

Microsoft

微软技术丛书

- ★ 权威力作 再度新鲜亮相
- ★ 名著名译 彰显经典魅力
- ★ 中英联动 丰富阅读体验
- ★ 深刻剖析底层实现机理
- ★ 直击Windows编程精髓

Fifth Edition

Windows核心编程 (第5版)

Windows via C/C++

(美) Jeffrey Richter (法) Christophe Nasarre 著

葛子昂 周靖 廖敏 译



Microsoft®
Most Valuable
Professional

微软最有价值
专家联袂推荐



清华大学出版社

微软技术丛书

Microsoft

5

Fifth Edition

Windows核心编程
(第5版)

Windows via C/C++

(美) Jeffrey Richter (法) Christophe Nasarre 著
葛子昂 周靖 廖敏 译

清华大学出版社
北京

内 容 简 介

这是一本经典的 Windows 核心编程指南,从第1版到第5版,引领着数十万程序员走入 Windows 开发阵营,培养了大批精英。

作为 Windows 开发人员的必备参考,本书是为打算理解 Windows 的 C 和 C++ 程序员精心设计的。第5版全面覆盖 Windows XP, Windows Vista 和 Windows Server 2008 中的 170 个新增函数和 Windows 特性。书中还讲解了 Windows 系统如何使用这些特性,我们开发的应用程序又如何充分使用这些特性,如何自行创建新的特性。

Windows via C/C++ (978-0-7356-2424-5)

Copyright © 2008 by Jeffrey Richter and Christophe Nasarre

Original English Language Edition Copyright © 2008 by Jeffrey Richter and Christophe Nasarre

Published by arrangement with the original publisher, Microsoft Press, a division of Microsoft Corporation, Redmond, Washington, U.S.A.

本书中文简体版由 Microsoft Press 授权清华大学出版社出版发行,未经出版者书面许可,不得以任何方式复制或抄袭本书的任何部分。

北京市版权局著作权合同登记号 图字:01-2007-5032

版权所有,翻印必究。举报电话:010-62782989 13701121933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

Windows 核心编程(第5版)/(美)杰夫瑞(Jeffrey, J.), (法)克里斯托夫(Christophe, N.)著;葛子昂,周靖,廖敏译. —北京:清华大学出版社,2008.9
(微软技术丛书)

书名原文:Windows Via C/C++

ISBN 978-7-302-18400-3

I. W… II. ①杰… ②克… ③葛… ④周… ⑤廖… 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(2008)第 123109 号

责任编辑:文开琪

装帧设计:杨玉兰

责任印制:孟凡玉

出版发行:清华大学出版社

<http://www.tup.com.cn>

社 总 机:010-62770175

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

地 址:北京清华大学学研大厦 A 座

邮 编:100084

邮 购:010-62786544

印 刷 者:北京密云胶印厂

装 订 者:三河市漂源装订厂

经 销:全国新华书店

开 本:185×260 印 张:49.75 字 数:1210 千字

版 次:2008 年 9 月第 1 版 印 次:2008 年 9 月第 1 次印刷

印 数:1~5000

定 价:99.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:026882-01



Dedication

*To Kristin, words cannot express how I feel about our life together.
I cherish our family and all our adventures.
I'm filled each day with love for you.*

*To Aidan, you have been an inspiration to me and have taught me
to play and have fun. Watching you grow up has been so rewarding
and enjoyable for me. I feel lucky to be able to
partake in your life; it has made me a better person.*

*To My New Baby Boy (shipping Q1 2008),
you have been wanted for so long it's hard to believe
that you're almost here. You bring completeness and balance
to our family. I look forward to playing with you,
learning who you are, and enjoying our time together.*

— Jeffrey Richter

To my wife Florence, au moins cette fois c'est écrit: je t'aime Flo.

