21世纪高等学校计算机规划教材

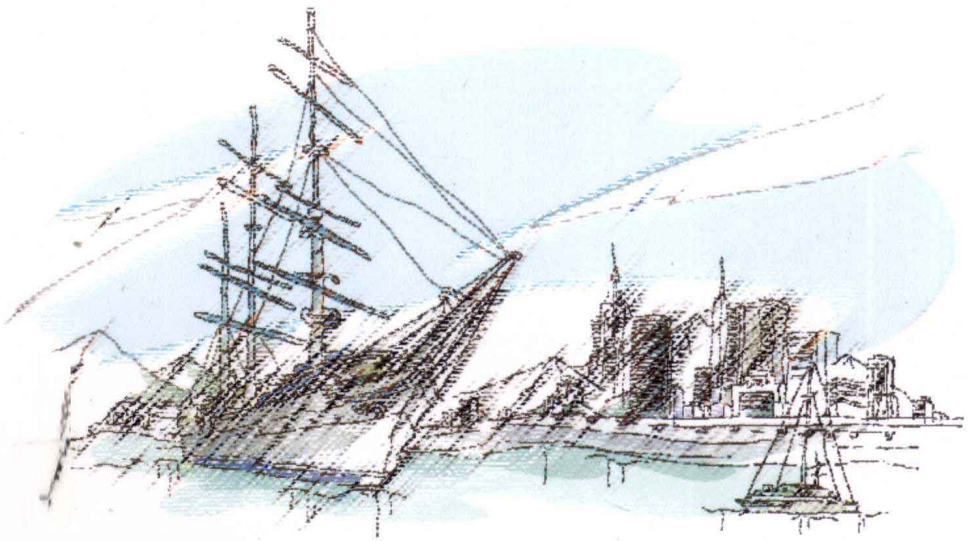
21st Century University Planned Textbooks of Computer Science

Windows系统编程

Windows System Programming

李晓黎 编著

- 体现作者多年的Windows系统编程开发经验
- 讲解大量实用技巧，重点突出，便于灵活掌握
- 提供典型应用实例及其源代码，分析详细，实用性强



高校系列

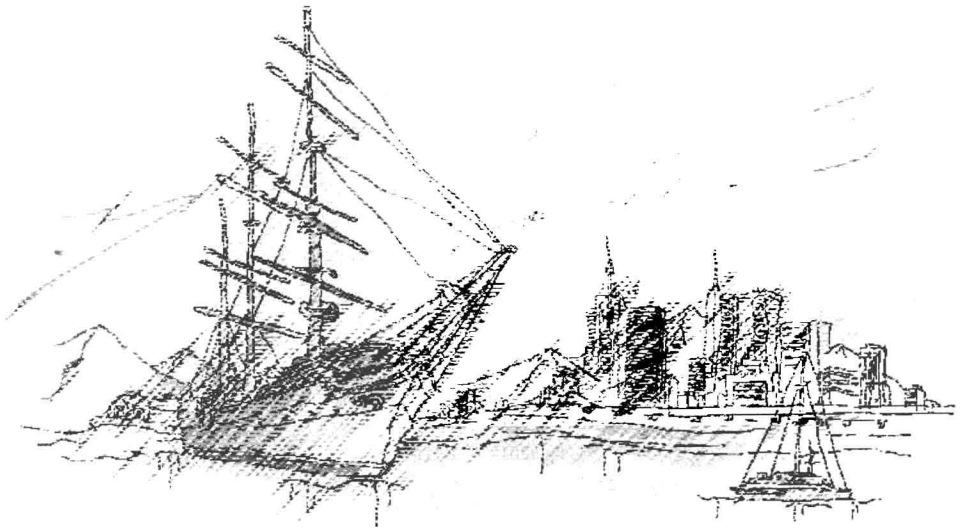
人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

Windows系统编程

Windows System Programming

李晓黎 编著



高校系列

人民邮电出版社
北京

图书在版编目（CIP）数据

Windows系统编程 / 李晓黎编著. -- 北京 : 人民邮电出版社, 2012.1
21世纪高等学校计算机规划教材
ISBN 978-7-115-26702-3

Ⅰ. ①W… Ⅱ. ①李… Ⅲ. ①Windows操作系统－程序设计－高等学校－教材 Ⅳ. ①TP316.7

中国版本图书馆CIP数据核字(2011)第233957号

内 容 提 要

Windows 是个人计算机上最流行的操作系统，在国内外拥有众多家庭和商务用户，因此 Windows 系统编程可以说是程序员的基础必修课。各高校许多专业都开设了相关的课程。本书结合大量的例子，介绍 Windows 系统编程的经典技术，并完整地介绍几个 Windows 系统编程的实例，包括安装和卸载软件管理、Windows 服务状态监视器、键盘监视程序、进程保护器、系统信息查看程序等。

本书可以作为大学本科教材，也可供大专、高职相关专业使用，或作为广大 Windows 系统编程开发人员的参考资料。

21 世纪高等学校计算机规划教材

Windows 系统编程

◆ 编　　著　李晓黎
责任编辑　邹文波

◆ 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　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市潮河印业有限公司印刷

◆ 开本：787×1092　1/16
印张：20.25　　2012 年 1 月第 1 版
字数：536 千字　　2012 年 1 月河北第 1 次印刷

ISBN 978-7-115-26702-3

定价：45.00 元

读者服务热线：(010)67170985　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

所有应用程序都必须在特定的操作系统下运行，不同的操作系统都提供了很多开发包和开发接口。在开发应用程序之前应该充分了解其所运行的操作系统的体系结构、特性和开发接口，这样才能最大限度地满足开发应用程序的需求。

