开发人员专业技术丛书

HZ BOOKS 华章科技

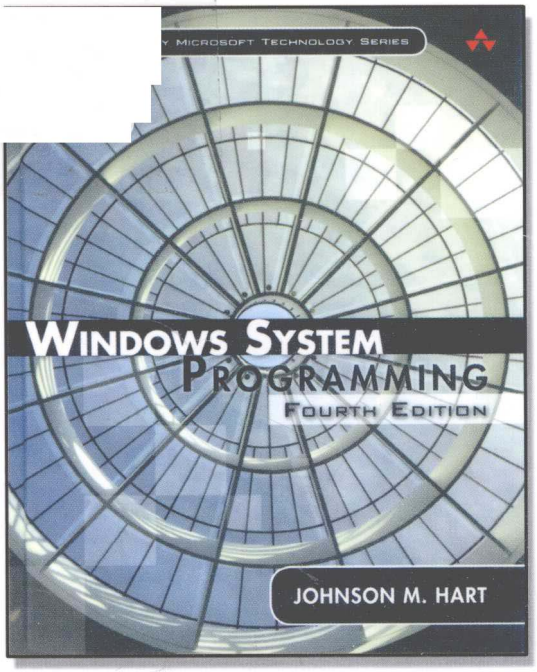
PEARSON

（原书第4版）

Windows系统编程

Windows System Programming (4th Edition)

- 经典Windows系统编程教科书
- Windows API编程权威指南
- 针对Windows 7、Windows Server 2008和Windows Vista全面更新



（美） Johnson M. Hart 著

戴锋 陈征 等译

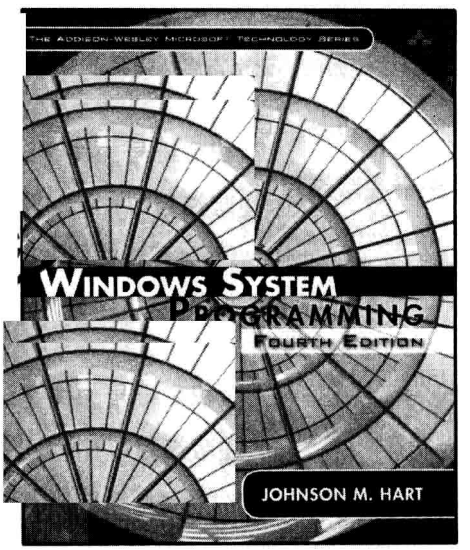
机械工业出版社
China Machine Press

开发人员专业技术丛书

（原书第4版）

Windows系统编程

Windows System Programming (4th Edition)



（美）Johnson M. Hart 著

戴锋 陈征 等译

机械工业出版社
China Machine Press

本书是介绍使用 Microsoft Windows 应用程序编程接口进行应用程序开发的专著，专注于文件系统、进程和线程管理、进程间通信、网络编程以及同步等核心系统服务。本书的示例都来自现实场景，其中有许多是基于作者在实践中所开发的真实应用程序。本书的目的在于展示如何高效地在现实中使用 Windows API 特性来开发高质量、高性能的应用程序。

本书适合以下读者阅读：任何有 C 或C++ 编程知识且想快速学习 Windows 应用程序开发的开发人员、计算机科学专业学习系统编程或应用程序开发课程的高年级学生以及 UNIX/Linux 程序员。

本书版权登记号：图字：01-2010-2461

图书在版编目（CIP）数据

Windows 系统编程（原书第 4 版）/（美）哈特（Hart J. M.）著；戴锋，陈征等译. —北京：机械工业出版社，2010.10

（开发人员专业技术丛书）

书名原文：Windows System Programming, Fourth Edition

ISBN 978-7-111-31668-8

Ⅰ.W…　Ⅱ.①哈…　②戴…　③陈…　Ⅲ.视窗软件，Windows－程序设计　Ⅳ.TP316.7

中国版本图书馆 CIP 数据核字（2010）第 166806 号

机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码　100037）

责任编辑：秦　健

北京京北印刷有限公司印刷

2010 年 10 月第 1 版第 1 次印刷

186mm×240mm · 25 印张

标准书号：ISBN 978-7-111-31668-8

定　价：65.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991；88361066

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

译 者 序

在当今的应用程序开发领域，Web 方面热闹非凡，大量基于 Web 的应用层出不穷，大有替代桌面系统的趋势。但在其热闹的背后，只要我们冷静下来就不难发现，无论世界如何变化，应用软件怎能离开操作系统的支持？而当我们提及 Windows 操作系统时，则无法不冠之以“优秀”两个字。

