

Win32 System Programming

(Second Edition)

Win32 系统编程

——Windows 2000 应用程序开发指南

第二版

Johnson M. Hart 著

刘立业 赵忠宇 齐占杰 译

Win32 System Programming (Second Edition)

Win32 系统编程 (第二版)

—— Windows 2000 应用程序开发指南

Johnson M. Hart 著

刘立业 赵忠宇 齐占杰 译

中国电力出版社

Win32 System Programming Second Edition

(ISBN 0-201-70310-6)

Johnson M. Hart

Authorized translation from the English language edition, entitled **Win32 System Programming Second Edition**, published by Addison Wesley, Copyright©2001

All rights reserved. NO part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the Publisher.

CHINESE SIMPLIFIED language edition published by China Electric Power Press Copyright©2003

本书由美国培生集团授权出版。

北京市版权局著作权合同登记号 图字：01-2001-2221 号

图书在版编目（CIP）数据

Win32 系统编程：第 2 版/（美）哈特著；刘立业等译。—北京：中国电力出版社，2003

ISBN 7-5083-1520-0

I. W... II. ①哈...②刘... III. 窗口软件, Win32—程序设计 IV. TP316.7

中国版本图书馆 CIP 数据核字（2003）第 048300 号

责任编辑：乔晶

书 名：Win32系统编程（第二版）

编 著：（美）哈特

翻 译：刘立业 赵忠宇 齐占杰

出版发行：中国电力出版社

地址：北京市三里河路6号 邮政编码：100044

电话：（010）88515918 传真：（010）88518169

印 刷：北京市铁成印刷厂

开 本：787×1092 1/16

印 张：24

字 数：534 千字

书 号：7-5083-1520-0

版 次：2003年8月北京第一版

印 次：2003年8月第一次印刷

定 价：45.00 元

译 者 序

随着 Windows 95 和 NT 的推出, Windows 操作系统家族实际上已经占据了桌面电脑和部分服务器的市场, 32 位的 Windows 操作系统已经成为个人计算机操作系统事实上的标准。而且已经出现的 Windows 2000 可以支持适应未来发展所需要的 64 位程序。所以每个从事程序设计的人都想要快速了解和掌握 Win32 程序设计。

要进行 Windows 程序设计是一件比较棘手的事情, 尤其对于想要快速了解和熟悉 Windows 编程的人员。他们经常不得不在厚厚的随机资料或者是英文文档中寻找可能只是一点点的有用的东西。事实上, 有很多非专业的, 甚至有很多原来是从事其他操作系统编程的人员, 由于需要, 他们非常希望将代码移植到 Windows 操作系统下。本书就是这样一本书, 能帮读者实现这个目的。

本书对于已经熟悉 Windows 操作系统编程的人来说也是非常有用的。它可以使读者更加深入了解 Windows 程序的运行机制, 从而编写出更加有效和简洁的应用程序。本书通过大量的例子和代码来说明如何实现这一点。

本书由孙兆林组织翻译, 参加主要部分翻译工作的还有刘立业、赵忠宇、邱兆坤、黄海雄、严朝晖等, 全书由刘立业统校, 灯芯工作室的其他工作人员完成全书的录入、校对和审稿等工作。本书的出版是集体劳动的结晶, 在此特别感谢工作室的全体工作人员。由于时间仓促, 且译者经验和水平有限, 译文难免有不妥之处, 恳请读者批评指正!

译 者

序 言

本书主要展示怎样应用 Win32 应用程序编程接口 (API) 和新出现的 Win64 API, 着重讲述核心系统服务, 包括文件系统、进程和线程管理、进程间的通信、网络编程以及同步。虽然用户界面、内部机制和 I/O 驱动程序是重要和有趣的话题, 但超出了本书的范围。本书的例子主要集中于有可能在现实中出现的情况, 许多情况下, 这些范例可用作实际应用程序的基础。

Win32 API 可以被微软的 32 位操作系统家族所支持, 这些操作系统是 Windows 2000、Windows NT、Windows 98、Windows 95 和 Windows CE。将会被 Windows 2000 当作 64 位接口所支持的 Win64 与 Win32 非常相似, 在需要的时候, 我们将会讨论到它的移植问题。毫无疑问, 对于应用开发者来说, Win32 与 Win64¹ 都非常重要, 对于桌面电脑和服务器的应用程序而言, Win32 在很多情况下正取代 UNIX 而成为一种首选的 API。许多观察家预测, 虽然看起来 UNIX 和 Win32 将会共存并且都有各自的生存环境, 但是 Win32 将会成为占统治地位的编程接口。