*To my parents who cannot believe that learning English
with Dungeons & Dragons rules could have been so efficient.*

— Christophe Nasarre



寄语中国读者 1

Hello to my Chinese readers,

I have visited your country several times now and am always in awe at the beauty and rich culture that you have.

On each visit, I have made some friends and everyone has always been very gracious towards me.

I am very pleased to know that my book has been translated into your native language and I hope that you enjoy reading it as much as I enjoyed producing it.

Many years ago, operating systems were small enough that it was possible to understand the entire system.

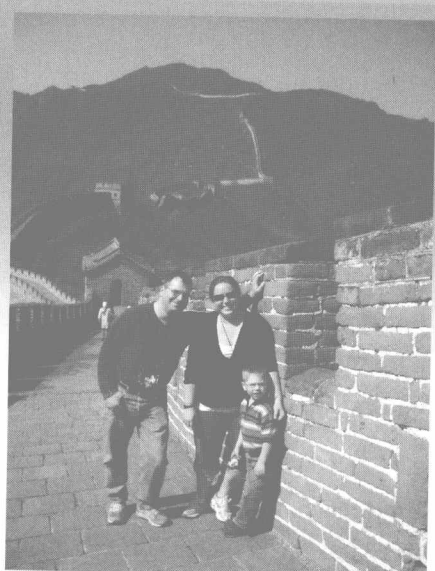
But today, operating systems are so big and are changing so quickly that understanding the whole system is not something that any single person can accomplish.

For those of you pursuing a career in programming, I'd recommend you learn as much about the base systems (such as Windows and the CLR) as possible.

Then, I recommend that you specialize in a particular area such as graphics, web services, web sites, GUI programming with Windows Forms or Windows Presentation Foundation, etc.

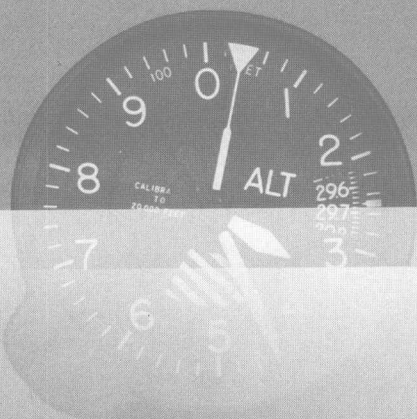
I believe that following this path will be the best way for you to become successful as a software developer.

Best wishes!



Jeffrey Richter

Jeffrey Richter (<http://wintellect.com>)



寄语中国读者 2

Hello, Chinese readers,

In the last couple of years, I've been lucky enough to work closely with one of our development teams in Shanghai. It has been surprising to me that we were all sharing almost the same way of working and thinking. The only one "tiny" obstacle in our perfect team was... the language: all of us had to use a language different from our own native language. Even if we were all fluent in English, it would have been so efficient if we could have all spoken... Chinese. This is the exact purpose of this updated Chinese version of "Windows Via C++": allows you to much more easily understand how to leverage the power of Windows for your own applications and software systems.

I'm French and writing in English is not... well... it would have been much more easier to write this book in French. You can imagine that writing it in Chinese would have been impossible based on the only two words I have memorized ("Hello" and "Thanks" if you wonder). I would like to thank SGuy^① for his very scrupulous work on the Chinese version of the book which allows you to have in your hands a real update with technical changes.

Good luck for your travel into Windows and "Xie xie" for reading this book!

