

# Visual C++ 6.0 网络及 Internet 开发指南

李博轩 等 编著



专题  
开发指南



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

# Visual C++ 6.0 网络及 Internet 开发指南

李博轩 等 编著

清华大学出版社

(京)新登字 158 号

### 内 容 简 介

本书通过大量实例深入浅出地介绍了 Visual C++ 6.0 网络及 Internet 开发技术。全书共 8 章, 主要内容包括: Internet 应用程序开发基础、WinInet 类编程、WinInet 客户应用程序开发、WinInet API 编程、WinInet API 客户应用程序开发、Windows Sockets 类编程、Windows Sockets 应用程序开发和电子邮件程序设计。

全书对每种编程技术都给出了具有代表性的应用实例, 而且将 Visual C++ 6.0 的 Internet 特性和技术难点融入到具体的实例中, 使读者通过实例的学习, 能够迅速掌握网络及 Internet 开发技术。本书所附的光盘中含有全部实例的源代码。

本书内容全面、深入, 适合中高级编程技术开发人员, 以及大专院校师生学习参考, 也适合各类培训学员学习 Visual C++、网络及 Internet 开发技术。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

书 名: Visual C++ 6.0 网络及 Internet 开发指南

作 者: 李博轩 等

出版者: 清华大学出版社 (北京清华大学校内, 邮编: 100084)

<http://www.tup.tsinghua.edu.cn>

印刷者: 北京市清华园胶印厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 23.25 字数: 578 千字

版 次: 2000 年 7 月第 1 版 2000 年 7 月第 1 次印刷

书 号: ISBN 7-900625-68-2

印 数: 0001~8000

定 价: 42.00 元

# 前 言

Visual C++是一种十分流行的软件开发平台,但是多数有关 Visual C++的书籍都是快速入门一类,或是对软件环境本身的介绍,很少涉及到实际应用。这样,读者在阅读完之后,只能对软件有一些零散的认识,而很难实际用它去完成独立应用程序的设计,尤其是专业应用软件的开发。

本书侧重 Visual C++在网络尤其是 Internet 方面的应用,重点介绍如何利用 Visual C++实现网络及 Internet 应用系统的开发。

全书以网络及 Internet 应用程序开发的不同主题来编排内容,分别讲述下列主题:

- Visual C++ Internet 应用程序开发基础
- WinInet
  - ◇ 类编程基础
  - ◇ 客户应用程序开发
- WinInet API
  - ◇ 编程基础
  - ◇ 客户应用程序开发
- Windows Sockets
  - ◇ 类编程基础
  - ◇ 应用程序开发
- 电子邮件程序设计

对于每个主题,都给出了 Visual C++在这一方面的要领,以及运用这类技术的实例和技巧,使读者能够通过实例的学习,迅速掌握 Internet 编程技术。

本书所附的光盘包括涉及实例的全部源代码和可执行文件,它们分别可以在光盘内对应目录下找到。读者可以从光盘上打开文件和工程浏览程序,或直接运行可执行文件以测试程序的功能,具体用法请参考光盘中的 readme 文件。

本书的特点在于讲述如何用 Visual C++ 6.0 实现网络及 Internet 应用程序,而且将 Visual C++ 6.0 的 Internet 特性和技术难点融入到具体的实例中,不强调“大而全”,而侧重“专而精”。

除封面署名作者外,参与本书编写的还有吴灵、刘秀蓉、吉尚戎、吉二源、杜丽、何震声、宋淼、陈明、李洪声、刘海涛、李敏、刘志诚、朱志言、刘兵等人,另外,王宇红、李兵、刘海兰、姚文龙、李晓霞、向文兵、刘斌、张勇、张碧霞、孟文征参与了校对与录排工作,在此对他们的辛勤劳动表示感谢。

作者

2000年2月

# 目 录

|                                |    |
|--------------------------------|----|
| 第 1 章 Internet 应用程序开发概述.....   | 1  |
| 1.1 计算机网络概述.....               | 1  |
| 1.1.1 计算机网络的分类.....            | 1  |
| 1.1.2 层次网络模型.....              | 4  |
| 1.1.3 网络协议.....                | 6  |
| 1.2 Internet 基础.....           | 6  |
| 1.2.1 Internet 的发展.....        | 7  |
| 1.2.2 Internet 的特点.....        | 10 |
| 1.2.3 Internet 的几个重要问题.....    | 11 |
| 1.2.4 Internet 的管理.....        | 12 |
| 1.2.5 Internet 的连接方式.....      | 14 |
| 1.3 TCP/IP 协议.....             | 15 |
| 1.3.1 IP 协议.....               | 16 |
| 1.3.2 TCP 协议.....              | 18 |
| 1.3.3 端口号.....                 | 19 |
| 1.4 域名系统.....                  | 20 |
| 1.4.1 域名.....                  | 21 |
| 1.4.2 名称服务器.....               | 21 |
| 1.4.3 DNS 客户/服务器机制.....        | 22 |
| 1.5 WWW 基础.....                | 22 |
| 1.5.1 WWW 概述.....              | 22 |
| 1.5.2 WWW 运行机制.....            | 23 |
| 1.6 VC 与网络开发.....              | 24 |
| 1.7 小结.....                    | 25 |
| 第 2 章 掌握 WinInet 类编程技术.....    | 26 |
| 2.1 WinInet 类概述.....           | 26 |
| 2.2 CInternetSession 类.....    | 27 |
| 2.2.1 构造函数.....                | 28 |
| 2.2.2 属性函数.....                | 29 |
| 2.2.3 操作函数.....                | 37 |
| 2.2.4 重载函数.....                | 39 |
| 2.2.5 运算符.....                 | 40 |
| 2.3 CInternetConnection 类..... | 40 |
| 2.3.1 构造函数.....                | 40 |