不论操作系统战争的结果如何, 许多有经验的编程者将想要快速地学习 Win32 API, 本书将会帮助他们实现这一点。

第一个目的是解释 Win32 是什么, 说明在实际情况下如何使用 Win32, 并且不会给读者在一些不必要的细节上加重负担, 从而使他们能够尽快地使用 Win32。因此, 虽然本书并不十分全面, 但它解释了最重要函数的主要特征以及在实际的程序设计中如何使用这些函数。只要具备了这些知识, 读者就能按照需求或兴趣利用复杂的 Microsoft 帮助文档研究其细节、高级选项和更加晦涩的函数。我发现用这种方法很容易学习 Win32 API, 尽管偶尔也会受挫, 但我已经很喜欢开发 Win32 程序了。这种对 Win32 的热情在学习的过程中不时表现出来。这并不意味着我感到 Win32 比其他操作系统 API 更加必要, 但是 Win32 的确有许多有吸引力的特色。

许多 Win32 的书籍假设读者只熟悉 16 位 Windows 3.1 程序设计。这些书花费大量的时间来说明进程、虚拟内存、进程间通信和抢占式调度 (preemptive scheduling) 如何工作, 然而并没有介绍在实际情况下如何使用它们。具有 UNIX、VAX VMS、IBM MVS 或其他高端操作系统 (也就是说, 除了很长时间不用的 Windows 3.1 Win16 API 之外的任何操作系统) 经验的程序员应该对这些概念很熟悉, 并且急于找出它们在 Win32 中如何实现。大多数 Win32 的书也在用户界面程序设计上花费了许多时间。本书避开用户界面, 不讨论简单的基于字符的控制台 I/O, 而是将注意力集中在重要的核心特征上。

本书接受 Win32 只是一个 OS (operating system, 操作系统) API 的观点, 其中它用来提供容易理解的特征集。许多程序员, 不管他们处于什么水平, 都需要快速地学习 Win32

¹ 只有在二者的区别比较明显时, 才会提到 Win64; 也就是说几乎所有的语句都能够很好地应用到 Win64 中, Win64 有可能成为服务器和某些工作站上大型应用程序的 API。

和 Windows 操作系统。理解 Win32 对于讨论如 Microsoft COM (Component Object Model, 组件对象模型) 这样的问题所起的作用是无法衡量的。同其他操作系统相比, Windows 操作系统有优点、缺点以及一般水平的特征和质量。本书的目的是如何有效地使用这些特征以及在实际情况中如何开发有用的、高质量的和高性能的应用程序。

读者对象

- 在 UNIX 或其他操作系统甚至是 Windows 3.1 上有编程经验以及想要快速了解 Win32 的任何读者。
- 为在 Windows 2000 和其他任何一种 Windows 平台上执行操作, 所以必须要将现有的应用程序 (通常是 UNIX 的程序) 转向 Win32 的程序员和软件工程师。本书包含了许多 Win32、UNIX 以及 C 库的函数和程序设计模型的比较。包括进程管理、同步、文件系统和进程间通信在内的所有常见的 UNIX 功能都能用 Win32 代替。
- 开发服务器或其他系统或组件的程序员, 其中首要的是管理资源, 如进程和线程, 其次是用户界面。
- 使用 COM 和 DCOM 的程序员将在此发现许多有助于理解 COM 的 DLL 用法、线程模型、界面和同步的信息。
- 将要开始新项目, 并且还没有必要转换现有代码的读者。本书覆盖了程序设计和实现方法的许多方面, 并且还涉及了使用 Win32 函数来创建可用的应用程序和解决一般的程序设计问题。
- 学习系统程序设计或应用程序开发的计算机专业高年级的大学生或低年级的研究生。本书对那些正在学习多线程或需要建立网络应用程序的人也是有用的。为了让研究人员可以比较 Win32 和 UNIX, 对于像 W. Richard Steven 的《Advanced Programming in the UNIX Environment》那样的书来说, 本书应该是一本与其互补的教材。学习操作系统课程的研究人员会发现本书是一本有用的补充教材, 因为它举例说明了一个商业化的重要操作系统如何提供基本的 OS 功能。