Christophe Nasarre

① 注：葛子昂。



玉不琢，不成器

——《Windows 核心编程(第5版)》出版前言

在刚刚过去的第29届奥运会上，气势恢弘、美轮美奂的开幕式，精彩纷呈、欢呼不断的赛场，让我们度过了多少个激动人心的不眠之夜！开幕式上，以宏伟气势演绎的各种形式的“和”给人留下深刻的印象，同时也凸现了我们这个时代的主旋律。是的，“事成于和睦，力生于团结。”电影《赤壁》中，饰演周瑜的梁朝伟同样也以一个简单的动作强调了“和”的重要作用。

值此《Windows 核心编程(第5版)》出版之际，心里也同样洋溢着一样的情结。作为此书的出版者，我们得到了很多方面的支持，从拿到选题、遴选译者，再到后期的反复审稿，来自社内领导和同仁，来自兄弟出版社，来自微软公司，来自译者，来自读者的支持和热情，大大增强了我们出好书，出精品书的信心和决心，能得到这么多的支持和关注，实乃三生有幸！在此特别向各位表示由衷的感谢！

正所谓“玉不琢，不成器”，一本经典著作，不仅要内容精确，还少不了版式和其他细节的精心雕琢与编排。

- **优秀译者是高品质译著的源泉** 甄选译者的时候，很多读者朋友踊跃报名，他们都能希望能参与此书的工作，为 Windows 编程领域作出贡献。在此，向他们表示由衷的感谢。他们当中，有70后的资深程序员，也有80后的高手程序员，更有朝气蓬勃的在校学生。考虑到质量和时间的因素，最后确定由微软中国研发集团服务器及开发工具事业部的葛子昂主译，周靖和廖敏参与。优秀译者是高品质译著的源泉，这一点在本书上得以充分的验证。他们在翻译过程中，秉承一贯的严谨风格，在字里行间进行认真的推敲，并认真对代码进行了相应的调试和实践。葛子昂作为主译，不仅精心规划进度，还牺牲了休息时间主动承担了大部分工作，责无旁贷地对全书进行统稿，他对本书的认真程度，让人钦佩。兄弟出版社朋友对本书翻译的参与，也使我大受鼓舞。从他们身上，能够深切地体会到他们的敬业精神，他们是真正的爱书人，全心全意为读者着想的人。为读者奉献更丰盛的技术图书盛宴，促进技术图书的健康发展，这是我们共同的心愿。
- **来自微软的支持** 本书由微软中国区微软最有价值专家(Most Valuable Professional, MVP)集体审阅，他们对本书给与高度评价，在此感谢刘彦博、蒋晟和王涛为代表的 Microsoft MVP。他们是一群令人敬佩的人，他们具备一种或多种微软技术专业知知识，积极参与在线或离线的社群活动，经常与其他专业人士分享知识和专业技能。

他们平易近人，是受人尊敬和信任的专家。在全球 90 多个国家，这样的 MVP 有近 4000 名。目前中国区就有 215 名。

- **大人物的推荐序** 本书由 Windows Internals(深入解析 Windows 操作系统)作者提笔所写，他对本书的内容和结构给予了充分的肯定。
- **精美、雅致的版式** 在版式设计方面，我们突破技术类图书的常规，精心设计了页眉和章首页，为读者提供更赏心悦目的阅读感受。
- **Jeffrey 照片和亲笔签名** Microsoft Press 的陈芃帮我们穿针引线，引入 Jeffrey 的“大头照”、家庭照和亲笔签名，这些都会在书中呈现出来。此外，Jeffrey 还计划在 2009 年 4 月访问中国，届时欢迎读者朋友参与他的见面会，与他一起探讨和体验 Windows 编程乐趣。
- **索引** 为方便读者查阅，我们在译文相应部分添加了原书页码，同时保留原书索引，并将其附于书后。
- **读后感分享** 对于一本好书，很多读者都有自己的心得和感想，并希望和同道中人切磋，这一点，我们将成立一个专门的书友会，读者可以在其中畅所欲言，在《Windows 核心编程》书友会这个大家庭中共同成长。详情请关注我社网站首页上的公告。

作为出版工作者，我们要感谢来自译者和读者的支持和鼓励，没有他们，就没有佳作的诞生。关于本书的任何意见和建议，欢迎发送邮件到 wenkq@tup.tsinghua.edu.cn，我们一直在聆听大家的心声。

