

INTERNET编程

INTERNET PROGRAMMING

〔美〕 Kris Jamsa博士和Ken Cope 著

刘素丽 李彤红 等译

罗四维 审校



- 编写通过Windows 访问Internet 的程序
- 创建你自己的ftp、finger、ping、mail等程序
- 理解Winsock API
- 掌握TCP/IP编程和Internet协议
- 使用C、C++、Visual C++和Visual Basic进行Internet编程



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY



INTERNET 编程

[美] Kris Jamsa 博士和 Ken Cope 著

刘素丽 李彤红 等译

罗四维 审校

电子工业出版社

内 容 简 介

本书全面、系统地阐述了 Internet 编程方法及其相关问题。全书共十九章:第一、二章简述计算机网络的基本概念和网络体系结构;第三章介绍建立在 TCP/IP 协议组基础上的网络的原理与特点;第四章~第六章分别介绍 Internet 协议、传输协议、串行链路 Internet 协议(SLIP)和点对点协议(PPP);第七章介绍了 TCP/IP 网络最通用的应用程序接口——Socket 接口;第八章介绍如何编写基于 Windows 的 Internet 程序;第九章介绍域名系统(DNS);第十章~第十四章分别介绍 Finger 用户信息协议、异步 Windows Sockets、Windows 任务句柄、时间和网络字节序、Raw Sockets;第十五章专门介绍 Internet E-mail 基本组成、简单邮件传输协议,以及其它与 Internet 相关的协议;第十六章介绍文件传输协议(FTP);第十七章~第十九章分别介绍使用 DLL 的 Internet 编程、Internet 的可视化编程和 Internet 的全球网(World Wide Web)编程方法。附录 A 介绍了防火墙与 Internet 的安全性问题;附录 B 介绍了随书磁盘中编程实例的设计思想。

附赠磁盘给出了多个编程实例,以及 SLIP 和 PPP 连接的 Winsock 软件。

Copyright © 1995 by Jamsa Press, All Rights Reserved.

本书英文版权归美国 Jamsa 出版社所有。电子工业出版社已从 Jamsa 出版社购得中文版简、繁体的专有出版权和全球发行权。版权所有,侵权必究。

INTERNET 编程

[美] Kris Jamsa 博士和 Ken Cope 著

刘景丽、李彤红、李译译

罗四清 审校

责任编辑 史明生

电子工业出版社出版

北京市万寿路 173 信箱(100036)

电子工业出版社发行 各地新华书店经销

电子工业出版社计算机排版室 排版

北京科技大学印刷厂印刷

*

开本:787×1092 1/16 印张:30.25 字数:790 千字

1996 年 5 月第一版 1996 年 5 月北京第一次印刷

印数:5000 册 定价:58.00 元

ISBN 7-5053-3495-6/TP·1398

著作权合同登记号:01-95-583

前 言

Internet 是目前世界上连接用户最多的国际交互式计算机网络。对于众多的网络程序员来说,本书可以帮助你成为一名优秀的网络程序员。

本书译自美国 Jamsa 出版社的《Internet Programing》。全书共十九章:第一、二章介绍网络基本原理;第三~六章分别剖析 TCP/IP 网络原理、互连网协议、传输协议和基于 Windows 环境的 SLIP、PPP 协议;第七章重点介绍 TCP/IP 网络协议最通用的 API 之一——Socket 接口;第八章专为 Windows PC 机用户介绍如何编写基于 Windows 的 Internet 程序;第九~十四章分别介绍 DNS、Finger 用户协议、异步 Windows Sockets、更好地处理异步数据库函数、Time 协议、Raw Sockets;第十五、十六章阐述电子邮件和文件传输协议;第十七、十八章分别介绍 DLL 和 Internet 的可视化编程;第十九章介绍超级文本传输协议及 Web 编程。附录介绍了“防火墙”问题,并详细讲述了随书磁盘中实例程序的设计思想与结构。

参加本书翻译工作的有刘素丽、李彤红、韩臻、孙安瑜、蔡天芳、张爱军、杨武杰、平先福、韩飞、黄亚平、武娟娟、黄岸风、徐加星、孙仲阳、田霞、徐止铎、樊崇艺。罗四维对全书进行了审校。