本书仅有的一个前提是读者需要具有一定的 C 程序设计知识。

组织结构

本书在组织章节时是按照主题进行组织的。因此, 首先讲述即使是单线程应用程序也必须包含的特性, 接着讲述进程和线程管理特性, 最后讲述多线程环境下的网络编程。这种组织方式允许读者按照从文件管理到安全性、内存管理和文件映射, 然后到进程、线程和同步, 接着再到进程间和网络通信的顺序依次提高。这种结构也允许范例以自然方式发展, 就像开发者先创建简单的原型, 然后再加入另外的性能。像异步 I/O 那样的高级特性会在最后出现。

每章中, 在介绍像进程管理或内存映射文件那样的功能之后, 我们会详细讨论重要的 Win32 函数及它们之间的关系。接着是举例说明。在本书中, 只列出程序的基本部分; 完整的程序和所需的包含文件、实用函数和类似的函数位于附录或 <http://www.infopower.com.cn> 中。我们会标识那些只被 Windows 2000 和 NT 支持的特性, 因为 Windows 98、95 和 CE 不能实现许多高级特性。每一章都给读者推荐其他相关的读物。

第 1 章是 Windows OS 家族和 Win32 的高级介绍。其中一个简单的示例程序展示了 Win32 程序设计风格的基本构成, 并为更高级的 Win32 特性铺设了基础。第 1 章中介绍了 Win64

和移植问题，这将在第 16 章中进一步介绍，并在本书其他必要的地方也会涉及到。

第 2 章和第 3 章处理文件系统、控制台 I/O、文件锁定和目录管理。Windows 2000/NT 使用的扩展字符集 Unicode 也在第 2 章介绍。范例包括顺序文件处理和直接文件处理以及目录遍历程序。第 3 章以注册表管理程序设计结束。注册表管理在许多方式上与文件和目录管理相似。

第 4 章介绍 Win32 的 SEH (Structured Exception Handling, 结构化异常处理) 性能。SEH 在本书中被广泛使用。许多书都把 SEH 推迟到最后一章介绍，但早点介绍它，我们就可以在整本书中使用，从而简化一些程序设计的任务并提高性能。

第 5 章解释了 Windows NT 对象安全性，并在一个范例中说明如何列举 UNIX 风格的文件权限。在适当的时候，还可以将安全性升级用于范例。尽管在后面的章节中将会用到安全性，但是对这个话题不感兴趣的用户可以跳过这一章。

第 6 章处理 Win32 内存管理以及说明如何使用内存映射文件来简化程序和提高性能。这一章也涉及到 DLL (dynamic link library, 动态链接库)。

第 7 章介绍 Win32 进程、进程管理和简单的进程同步。然后第 8 章类似地介绍线程管理。每一章的范例都说明了进程和线程的许多优点，包括程序的简易性和性能。

第 9 章和第 10 章提供了深层次展开的 Win32 线程同步处理，这是 Win32 强大的特性之一。同步是一个复杂的话题，为了帮助用户在避免许多缺陷时获得线程的编程和性能上的优点，这两章使用扩展的范例和易懂的模型。

第 11 章和第 12 章着重于进程和线程间的通信以及网络技术。第 11 章集中介绍 Win32 的关键特性，即管道 (pipe)、命名管道和邮槽 (mailslot)。第 12 章处理 Windows 套接字 (socket)。套接字主要使用 TCP/IP 的工业标准协议与非 Windows 进行通信。严格来说，Windows Socket 不是 Win32 的组成部分，它针对网络 and Internet 通信以及互用性，本书的第 12 章将讨论这个问题。多线程客户/服务器系统解释如何使用进程和线程通信。

第 13 章介绍 Windows 2000 和 Windows NT 如何提供“NT 服务”。NT 服务允许用户管理后台服务器，如第 11 和 12 章中创建的服务器。某些小的编程改动会将服务器变为 NT 服务。

第 14 章说明如何通过事件和完成例程来使用重叠 I/O。由于文件系统的原因，这个特性主要应用于 Windows 2000 和 NT，用户可以利用线程做到许多同样的事情。然而，紧密相关的 I/O 完成端口对可升级的多线程服务器是必要的，所以用第 11、12 和 13 章中所创建的服务器举例说明了这种特性。第 14 章还介绍了等待计时器，因为它们需要第 14 章介绍的概念。