学习 Windows 编程接口、熟悉 Windows 的编程特性，是想为 Windows 编写应用程序的所有程序员不能不做的事情。本书就是一本介绍 Windows 编程接口的书，涵盖 Windows 内核的各个方面，从简单的文件系统到颇为复杂的进程间通信和同步问题，甚至是 NT 6 内核的新特性。

无论读者使用的是 Windows API 还是 MFC/ATL 等类库甚至是. NET Framework 来编写Windows 应用程序，本书都是极为难得的参考材料。对于 UNIX/Linux 程序员，本书也有一定的参考价值。

参加本书翻译的人员有：戴锋、孟庆麟、陈征、陈宗斌、许瑛琪、张景友、易小丽、陈婷、管学岗、王新彦、金惠敏、张海峰、徐晔、张德福、张士华、张锁玲、杜明宗、高玉琢、王涛、申川、孙玲等。

由于时间紧迫，加之译者水平有限，错误在所难免，恳请广大读者批评指正。

前　言

本书描述使用 Microsoft Windows 应用程序编程接口（Application Programming Ingterface，API）进行应用程序开发的方法，专注于文件系统、进程和线程管理、进程间通信、网络编程以及同步等核心系统服务。本书的示例都来自现实场景，其中有许多是基于作者在实践中所开发的真实应用程序。

Microsoft 的 32 位和 64 位操作系统家族（目前广泛使用的版本包括 Windows 7、Vista、XP、Server 2003、Server 2008 以及 CE）支持 Win32/Win64 API（也就是 Windows API）。更早一些的 Windows 家族成员有 Windows 2000、NT、Me、98 和 95，虽然这些系统已经过时，但本书中的许多主题仍旧适合于这些老系统。

Windows API 是应用程序开发的重要因素，经常代替 POSIX API（UNIX 和 Linux 支持它）作为桌面、服务器和嵌入式系统应用程序开发的 API，不仅现在如此，将来也是如此。许多程序员，无论经验如何，都想快速学习 Windows API，而本书正是为他们所写的。

目标和方法

本书的目标是：在不给读者过多不必要的细节的情况下尽可能快地讲解 Windows 是什么以及如何在实际中使用它。本书不是一本参考指南，而是讲解那些最重要的函数的核心特性，并且展示如何在实际编程中将它们一起使用。有了这些知识，读者就可以通过广泛的 Microsoft 参考文档来探究更多细节和更高级的选项，根据需求或兴趣探究更为隐秘的功能。这种方法会让 Windows API 的学习更为容易，而且让开发 Windows 程序成为一种享受，即使偶尔也会有挫折。这种热情将时不时地显露出来。当然，Windows 并不一定比其他操作系统 API 更好，但它的确有许多引人入胜的特性，而且随着每个新版本的推出都得到极大的改进。

许多 Windows 书籍用大量篇幅讲述进程、虚拟内存、进程间通信和抢占式调度，却没有展示在实际情况下如何使用它们。有 UNIX、Linux、IBM MVS 或其他操作系统经验的程序员对这些概念并不陌生，他们会急于了解如何在 Windows 中实现它们。大多数 Windows 书籍也会在用户界面编程这一重要主题上占用大量篇幅。本书有意避开用户界面的问题，而仅讨论简单的基于字符的控制台 I/O，以便专注于重要的核心功能。

