户提供文件、资源和服务。事实上，在客户机/服务器系统中，一台计算机 (客户机) 向另外一台计算机 (服务器) 发送提供某种服务的请求，通常情况下，一台计算机既可以是客户机又可以是服务器。对服务器的最初定义是在局域网上，主要指放置大量文件和应用程序的高性能计算机，而其他的计算机通过网络访问服务器上的文件。在这种类型的网络中，一般只有一台计算机作为服务器，其他计算机作为客户机。大的基于服务器的网络有专用服务器，例如，有的服务器可能用来放置文件 (称作文件服务器)，有的服务器用来处理打印需求 (称作打印服务器)，有的服务器可能用来处理连接外部 (广域网) 的服务 (称作通信服务)，而在广域网或者局域网中，都可能采用 HTTP 提供 WWW 服务的服务器 (称为 WWW 服务器) 等。有些特殊的应用可能需要特殊的服务器，如有些大型软件，为了防止盗用，可能设置 License 服务器，每个使用其应用程序的计算机，在使用前都必须通过 License 服务器进行验证，从而保护软件不被盗版使用。

通常情况下，在有多台 PC 机的局域网中，可能有一台比较大的服务器，提供对大量文件和数据库的服务，而其他连接在它上面的计算机称作客户机。从服务器的定义可以看出，客户机是向服务器发出请求的计算机。通常意义上的客户机是指服务器为之提供文件和处理能力的计算机。在 TCP/IP 网络中经常使用客户机/服务器模式。

在规模很小的网络中通常会包含一台服务器和一定数量的 PC 机或者 Macintosh 计算机。每一个连接在网络上的计算机都可以称作一个节点 (Node)。节点是指连接到网络上的设备，这些设备在网络上有独一无二的名字 (地址)。规模很大的网络可能包括上千个节点，这些节点可能是计算机、打印机，或者网络设备等。

### 1.1.2 本地资源和远程资源

本地资源 (Local Resource) 是指连接到本地计算机上的设备，如打印机、调制解调器、扫描仪以及硬盘等。计算机访问本地设备无须通过网络，这样的设备就被称作本地设备或者本地资源。如果要访问某个设备，必须通过网络，则这样的设备就称为远程资源 (Remote Resource)。例如，对一般用户而言，那些连接到服务器上的设备就是远程资源。直接连接在网络上的网络打印机也可称为远程资源。

### 1.1.3 网络操作系统

网络操作系统 (Network Operating System) 的简称为 NOS, NOS 负责控制网络媒体中信息的发送和接收。它处理诸如下面的事务: 来自一台计算机的报文如何到达另一台计算机, 如果多台计算机同时发送报文和如何处理报文冲突, 以及多台计算机如何协调使用资源等。NOS 还要负责对外共享外设 (如网络上的打印机、扫描仪等) 的控制, 因为这些外设要被网上的用户共同使用, 不可避免地存在资源的分配问题。

在只有一台服务器和几个客户机的小型局域网中, NOS 位于服务器上, 如 Novell 的 Netware 网络就是采用这样的处理方式, 其 NOS 的主要部分驻留在服务器上, 而每个客户机中只有很少的客户软件。

对于拥有超过一台服务器的较大型网络, NOS 是每台计算机中软件的一部分。任何需要连上 TCP/IP 网络的 PC 机都要安装 TCP/IP 协议。Linux 有一些用于建造 TCP/IP 网络的代码, 集成在操作系统的内核中。而另外一些以 Microsoft Windows 98, Artisoft 的 LANtastic 等作为操作系统的网络, 并没有单个的主服务器 (当然, 它们也可使用单个服务器方式)。在这样的网络中, 每台计算机都拥有 NOS 中用来与其他计算机进行通信的功能。但是, Windows 2000 就包含了自己的主服务器。

### 1.1.4 网络协议

共享计算机的资源以及在网上交换信息, 都需要在不同系统的实体之间进行通信。两个实体 (实体间对等的应用程序等) 要成功地交换信息, 就必须具有同样的语言。交流什么、怎样交流以及何时交流, 都必须遵从有关实体间某种相互能接受的规则。这些规则的集合称为协议 (Protocol), 它可以定义为两个实体间控制数据交换的规则集合。协议的关键部分包括: 语法 (数据格式、编码及信号电平等)、语义 (包括用于协调和进行差错处理的控制信息) 和定时 (包括速度匹配和排序) 等。

由于不同系统中实体间通信的任务十分复杂, 相互不可能作为一个整体来处理, 否则任何小部分的改变, 就需要修改整个软件包。一种替代的方法是使用结构式的设计和实现技术, 用分层或层次结构的协议集合。

TCP/IP (Transmission Control Protocol/Internet Protocol) 是最为通用的网络协议, 在各种系统中都得到了广泛应用。事实上, TCP/IP 是一个协议的集合 (称作协议族), 而 Novell Netware 则使用 IPX 协议 (InterPacket Exchange)。

