求是科技

核心编程系列

Windows 系统编程

求是科技　周金萍 徐丙立 姜小光 马小亮 编著

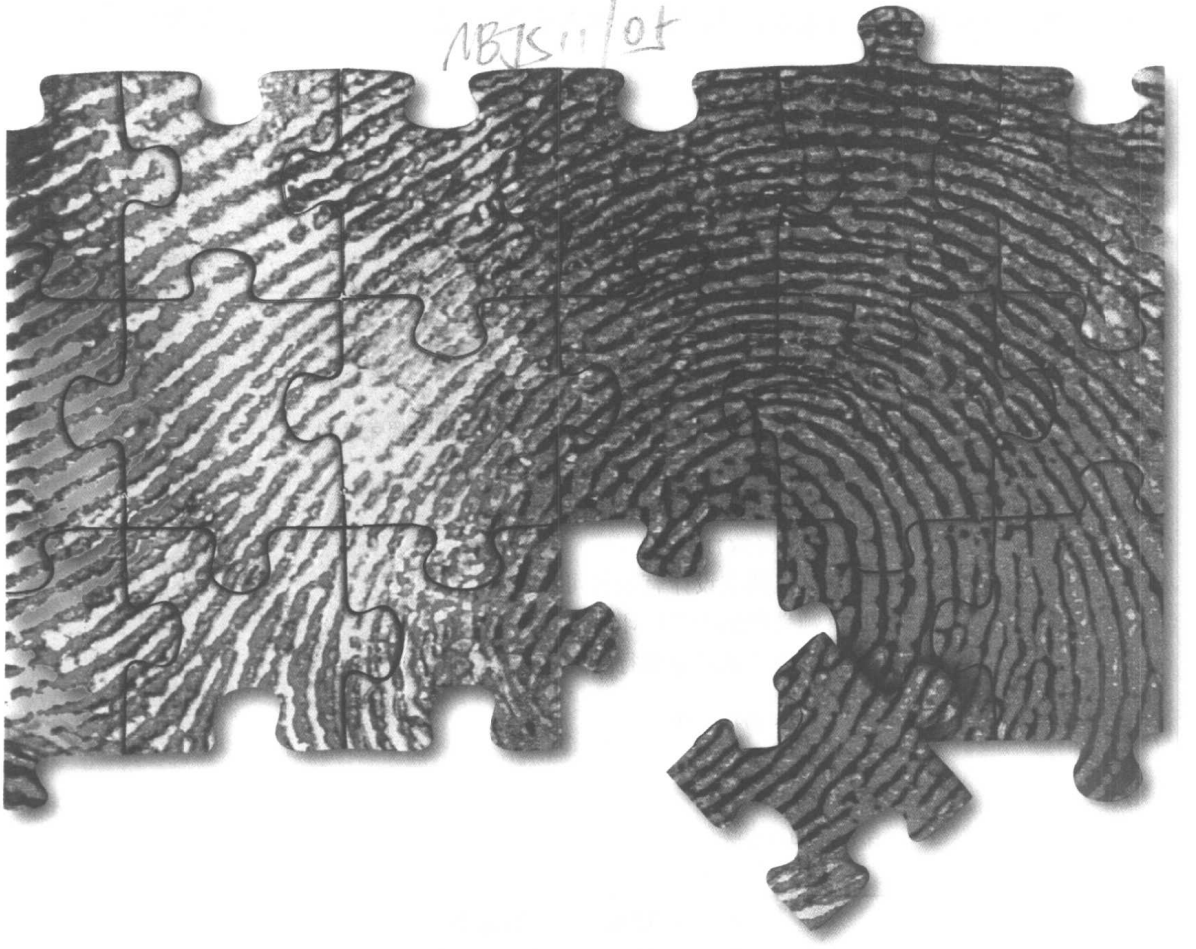
源代码光盘
CD-ROM

人民邮电出版社
POSTS & TELECOMMUNICATIONS PRESS

核心编程系列

Windows 系统编程

求是科技　　周金萍 徐丙立 姜小光 马小亮 编著



人民邮电出版社

图书在版编目(CIP)数据

Windows 系统编程/求是科技编著—北京：人民邮电出版社，2002.7
ISBN 7-115-10322-4

I. W... II.求...III.窗口软件，Windows-程序设计—教材 IV. TP316. 7

中国版本图书馆 CIP 数据核字(2002)第 036546 号

内容提要

本书讲解了 Windows 系统编程方面的内容。按照“由浅入深”、“相互贯穿”、“重点突出”、“文字叙述与典型代码实例相结合”的原则，本书首先介绍了不同版本的 Windows 系统的内核和编程环境，接着介绍了 Windows 程序设计的特点和程序员所应掌握的基础知识，然后详细讲述了 Windows 的基本构件（如进程、线程、系统信息、内存管理和动态链接库等）以及它们在 Windows 系统中和在实际的应用程序中是如何使用的，此外书中的各章配以丰富而恰当的实例，帮助读者深入理解系统编程的内容。

对于 Windows 编程人员来说，本书极具参考价值，是一本不可多得的参考书。

核心编程系列

Windows 系统编程

◆ 编　　著　求是科技 周金萍 徐丙立 姜小光 马小亮
责任编辑　张立科

◆ 人民邮电出版社出版发行　北京市崇文区夕照寺街 14 号
邮编　100061　电子函件　315@ptpress.com.cn
网址　http://www. ptpress.com.cn
读者热线　010-67180876
北京汉魂图文设计有限公司制作
北京顺义向阳胶印厂印刷
新华书店总店北京发行所经销

◆ 开本：787×1092　1/16
印张：44
字数：1080 千字　　2002 年 7 月第 1 版
印数：1-4 000 册　　2002 年 7 月北京第 1 次印刷

ISBN 7-115-10322-4/TP • 2887

定价：72.00 元　（附光盘）