|        |                            |    |
|--------|----------------------------|----|
| 2.3.2  | 操作函数.....                  | 41 |
| 2.3.3  | 运算符 .....                  | 41 |
| 2.4    | CFtpConnection 类 .....     | 42 |
| 2.4.1  | 构造函数.....                  | 42 |
| 2.4.2  | 操作函数.....                  | 42 |
| 2.5    | CGopherConnection 类.....   | 48 |
| 2.5.1  | 构造函数.....                  | 48 |
| 2.5.2  | 操作函数.....                  | 49 |
| 2.6    | CHttpConnection 类 .....    | 51 |
| 2.6.1  | 构造函数.....                  | 51 |
| 2.6.2  | 操作函数.....                  | 51 |
| 2.7    | CInternetFile 类 .....      | 53 |
| 2.7.1  | 构造函数.....                  | 53 |
| 2.7.2  | 操作函数.....                  | 53 |
| 2.7.3  | 重载函数.....                  | 55 |
| 2.7.4  | 运算符 .....                  | 57 |
| 2.7.5  | 数据成员.....                  | 57 |
| 2.8    | CGopherFile 类.....         | 57 |
| 2.8.1  | 构造函数.....                  | 58 |
| 2.8.2  | 操作函数.....                  | 58 |
| 2.9    | CHttpFile 类 .....          | 58 |
| 2.9.1  | 构造函数.....                  | 59 |
| 2.9.2  | 操作函数.....                  | 59 |
| 2.10   | CFileFind 类 .....          | 65 |
| 2.10.1 | 构造函数.....                  | 65 |
| 2.10.2 | 属性函数.....                  | 65 |
| 2.10.3 | 操作函数.....                  | 70 |
| 2.11   | CFtpFileFind 类 .....       | 71 |
| 2.11.1 | 构造函数.....                  | 71 |
| 2.11.2 | 操作函数.....                  | 71 |
| 2.12   | CGopherFileFind 类 .....    | 73 |
| 2.12.1 | 构造函数.....                  | 73 |
| 2.12.2 | 属性函数.....                  | 73 |
| 2.12.3 | 操作函数.....                  | 74 |
| 2.13   | CGopherLocator 类.....      | 75 |
| 2.13.1 | 构造函数.....                  | 75 |
| 2.13.2 | 属性函数.....                  | 75 |
| 2.13.3 | 操作符 .....                  | 76 |
| 2.14   | CInternetException 类 ..... | 77 |

|        |                           |     |
|--------|---------------------------|-----|
| 2.14.1 | 构造函数.....                 | 77  |
| 2.14.2 | 数据成员.....                 | 77  |
| 2.15   | 全局 WinInet 函数.....        | 77  |
| 2.16   | 小结 .....                  | 79  |
| 第 3 章  | WinInet 类客户应用程序编程 .....   | 80  |
| 3.1    | WinInet 类编程概述.....        | 80  |
| 3.1.1  | WinInet 类通用编程步骤.....      | 80  |
| 3.1.2  | 状态回调程序.....               | 81  |
| 3.2    | HTTP 客户实例——我的浏览器 1.0..... | 82  |
| 3.2.1  | HTTP 协议概述.....            | 82  |
| 3.2.2  | HTTP 客户一般编程步骤.....        | 82  |
| 3.2.3  | 创建应用程序框架.....             | 84  |
| 3.2.4  | 制作应用程序界面.....             | 84  |
| 3.2.5  | 完成应用程序功能设计 .....          | 85  |
| 3.3    | FTP 客户实例——FTP 结构精灵 .....  | 89  |
| 3.3.1  | FTP 协议概述.....             | 89  |
| 3.3.2  | FTP 客户一般编程步骤.....         | 90  |
| 3.3.3  | 创建应用程序框架.....             | 91  |
| 3.3.4  | 制作应用程序界面.....             | 91  |
| 3.3.5  | 定制树状控件类.....              | 92  |
| 3.3.6  | 完成应用程序功能设计 .....          | 101 |
| 3.4    | Gopher 客户编程.....          | 106 |
| 3.4.1  | Gopher 协议概述.....          | 106 |
| 3.4.2  | Gopher 客户一般编程步骤.....      | 107 |
| 3.5    | 小结 .....                  | 108 |
| 第 4 章  | 深入 WinInet API .....      | 109 |
| 4.1    | HINTERNET 句柄 .....        | 109 |
| 4.2    | 常规 WinInet API 函数 .....   | 112 |
| 4.3    | 自动拨号函数 .....              | 124 |
| 4.4    | URL 函数 .....              | 127 |
| 4.5    | FTP 函数 .....              | 132 |
| 4.6    | Gopher 函数 .....           | 137 |
| 4.7    | HTTP 函数 .....             | 141 |
| 4.8    | Cookie 函数.....            | 149 |
| 4.9    | 缓存函数 .....                | 150 |
| 4.10   | WinInet API 宏 .....       | 159 |
| 4.11   | WinInet API 结构 .....      | 160 |
| 4.12   | WinInet API 错误码 .....     | 169 |