译者水平有限,书中不妥之处,恳请读者指正。

译 者
1995 年 10 月

目 录

第一章 计算机网络导引	(1)
计算机网络定义	(1)
连接两个或多个网络	(2)
计算机怎样通信	(3)
理解网络的拓扑结构	(11)
计算机网络的连接	(15)
理解网络结构	(19)
本章总结	(20)
第二章 理解网络体系结构	(22)
理解网络术语和概念	(22)
简要的历史回顾	(23)
路径表	(23)
理解分层	(24)
理解网络通信	(24)
理解通信服务	(27)
理解设计问题	(32)
定义网络分层	(33)
理解网络分层	(36)
理解客户/服务器模型	(45)
本章总结	(47)
第三章 TCP/IP 导引	(48)
理解 TCP/IP 协议的重要性	(48)
理解 TCP/IP 术语	(48)
理解数据流	(50)
理解 TCP/IP 框架	(50)
通过 Internet 传送数据	(51)
研究 2TCP/IP 协议栈	(51)
理解物理层	(52)
理解链路层	(53)
增加 TCP/IP 词汇	(56)
本章总结	(58)

第四章 理解互连网协议	(60)
理解网络层	(60)
理解 Internet 地址	(62)
理解 Internet 地址协议	(67)
理解 IP 数据报	(68)
理解 IP 头	(70)
理解分割	(76)
理解 IP 路由选择	(80)
本章总结	(82)
第五章 理解传输协议	(83)
理解传输层	(83)
理解传输层端口	(84)
理解 UDP 端口使用	(85)
理解 TCP 端口使用	(85)
程序怎样使用端口号	(86)
理解用户数据报协议	(86)
理解传输控制协议	(87)
理解封装	(97)
理解应用层	(98)
本章总结	(98)
第六章 SLIP 与 PPP	(99)
回顾串行数据通信协议	(99)
理解 SLIP 连接	(101)
理解压缩的 SLIP(CSLIP)	(106)
理解点对点协议	(110)
本章总结	(122)
第七章 理解 Socket 接口	(123)
理解 Socket 实现	(123)
网络 I/O 与文件 I/O	(123)
理解 Socket 抽象	(125)
建立一个 Socket	(126)
定义程序的 Socket 使用	(130)
配置 Socket	(130)
通过 Socket 传输数据	(133)
通过 Socket 接收数据	(135)
综述此过程	(136)
服务器的 Socket 使用	(138)

综述服务器进程	(140)
本章总结	(142)
第八章 理解 Windows Sockets API	(143)
理解 Winsock 的起源	(143)
理解 Winsock 实现	(144)
研究总体结构	(148)
回顾 Socket 编程模型	(151)
Berkelay Sockets 与 Windows Sockets	(157)
理解阻塞	(162)
非阻塞函数与异步函数	(165)
再研究 Socket 函数	(166)
理解 WSAAsyncSelect 函数	(166)
Windows 3.1 环境下的阻塞问题	(168)
Windows Sockets 中的阻塞	(168)
本章总结	(172)
第九章 理解域名系统	(173)
将 DNS 当作一个实例使用	(173)
平坦名字空间与层次名字	(174)
理解域名系统	(175)
理解名字服务器	(177)
理解解析器	(179)
本章总结	(188)
第十章 Finger 用户信息协议	(190)
回顾表示层	(190)
理解网络虚拟终端	(191)
建立快速 Finger 程序	(193)
理解 Finger 用户信息协议	(210)
本章总结	(211)
第十一章 异步 Windows Sockets	(212)
建立 Sockman 模板	(212)
给 Sockman 增加查找功能	(213)
修改 DoWinsockProgram 函数	(214)
理解对话框	(215)
理解对话函数	(216)
理解对话过程	(218)
执行阻塞查找	(222)