前　言

1. 关于 Windows 系统

Windows 系统是由 Microsoft 公司出品的、性能复杂的一种操作系统。由于它极其丰富的特性和功能，其版本也不断更新，使得很多用户都无法全面了解和掌握整个系统，甚至不知道学习 Windows 系统究竟该从何处入手。

目前，Microsoft 公司仍在推广几种不同内核的操作系统，如 Windows 98、Windows 2000、Windows XP 以及为了满足小型硬件设备的需要而推出的小型操作系统 Windows CE 等。每种内核经过优化后，适用于某种特定的计算环境中。Microsoft 公司声称为每种平台提供了相同的 API（Application Programming Interface，应用程序编程接口），以便吸引软件开发人员使用 Windows。这意味着，当人们为一个内核编写 Windows 应用程序时，也同样知道如何为其他内核编写 Windows 应用程序。

2. 本书的结构和内容

本书以 Windows 2000 系统为主线（必要时也针对不同的系统讲述它们具体的差异），系统地讲解 Windows 系统编程方面的知识。本书以“由浅入深”、“相互贯穿”、“重点突出”、“文字叙述与典型代码实例相结合”为原则，向每一位 Windows 编程爱好者全面介绍 Windows 的基本构成、内部运行机制以及在应用程序编写中的具体运用。

全书总共分为 16 章，首先介绍了不同版本的 Windows 系统内核和编程环境，然后说明了 Windows 程序设计的特点和程序员所应掌握的基础知识，接着讲述了 Windows 的基本构件（如进程、线程、系统信息、内存管理和动态链结库等）以及它们在 Windows 系统中和在实际的应用程序的编写中是如何使用的。

本书各章的内容具体安排如下：

“第 1 章　Windows 编程开发环境”首先以最简单通俗的方式，循序渐进地介绍了 Windows 操作系统目前的几个平台和内核以及它们之间的区别。接着，介绍了 Windows 编程的程序开发环境 Developer Studio 以及如何自定义一些编程环境和有效地获取编程帮助。

“第 2 章　Win32 程序设计”针对刚接触 Windows 编程的用户，介绍消息驱动机制、32 位编程、进程、线程和内存管理等基本知识，以便为用户学习后续内容打下基础。