|                                           |            |
|-------------------------------------------|------------|
| 4.13 HTTP 状态码.....                        | 172        |
| 4.14 小结 .....                             | 174        |
| <b>第 5 章 WinInet API 客户应用程序编程.....</b>    | <b>175</b> |
| 5.1 WinInet API 编程预处理 .....               | 175        |
| 5.2 通用操作编程 .....                          | 176        |
| 5.2.1 通用文件检索.....                         | 176        |
| 5.2.2 通用文件下载.....                         | 179        |
| 5.2.3 设置异步操作.....                         | 183        |
| 5.2.4 锁定/解除锁定资源.....                      | 184        |
| 5.2.5 关闭 HINTERNET 句柄 .....               | 185        |
| 5.3 统一资源定位器编程.....                        | 185        |
| 5.3.1 标准化 URL.....                        | 186        |
| 5.3.2 创建 URL.....                         | 188        |
| 5.3.3 直接访问 URL.....                       | 188        |
| 5.4 身份验证 .....                            | 191        |
| 5.4.1 HTTP 身份验证.....                      | 191        |
| 5.4.2 处理 HTTP 身份验证.....                   | 193        |
| 5.5 使用缓存 .....                            | 195        |
| 5.5.1 枚举缓存.....                           | 195        |
| 5.5.2 获得缓存对象信息.....                       | 198        |
| 5.5.3 获取缓存对象流.....                        | 199        |
| 5.5.4 创建缓存对象.....                         | 200        |
| 5.5.5 删除缓存对象.....                         | 203        |
| 5.5.6 缓存组 .....                           | 203        |
| 5.6 处理错误 .....                            | 203        |
| 5.7 WinInet API HTTP 客户实例——我的浏览器 2.0..... | 205        |
| 5.7.1 WinInet API HTTP 客户编程步骤.....        | 205        |
| 5.7.2 创建应用程序框架.....                       | 206        |
| 5.7.3 制作应用程序界面.....                       | 206        |
| 5.7.4 设计工作线程类.....                        | 209        |
| 5.7.5 实现网页下载功能.....                       | 216        |
| 5.7.6 实现工作线程与主程序之间的通信 .....               | 219        |
| 5.8 WinInet API FTP 客户实例——超级 FTP 代理.....  | 221        |
| 5.8.1 WinInet API FTP 客户编程步骤.....         | 221        |
| 5.8.2 创建应用程序框架.....                       | 230        |
| 5.8.3 制做应用程序界面.....                       | 230        |
| 5.8.4 同步对象和回调函数.....                      | 233        |
| 5.8.5 实现异步通信.....                         | 240        |