清华大学出版社



Windows 编程之道

——译序 1

你是否在 Windows 平台上开发应用程序？你是否希望对 Windows 提供的各项服务有更深入的理解？你是否需要使用 Windows 提供的高级功能？你是否在找寻一些 Windows 实用技术来解决项目中的难题？如果对上述问题的回答是肯定的，那么你绝对不应该错过本书。

自这本《Windows 核心编程(第4版)》出版至今，转眼间已经过去了八年。在这八年期间，Windows 世界已经发生了许多变化，Windows XP、Windows 2003、Windows Vista 以及 Windows Server 2008 相继推出。与之对应的，《Windows 核心编程》(第5版)也与时俱进，针对最新的操作系统进行更新，为我们了解并发挥最新的平台特性提供了宝贵的指引。

这是一本 Windows 系统编程的权威专著，许多年之前我就已经读过它的第4版；这是一本生动的教科书，它帮助我更深入地了解 Windows 系统；同时它又是一本参考书，在我开发 Windows 应用程序的过程中遇到问题时，我会到本书中寻找答案。希望它同样能够为你答疑解惑，并解决你的实际问题。

本书由葛子昂、周靖、廖敏共同翻译：第8~22章由葛子昂翻译，第1~6章由周靖翻译，第23~26章及附录由廖敏翻译，第7章由刘江友情客串翻译，全书由葛子昂最终审阅和统稿。鉴于时间和精力有限，一些翻译难免存在不足甚至错误之处，为此我建立了一份网上勘误表。如果读者发现任何错误，都可以通过该网页与我联系，一旦确认，我会立即将其更新到勘误表中。勘误表的网址为 www.gesoftfactory.com/ge/WindowsViaCpp。

最后，感谢 Jeffrey 和 Christophe 在本书翻译过程中不厌其烦地解答我的问题，核实并纠正英文版中发现的一些错误。感谢李劲松、王渊峰、张羿、孙展波、谭映辉以及孙祺对中文版的全部或部分章节进行了审阅并提出宝贵的意见。感谢潘爱民在百忙中抽空阅读部分译稿，并给以反馈。感谢妻儿和家人对我的理解和支持，本书的翻译工作占用了本应属于你们的时间，现在终于可以有更多时间来陪伴你们了。



附：作译者近照(左为葛子昂，右为 Jeffrey)



继往开来创新高，推陈出新品佳酿

——译序 2

1993 年，本书第 1 版 *Advanced NT* 出版的时候，我和三个朋友一起成立了一个“四喜工作室”。由于四个人只有一台计算机，所以我们几个每天一睁眼，第一件事情便是抢占计算机。这台 286 配置简单，根本无法与现在的计算机相提并论，而且当时也没有网络，所以计算机的用途非常有限，主要也就是文字处理，玩游戏，编简单程序等，但它带给我们的乐趣至今难以忘怀。受限于当时的环境，数据和游戏的交换也基本上在圈内好友之间进行，就像搞地下活动一样约好时间地点碰头。幸运的是，由此结交了一大批计算机爱好者，后来他们大多成为 IT 届的领军人物。

其时，从大环境看，我国网络也开始悄然起步。1993 年年初，中国科学院高能物理研究所接入斯坦福大学线性加速器中心的 64K 专线开通，国内科学家开始在国内使用电子邮件。随后几个月的时间，金桥工程和域名体系的确立和部署，三大院校网的连接，最终将我国带入信息高速公路，推动我国 IT 业的迅猛发展。

由此而来的便是计算机类图书和报纸期刊的炙手可热，《电脑报》等 IT 媒体相继崛起，计算机图书更是出现供不应求的现象。在当时，即便是从国外引进版权再翻译出版的图书，也能轻松突破几万册的销量，计算机图书的发展达到全盛时期。