“第 3 章　Windows 用户界面”实际上是对第 2 章的进一步补充，它讲述了与 Windows 用户界面有关的各要素，如控件、资源、用户输入和窗口等的编程内容。

“第 4 章　Windows 程序员基础”讲述了合格的 Windows 程序员应该掌握的一些要点。这些要点包括：如何对 Windows 错误进行处理，如何使开发的程序适应 Unicode 字符扩展的需要，如何准确理解内核对象以便对 Windows 的各种资源进行有效的管理。

“第 5 章　进程”是对进程的全面讲解，它首先讲述进程的句柄、环境变量、命令行、亲缘性以及错误模式，而后讲述如何创建一个新的进程，在新的进程中如何设置各个参数以

及如何结束进程。

“第 6 章　进程的作业”主要讲述作业进程的限制、如何将进程放入作业、终止作业中所有进程的运行、查询作业统计信息以及作业的通知信息。

“第 7 章　线程基础”介绍线程的基础知识。线程是一个非常重要的概念，因为每个进程至少需要一个线程。本章讲述了进程与线程之间的差别，系统如何使用线程内核对象来管理线程，如何查询和修改线程内核对象属性，以及如何在进程中生成多线程。

“第 8 章　线程的调度、优先级和亲缘性”详细阐述了有关线程的三个重要概念：线程的调度、优先级和亲缘性。这些问题是线程的基本属性的问题。

“第9章　线程的同步”介绍了线程的同步方面的内容，很少有线程能够在所有的时间都独自运行；通常情况是，系统中有好多线程都拥有对各种系统资源的访问权；要想让这些线程能够正常工作而又不破坏共享的资源，就必须协调它们的工作。

“第 10 章　线程的堆栈与纤程”讲述了两个方面的内容：一个是线程的堆栈，每个线程有它自己的堆栈；另一个是纤程，它是一个必须由应用程序手动调度的执行单元，它运行于线程的上下文中。

“第 11 章　系统信息与注册表”内容一方面是系统信息，它是描述关于计算机硬件、计算机名和用户名等方面的信息；另一方面是注册表，它实际上是一个系统定义的数据库，其目的是为应用程序和系统的组成部分能够用于存储和检索相关的配置信息。

“第12章　虚拟内存”讲述了Windows提供的3种内存管理机制之中的一种，即虚拟内存，它最适合用来管理大型对象或结构数组。

“第 13 章　文件映射”讲述了第二种内存管理机制，即文件映射。文件映射就是使文件内容与进程的虚拟地址空间中的部分区域关联起来，它最适合用来管理大型数据流（通常来自文件）以及在单个计算机上运行的多个进程之间共享数据。

“第 14 章　内存堆栈”讲述了第三种内存管理的机制，即内存堆栈。内存堆栈最适合用来管理大量小的数据对象。

“第 15 章　动态链接库”除了对 DLL 的基础知识作了详细的阐述，还讲述了有关 DLL 的高级应用，如 DLL 的延迟加载、DLL 的转移、DLL 的移位和绑定、线成本地存储器（TLS）等。

“第 16 章　结构化异常处理”针对 Windows 编程的异常处理进行了介绍，主要包括两种异常（硬件异常和软件异常）和两种处理（结束处理和异常处理）。在 Windows 中和自己设计的应用程序中引入了结构化异常处理机制保证了程序本身的稳健性和强壮性。

以上各章中的每一章（第 1 章和第 6 章除外）都配以丰富而恰当的实例。实例的完整代码存于本书所附带的光盘里，以供感兴趣的读者仔细地学习和研究。每个代码例子所在的目录由该例子在书中的目次和例子名称所构成，如目录“2-5 spy”表示该例子位于本书第 2 章的 2.5 节，例子的名称为 spy。