|                                          |            |
|------------------------------------------|------------|
| 5.9 WinInet API Gopher 客户编程 .....        | 245        |
| 5.10 小结 .....                            | 246        |
| <b>第 6 章 Windows Sockets 类编程基础 .....</b> | <b>247</b> |
| 6.1 Windows Sockets 概述 .....             | 247        |
| 6.1.1 套接字 .....                          | 247        |
| 6.1.2 Windows Sockets 规范 .....           | 248        |
| 6.1.3 Windows Sockets 的基本概念 .....        | 249        |
| 6.2 CAsyncSocket 类 .....                 | 251        |
| 6.2.1 构造函数 .....                         | 251        |
| 6.2.2 属性函数 .....                         | 253        |
| 6.2.3 操作函数 .....                         | 259        |
| 6.2.4 重载函数 .....                         | 271        |
| 6.2.5 数据成员 .....                         | 273        |
| 6.3 CSocket 类 .....                      | 273        |
| 6.3.1 构造函数 .....                         | 274        |
| 6.3.2 属性函数 .....                         | 274        |
| 6.3.3 操作函数 .....                         | 275        |
| 6.3.4 重载函数 .....                         | 275        |
| 6.4 小结 .....                             | 276        |
| <b>第 7 章 Windows Sockets 类程序设计 .....</b> | <b>277</b> |
| 7.1 使用 CAsyncSocket 类 .....              | 277        |
| 7.1.1 CAsyncSocket 类编程步骤 .....           | 277        |
| 7.1.2 处理字节排序 .....                       | 278        |
| 7.2 使用 CSocket 类 .....                   | 280        |
| 7.2.1 CSocket 类编程步骤 .....                | 280        |
| 7.2.2 CSocket 对象与串行化技术 .....             | 283        |
| 7.2.3 使用归档的例子 .....                      | 286        |
| 7.3 通告回调函数重载 .....                       | 288        |
| 7.4 创建聊天客户 .....                         | 289        |
| 7.4.1 创建应用程序框架 .....                     | 290        |
| 7.4.2 制作应用程序界面 .....                     | 290        |
| 7.4.3 创建对话编辑和对话浏览窗口 .....                | 292        |
| 7.4.4 创建客户套接字类 .....                     | 295        |
| 7.4.5 创建串行化对象类 .....                     | 296        |
| 7.4.6 处理套接字通信 .....                      | 298        |
| 7.4.7 编辑和发送对话 .....                      | 304        |
| 7.4.8 显示对话内容 .....                       | 307        |
| 7.5 创建聊天服务器 .....                        | 307        |

|                          |            |
|--------------------------|------------|
| 7.5.1 创建应用程序框架.....      | 308        |
| 7.5.2 制作应用程序界面.....      | 308        |
| 7.5.3 创建服务器套接字类.....     | 310        |
| 7.5.4 管理通信.....          | 312        |
| 7.6 小结.....              | 314        |
| <b>第 8 章 发送电子邮件.....</b> | <b>315</b> |
| 8.1 电子邮件概述.....          | 315        |
| 8.1.1 电子邮件的产生.....       | 315        |
| 8.1.2 电子邮件的结构.....       | 316        |
| 8.1.3 电子邮件的地址.....       | 317        |
| 8.1.4 电子邮件的优越性.....      | 317        |
| 8.2 电子邮件的工作原理.....       | 318        |
| 8.3 电子邮件的传送过程.....       | 319        |
| 8.4 电子邮件协议.....          | 320        |
| 8.4.1 SMTP 协议.....       | 321        |
| 8.4.2 POP 3 协议.....      | 323        |
| 8.4.3 MIME 协议.....       | 323        |
| 8.4.4 NVT.....           | 324        |
| 8.5 邮件代理.....            | 325        |
| 8.5.1 创建应用程序框架.....      | 325        |
| 8.5.2 制做应用程序界面.....      | 325        |
| 8.5.3 格式化邮件信息.....       | 328        |
| 8.5.4 定制 SMTP 类.....     | 335        |
| 8.5.5 定制 POP 3 类.....    | 342        |
| 8.5.6 粘贴附件.....          | 348        |
| 8.5.7 完成功能集成.....        | 360        |
| 8.6 小结.....              | 361        |

# 第 1 章 Internet 应用程序开发概述

像当年 PC 机取代终端和大型机成为主流一样，网络正在逐步而又迅速地渗入社会生产和日常生活的各个领域，而网络应用程序的开发是随着网络技术的发展和需要而产生的。基于网络的程序开发是指在网络环境下运行和开发应用程序，它与传统意义上的程序开发有很大的不同。在掌握网络程序开发特点之前，首先简单回顾一下计算机网络的有关知识。

**本章要点：**

- 计算机网络基础知识
- Internet 和 WWW 基础知识
- Visual C++ 6.0 与网络应用程序开发

## 1.1 计算机网络概述

计算机网络是指将分布在不同地点且具有独立功能的多个计算机系统，通过通信设备和线路连接起来，在功能完善的网络软件运行下，以实现网络中资源共享为目的的系统。所谓独立是指每台计算机的工作是独立的，任何一台计算机都不能干预其他计算机的工作。例如，启动、关机等，并且任意两台计算机之间没有主从关系。

要组成一个计算机网络，首先应该把相关的计算机用专用的网络电器设备（如网卡、调制解调器、网线、集线器和路由器等）物理地连接在一起。其次，应该在所有连入网络的计算机中运行相关的网络软件，如支持网络连接的操作系统、驱动网卡等设备的专用软件等，同时应该利用这些网络支持软件对网络进行相应的合理配置，如常见的 IP 地址、网关等，只有在所有软硬件都正常工作并且网络配置正确的情况下，计算机才能真正地在逻辑上加入网络，为网上的其他计算机所承认，并能够与它们在规定的权限级别下实现通信和数据共享。