在这个时期，国内开发人员先后成为 Jeffrey 和 McConell 等大师的拥趸。因为在 IT 界，虽然资深程序员不胜枚举，但同时又是深受程序员喜爱的技术图书作家的乏善可陈。而像他们那样，曾经写过多部书，部部都引人入胜，令人醍醐灌顶，就更是凤毛麟角。他们是 Windows 编程世界中的中流砥柱，也是 Windows 技术当之无愧的布道者。曾有不少读者放言，只要是 Jeffrey 的书，他们必定会花时间研读，并加以收藏。这一点都不夸张，我们同时代的很多人都是在这批书的滋润下成长起来的。他们熟读了 *Advanced NT* 之后，又如痴如狂地捧起了 *Advanced Windows* 和 *Programming Application for Microsoft Windows* 等续作。他们是 Jeffrey 的粉丝，同时也是微软开发阵营的主力军。

随着微软宣布放弃对 Windows XP 以及以前版本的支持，Windows Vista 的普及势在必行，迟早会安装到普通用户的计算机上。Windows Vista 有很多吸引人的新特性，相信大家不用不知道，一用忘不了。（在翻译 Microsoft Press 的 *Windows Vista Inside Out* 一书的过程中，我已经深切体会到她的妙处。）作为一名程序员，有必要在第一时间适应并习惯于为新的操作系统编写程序。历经 15 年，《Windows 核心编程》也随着 Windows 操作系统的“改朝



换代”，升级到第 5 版，即 *Windows via C/C++*。如果您要用 C/C++ 开发 Windows 应用程序，那就不要走弯路，直接让 Jeffrey 告诉您如何利用 Windows 的新特性和新函数来编写出高效、优美的 Windows 应用程序。

对于本书的学习，谨以《史记·孔子世家》中孔子学琴一文与大家共勉(请原谅，这里引用了我另一本书的译序，因其寓意深刻，故忍不住又拿出来与大家分享☺)：

孔子学鼓琴师襄子，十日不进。师襄子曰：“可以益矣。”孔子曰：“丘已习其曲矣，未得其数也。”有间，曰：“已习其数，可以益矣。”孔子曰：“丘未得其志也。”有间，曰：“已习其志，可以益矣。”孔子曰：“丘未得其为人也。”有间，有所穆然深思焉，有所怡然高望而远志焉。曰：“丘得其为人，黯然而黑，几然而长，眼如望羊，如王四国，非文王其谁能为此也！”师襄子辟席再拜，曰：“师盖云文王操也。”

期望读者朋友也能达到学习的三大境界：学习掌握演奏(编程)的技巧；领会其中的志趣；熟悉乐曲(程序)的作者。

翻译过程中，感谢我的家人和朋友的诸多帮助和理解，他们是文天山、曹伟、文娟、马文喜、张旭等。尤其要感谢我的乖女儿。这个暑假，她的成长令人激赏！

同时，还要感谢其他几名译者以及微软公司的技术专家们，感谢他们为本书的质量进行了严格的把关。最后，欢迎读者指出本书的疏漏和不足之处，如果对我翻译的部分(第 1~6 章)有什么意见和建议，请访问我的博客(transbot.blog.163.com)，在此留言指出。我在这里为自己翻译的图书开辟了专栏，专门用于和读者们分享勘误和其他有用的信息。

周 靖



推荐序

当 Jeffrey 请我为本书写一篇序的时候，我爽快地答应了。我还记得在上个世纪 90 年代早期寻找本书第 1 版《高级 Windows NT 编程》(Advanced Windows NT)时的情形。那时我刚开始开发 Windows 应用程序。在我开发诸如 Filemon, Regmon 以及 HandleEx(现在叫 Process Explorer)之类的应用程序时，它既是一本指南，又是一本参考手册。本书第 1 版面向 Windows 系统编程，而 Charles Petzold 的《Windows 编程》(Programming Windows)则面向用户界面编程。