Windows 只是一套操作系统 API，提供了清晰明了的功能集合。许多程序员，无论经验如何，都需要快速学习 Windows。而且，对于在 Microsoft. NET Framework 上进行开发的程序员而言，如果能够理解 Windows API，那将是非常宝贵的背景知识。

当我们将 Windows 系统与其他系统做比较时，从特性和质量上看会有好的、坏的以及不相上下的结果。最新的版本（Windows 7、Vista、Server 2008）提供了新的特性，比如条件变量（condition variable），既改进了性能也简化了编程。本书的目的在于展示如何高效地在现实中使用这些特性来开发高质量、高性能的应用程序。

读者对象

感谢不同的目标读者提供了有价值的建议、想法和反馈，这些读者包括：

- 任何想快速学习 Windows 应用程序开发的人，无论以前有无经验。
- 想要将现有 Linux 或 UNIX（POSIX API）应用程序移植到 Windows 的程序员与软件工程师。源代码经常需要继续支持 POSIX，也就是说，源代码需要有可移植性。本书经常比较 Windows、POSIX 和标准 C 库函数和编程模型。
- 开始进行新项目的开发人员。他们不受移植现有代码的限制，程序设计和实现的许多方面都包括在内，可以使用 Windows 函数来创建有用的应用程序和解决常见的编程问题。
- 需要理解 Windows 能力和原理的应用程序架构师和设计师。
- 使用 COM 和. NET Framework 的程序员。如果需要理解动态链接库（DLL）、线程的使用和模型、接口以及同步问题，他们将从本书中受益良多。
- 计算机科学专业学习系统编程或应用程序开发课程的高年级本科生或低年级研究生。本书对于那些正在学习多线程编程或需要构建联网应用程序的人也有帮助。本书将与《Advanced Programming in the UNIX Environment》（由 W. Richard Stevens 和 Stephen A. Rago 著）这样的经典书籍互补，学生可以借此比较 Windows 和 UNIX。学习操作系统课程的学生会发现本书是有用的补充，因为它阐明了重要的商业操作系统提供实质功能的方法。

本书惟一的假定是读者要有 C 或C++ 编程知识。

Windows 自上个版本以来的发展

本书第 1 版名为《Win32 System Programming》，出版于 1997 年，后来又出版了第 2 版（2000 年）和第 3 版（2004 年）。自这些版本出版至今，很多内容发生了变化，而 Windows 则是快速发展中的计算技术的一部分。编写第 4 版的主要因素如下：

- Windows API 极为稳定。编写于 1997 年的程序仍能运行于最新的 Windows 版本上，现在甚至是多年前学习的 Windows 技能对于未来的数十年而言仍将是有价值的。
- 无论如何，API 扩展了，有了新的功能与函数，有时是强制性的。例如，在以下几个方面：1）易于处理大文件和大的 64 位地址空间的能力；2）线程池；3）新的能够高效解决重要的同步问题的条件变量。
- Windows 的规模小至电话、手持和嵌入式设备，大到笔记本电脑和桌面系统，甚至是最大的服务器。
- Windows 从 1997 年所需要的不算苛刻的资源（16MB 内存和 250MB 空闲硬盘空间）开始扩展，在数量级更大、更快却更便宜（通常来说）的系统上高效运转。
- 64 位系统、多核处理器和大文件系统随处可见，应用程序必须能够用上这样的系统。通常，