执行异步查找	(224)
修改 WndProc 函数	(228)
理解 DisplayHostEntry 函数	(229)
本章总结	(230)
第十二章 使用 Winsock 任务句柄	(232)
把 Finger 加到 Sockman 中	(232)
为 Finger 修改 DoWinsockProgram	(233)
理解 Finger 对话框	(234)
理解 AsyncGetServiceInfo 函数	(237)
修改 WndProc 函数	(240)
理解 LookupFingerHost 函数	(241)
再次修改 WndProc 函数	(242)
执行异步 Finger 操作	(244)
执行阻塞 Finger 操作	(245)
理解 DoFingerOperation 函数	(246)
综述	(248)
本章总结	(249)
第十三章 Time 协议和网络字节序	(250)
理解网络字节序	(250)
理解 Internet Time 协议	(251)
理解 Time 协议	(252)
理解字节序	(253)
使用 Time 协议	(257)
创建快速 Time 程序	(257)
把 Time Sever 查询加到 Sockman 中	(266)
本章总结	(276)
第十四章 理解 Raw Sockets	(277)
理解 ICMP	(277)
使用 Raw Sockets	(290)
把其它的应用程序加到 Sockman 中	(302)
本章总结	(303)
第十五章 理解 Internet E-Mail	(304)
概述	(304)
理解简单邮件传输协议	(306)
定义 E-Mail 的组成部分	(315)
改善 E-Mail 各部分	(316)

考虑影响	(325)
理解邮局协议 (POP)	(325)
综合使用	(329)
本章总结	(331)
第十六章 文件传输协议	(332)
FTP 基础知识	(332)
识别文件传输协议	(335)
理解 FTP 模型	(336)
管理数据	(337)
理解 FTP 和 TELNET 之间的关系	(339)
连接管理	(345)
定义 FTP 命令	(348)
定义 FTP 应答码	(356)
本章总结	(357)
第十七章 用 DLL 进行 Internet 编程	(359)
策略综述	(359)
FTP 快速回顾	(359)
第一阶段:服务器控制	(360)
第二阶段:传输数据	(366)
第三阶段:DLL 实现	(378)
本章总结	(391)
第十八章 Internet 的可视化编程	(392)
观察 SockFTP 的实际运作	(392)
理解设计目标	(395)
选择综述	(397)
QFTP 阶段 4:获得目标的可视化结果	(398)
综述 Visual Basic 到 DLL 接口	(398)
理解 SockFTP	(402)
定义 SockFTP 过程	(412)
本章总结	(424)
第十九章 Web 网上的 Spiders 工具	(426)
综述超级文本传送协议	(427)
理解 HTTP 客户请求	(427)
综述 URIs 和 URLs	(427)
理解 HTTP 方法	(429)
了解机会	(430)

了解问题	(430)
Web 编程	(431)
创建一个 Web 服务器程序	(438)
本章总结	(457)
 附录 A 防火墙和 Internet 安全性	 (459)
理解防火墙	(460)
总结	(462)
 附录 B 理解实例程序	 (463)
定义种类	(463)
综述快速程序	(464)
综述 Sockman 程序设计	(464)
编译实例程序	(472)

第一章 计算机网络导引

如果你已经很了解计算机网络,那么通过这一章可对熟悉的内容复习一遍。如果你对网络术语很熟悉,但不知道其原理,本章可以让你了解网络的工作原理。如果你对网络完全陌生,那么这章将一步一步地告诉你,作为一个 Internet 程序员应该掌握的网络基本知识。在 1981 年出版的由 Andrew S Tanenbann 著的《Computer Network》这本书的前言中,作者说过:“设计计算机网络的方法最先是由 Julius Caesar 提出来的,其方法是分而治之。”

了解计算机网络也可采用同样方法。初看计算机网络,你会发现计算机网络尤其是巨大的、世界范围的 Internet 网络非常可怕,以致于认为了解网络简直是一件不可能的事情。但是,如果采用 Tanenbaum 的“分而治之”的方法,把每个任务(或主题)分为许多小任务(或子主题),你就会发现子主题要容易理解得多。并且经过一段时间对这些子主题了解以后,你就可以掌握大的主题,进而掌握整个网络的原理。

要了解如何使 200 台计算机进行网络通信,必须先了解两台计算机如何通信。同样,你不用一开始就了解上千个计算机网络如何工作,而只需研究两个或三个计算机网络是如何通信的。

开始看起来,要掌握的网络知识大得可怕,但是就像吃饭是一口一口地吃,学习网络也可以一次只熟悉一个主题。因此本书第一章和第二章将网络主题分为几部分,当你对这几部分了解以后,你就会对网络有较深的了解,在你看来学习网络不再是一件高不可攀的事情。到你学完本章时,你将会掌握如下几个关键概念:

- ◆ 什么是计算机网络
- ◆ 网络和互连网之间有什么差别
- ◆ 计算机如何互相通信
- ◆ 通信交换使用交换箱为电信号选择路由
- ◆ 分组交换在传送时把报文信息分成小块
- ◆ 网络拓扑结构规定网络中计算机怎样连接
- ◆ 网络管理员怎样连接网络
- ◆ 网络的基本组成部分

计算机网络定义

计算机网络最基本的形式是两台计算机相互连接进行通信。当然,大多数网络的计算机数目远远超过两台。但是网络通信的原理,不管网络大小是 2 或 3 甚至 1000,其原理都是一样的。如果你掌握了两台计算机怎样通信,那么就为理解 1000 台甚至更多的计算机通信奠定了

基础。

网络通常分为两大范畴:局域网和广域网。局域网(LAN)把相距比较近的计算机连在一起。在一些情况下,“局部”意味着计算机是在一个房间里或在一座楼中。在另外一些情况下,计算机也可以相距几英里远。与局部网相比,广域网(WAN)包括不同城市,不同国家连在一起的计算机。你可以把广域网看作是远距离网络,因为网络交换的信息常常要传输很远的距离。你知道,Internet 包含全世界范围的几千个网络。但作为一个程序员,你可以将 Internet 看作单个网络。

连接两个或多个网络

如果你把两台或者更多的计算机连接在一起,使它们能够互相通信,这样就建立了一个网络。同样,你也可以把两个或更多的网络连接在一起,建成互连网(internet)。图 1.1 说明了网络和互连网的关系。Internet(以大写 I 开头)是全世界范围内现有最大的和最通用的互连网。Internet 连接 130 个国家的 20,000 多个计算机网络。这些网络使用许多种软件。很幸运,当你使用 Internet 网或者对它进行编程时,可以忽略这些差别。你将使用的软件和 Internet 程序将要访问的软件库会为你消除这些差别。

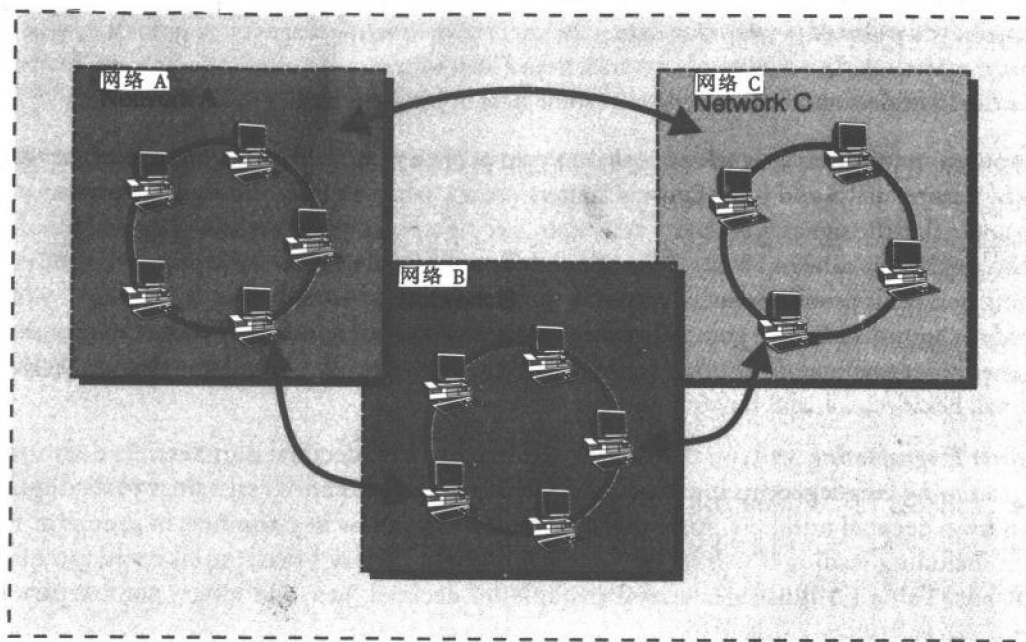


图 1.1 网络和互连网的关系

开始学习 Internet 编程时,你可以把 Internet 看成一个网络。当你掌握了单个网络的基本原理后,就可以把这些知识应用到了解互连网——如 Internet 上。