### 1.1.1 计算机网络的分类

对于计算机网络进行分类的标准很多。例如，按照拓扑结构、网络协议、信道访问方式或数据传输方式等分类。但是，按照这些标准都只能给出网络某一方面的特征，这里介绍常见的两种。

#### 1. 按地域范围划分

按照网络覆盖的地域范围，可以将其划分为局域网（Local Area Network, LAN）、城域网（Metropolitan Area Network, MAN）、广域网（Wide Area Network, WAN）和互联网（Internet）。

### ● 局域网

局域网的分布范围一般在 10km 以内, 最大距离不超过 10 km, 它属于一个部门或一个单位之间的网络。它是在小型计算机和 PC 机大量推广之后才逐渐发展起来的。由于局域网的覆盖范围小, 因此其传输速率大、延迟小, 而且容易对其进行管理、配置和构成简洁整齐的拓扑结构, 从而使网络站点往往能对等地参与对整个网络的使用和监控。此外, 由于局域网的造价低、组网方便和使用灵活等特点, 使其深受用户的欢迎, 是目前计算机网络技术发展最活跃的一个分支。

### ● 城域网

城域网覆盖一个城市, 其直径在 10 km 到 100km 之间, 它是一种大范围的高速网络。由于局域网所具有的种种优势, 用户需要扩大局域网的范围和将局域网相互连接起来, 使其成为一个规模较大的城市范围内的网络。因此, 城域网设计的目标就是要满足几十公里范围内的大量企业、机关、公司与社会服务部门的计算机联网需求, 实现大量用户、多种信息传输的综合信息网络, 其所采用的标准是 IEEE802.6。

### ● 广域网

广域网也称远程网 (long haul network), 其覆盖范围一般跨城市、地区甚至国家。此类网络的出现是由于军事、国防和科学研究的需要, 发展较早。例如, ARPAnet 网络, 在 1971 年从全美推广使用并已延伸到世界各地。由于广域网分布距离太远, 其速率要比局域网低得多, 一般为 64 kbps 左右。另外在广域网中, 大多使用专线进行网络之间的互联。物理网络本身往往包含了一组复杂的分组交换设备 IMP (Interface Message Processor, 接口信息处理机), 通过通信线路连接起来, 构成网状结构。由于广域网一般采用点对点的通信技术, 所以必须解决寻径问题, 这便是广域网的物理网络中必须包含网络层的原因。IMP 的主要功能之一就是寻径。广域网的覆盖范围大, 包容的信息量大。目前, 许多全国性的计算机网络都属于广域网, 如 ChinaDDN 网。

### ● 互联网

互联网实际上并不是一种具体的物理网络技术, 而是将不同的物理网络技术按照某种协议统一起来的一种高层技术。它是广域网、局域网以及城域网之间的相互连接, 形成局部处理与远程处理、有限地域范围资源共享与广大地域范围资源共享相结合的互联网。目前, 世界上发展最快、最热门的网络就是 Internet 网, 它是世界上最大的互联网。

上述这些网络各自的特点总结如表 1-1 所示。

表 1-1 各类计算机网络的特点

| 网络类别 | 分布距离        | 计算机位置                | 传输速率             |
|------|-------------|----------------------|------------------|
| 局域网  | 10~10000 m  | 网络中的计算机处于同一房间、建筑物或校园 | 4 Mbps~2 Gbps    |
| 城域网  | 10~100 km   | 网络中的计算机处于同一城市        | 50 kbps~100 Mbps |
| 广域网  | 100~1000 km | 网络中的计算机处于同一国家        | 9.6 kbps~45 Mbps |
| 互联网  | 1000 km     | 网络中的计算机处于同一洲或洲际      | 9.6 kbps~45 Mbps |

由表 1-1 可见,网络所覆盖的范围越大,其传输速率越低。一般来说,传输速率是网络的关键因素,它极大地影响着计算机网络硬件技术的各个方面。例如,广域网一般采用点对点的通信技术,而局域网一般采用广播式的通信技术。在距离、速率和技术细节的相互关系中,距离影响速率,速率影响技术细节。因此可以说这种分类标准能够反映网络技术的本质特征。

目前流行的“国家信息基础结构”(National Information Infrastructure, NII),也就是通常所说的“信息超高速公路”,是由美国政府首次提出的,它的目的是提供“一个通信网络、计算机、数据库和消费者电子设备组成的无缝连接网”,使用户可以方便地获得信息。NII 的具体实现方式还处于朦胧状态,但是可以预料的是 NII 将运行得足够快(55 kbps~150 Mbps),以便在同一个网络中提供充分集成的数字服务,例如声音、图像以及数据等。这意味着,用户能够使用同一条线路来接收电视(无线和有线)信号、打电话或上网,当然在付费时每种服务的标准可能不相同。现在我们所使用的 Internet 与 NII 的最大不同在于前者与能够提供全动感视频图像的网络相比,速度太慢而且很不方便。由于 NII 能够将网络性能提高许多倍,因此有可能取代 Internet。如果读者希望更多地了解有关 NII 的相关信息,可以查看 <http://sunsite.unc.edu/nii/toc.html>。