第 15 章包括两个专业话题的概述：RPC (Remote Procedure Call, 远程过程调用) 和集成了本书许多概念的 Microsoft COM 对象模型。远程过程和 COM 对象经常使用本书所描述的特征。这一章包括简单的例程，并且还举例说明读者现在已经具有一些必要的信息来学习超出核心系统服务的其他 Win32 主题。

第 16 章描述 Win64 程序设计问题和如何做出应用程序的植入和移植计划。

本书有 3 个附录。附录 A 介绍了光盘中的程序以及如何使用它们。附录 B 包含用于比较 Win32 函数和 UNIX 以及标准 C 库中对应函数的几个表。附录 C 比较本书某些范例的可选实现方式之间的性能差别，从而用户能够在 Win32 特性 (包括基本特性和高级特性) 和 C

库之间进行权衡。

第二版的注意事项

第二版除了进行重要的更新和重新组织之外，还包括扩充的新资料。第二版的目的包括：

- 介绍 Windows 2000 和 Win64 移植。
- 扩充对线程和同步的讨论。第 9 章的例子和第 10 章都是新加入的。
- 增加对套接字和远程调用的介绍（第 12 和 15 章），以及增加对 NT 服务的介绍（第 13 章）。从而在开发服务和服务器时，特别是开发 Windows 2000 的服务和服务服务器时，可以处理重要的问题。
- 更加深入地介绍了线程和同步运行的含义（第 9 章和第 10 章）。
- 根据读者和学生的反馈来修正书中的不足，并且增加了注解，改进了书的结构并加入大量的细节。

UNIX、C 库注释和表

我们在书中的适当部分对 Win32 风格和功能性与相应的 UNIX（以及 LINUX）和 ANSI 标准 C 库特性做了比较。附录 B 的表用于比较函数。之所以包括此信息是因为许多读者熟悉 UNIX 并对这两个操作系统进行比较感兴趣。没有 UNIX 经验的读者可随意略过这些段落。用小字体表示这种讨论。

范例

设计的范例用于处理以下事情：

- 举例说明 Win32 函数一般的、具有代表性的和有用的应用。
- 与在程序开发、咨询和教授中所遇到的真正的编程情形相符。我的一些客户和课程参加者已经使用这些代码例子作为他们自己系统的基础。在我做咨询活动期间，我经常遇到与书中的例子相似的代码，在偶然情况下，我甚至看到直接从第一版中摘录的代码（你可以自由地摘录代码，而且还要很感谢你文档中的致谢）。这样的代码经常是 COM 和 C++ 对象的一部分。书中的范例是“现实世界”的例子并解决“现实世界”的问题。
- 重点是介绍函数如何发挥功能及其相互之间的作用，这通常不是你在阅读文档之后的第一个期望。整本书的文章和范例主要集中于函数间的相互作用而不是函数本身。
- 增加和扩展，为以前的简单和自然的解决方案增加了新性能以及研究了可选的实现方法。
- 在前面的几章中，许多范例实现了 UNIX 命令，如 ls、touch、chmod 和 sort，从而在创建有用的实用程序集²时，以熟悉的上下文显示 Win32 函数。同一命令的不同实现方式也将给我们一种与高级 Win32 特性比较性能优劣的方法。附录 C 包含了这些性能测试的结果。

² 一些商业和免费产品提供了完整的 UNIX 实用程序集；在此将不会试图对它们进行补充。虽然这些范例有用，但是它们主要是用于说明 Win32 特性的用法。然而，不熟悉 UNIX 的读者在理解这些程序及其功能时应该没有任何困难。

前面章节中的范例通常比较短小，但是后面章节在适当时会给出较长的范例。

所有的范例均在 Windows 2000 和 Windows NT 下经过调试和测试，在适当的时候也在 Windows 98 和 95 下经过测试。由于我们使用 Windows NT 4.0 版本，所以虽然大多数开发是在单处理器（基于 Intel 系统）上运行，但是许多程序也在多处理器上经过了测试。已经使用多个客户同时与服务器发生作用来测试客户机/服务器应用程序。即使是这样，仍不能保证程序的正确性、完整性和适于任意的用途。毋庸置疑，即使最简单的程序也可能有缺陷或在某些情况下失败；这是几乎所有软件的致命缺陷。然而，我将对反映关于程序不足的信息表示感谢。