计算机怎样通信

一个网络最简单的形式是两台计算机连在一起互相通信。因此要了解网络,必须了解计算机如何通信。下面几段将告诉你许多通信术语和基本概念。

从原理上讲,计算机通信同人类交流信息是一样的。人使用语言交流信息,所有语言都有相同的基本组成部分,即字母和符号。字母或符号组合在一起构成单词。不管你使用什么语言交流信息,必须把单词组合成句子。

同样,计算机使用二进制数(1 和 0 这些符号)构成语言。计算机将 01 组合在一起形成字节。计算机通过传输许多字节的组合来交换信息。要真正了解计算机如何通信,必须先了解握二进制数。

计算机通信的最终目的是把信息传给人。虽然计算机很容易理解 1 和 0,但二进制数对人来说几乎是不可理解的。因此计算机必须把二进制数转换成人能够理解的字母和数字。下面几段将告诉你计算机如何进行这些转换工作。

理解计算机语言

当你编写程序时,计算机把你的数据表示成一串二进制比特位,每位要么是 1,要么是 0。作为一个应用程序员,可能不用处理二进制数,但是作为网络或 Internet 程序员,可能偶尔需要读取、解释和保存二进制信息。数据通过网络进行分组传输时,通常是工作在比特位级。也就是说,要了解怎样通过 Internet 传输数据,必须了解数据怎样用二进制表示。

计算机使用电信号来表示二进制数。每一个二进制数由一个 0 和 1 的串组成。许多计算机规定:线路上有电信号时表示 1,没有电信号时表示 0。与其相反,人常采用十进制数。虽然可以将二进制数转换为十进制数,但这很费时,也容易发生错误。鉴于此,可以使用十六进制数。十六进制数转换成十进制数和二进制数都比较方便。如果需要对 Internet 进行编程,你就有必要花点钱买一个计算器,这样可以方便地进行上面这些转换工作。

《Internet 编程》使用 hex 这个术语表示十六进制数。与 C 语言相似,本书在十六进制数前面加一个前缀 0x,用来区别其它进制数。本书按两位数字为一组显示十六进制。如果不够两位数,在前面加一个 0。本书按八比特位为一个组显示二进制数。表 1.1 显示了几个十六进制数、十进制数、二进制的例子。

许多 Internet 文档都是按位来确定和规定数据的。例如:注解规范(RFC)可能告诉你一个特定的数据域是十六位的,也可能告诉你一个特定的位模式表示一个特定的网络报文信息。当编写 Internet 程序时,要经常参考此类文件。必须学会怎样按位或者位模式表示你的数据,否则你会发现许多重要的 Internet 参考文献非常难懂。

表 1.1 十进制、十六进制、二进制数格式例子

十进制	十六进制	二进制
10	0x0A	00001010
100	0x64	01100100
255	0xFF	11111111
512	0x0200	0000001000000000
500	0x01F4	000000011110100

复习十进制、十六进制和二进制数

每种程序语言允许你对数字和字符进行操作。你可能对 ASCII 码(美国国家标准信息交换码)非常熟悉,ASCII 对每个字母、每个标点符号等等都规定一个值。程序在计算机屏幕上显示信息(例如一个名字)时,使用 ASCII 码来表示信息。同样程序向打印机发送信息时,也是发送 ASCII 码。

查看 ASCII 码表,将 ASCII 数字值对应为相应符号时,你会发现 ASCII 码数字值通常用十进制和二进制数表示。之所以采用这种表示方法,主要是由于程序员经常使用此表。虽然许多程序员把每一个 ASCII 符号用一个字节表示(8 位)。但 ASCII 码对计算机来说实际上是一个 7 位编码方案。表 1.2 显示了英语前五个字母的 ASCII 码。此表显示每一个编码时,按字母、十进制、十六进制、二进制格式进行显示。

表 1.2 ASCII 码按照字母、十进制、十六进制、二进制格式表示的例子

字符	十进制	十六进制	二进制
A	65	0x41	01000001
B	66	0x42	01000010
C	67	0x43	01000011
D	68	0x44	01000100
E	69	0x45	01000101