程序还必须能够继续运行于32位系统上。

第4版中的变化

第4版提供了大量新材料，更新并重新组织了原来的内容，以便适应最新的进展，主要表现在以下各方面：

- 覆盖了Windows 7、Vista和Server 2008的新特性。
- 通过屏幕截图来演示示例程序的操作和性能。
- 描述并阐明能够确保相关应用程序伸缩到64位系统上运行并使用大文件的技术。全书中的改进都涉及这个问题。
- 去除了对Windows 95、98和Me（即“Windows 9x”家族）和NT等其他过时系统的讨论。程序示例默认采用当前Windows版本所支持的特性。
- 对线程、同步和并行计算的讨论做了改进，包括性能、可伸缩性和可靠性方面的考虑。
- 强调运行高性能、可伸缩、多线程应用程序的Windows服务器的重要角色和新特性。
- 学习不同的程序设计之间的性能问题（performance implication），尤其是运行于多核系统上的带有同步和并行能力的文件访问和多线程应用程序。
- 讲解源代码的可移植性，以便确保能在Windows、Linux和UNIX操作系统上运行。附录B对原来的版本做了改进，以便为需要构建运行于多目标平台上的代码（通常是服务器应用程序）的那些人提供帮助。
- 加入了大量来自优秀读者和评审者的反馈，减少了瑕疵，改进了讲解方式、组织结构以及大量细节信息。

组织结构

本书按主题来组织章节，首先讲解单线程应用程序所需的功能，而后讲解进程和线程管理功能，最后讲解多线程环境中的网络编程。这样组织使得读者可以按以下逻辑顺序阅读：先是文件系统、内存管理和文件映射，然后是进程、线程和同步，最后是进程间通信、网络通信与安全。这样组织也让我们可以以自然的方式演示示例，就如开发人员可能一开始创建一个简单的原型而后添加更多功能那样。高级特性，比如异步I/O和安全性，将在最后出现。

每一章都是先对功能域进行介绍，比如进程管理或内存映射文件，然后再详细讨论重要的Windows函数及其关系。随后是说明性的示例。文中仅列出关键程序段，完整的项目、程序、包含文件、实用函数和文档可在本书的Web站点（www.jmhartsoftware.com）下载。本书仅给出当前Windows版本所支持的功能。每章都会给出相关的阅读建议并提供一些习题，许多习题涉及的是文中不包含但有趣且重要的问题，而另外一些习题则为读者探究高级或特定主题给出了建议。

第1章是对Windows操作系统家族和Windows的初步介绍。我们通过一个简单的示例程序来展示Windows编程风格的基本元素，并且为讲述更高级的Windows特性奠定基础。Win64兼容性问题从第1章就开始引入，并将贯穿全书。

第2章和第3章的内容与文件系统、控制台I/O、文件锁和目录管理有关。Unicode（Windows所用的扩展字符集）也在第2章介绍。本章的示例包括顺序和直接文件处理、目录遍历和管理。第

3 章最后讨论注册表管理编程，它在许多方面与文件管理和目录管理相似。

第 4 章介绍 Windows 异常处理，包括在本书中广泛使用的结构化异常处理（Structured Exception Handling，SEH）。早一些介绍 SEH，就可以在全书中使用 SEH 简化一些编程任务并提高质量。本章也描述了向量化异常处理（vectored exception handling）。

第 5 章探讨 Windows 内存管理并展示使用内存映射文件来简化编程、提高性能的方法。本章也讲解 DLL。本章还有一个在 32 位和 64 位系统中分别比较内存映射文件访问与普通文件 I/O 之间的性能与可伸缩性的示例。

第 6 章介绍 Windows 进程、进程管理和简单的进程同步。第 7 章描述线程管理并介绍并行性，以便利用多处理器系统。每一章的示例展示了使用线程和进程所带来的许多好处，其中包括程序简单性和性能改进。

第 8、9 和 10 章广泛、深入地探讨了 Windows 线程同步、线程池和性能方面的问题。这些主题比较复杂，所以这些章节使用广泛的示例和清晰明了的模型来帮助读者既获得线程所带来的编程和性能上的好处，又避免落入许多陷阱之中。新加入的材料包括新的功能以及性能和可伸缩性问题，这在构建包括那些将在多处理器系统上运行的基于服务器的应用程序时非常重要。

第 11 章和第 12 章讨论进程间和线程间的通信以及联网。第 11 章关注完全属于 Windows 的部分，也就是匿名管道、命名管道和邮槽。第 12 章讨论 Windows 套接字，这是可以与使用业内标准协议（主要是 TCP/IP）的非 Windows 系统互操作的技术。Windows 套接字严格上说并不属于 Windows API 的一部分，它提供网络和 Internet 通信以及可互操作性，这也与本书剩余部分的主旨一致。我们使用一个多线程的客户/服务器系统来演示使用进程间通信和线程的方法。