### 1.1.5 网卡、桥和路由器

网络接口卡 (Network Interface Card, NIC), 即网卡, 这是插在 PC 机的 I/O 槽上的一种适配器。有些 NIC 可能是插在并口或 SCSI 口上, 这样的 NIC 适用于笔记本等移动式 PC 机。

网卡一般处理的是协议族的底层协议。底层协议是与网络介质直接相关的。因此不同网络使用不同网卡。如 FDDI 网络有 FDDI (Fibre Distributed Data Interface, 光纤分布式数据接口) 的网卡、以太网有以以太网的网卡、ATM (Asynchronous Transfer Mode, 异步传输模式) 有 ATM 的网卡等。

桥 (Bridge, 或者称为网桥) 和路由器 (Router) 二者都是用于互联局域网 (LAN) 的专门设备, 它们是把不同的 LAN 或 LAN 分段连接在一起的设备。路由器最初的设计是让用户能够通过广域网把这些远程 LAN 连接起来, 网桥也可用于这个目的。把路由器或者网桥放在两个不同地点的 LAN 上, 再通过广域网或者远程通信把它们连接起来, 一个 LAN 上的用户就可以访问另一个 LAN 上的资源, 就好像这些资源是在本地一样。

网桥比较简单, 它能对在网络传递接受过程中接收到的每一个数据包作出简单的转发/不转发决定, 在 LAN 之间过滤数据包。过滤是根据包的目的地来执行的, 如果包的目的地是发送点所在网段上的一个站, 就不进行转发; 如果目的地是另一个 LAN 上的一个站, 它就被连接到不同的网桥端口, 并转发到该端口。当前许多网桥过滤和转发数据包只有很小的延迟, 故适用于大的流量。



路由器是更为复杂的网络互连设备，通常要比网桥昂贵。它利用每个数据包中的“网络层协议信息”，把数据从一个 LAN 传送到另一个 LAN。这意味着路由器必须能够识别与之连接的各个网络上可能使用的各种不同的网络层协议。这就是术语“多协议路由器”的来源——一种可以使用许多种不同协议进行路由的设备。路由器之间互相通信并共享信息，使它们能够决定通过连接许多 LAN 的复杂互联网的最佳路由。

### 1.1.6 Intranet

Intranet 是指在组织和企业内部采用与 Internet 相似的协议实现的网络，其主要目的是以 Internet 的方式在组织和企业内部发布信息。Linux 支持的 Internet/Intranet 服务包括电子邮件、新闻、WWW 服务等。

### 1.1.7 拓扑结构

常用计算机拓扑结构共 5 种：总线型、星型、环型、树型和网型。它们又可以按信道拓扑分为两类：点到点信道和广播信道。点到点信道是指网上主机之间都存在一条物理信道，网型，星型都是典型的点到点信道；广播型拓扑结构中，所有主机共享一条信道，其典型代表是总线型，而树型结构则具有层次性。

“连接”有 3 种层次上的意义，一是实际的物理线；二是建立在硬件基础上的虚电路；三是面向连接的服务，这是虚拟的、形式上的连接。通常所说的 TCP/IP 协议通过无连接的 IP 层实现了面向连接的 TCP 服务，指的就是这三种层次。

“端到端”与“点到点”是描述网络传输中对实体之间的关系的概念。前者是信号发送者与接收者直接通信，而不管中间经过多少路由；后者，对等实体的通信是一段一段直接相连的计算机所组成的，建立在 TCP/IP 协议上的网络是典型的端到端系统。

## 1.2 开放系统互连参考模型

开放系统互连 (Open System Interconnection, OSI) 参考模型是一个多层的通信协议，最初由国际标准化组织 (International Standard Organization, ISO) 开发，1983 年正式成为国际标准。所谓开放系统是指允许任意两个具有不同基本体系结构的系统进行通信的一套协议集。ISO 一直致力于允许多种设备相互通信的研究，并制定了开放系统互连模型。如果发展完善的话，OSI 将允许任意两台连接的计算机实现通信。

OSI 模型将网络划分为 7 层模型，分别用以在各层上实现不同的功能，这 7 层从上至下分别是：应用层 (Application Layer)、表示层 (Presentation Layer)、会话层 (Session Layer)、传输层 (Transport Layer)、网络层 (Network Layer)、数据链路层 (Data Link Layer) 及物理层 (Physical Layer)，其通信模型可参考图 1-1 所示。

OSI 模型中每一层只与其上下两层直接通信。高层协议偏重于处理用户服务和各种应用请求，低层协议偏重于处理实际的信息传输。分层的目的在于把各种特定的功能分离开来，并使其实现对其他层次来说是透明的。这种分层结构使各个层次的设计和测试相对独立。第  $n$  层为  $n+1$  层提供服务，第  $n+1$  层不必理会下层服务是如何实现的，因此，第  $n$  层实现方式的改变将不会影响第  $n+1$  层。