本书的勘误表、其他的程序、读者贡献、另外的注解以及其他更多的信息包括在我的主页 <http://world.std.com/~jmhart> 中。这些代码将在发现和修订错误以及收到读者的输入信息时更新。如果你在书中遇到程序或任何资料上的困难时，你也可以访问我的主页，因为或许它们已被修订或解释了。如果没有你问题的答案，可以给我发 e-mail: jmhart@world.std.com。

致谢

在本书的第二版准备期间，有许多人提供了帮助和建议。特别是，我从读者和评论者那里收到了极好的非常宝贵的建议；他们的贡献大大地增加了本书的精确性和完整性。

第二版中资料、观点、洞察力、解决方案和说明的主要贡献者包括 Steve Evans、Mike Francis、Craig Hill、Mike Innes、Vadim Kavelerov、Thomas Ollson、David Poulton、Donna Reese、Andrew Tucker 和 Christian Vogler。许多读者和学生在使用第一版学习 Win32 课程的过程中也提供了建议，修正了本书的错误，他们所做的无形贡献实在太多而不能一一提及。本书的最好部分应属于其他人的影响，而错误部分则属于作者。Michael Slaughter 和 Gary Clarke 是杰出的编辑，他们提供恰当的鼓励和督促。Addison-Wesley 制作职员给出了极好的支持。在准备手稿排版期间，排版者 Elissa Armour 以不能代替的高水平的技能和耐性为本书作出了很大的贡献。

如果没有第一版，本书的第二版也不可能写成，为此我很感激（以字母排序）Ralph Davis、Ed Dekker、Bjørn Elstad、Alan Feuer、Peggy Harris、Carl Anton Holmboe、Shin-Wei Hwang、Shirley Kaltenbach、Joe Newcomer、Ali Rafieymehr、Edward Schiebel 和 Larry Schmuhl。

编写本书真正的鼓励和信任最终来自于我的父母，他们教会我笔下世界的重要性、力量和快乐。为此我永远感激他们。

Johnson (John) M. Hart
jmhart@world.std.com

目 录

译者序 序 言

第 1 章 从 Win32 和 Win64 开始	1
1.1 操作系统的本质	1
1.2 Win32 与 Windows 2000、NT、9x 以及 CE	3
1.3 Win32、标准和开放式系统	4
1.4 Win32 准则	5
1.5 为 Win64 做好准备	7
1.6 标准 C 库函数：何时用于文件处理	7
1.7 使用本书的前提	8
1.8 范例：简单顺序文件的复制	9
1.9 小结	13
1.10 练习	14
第 2 章 Win32 文件系统操作和字符 I/O	15
2.1 Win32 文件系统	15
2.2 文件命名	16
2.3 打开、读、写和关闭文件	16
2.4 插曲：Unicode 字符和普通字符	20
2.5 Unicode 方案	22
2.6 标准设备和控制台 I/O	23
2.7 范例：打印和提示	26
2.8 范例：错误处理	28
2.9 范例：复制多文件到标准输出	29
2.10 范例：ASCII 到 Unicode 的转换	30
2.11 文件和目录管理	32
2.12 范例：打印当前目录	35
2.13 小结	35
2.14 练习	36
第 3 章 高级文件和目录处理以及注册表	38
3.1 64 位文件系统	38
3.2 文件指针	38
3.3 获取文件大小	40
3.4 范例：查看文件结尾	41
3.5 文件属性和目录处理	43
3.6 范例：列出文件属性	46