第 13 章描述 Windows 如何允许服务器应用程序（比如，第 11 章和第 12 章创建的）转换成可以作为后台服务程序管理的 Windows 服务（Windows Service）。进行一些小改动就可将服务程序转换成服务。

第 14 章展示使用带有事件和完成例程的重叠 I/O 执行异步 I/O 的方法。这也可以使用线程来实现，所以我们提供了比较不同解决方案的简单性和性能的示例。此外，对于 Windows Vista 而言，完成例程提供了非常优秀的性能。对于某些可伸缩的多线程服务器而言，与此紧密相关的 I/O 完成端口很有用，所以前面章节中的服务器程序就阐明了这个特性。最后一个主题是可等待的计时器，它需要本章前面介绍的概念。

第 15 章简要讲解了 Windows 对象安全性，并且以一个示例展示了模拟 UNIX 风格的文件权限的方法。另外还有一些示例展示如何为进程、线程和命名管道提供安全。如此一来，就可以酌情对早先的示例进行安全升级。

本书有三个附录。附录 A 描述可从本书 Web 站点（www. jmhartsoftware. com）下载的示例代码。附录 B 展示如何创建可以运行于 POSIX（Linux 与 UNIX）系统中的源代码，对于服务器应用程序和那些不仅仅需要支持 Windows 系统的组织来说很有用。附录 C 比较了文中一些示例的不同实现的性能，以便读者在 Windows 的基本特性和高级特性之间取舍。

UNIX 和 C 库的注释和表

我们会在书中适当的地方将 Windows 风格和功能与同类的 POSIX（UNIX 和 Linux）和 ANSI

标准 C 库功能进行对比。附录 B 回顾源代码的可移植性并包含一个列出这些同类函数的表。包含这些内容有两个主要理由：

- 许多人熟悉 UNIX 或 Linux，而且有兴趣对比这两个系统。如果读者没有 UNIX/Linux 背景，可跳过书中的这些内容。
- 对于许多开发人员和组织而言，源代码可移植性很重要。

示例

示例的设计旨在：

- 演示 Windows 函数常见的、有代表性的以及有用的应用。
- 与在程序开发、咨询和培训中所遇到的真实编程情况相联系。一些客户以及参加我的课程的人基于这些示例开发他们自己的系统。在提供咨询时，我经常碰到与示例中所用的代码相似的代码，有几次还碰到从前一版本的书中直接取来或经过修改的代码（读者可随意这样做，如果能在文档中致谢我将不胜感激）。此代码经常作为 COM、.NET 或C++ 对象的一部分。由于时间和空间的限制，这些示例是“真实世界”的示例并且解决“真实世界”的问题。
- 强调函数的实际行为以及与实际应用程序的交互，这与读者在阅读了文档之后所想到的并不总是一致。本书中的正文和示例都专注于函数间的交互，而不是函数本身。
- 增长与扩张，既以自然的方式对前面的解决方案添加新功能，又探究可选的其他实现技术。
- 实现 UNIX/Linux 命令，比如 ls、touch、chmod 和 sort，这样既以熟悉的上下文展示 Windows 函数，又创建一组有用的工具[⊖]。对相同命令的不同实现也让我们可以很容易地对高级 Windows 特性所具有的性能优势进行比较。附录 C 包含性能测试结果。

前面章节中的示例通常较短，后面章节中的示例会适当长一些。

每章末尾的习题提供可研究的另一种设计、主题，以及额外的、超出本书范围但却是重要的功能。有些习题很容易，而有一些则颇具挑战性。我们经常给出清楚标注的有瑕疵的解决方案，因为解决错误是提高技能的好方法。