本书为立志掌握 Windows 编程技术的人提供了一条有效的捷径。对于不同层次的 Windows 编程人员来说，本书都极具参考价值，是一本不可多得的参考书。

由于 Windows 系统涉及到的知识面极为广泛，编者的知识有限，尽管我们对本书中所涉及的内容一再推敲和仔细调试，仍有可能出现错误和纰漏，希望广大读者批评指正。

编者

目　录

第 1 章　Windows 编程开发环境

1.1　Windows 操作系统及其内核

1.1.1　已有的 Windows 平台

目前，Microsoft 公司有几种不同的操作系统内核。每种内核经过优化后，适用于某种特定的运行环境。Microsoft 公司声称为每种平台提供了相同的 API（Application Programming Interface，应用程序编程接口），以便吸引软件开发人员使用 Windows 操作系统。这意味着，当人们学习为一个内核编写 Windows 应用程序时，他也同样知道如何为其他内核编写 Windows 应用程序。

然而，由于各操作系统的不同功能是用不同的方法来实现的，因此各个内核是有些差异的。下面就分别对 Windows 2000、Windows 98 和 Windows CE 三种操作平台的内核作一简要介绍。

1. Windows 2000 的内核

Windows 2000 是 Microsoft 推出的高端操作系统，其主要特性如下：

- 可以作为工作站、服务器核数据中心来运行。
- 能避免运行一些编写得不好的应用程序后所导致的系统崩溃。
- 能防止未经授权的用户访问系统管理的资源（如文件和打印机）。
- 非常丰富的工具和实用程序，可用于管理人员对操作系统的管理。
- 它的内核基本上是用 C 和 C++编写的，可以很容易移植到其他 CPU 结构中去。
- 系统本身支持 Unicode，因此，很容易用其他国家的语言进行系统的本地化。
- 内存管理特性，提供了极其丰富的功能，效率很高。
- 运用了 SEH（结构化异常处理）特性，可以方便地进行错误的修复。
- DLL（动态链接库）使得系统的扩展非常方便。
- 多线程，支持多处理器，因此系统具备很好的伸缩性，便于性能的改进。
- 文件系统的出色特性，可以用来很好地跟踪用户是如何处理他们计算机上的数据的。

2. Windows 98 的内核

Windows 98 具有 Windows 2000 的许多特性，但是却不具备它的若干关键特性。例如：

- 不够健壮，一个应用程序就能够导致系统崩溃。
- 不够安全，没有用户登录密码设置等。

- 伸缩性受限，是单处理器内核，不支持多处理器。
- 系统对 Unicode 不全面支持。

如果要使 Windows 98 的内核也支持 Windows 2000 的上述特性，则必须把 Windows 98 的内核换成 Windows 2000 的内核。因此，Windows 2000 的内核会很快取代 Windows 98 的内核。

Windows 98 的内核在短期内之所以还能够存在下去，原因是它比 Windows 2000 具有较强的用户友好特性（比如说很多用户不喜欢登录计算机，不喜欢管理计算机等）。此外，消费者往往比企业员工更喜欢玩游戏。不过，Microsoft 公司的积极努力也将使 Windows 2000 的内核很快推向消费市场。

本书对操作系统的叙述所针对的是 Windows 2000 的内核，如果需要标明 Windows 98 的不同之处，则会特地指出是针对 Windows 98 的。

3. Windows CE 的内核

Windows CE 是 Microsoft 公司为了满足小型硬件设备的需要而推出的新型操作系统。这些设备包括手提式计算机、汽车用 PC、智能终端、电烤箱、微波炉和自动售货机等。这些设备的用电量通常比较小、内存量不大，并且拥有很小的经久性存储器（如磁盘驱动器）。由于这些硬件的限制，Microsoft 公司不得不开发一种新型操作系统内核，它的规模既小于 Windows 2000，也小于 Windows 98。

不过，Windows CE 功能非常强大，提供的特性非常多。由于 Windows CE 的机器主要是供个人使用的，因此，它的内核不需要对管理和伸缩性等许多特性的支持。虽然本书并没有专门介绍 Windows CE，但是书中介绍的许多概念都适用于该平台。