在过去的几年中，Jeffrey 已经将他关注的重点转向了托管代码，这使得为本书作序对我来说大有意义。Jeffrey 已经成为 .NET CLR 和 C# 系统编程的权威，当 Microsoft 还在设计 .NET 的时候，他的这种转换就已经开始了。他还不时地“嘲弄”我和我的搭档 David Solomon——我们合著了《深入解析 Windows 操作系统》(Windows Internals)系列，说本机代码编程(native code programming)已经日暮西山。我和 David 始终坚信本机代码编程由于具有不同的 API、不同的失败语义(failure semantics)、不同的部署方式以及不同的运行时特征，因此将继续满足某一类应用程序的需求，或者至少是满足部分应用程序的需求。相当一部分 Windows 客户端应用程序仍然是用本机代码编写的，甚至 Microsoft 的旗舰服务器端应用程序，包括 IIS、SQL Server 以及 Exchange，几乎完全是用本机代码编写的，或者使用了用本机代码编写的组件。因此，我将本书视作 Jeffrey 对我们这种信念的无言的肯定。

Windows 一直在演化，本书亦是如此。它之前已经经历了许多个版本，甚至书名也一变再变。比如，本书上一版名为 Programming Applications for Microsoft Windows。但无论书名如何改变，它的每一版在我的书架上都占有一席之地。(这与 Jeffrey 给我在书上签名不无关系，不仅如此，而且所签的内容还是我选的，比如“我从 Mark 那里学到了一切”之类可供炫耀的话。)这一个版本不仅只是换了一个新的书名，它还介绍了 Windows Vista 和 Windows Server 2008 中引入的最新 API，包括线程、安全、异常处理以及 I/O。

本书提供了大量的编程实例，既紧凑又有效地展示了各种关键概念，这使得它尤其有用。许多作者在将代码与读者分享的时候，并没有认识到许多开发人员会直接将代码复制和粘贴到将要发行的应用程序中去。但与我们在互联网上搜索到的代码不同，Jeffrey 和 Christophe 已经确保书中的代码在编写时兼顾了高质量和高准确性。

我相信，无论您是 Windows 应用程序编程的新手，还是完全用本机代码来编写应用程序或



通过 P/Invoke 来调用 .NET Framework 未提供的 API 的老手，都会发现本书的价值。它将有助于发挥强大的平台特性并降低其复杂度，帮助开发人员开发伸缩性更好、更可靠、更安全以及性能更高的应用程序。显然，它依旧会出现在我的书架上，希望这一次 Jeffrey 和 Christophe 都能为我签上“我从 Mark 那里学到了一切”。

Mark Russinovich
(*Windows Internals* 作者)
Technical Fellow
Microsoft Corporation



致 谢

没有其他人的帮助和技术上的支持，我们不可能写就本书。具体说来，我们要向以下人士表达我们由衷的感激之情。

Jeffrey 的家人

Jeffrey 要向他的妻儿(Kristin 和 Aidan)表示感谢，感谢他们绵延不尽的爱和支持。

Christophe 的家人

没有妻子 Florence 的爱和支持，没有女儿 Celia 永不满足的好奇心，没有爱猫 Canelle 和 Nougat 时不时发出的咕噜声，Christophe 可能无法写就本书第 5 版。现在，我没有任何好的借口可以逃避对你们的照顾！

技术支持

对于写就这样一本经典的图书，单凭个人的研究是不够的。我们要向 Microsoft 曾经帮助过我们的员工表示诚挚的谢意。具体说来，我们要感谢 Arun Kishan，对于我们提出的怪问题或者复杂问题，他总能迅速找到答案或者立即找到 Windows 开发团队中合适的人提供更详细的解释。我们还要感谢 Kinshuman Kinshumann, Stephan Doll, Wedson Almeida Filho, Eric Li, Jean-Yves Pouban, Sandeep Ranade, Alan Chan, Ale Contenti, Kang Su Gatlin, Kai Hsu, Mehmet Iyigun, Ken Jung, Pavel Lebedynskiy, Paul Sliwowiecz 和 Landy Wang。此外，还有负责解答 Microsoft 内部社区所贴问题和分享其渊博知识的其他人士，如 Raymond Chen, Sunggook Chue, Chris Corio, Larry Osterman, Richard Russell, Mark Russinovich, Mike Sheldon, Damien Watkins 和 Junfeng Zhang。最后要特别感谢 John “Bugslayer” Robbins 和 Kenny Kerr，他们对本书各章提供了出色的反馈意见。