## 2. 按拓扑结构划分

将网络中的计算机抽象成点,连接计算机的网络抽象成连线,所形成的几何拓扑图形成计算机网络的拓扑结构。按照拓扑结构的不同,计算机网络可以划分为总线网、星型网和环形网。

总线网又称为 BU,如图 1-1 所示,它是在一条主干网线上分别连入不同的计算机所形成的网络。总线型的网络结构类似于干路与支路的关系,所有的计算机连接于一个信息传递的总干路上,总线中的数据可以向任何方向传递,两端都必须联结到专用的终端器上。每一台计算机的信息都通过网络干线传递到另一台计算机。总线网便宜,但是容错性差。

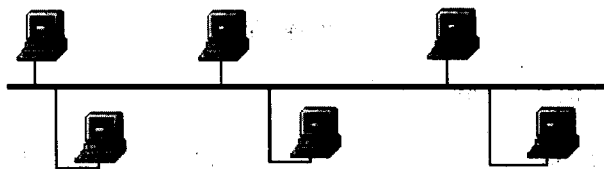


图 1-1 总线网

将总线的主干网线首尾相连,就形成了环形网,如图 1-2 所示,网络中的每台计算机都挂接在这个环线上。环形网又被称为 RING,是一种技术性和专用性都比较强的网络。环形网络结构中所有的计算机构成了一个环形通道,规定其中的数据只能向同一方向传递,并可以沿网络循环一周回到起点。网络中各台计算机都进行信息的接受与转发,因此任何一台计算机都可以获取网络中的所有信息。环形拓扑结构的电气原理决定整个通道必须全部畅通,任一点产生故障都会使整个网络瘫痪。

把环形网的网环或者总线网的主干网收缩为一个点,成为一个交换中心,使每台计算机直接与此点相连,就形成了星型网,如图 1-3 所示。星型网络结构中有一个中心节点,

所有的计算机都连接到这个计算机上，彼此之间不进行互连。网络中的计算机构成放射状星形结构，网络的所有信息都经过这个中心进行转发。在这种结构中，单一的计算机与中心节点之间的线路故障不会影响整个网络的使用，但中心节点的故障会使整个网络瘫痪。星型网又被称为 STAR，它的容错性较好，但是开销较大。

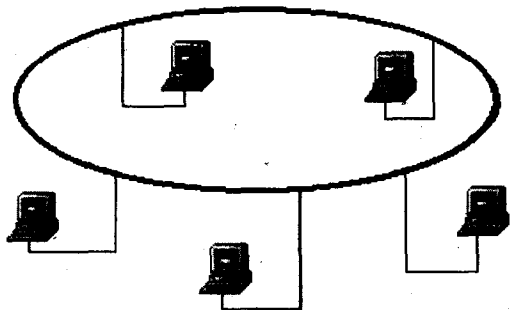


图 1-2 环形网

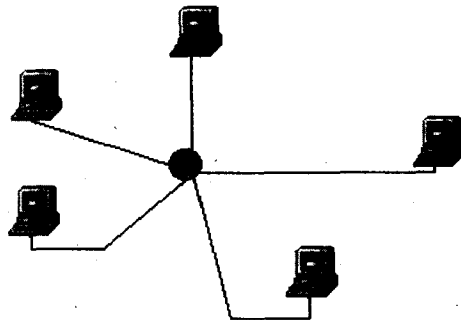


图 1-3 星型网

### 1.1.2 层次网络模型

ISO/OSI 是国际标准化组织制定的开放式系统互连参考模型。它将网络按功能分为七个层次，每个层次提供一定的服务，并提供与相邻层的接口，隐藏其下各层的细节。这七个层次自上而下分别是：应用层、表示层、会话层、传输层、网络层、数据链路层和物理层。

#### 1. 应用层 (Application Layer)

应用层包括与专门的用户应用程序的所有相关细节，实现具体的网络应用。它是 OSI 模型的最高层，负责网络中应用程序与网络操作系统之间的联系，并为用户提供各种服务，例如文件传输、远程登录、电子邮件以及网络管理等。

#### 2. 表示层 (Presentation Layer)

表示层主要用于处理两个通信系统中信息的表示方式，完成数据格式的转换，并对数据进行加密/解密、压缩/恢复等操作。它包含网络通信时重复使用的公共函数，提供与网络相关的文件格式或视频显示等的接口。