作为一个 PC 程序员,你可能对扩展 ASCII 字母集很熟悉。扩展 ASCII 码使用 8 位模式表示数据。程序可使用扩展 ASCII 码显示,像制表符(「,」,「,」等)这样的特殊字符。为了方便起见,本书用 ASCII 这个术语表示扩展 ASCII 码。

当前,许多计算机用八位表示一个字节。上面讲过,十六进制为表示数据提供了方便。十六进制是以 16 为基的数制系统。十六进制包含数字 0 到 9 以及字母 A 到 F(或者 a 到 f)。如表 1.3 所示,它使用四位二进制数表示十六个数值。你可使用两个十六进制数(每个四位)表示一个 8 比特字节。

表 1.3 十六进制系统

十进制	十六进制	二进制	十进制	十六进制	二进制
0	0X00	0000	8	0x08	1000
1	0X01	0001	9	0x09	1001
2	0X02	0010	10	0x0A	1010
3	0X03	0011	11	0x0B	1011
4	0X04	0100	12	0x0C	1100
5	0X05	0101	13	0x0D	1101
6	0X06	0110	14	0x0E	1110
7	0X07	0111	15	0x0F	1111

由于字节是程序访问的最小数据单位,所以程序只能对 8 位数据单位的组合进行操作。八比特表示的最大数(11111111)是 255 (十进制)或者 0xFF(十六进制)。两个十六进制数能够表示一个字节的 256 个不同数,它和扩展 ASCII 码表示的数一样多。

如果你不必处理二进制信息,可以将字节数据看成两位十六进制数。使用 ASCII 码,你可用一个字节可以表示英语中的所有字母。如果你对 ASCII 码、两位十六进制数、一个字节、八比特位之间的隐含关系比较熟悉,就会很容易地对网络数据进行打包操作。

例如,你知道两个十六进制数可以表示一个字节数据。因此当你需要使用二进制数时,可以很容易把两个十六进制数表示成八比特二进制数。如果你的数据是文本的,可以使用十六进制 ASCII 码很容易地把每一个文本字符表示成八位二进制数。

当从相反方向,即从二进制数转换到文本时,你会发现十六进制更有用。当使用十六进制数把二进制转换成文本时,每两个十六进制数表示一个字节数据。使用 ASCII 码时,每个字节表示一个文本字符。当把数据转换成文本或字符串后,程序使用这些数据就很方便了。

图 1.2 显示了二进制数(计算机理解的)如何转变为人所能理解的信息。计算机用 0 和 1 组成的串表示数据。开始,你可以将八位二进制数转换为两位十六进制数,然后使用 ASCII 码将两位十六进制数翻译成一个文本字符。

二进制计算机数据	01001000	01001001	00100000	01001011	01000101	01001110
十六进制程序员数据	0x48	0x49	0x20	0x4B	0xB5	0x4E
ASCII 文本数据	H	I		K	E	N

图 1.2 将二进制数翻译成文本字符

人们每天使用十进制数来表示钱的数目、实验结果等等。因此,人们更习惯使用十进制数。刚开始,你可能对使用二进制和十六进制表示数据,尤其是表示文本数据时感到不方便,但是随着使用次数的增加,你会感觉到使用它们很自然。

作为一个 Internet 程序员,你必须使用二进制数进行工作。对 Internet 编程时,你会发现使用十六进制数对二进制数和字节数据操作起来更加方便。如果你对二进制和十六进制数非常熟悉,将有助于更好地理解 Internet 编程的底层原理。

理解通信信号

计算机只使用包括 0 和 1 的二进制语言进行通信。作为 Internet 程序员,你必须掌握怎样使用这种语言进行通信。下面段落简单描述了计算机如何使用电信号表示二进制数据。对于你来说,详细了解这些信息并不是必须的。如果你很想知道计算机硬件怎样传输二进制数,下面段落将会很有用。

不管使用什么语言,要与其它人交换信息必须使用单词和句子。计算机不使用单词和句子而是使用电信号。单词和句子包含字母,数字和符号。计算机使用表示 0 和 1 的电信号进行通信。