Microsoft Press 编辑团队

我们要感谢本书组稿编辑 Ben Ryan 对 Christophe 这样一个法国小伙儿的信任，感谢经理 Lynn Finnel 和 Curtis Philips 的耐心。感谢 Scott Seely 为保证技术准确性而进行的搜索查找，感谢 Roger LeBlanc 在将 Christophe 的法国式英语变得更容易理解这方面所表现出来的天才，感谢 Andrea Fox 字斟句酌的校对。除了 Redmond 团队成员外，我们还要感谢 Joyanta Sen 花费大量宝贵的私人时间为我们所提供的帮助。

互致谢意

Christophe 要向 Jeffrey Richter 表示诚挚的谢意，感谢他相信自己不会弄砸第 5 版。

Jeffrey 也要感谢 Christophe 兢兢业业，恪尽职守，在研究、重新组织、重写和重修本书内容以期接近 Jeff 完美要求所做的努力。



前言

Microsoft Windows 是一个复杂的操作系统。它提供的特性非常丰富，对任何个人而言，要想完全透彻地理解整个系统几乎不可能。其复杂性也使我们很难确定应该把主要精力集中在哪些地方。万丈高楼平地起，就我个人而言，我始终倾向于从最底层开始，深刻理解系统的基本构件。一旦掌握了最基本的东西，就很容易向自己的知识库中逐步添加更多高层的东西。所以本书将集中介绍 Windows 的基本构件和基础概念，对于构架和实现 Windows 应用程序来说，这些都是必须掌握的。简而言之，本书是向读者介绍各个 Windows 特性，并介绍如何通过 C 和 C++ 语言来了解并使用这些特性。

虽然本书不会涵盖某些 Windows 概念，比如组件对象模型(Component Object Model，后面简称为 COM)，但 COM 是在进程、线程、内存管理、DLL、线程本地存储区以及 Unicode 等基本构件的基础上构建的。如果了解了这些基本构件，那么要理解 COM 就相当于理解如何使用这些基本构件。对于那些试图跳过这些基本构件，悉心钻研 COM 体系结构的朋友，我深表同情。要完善知识库，他们还有很长一段路要走，这必然会对他们的代码和软件开发进度产生很大的负面影响。

Microsoft .NET Framework 的公共语言运行时(Common Language Runtime, CLR)是本书未涉及的另一种技术(感兴趣的朋友可以阅读我的另一本专著《框架设计：CLR via C#》)。不过，CLR 是作为动态链接库(dynamic link library, DLL)中的一个 COM 对象来实现的，它需要被载入到进程中，它需要使用线程来执行代码，它需要处理 Unicode 字符串，它还需要在内存中对这些字符串进行管理。因此，本书所介绍的基本构件仍然有助于软件开发人员编写托管代码。此外，借助于 CLR 的平台调用(Platform Invocation, P/Invoke)技术，我们可以调用本书所介绍的各种 Windows API。

如此一来，本书的宗旨是介绍每一个 Windows 开发人员都应该掌握的基本的 Windows 构件(至少在我看来如此)。在介绍每一个构件的时候，我还会描述系统是如何使用这些构件的，我们自己的应用程序如何充分利用这些构件。在很多章中，我会介绍如何创建自己的构件。我们自己的构件一般都实现为普通函数或 C++ 类，它们把一些 Windows 构件组合在一起，从而创造出一个整体，其结果远远大于各部分之和。