#### 3. 会话层 (Session Layer)

会话层为每一个网络会话建立和协调不同主机之间进程和应用程序的连结。它负责控制每一站究竟什么时间可以传送和接收数据，为不同的用户提供建立会话关系，并对会话进行有效管理。

#### 4. 传输层 (Transport Layer)

传输层负责提供两节点之间数据的传输，当两节点已经确定建立联系之后，传输层则

负责监督，以确保数据能够正确无误地传送。传输层的目的是向用户提供可靠的端到端服务，透明地传送报文，它向高层屏蔽了下层数据通信的细节，因而，它是计算机网络体系中最关键的一层。

### 5. 网络层 (Network Layer)

网络层是网络的传输系统，决定了主机和所有源地址和目的地址之间包交换点的接口。它的主要功能是通信子网内的寻径、流量、差错、顺序、进/出路由等控制，即负责将数据从物理连接的一端传送到另一端，实现点到点的通信。通过执行路由算法，网络层能够为报文分组在经过通信子网时选择最适当的路径。由于网络层需要执行路径选择、拥挤控制和网络互连等功能，是 OSI 参考模型中最复杂的一层。

### 6. 数据链路层 (Data Link Layer)

数据链路层处理二进制流到电气信号 (FM 信号或电平信号) 的转换，确保网络主机间的二进制信息不会发生错误。它负责相邻节点之间链路上的帧传输控制，通常被分为介质访问控制 (MAC) 和逻辑链路控制 (LLC) 两个子层。MAC 主要用于共享型网络中多用户信道竞争问题。而 LLC 的主要任务则是提供数据或帧、差错控制、流量控制和链路控制等功能。

### 7. 物理层 (Physical Layer)

物理层是 OSI 模型的最底层，利用物理传输介质为数据链路层提供物理连接，主要任务是在通信线路上传输数据比特电信号。它涉及到处理与传输介质有关的电气、机械等接口，以及通信方式 (单工、半双工和全双工) 等问题。

这七层模型提供了网络功能划分的详细方法，在实际使用中，一个网络协议并不一定要严格按照这个模型进行设计，很多通信协议在此基础上进行了一定的简化处理。每个层次都在其下层提供的服务的基础上工作，所以每一个层次在工作时都好像直接与其他计算机的同类层进行通信，而无须了解其下各层次的具体细节，这种同层次的连接被称为“虚连接”。例如，我们无须了解 E-mail 的具体传输过程，而只需关心电子邮件是否送达，其内容是什么等，这实际就是一个应用层之间的连接。一般来说，物理层和数据链路层的任务由硬件来实现，合称硬件层；网络层以上的各层功能一般由软件来实现，合称软件层。不过为了提高网络速度，有些网络层的功能也被固化在硬件中实现。例如，对于局域网来说，物理层、数据链路层和网络层都被直接固化在网卡中，而其余四层则由网络操作系统控制。

计算机间进行通信时，通常是一台计算机上运行的应用程序要求将数据传送给其他计算机上正在运行的程序。此时数据从一台计算机的最高层出发向下传输，通过表示层、会话层等直到物理层，在这个过程中，每通过一层，都会对上层传送下来的数据进行加工，如数据打包、附加信息等等，这种加工的目的是为提高数据传输的准确性和有效性，就好像发信时首先将要传递的信息写在信纸上，然后将信装入信封封好寄出，邮局再将信分好装入邮包，向不同的目的地发送。在计算机网络通信中，最后传送到物理层的数据被理解成二进制的比特信息，其具体含义在此时已经不再重要，这些信息被分装成小的数据包，

像邮包一样发送到不同的计算机节点。

当信息包沿网络传送到目的地计算机的物理层时,就像寄达的邮包分拆,信封被撕开,信纸被阅读一样,这些信息从目的地计算机的物理层向上传输,每传输一层就拆掉一层用于保证传输的附加包装,最后到达目的地的应用层,被重新组合,恢复成原始意义的数据,而被目的方应用程序所接收、理解和利用。这就是信息传递的大致过程。

### 1.1.3 网络协议

协议是网络中计算机交换信息时所遵守的规则,这些规则定义了网络所提供的服务。举个例子来说,如果你要使用寄一封信的服务,就必须遵守一定的规则,按一定的格式填写信封上的各种项目,这样,服务的提供者(邮局或邮递员)才可以顺利地完成任务。在网络中也是一样,为了使两台计算机之间能够顺利地交换信息,它们必须了解彼此所采用的信息组织结构。

网络协议一般以软件的形式存在和工作,但有时也将一些底层协议固化在硬件中以提高运行速度。在 OSI 网络模型的不同层次定义有不同的协议,如应用层上有 FTP、HTTP,传输层上有 TCP、SPX,网络层上有 IP、IPX 等。显然,不同层次上的协议应该相互支持配合才能完成完整的网络通信,为此,通常将在 OSI 模型的不同层次上共同工作的多个协议捆绑在一起,形成协议软件的集合,称为协议集。较常见的协议集有 TCP/IP、IP/SPX、NetBEUI 以及 AppleTalk 等等。