1.1.2 新一代 Windows 平台——Windows XP

跟 Windows 2000 一样，Windows XP 基于 NT 技术，是纯 32 位操作系统，而不像 Windows 9x 那样是 16/32 位操作系统。相比之下，Windows XP 更健壮、更稳定。系统支持 Unicode、DLL、SHE、多线程和多处理器等。

Microsoft 公司的策略是想让 Windows XP 来取代所有 Windows 的前期版本。他们使 Windows XP 具有了如下一些新的特性：

- 外观上的新特征：包括更为赏心悦目的界面、高品质的图标、任务条、菜单等。Windows XP 视觉风格变化很大，但仍然可以保持与以前版本的 Windows 界面风格兼容。由于界面的改变，原有在传统风格中得心应手的主题界面开发工具，将不再适用于 Windows XP。
- 更丰富的人机交互功能：实时的声音、视频，以及不用重新启动计算机就可以在各个用户之间进行快速切换、并行部件共享等。
- 更强的移动性和更完善的电源管理机制：移动用户随时随地访问信息的能力进一步增强。
- 更好的帮助与支持：用户遇到困难可以方便地链接到其他用户或帮助资源，及时获得帮助与支持。
- 简洁的数码影像：利用 Windows XP，创建、管理、共享数码影像将变得非常容易。

- 令人激动的音乐和娱乐：Windows XP 将为寻找、下载、个性化和回放高质量的音频、视频带来最佳体验。
- 建设“互联家庭”：Windows XP 为人们提供了在家中共享信息、设备和 Internet 链接的简易途径。

在服务器版本领域里，Windows XP 至少具备三个版本，分别取代目前 Windows 2000 的服务器版本、高级服务器版本和资料中心服务器版本。

1.1.3　未来的 Windows 平台——64 位 Windows

目前，Microsoft 公司正在加紧开发新的 Windows 系统，使其成为真正 64 位 Windows。它的特点主要有：

- 64 位 Windows 内核是现有的 32 位 Windows 内核的一个端口。这意味着关于 32 位 Windows 系统的所有详细信息和复杂结构仍然适用于 64 位的运行环境。实际上，Microsoft 公司已经修改了 32 位 Windows 的源代码，这样就可以在编译后产生 32 位或 64 位的系统。由于它们只有一个基础源代码，因此，新的特性和错误调试程序可以同时适用于两种系统。
- 由于这两种内核使用相同的代码和基本概念，因此 64 和 32 位两种平台上的 Windows API 是相同的。这意味着不必重新设计或实现应用程序以使之能在 64 位 Windows 上运行。只需要对原来的代码稍加修改，然后重新创建应用程序即可。
- 由于转成 32 位应用程序非常容易，因此很快将会有支持 64 位应用程序开发的工具（如 Microsoft Developer Studio）。
- 64 位的应用程序的运行性能将得到大幅度的提高。为了实现向后兼容，64 位 Windows 能够执行 32 位的应用程序。
- 64 位的 Windows 内核新东西并不多。大多数数据类型（包括 int、DWORD、LONG 和 BOOL 等）的长度仍然是 32 位。主要的变化在那些现在变成了 64 位值的指针和某些句柄。

每一个应用程序和系统都有一个抽象数据模型，这个模型指导着应用程序代码的编写。在 Win32 编程模型（也叫 ILP32 模型）中，整数、长整数、指针数据类型都是 32 位长度。在 64 位系统中采用的抽象数据模型叫 LLP64 模型或叫 P64 模型，在该模型中只有指针类型被拓展到 64 位，其他所有类型（如整数和长整数等）仍然是 32 位长度。