64 位 Windows 操作系统

Microsoft 发售支持 x86 CPU 体系结构的 32 位 Windows 操作系统已经有多年的历史。如今，Microsoft 还提供支持 x64 和 IA-64 CPU 体系结构的 64 位 Windows 操作系统。基于这些 64

位 CPU 体系结构的计算机迅速为市场所接受。事实上,在不远的将来,所有台式机和服务器都将使用 64 位 CPU。为此,Microsoft 已发表声明,Windows Server 2008 将是最后一款 32 位版本的 Windows 操作系统!对于开发人员而言,现在是时候集中精力让自己的应用程序能在 64 位 Windows 操作系统上正常运行了。贯穿本书,将全面覆盖相关的知识,让开发人员的应用程序能够在 64 位 Windows 操作系统(同时也包括 32 位 Windows)上正常运行。

应用程序从 64 位地址空间所获得的最大的好处是,能够轻松处理大量数据,因为应用程序的进程不再受限于 2GB 可用地址空间。即使应用程序并不需要所有这些空间,Windows 本身也可以利用这个显然大得多的地址空间(约 8TB)来加快运行速度。

对于 64 位 Windows 操作系统,我们需要了解哪些呢?下面简单看一看。

- 64 位 Windows 内核是 32 位 Windows 内核的移植版本。这意味着以前所学的 32 位 Windows 所有细节和难点均适用于 64 位 Windows。事实上,Microsoft 已经修改了 32 位 Windows 的源代码,使其既可以编译生成 32 位系统,也可以编译生成 64 位系统。两个系统使用同一个源代码库,所以新的特性和缺陷的修复会同时应用于这两个系统。
- 因为内核使用相同的代码和基本概念,所以在两个平台上的 Windows API 是一样的。这意味着我们不必重新设计或实现应用程序,就能让它在 64 位 Windows 操作系统上运行。我们只需要对源代码做少许改动,然后重新构建应用程序即可。
- 为了保持向后兼容性,64 位 Windows 操作系统是可以执行 32 位应用程序的。但是,如果应用程序是作为 64 位应用程序来生成的,那么它的性能会有显著的提高。
- 因为移植 32 位代码非常容易,所以 64 位 Windows 操作系统中已经有很多的设备驱动程序、工具和应用程序了。遗憾的是,Visual Studio 是 32 位的,而且 Microsoft 似乎还并不急于将其移植为 64 位的。不过,好消息是 32 位的 Visual Studio 在 64 位 Windows 操作系统上的确运行得非常好,只不过其数据结构的地址空间有限。而且,我们还可以用 Visual Studio 来调试 64 位应用程序。
- 并不需要学习太多新知识。大多数数据类型仍然保持 32 位宽度,这是很多读者很高兴看到的。它们是 INT, DWORD, LONG 和 BOOL 等。事实上,大多数情况下我们只需要担心指针和句柄,因为它们现在变成 64 位了。

对于如何将现有源代码修改为 64 位,Microsoft 已经提供了相当丰富的信息,所以我们不打算在本书中深入这些细节。不过,在我写每一章的时候,我都会考虑到 64 位 Windows 操作系统。在适当的时候,我会加入 64 位 Windows 的特定信息。同时,我还在 64 位 Windows 中编译和测试了本书中的所有示例程序。所以,如果读者效仿书中的示例程序和我的做法,那么创建一个能够为 32 位或 64 位 Windows 编译的源代码库应该完全不成问题。

《Windows 核心编程(第 5 版)》中有哪些新内容

在过去,本书曾经被冠以“Advanced NT”,“Advanced Windows”和“Programming Application for Microsoft Windows”等名字。为保持这个传统,本书的第 5 版也有一个新的书名,即 Windows via C/C++。新的书名表示本书是为打算理解 Windows 的 C 和 C++程序员设计的。