网络可以有种种不同的管理模式,网络上又经常联接着各种不同的计算机,随着计算机网络的发展,实现不同计算机网络及计算机之间的互连成为一个非常关键的问题,而 TCP/IP 协议由于有着很强的异种机互连的能力,使得它在 Internet 上得到了广泛的应用,成为事实上的标准。

网络程序开发可能涉及到网络模型的各个层次,即可能是较高的应用层,也可能是较低的网络层和传输层,但无论在哪一层工作,都必须跟其他的层次配合,如利用较低的层次的服务和支持较高层次的功能。所以与开发单机程序时需要了解操作系统的有关知识一样,开发网络应用程序应该了解层次模型和网络协议的基本知识。

## 1.2 Internet 基础

80 年代以来,随着计算机技术的发展和完善,特别是 PC 的出现,全世界越来越多的计算机采用各种通信媒体相互进行连接,组成了一个超级网络,这就是所谓的 Internet。1997 年 7 月,全国科学技术名词审定委员会推荐使用的中文译名为“因特网”(本书中为了叙述方便一律采用 Internet)。

Internet 是目前国际上最大的异构互联广域网和最大的信息集散地,采用 TCP/IP 协议和其他统一的标准,将世界各地业已存在的大量局域、广域网络连接起来,形成一个巨大的网中之网,将全球联系起来,实现了世界范围内的信息、资源和服务共享,极大地扩展了计算机网络应用的外延和内涵。

## 1.2.1 Internet 的发展

Internet 从 60 年代末诞生以来的 20 多年, 经历了 ARPAnet 网的诞生、NSFnet 网的建立、美国国内互联网的形成以及 Internet 在全世界的形成和发展等阶段。

### 1. 计算机网络互联的初期

50 年代初, 由于军事上的需要, 美国在建成的半自动地面防空系统 SAGE 上进行了计算机技术和通信技术相结合的尝试。就是将远程雷达及其他测量设备测量得到的数据信息, 通过总长度为 240 多万公里的通信线路与一台 IBM 计算机连接, 集中对防空信息进行处理与控制。为了达到这个目的, 进行了许多关于数据通信方面的基础研究。在这些研究的基础上, 人们通过利用通信线路将计算机与远方的终端连接起来。在通信软件的控制下, 各个用户在自己的终端上分时轮流使用中央计算机系统的资源对数据进行处理。然后, 再将处理结果直接送回终端。这就形成了具有通信功能的终端——计算机网络系统。它首次实现了计算机技术与通信技术的结合, 属于计算机网络发展的初级阶段。从通信的角度来看, 这种系统只能说是一种计算机数据通信系统。随着这种面向终端网络系统的大量应用, 人们发现它存在着两个主要缺点:

- 主机系统负担较重, 它既要承担数据处理任务, 又要承担通信任务。
- 由于终端设备运行速度慢, 操作时间长, 每个用户独占一条长距离的通信线路, 使得线路利用率低。

针对单机系统存在的上述缺点, 人们对面向终端的计算机通信网络进行了改造。首先, 在主计算机之前增加了一台功能简单的计算机, 专门用于处理终端的通信信息和控制通信线路, 并能对用户的作业进行某些预处理操作, 这台计算机被称为“前置处理机”或“通信控制处理机 (Communication Control Processor, CCP)”。然后, 在终端设备较集中的地方设置一台集中器 (Concentrator)。终端通过低速线路先汇集到集中器上, 然后再用高速线路将集中器连接到主机上。这样就形成了: 终端群→低速通信线路→集中器→高速通信线路→前置处理机→主机。

但是, 由于此类系统缺乏对网上资源进行统一管理的系统软件, 所以, 它仍属于计算机网络的低级形式。美国在 1963 年投入使用的飞机订票系统 SABRE-1 就是这类系统的典型代表。

### 2. ARPAnet 网的诞生

随着计算机应用的发展, 出现了多台计算机互连的需求。60 年代中期发展了若干个计算机互连起来的系统, 即利用高速通信线路将多台地理位置不同, 并且具有独立功能的计算机连接起来, 开始了计算机与计算机之间的通信。此类网络有以下两种结构形式:

- 主机通过高速通信线路直接互连起来, 这里主机同时承担数据处理和通信工作。
- 通过通信控制处理机 (CCP) 间接地将各台主机连接起来。通信控制处理机负责网络上各个主机之间的通信处理与控制; 主机是网络资源的拥有者, 负责数据的处理。它们共同组成资源共享的高级形态的计算机网络, 这是计算机网络发展的