所有的示例都在 Windows 7、Vista、Server 2008、XP 以及更早的系统下调试、测试过。测试包括 32 位和 64 位两个版本。所有程序也都在单处理器和多处理器系统下测试过，最多用到 16 个处理器。客户/服务器应用程序的测试使用了多个客户同时与服务器交互。但是，我们并不对程序的正确性、完整性或者针对任何目的的适用性提供保证。毫无疑问，即使是最简单的示例也包含瑕疵或者在某些情况下会出错，这是几乎所有软件的共性。但是，如果能将任何与程序瑕疵有关的信息提供给我，我将甚为感激；如果能有这些瑕疵的修改建议则更为理想，我会将这些信息张贴在本书的 Web 站点上，以便让所有人受益。

Web 站点

本书的 Web 站点（www.jmhartsoftware.com）包含可下载的示例文件，其中包括书中所有示例

⊖ 许多商业和开源产品提供了完整的 UNIX/Linux 工具集，我们并不想对此补充。这些示例虽然有用，但主要目的是阐明 Windows 的使用。即使对 UNIX 或 Linux 不熟悉的人，在理解这些程序及其功能时也不会有任何困难。

的完整代码和项目、一些习题答案、可选的实现方法、教学以及性能评估测试。这些材料将定期更新，以便把新材料和修正包括进来。

这个站点还包括本书的勘误表、更多的示例等。本站点也包含 PowerPoint 幻灯片，可用于非商业性的教学目的。我在专业培训课程中多次使用这些幻灯片，它们也适用于大学课程。

在瑕疵得到修正以及收到新内容时，这些材料会进行更新。如果读者对程序或书中的任何材料有疑问，请先查看本站点上是否有修正或解释。如果没有，请发电子邮件至 jmh_assoc@ hotmail. com 或 jmhart62@ gmail. com。

致谢

在我准备第 4 版期间，许多人给了我帮助、建议和鼓励，读者提供了许多重要的想法并纠正了许多错误。本书 Web 站点对为第 4 版的编写做出重要贡献的人表示了感谢，而前三个版本则对早先有贡献的人表示了感谢。完整的列表请参阅 Web 站点。

对以下三位评审者的深度评论、耐心、优秀建议和深入的专业知识致以深深的谢意：Chris Sells、Jason Beres 和 Raymond Chen。他们（尤其是 Raymond Chen）对本书的改进做出了重大的贡献。我尽最大努力来展现他们的观点和无价的内容。

需要特别感谢的还有许多朋友和同事。多年以来，我从他们身上受益良多，他们的许多想法以各种形式融入到本书之中。他们也慷慨地为测试系统提供了访问权限。我尤其想感谢我在 Sierra Atlantic、Cilk Arts（现在是 Intel 的一部分）、Vault USA 和 Rimes Technologies 的朋友们。

排版员 Anne H. Smith 用她的技艺、执着和耐心为这一新版本的出版进行了准备，没有她的协助，本书就不可能完成。Anne 和她的丈夫 Kerry 也慷慨地在他们的设备上测试了示例程序。

Addison-Wesley 的员工展现出的职业精神和专业技能让作者的工作成为一种乐趣。编辑 Joan Murray 和主任编辑 Karen Gettman 从本项目之初就一起工作，确保项目按部就班地顺畅进行。助理编辑 Olivia Basegio 对整个过程进行了管理，生产部门的 John Fuller 和 Elizabeth Ryan 让本书的面世过程看起来是如此简单。项目编辑 Anna Popick 为最终的编辑工作和日程表给予了指导。文字编辑和校对人员 Carol Lallier 和 Lori Newhouse 为本书的可读性和一致性做出了重要贡献。

作 者 简 介

Johnson（John）M. Hart 是一位顾问、专攻 Microsoft Windows 和. NET 应用程序开发、开放系统计算、技术培训和写作以及软件工程等领域。作为软件工程师、经理、工程主管以及 Cilk Arts 公司、Sierra Atlantic、HP 和 Apollo Computer 的高级技术顾问，他有超过 25 年的经验。John 还开发并提供 Windows、UNIX 和 Linux 方面的专业培训课程，而且曾在肯塔基大学做过 9 年的计算机科学教授。他是技术、贸易和学术文章以及包括《Windows 系统编程》4 个版本在内的许多书籍的作者。

目　录