电信号有数字和模拟两种形式。由于计算机使用数字信号表示 0 和 1 很方便,因此常常使用数字信号在内部表示数据。作为 Internet 程序员,你会发现模拟信号很有意思,当通过网络传输数据时,计算机经常采用模拟信号。可以将模拟信号看作波,就像在海洋中看到的波那样。图 1.3 显示了一个简单的模拟信号。



图 1.3 一个简单的模拟信号

注:产生信号的过程是研究电子学的人员而不是编程人员感兴趣的课题。本书不讨论怎样产生电子信号。

在固定时间间隔内(如一秒)波重复出现的次数叫做波的频率。模拟波按频率变化。例如,一个电路每秒可产生 100 个模拟波,另一个电路每秒可产生 1000 个。因此,你可以说产生 1000 个波的电路其频率比产生 100 个波的电路频率要高。

许多电路能够产生两种信号频率。也就是说,一个电路每秒能产生 1000 个模拟波,也可改变其频率,每秒只产生 100 个波。模拟波的频率变化过程叫做频率调制(FM)或者叫作普通

调制。例如,图 1.4 显示了两个模拟信号。你可以看到,左边的信号比右边的信号波数目要多。也就是说,左边信号的频率比右边的高。当计算机用二进制编码传输数据时。人们经常用频率高的模拟信号表示 1,使用频率低的信号表示 0。

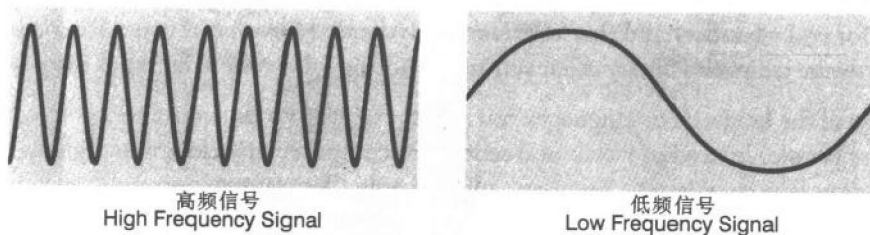


图 1.4 不同频率的模拟信号

电话线使用模拟信号携带信息。计算机使用调制解调器将二进制数转换为模拟信号,然后在电话线上传输。调制解调器使用频率调制以便表示二进制数据。modem 这个术语是调制/解调的缩写。调制解调器通过用 1 表示频率高的信号,用 0 表示频率低的信号来发送二进制数据。也就是说,为了表示二进制数,调制解调器对模拟信号的频率进行调制。图 1.5 显示了字母 A(ASCII 0x41 或者二进制 01000001)由调制解调器产生的模拟信号。当另一个调制解调器收到这个信号时,它对模拟信号进行解调,将模拟信号变为计算机理解 0 和 1 串。

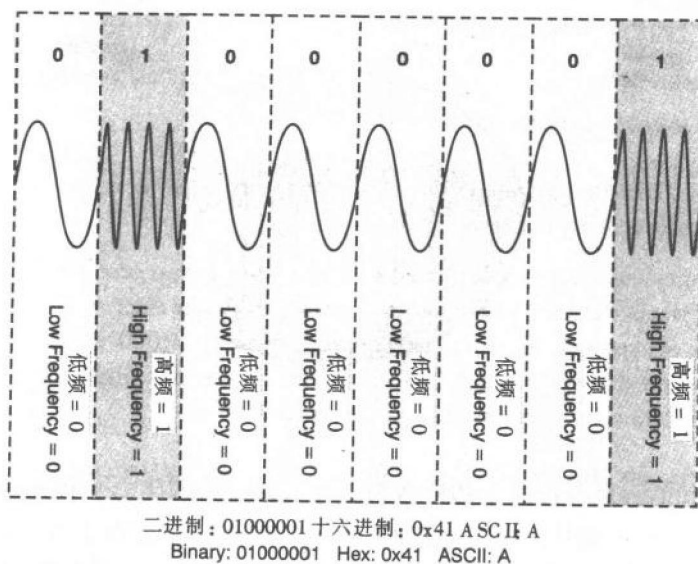


图 1.5 调制解调器产生的字母 A 的 FM 信号

计算机怎样传输数据

计算机与其它设备通信时,通常以串行或并行格式传输数据。大多数 PC 机使用并行方式将数据传送给打印机。并行通信意味着同时使用很多根线进行数据传输。为了发送一个字