3.7	设置文件时间	49
3.8	文件处理策略	50
3.9	文件锁定	50
3.10	注册表	53
3.11	注册表管理	55
3.12	范例：列举注册键和内容	58
3.13	小结	61
3.14	练习	61
第 4 章	结构化异常处理	63
4.1	异常和异常处理程序	63
4.2	浮点异常	68
4.3	错误和异常	69
4.4	范例：将错误当作异常	70
4.5	终止处理程序	71
4.6	范例：使用终止处理程序提高程序的质量	74
4.7	范例：使用 filter 函数	76
4.8	控制台控制处理程序	79
4.9	范例：控制台控制处理程序	80
4.10	小结	81
4.11	练习	81
第 5 章	Win32 对象安全性	82
5.1	安全属性	82
5.2	安全性概述：安全性描述符	83
5.3	安全性标识符	85
5.4	管理 ACL	86
5.5	范例：NTFS 文件的 UNIX 风格权限	87
5.6	范例：初始化安全属性	90
5.7	读取和改变安全性描述符	92
5.8	范例：读取文件权限	94
5.9	范例：改变文件权限	95
5.10	附加的安全性概述	96
5.11	小结	97
5.12	练习	98
第 6 章	内存管理、内存映射文件以及 DLL	99
6.1	Win32 内存管理体系	99
6.2	堆	101
6.3	管理堆内存	103
6.4	范例：用二分搜索树法排序文件	105
6.5	内存映射文件	110
6.6	范例：用映射文件进行顺序文件处理	114
6.7	范例：排序内存映射文件	116

6.8	范例：使用基指针	117
6.9	动态链接库	121
6.10	范例：显式链接文件转换函数	124
6.11	DLL 入口点	125
6.12	小结	126
6.13	练习	127
第 7 章	进程管理	128
7.1	Windows 进程和线程	128
7.2	进程创建	129
7.3	进程标识符	134
7.4	复制句柄	134
7.5	退出和终止进程	135
7.6	等待进程终止	136
7.7	环境块和字符串	137
7.8	范例：并行式搜索	138
7.9	进程运行时间	141
7.10	范例：进程运行时间	141
7.11	产生控制台控制事件	143
7.12	范例：简单的作业管理	144
7.13	作业对象	151
7.14	小结	152
7.15	练习	152
第 8 章	线程和调度	154
8.1	线程概述	154
8.2	线程基础	155
8.3	线程管理	156
8.4	在线程中使用 C 库	159
8.5	范例：多线程模式搜索	160
8.6	主机/工作者和其他线程模型	162
8.7	范例：合并排序——使用 SMP 的分而治之的方式	163
8.8	线程局部存储 (TLS)	167
8.9	进程与线程的优先权和调度	168
8.10	线程状态	169
8.11	缺陷和常见的错误	171
8.12	计时等待	171
8.13	Fiber	172
8.14	小结	175
8.15	练习	175
第 9 章	线程同步	177
9.1	线程同步的需求	177
9.2	线程同步对象	180

9.3	CRITICAL_SECTION 对象	181
9.4	用于保护共享变量的 CRITICAL_SECTION	182
9.5	范例：一个简单的生产者/消费者系统	183
9.6	互斥体	187
9.7	信号机	190
9.8	事件	193
9.9	范例：一个生产者/消费者系统	194
9.10	范例：同步性能影响	197
9.11	更多的互斥体和 CRITICAL_SECTION 原则	200
9.12	互锁函数深化	201
9.13	关于内存管理性能的考虑事项	202
9.14	小结	202
9.15	练习	203
第 10 章	高级线程同步	205
10.1	互斥体、事件和条件变量模型	205
10.2	范例：阈值屏障对象	209
10.3	队列对象	211
10.4	范例：在多级管道中使用队列	215
10.5	关于设计、调试和测试的提示	221
10.6	小结	223
10.7	练习	224
第 11 章	进程间通信	225
11.1	匿名管道	225
11.2	范例：使用一个匿名管道来实现 I/O 重定向	226
11.3	命名管道	228
11.4	命名管道事务函数	232
11.5	范例：一个客户端/服务器命令行处理器	235
11.6	客户端/服务器命令行处理器注释	240
11.7	邮槽	240
11.8	管道和邮槽创建、连接和命名	243
11.9	范例：客户端能够查找的服务器	243
11.10	线程模型注释	245
11.11	小结	246
11.12	练习	246
第 12 章	使用 Windows Socket 进行网络编程	248
12.1	Windows Socket	248
12.2	Socket 服务器函数	250
12.3	Socket 客户端函数	253
12.4	比较命名管道和 Socket	254
12.5	范例：Socket 消息接收函数	255