要想使 32 位代码能够在 32 位和 64 位系统上运行，必须注意以下两点：

- 确保数据模型变更的影响只涉及到指针数据类型。
- 定义一些新的数据类型，用于调整指针有关的数据的长度。

在 Win64 下将有 3 类新的数据类型，即固定精度数据类型、指针精度数据类型和指定精度指针类型。这些类型在 Windows 系统的头文件 Basetsd.h 中有定义，它们均是基于 C 语言的整数和长整数类型。下面是 Basetsd.h 文件中的一个代码片断：

```
...
#ifdef _WIN64

typedef __int64 INT_PTR, *PINT_PTR;
```

```
typedef unsigned __int64 UINT_PTR, *PUINT_PTR;

#define MAXINT_PTR (0x7fffffffffffffffI64)
#define MININT_PTR (0x8000000000000000I64)
#define MAXUINT_PTR (0xffffffffffffffffUI64)

typedef unsigned int UHALF_PTR, *PUHALF_PTR;
typedef int HALF_PTR, *PHALF_PTR;

#define MAXUHALF_PTR (0xffffffffUL)
#define MAXHALF_PTR (0x7fffffffL)
#define MINHALF_PTR (0x80000000L)
...
#else

typedef long INT_PTR, *PINT_PTR;
typedef unsigned long UINT_PTR, *PUINT_PTR;

#define MAXINT_PTR (0x7fffffffL)
#define MININT_PTR (0x80000000L)
#define MAXUINT_PTR (0xffffffffUL)

typedef unsigned short UHALF_PTR, *PUHALF_PTR;
typedef short HALF_PTR, *PHALF_PTR;

#define MAXUHALF_PTR 0xffff
#define MAXHALF_PTR 0x7fff
#define MINHALF_PTR 0x8000
...
#endif

//
// SIZE_T 用于计数或范围，因此需要定义成指针类型，以便扩大范围。而 SSIZE_T 为
// 有符号的指针型
//
typedef UINT_PTR SIZE_T, *PSIZE_T;
typedef INT_PTR SSIZE_T, *PSSIZE_T;

//
// 以下类型保证为有符号的和 64 位宽的
```

```
//
typedef __int64 LONG64, *PLONG64;
typedef __int64 INT64, *PINT64;

//
// 以下类型保证为无符号的和 64 位宽的
//

typedef unsigned __int64 ULONG64, *PULONG64;
typedef unsigned __int64 DWORD64, *PDWORD64;
typedef unsigned __int64 UINT64, *PUINT64;
...
```

以上只给出了 Basetsd.h 头文件中的几处关键代码，有关的省略部分请参见该头文件。

1.2　集成性开发环境 Developer Studio

1.2.1　Microsoft Visual C++和 Developer Studio

只要提到在 Windows 操作系统下进行 32 位的程序开发，就不能不提到 Visual C++。和其他编程工具相比，Visual C++在提供可视化编程方法的同时，也适用于编写直接对系统进行底层操作的程序。其生成代码的质量，也要优于其他许多开发工具。

Visual C++所提供的 MFC（Microsoft 基础类库），对 Windows 98/NT/2000 所用的 Win32 API（应用程序编程接口）进行了彻底的封装，这使得可以使用完全的面向对象的方法来进行 Windows 程序的开发，并且能够大大缩短程序的开发周期，降低开发成本，把 Windows 程序员从大量复杂的劳动中解放出来。

在一整套的 Microsoft Visual Studio 6.0 集成开发环境（IDE，Integrated Developing Environment）中，Visual C++ 6.0、Visual Basic 6.0 和 Visual InterDev 6.0 等都使用同一类集成开发环境，称作 Developer Studio，如图 1-1 所示。

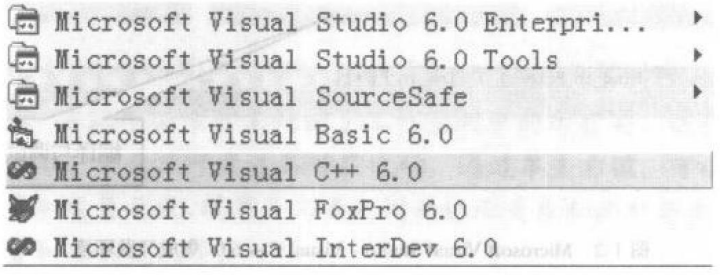


图 1-1　Visual Studio 集成开发环境组成

用户可以在 Developer Studio 中创建所开发的应用程序的源文件、各种资源文件和其他