12.6	范例：基于 Socket 的客户端	256
12.7	范例：具有新特性的、基于 Socket 的服务器	258
12.8	进程内服务	262
12.9	面向行的消息、DLL 入口点和 TLS	264
12.10	范例：一个用于 Socket 消息的线程安全的 DLL	264
12.11	范例：一个替代的线程安全的 DLL 策略	268
12.12	数据报	270
12.13	Berkeley 与 Windows Socket 比较	271
12.14	Windows Socket 的重叠 I/O	271
12.15	Windows Socket 2	271
12.16	小结	271
12.17	练习	272
第 13 章	NT 服务	274
13.1	编写 NT 服务——概述	274
13.2	main()函数	275
13.3	ServiceMain()函数	276
13.4	服务控制处理程序	279
13.5	范例：服务“包装”	279
13.6	管理 Windows NT 服务	282
13.7	总结：服务操作和管理	284
13.8	范例：服务控制命令处理程序	284
13.9	与服务共享内核对象	287
13.10	事件日志	287
13.11	调试服务的注意事项	287
13.12	小结	288
13.13	练习	289
第 14 章	异步输入/输出和完成端口	290
14.1	Win32 异步 I/O 概述	290
14.2	重叠 I/O	291
14.3	范例：文件句柄同步	294
14.4	范例：带有重叠 I/O 和多缓冲器的文件转换	294
14.5	带有完成例程的扩展 I/O	297
14.6	范例：带有扩展 I/O 的文件转换	301
14.7	带有线程的异步 I/O	303
14.8	可等待计时器	304
14.9	范例：使用可等待计时器	305
14.10	I/O 完成端口	306
14.11	范例：使用 I/O 完成端口的服务器	309
14.12	小结	312
14.13	练习	313

第 15 章 远程过程调用和 COM 概述	314
15.1 远程过程调用	314
15.2 基本的 RPC 体系结构	315
15.3 RPC 接口定义	316
15.4 范例：接口定义	317
15.5 范例：RPC 客户端	318
15.6 范例：RPC 服务器	320
15.7 COM 和 DCOM 简要概述	322
15.8 小结	324
15.9 练习	325
第 16 章 Win64 编程	326
16.1 64 位体系结构概述	326
16.2 Win64 编程模型	328
16.3 数据类型	328
16.4 3 个 Win64 编程模型	330
16.5 传统代码转换	331
附录 A 范例程序的使用	333
A.1 范例程序的组织结构	333
A.2 包含文件的清单	338
A.3 附加的实用程序	343
附录 B Win32、UNIX 和 C 库文件的比较	346
B.1 第 2 和 3 章：文件和目录管理	347
B.2 第 4 章：结构化异常处理	349
B.3 第 5 章：保护 Win32 对象	350
B.4 第 6 章：内存管理、内存映射文件和 DLL	351
B.5 第 7 章：进程管理	352
B.6 第 8 章：线程和调度	353
B.7 第 9 和 10 章：线程同步	354
B.8 第 11 章：进程间通信	354
B.9 第 14 章：异步 I/O	355
附录 C 性能结果	356
C.1 测试配置	356
C.2 性能测量	357
C.3 运行测试	364

第 1 章

从 Win32 和 Win64 开始

本章介绍微软 Windows 操作系统家族（Windows 2000、Windows NT、Windows 98、Windows 95 和 Windows CE）和这些操作系统所共同使用的 Win32 API（Application Programming Interface, 应用程序编程接口）。新的 64 位 Win64 API 以及 Win32 和 Win64 API 之间的可移植性在需要时也将讨论。从现在开始，我们将主要介绍 Win32，除了特别指出外，这些都可以应用于 Win64。

像其他的任何一种操作系统的 API 一样，Win32 API 有它自己的一系列的约定和编程方法，这些是从 Windows 操作系统的基本原理中推导出来的。一个简单文件复制的例子说明了 Win32 编程的风格，并且文件管理、进程管理、内存管理以及高级特性（如线程同步）也使用同样的风格。为了使 Win32 与更熟悉的编程风格形成对照，这个例子也采用标准 C 库实现。

第一步是回顾一下现代操作系统所必须提供的基本特性，从这里，将会学到如何在 Win32 中使用这些特性。

1.1 操作系统的本质

Win32 使核心的操作系统特征对从便携式电脑到企业级服务器整个范围都可用。可以通过考虑最重要的、必须进行管理的资源来描述操作系统的特性，这些资源包括：

- 内存。OS（operating system, 操作系统）管理着大量平面虚拟内存地址空间，并且能够在内存和磁盘存储器之间透明地移动信息。
- 文件系统。操作系统管理命名的文件空间，不仅能够提供直接和顺序访问目录和文件的功能，还能管理文件和目录。大多数的系统都有一个分层的命名空间。
- 资源命名和位置。文件命名可以采取较长的具有描述作用的名称，并且命名规则扩展到诸如设备、同步以及进程间通信等对象。操作系统也可以定位和管理命名对象的访问。
- 多任务。操作系统必须管理进程、线程以及其他独立的和同时执行的单元。操作系统按照动态决定优先权来优先执行和安排任务。
- 通信和同步。OS 既可以在单系统中管理任务间的通信和同步，也可管理网络系统间和 Internet 间的通信。
- 安全和保护。OS 提供了灵活的机制来防止对资源未经许可的、意外的访问和破坏。

Win32 API 能够支持以上这些特性，并使它们可用于满足某一范围平台上一般的 OS 要求。虽然在有些情况下会受到限制，但是 Win32 API 能够被所有的操作平台所支持。目前这些操作平台统称为“Windows”，包括：

- Windows 2000。Windows 2000 用来取代 Windows NT，并且它的目标是占领服务器和高端桌面电脑的市场。
- Windows NT 4.0（以前常用的版本是 3.51）。因为 3.51 版和早期的服务包¹不能支持一些新的特性，所以从现在开始，如果提及 Windows NT，我们认为是 NT Version 4.0 Service Pack 3（SP3）。NT 5.0 版实际上就是 Windows 2000。
- Windows 98 和 Windows 95（或简化为 Windows 9x，因为二者之间差别很小）。它们是桌面电脑的主要操作系统，其安全特性不如 NT 和 2000。
- Windows CE。它的目标是小型系统，如掌上电脑和嵌入式处理器，它能够支持 Win32 特性的一些子集。

如果只是支持以上的特性，Windows 并不是惟一的。毕竟，许多专有操作系统都有这些特性。在众多的系统中，UNIX 就长期存在。使用 Win32 会在商业和技术上占有其他的优势：

- Windows 在市场上占有统治地位，特别是在桌面电脑市场。因此，Windows 应用程序有着很大的目标市场，仅在数量上就有几千万，而包括 UNIX 和 Macintosh 在内的其他桌面电脑操作系统就相形见绌了。
- Windows 操作系统在市场上的统治地位意味着应用程序和软件开发工具可以广泛而廉价地应用于 Windows。而且，Windows 在不断地改革和创新。
- Windows 应用程序使用广为熟知的图形用户界面（GUI）。
- Intel X86、Pentium 和兼容处理器都可以运行 Windows。NT 也可以在 Digital Alpha 处理器上使用（但 2000 不能用于非 Intel 体系），Windows CE 可运行在各种处理器上。2000 和 NT 也能够支持 SMP（symmetric multiprocessor，对称多处理器）系统。Windows 2000 和 NT 可以运行在桌面电脑上，并且它们也能够支持部门和企业服务器以及高性能的工作站²。
- Windows 2000 和 NT（Windows 9x 和 CE 除外）的安全等级为国家安全局（NSA）C2 级，UNIX 和其他操作系统的某些方面并不是标准。
- 绝大多数操作系统（UNIX 除外）仅为少数系统体系所独有。
- Windows 操作系统有许多标准 UNIX 并不具备的特性，尽管它们可能在某些 UNIX 工具中是可用的。线程和 C2 级的安全特性就是两个例子。

总之，Windows 提供了现代操作系统的功能，并且可以通过采用友好的用户界面运行以前仅限于 UNIX、多用户或者便宜的 PC 机主框架系统上的大型应用程序。而且，Windows 操作系统可在从桌面电脑到企业服务器广泛的范围内运行。技术特点和商业需要使开发

¹ Microsoft 不时地发行服务包来修正错误、增加新功能以及提高性能。常见的是 SP4、SP5 有时称为“嵌入式 NT”。

² 本书的程序既可在陈旧的有 16MB RAM 的 486 上运行，又可在具有 4 个处理器、2GB RAM、500MHz 的 PIII 企业服务器上运行，这是由于 Win32 主系统覆盖的范围。