Windows 是个人计算机上最流行的操作系统，在国内外拥有众多家庭和商务用户，因此 Windows 系统编程可以说是程序员的基础必修课。各高校许多专业都开设了相关的课程。

Windows 系统编程不是简单地在对话框中摆放和使用控件，也不只是开发能在 Windows 下运行的应用程序。编者在多年从事 Windows 系统编程和研究相关课程教学的基础上设计本书结构并选择一些经典实例，力求做到理论与实际相结合。

本书首先介绍操作系统的工作原理和 Windows 体系结构，并在后面各章中结合 Windows 体系结构中的各主要部分介绍经典的 Windows 系统编程技术，包括 GDI 用户图形界面编程、文件系统编程、注册表编程、进程编程、多线程编程、Windows 服务编程、动态链接库编程、Windows 钩子编程和 WMI 编程等，内容涉及很多目前比较流行的经典 Windows 系统编程技术，对读者今后的实际工作有很强的指导作用。

为了提高读者的实用开发能力，本书提供了一些经典的案例，包括安装和卸载软件管理、Windows 服务状态监视器、键盘监视程序、进程保护器、系统信息查看程序，读者可以通过这些案例学习开发 Windows 应用程序的过程和技术，也可以在实例的基础上稍加修改，独立使用。本书所有实例都是使用 Visual Studio 2008 开发的 Visual C++应用程序，并在 Windows XP、Windows 2003 和 Windows 7 环境下调试成功，应该能够稳定地运行在目前流行的 Windows 操作系统中。另外，本书每章都配有相应的习题，帮助读者理解所学习的内容，使读者加深印象、学以致用。

因为 Windows 7 采用了新的用户权限管理方法，所以在 Windows 7 下运行本书中一些实例时，应首先生成文件，然后右击可执行文件，选择“以管理员身份运行”。

为了方便读者学习 WMI 编程，本书附录部分还介绍了常见的 CIM 类，同时提供 PPT 课件和程序源代码等，需要者可以登录人民邮电出版社教学服务与资源网（http://www.ptpedu.com.cn）免费下载。

本书在内容的选择、深度的把握上充分考虑初学者的特点，内容安排上力求做到循序渐进，不仅适合于教学，也适合于开发 Windows 系统编程的各类培训机构和个人用户学习与参考。

由于编者水平有限，书中难免有不足之处，敬请广大读者批评指正。

编　者

2011 年 11 月

目 录

第 1 章 操作系统概述

众所周知，操作系统是计算机的灵魂。没有操作系统，计算机就是一堆板卡和芯片的组合，做不了任何事情。任何应用程序都要依托一个操作系统。不同操作系统上运行的应用程序有不同的特点和开发方法。因此，在开发应用程序之前，有必要了解它所运行的操作系统的基本特点，从而在应用程序中充分利用和发挥操作系统的特性。本章简单介绍操作系统的发展历史和基本特性、Windows 操作系统的体系结构以及 Windows 系统开发。

1.1 Windows 操作系统概述

Windows 是在个人计算机上应用最广泛的操作系统，本节将介绍操作系统的基本概念和 Windows 操作系统的发展过程，使读者对 Windows 操作系统形成初步印象。

1.1.1 操作系统的功能和分类

操作系统是一种软件，由程序和数据组成，它运行在计算机上。操作系统在计算机中的地位如图 1.1 所示。

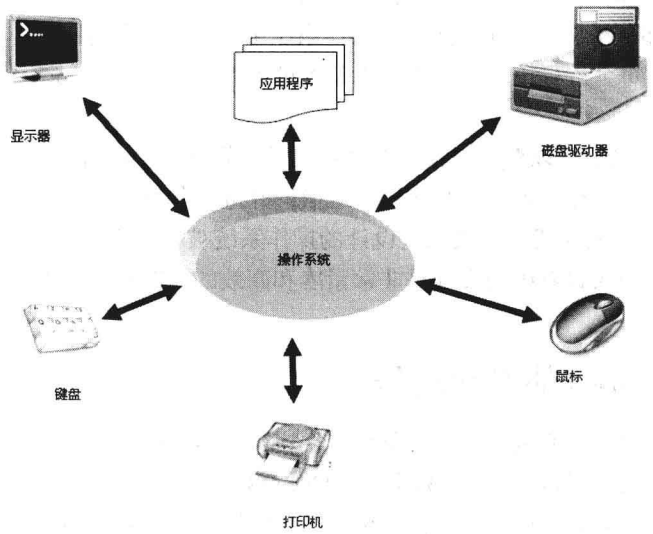


图 1.1 操作系统在计算机中的地位

操作系统主要功能如下。

- 管理计算机的硬件资源，如输入输出、内存分配等。操作系统是应用程序和计算机硬件之间的中介，尽管计算机有时会直接执行应用程序代码。
- 为各种应用软件的运行提供通用服务。

除了计算机外，很多诸如手机和游戏机等其他设备都安装了操作系统。

操作系统由许多部分组成，主要如下。

- 内核（kernel）：操作系统最重要的组件就是内核。它用于控制普通用户看不到的低级处理。例如，如何读写内存，进程执行的顺序，显示器、键盘和鼠标等设备如何接收和发送信息，以及决定如何解释从网络中接收到的信息。
- 用户接口：直接与计算机用户交互的组件，通过它，用户可以控制和使用程序。用户接口可以是带有桌面和图标的图形界面，也可以是带有命令行的界面。
- 应用程序编程接口：提供服务和代码库。 通过使用编程接口，开发人员可以编写出重用性很好的模块化代码。

不同的操作系统在实现这些组件时的侧重点也不同。比如，Windows 的特色在于图形化的用户接口，而很多版本的 Linux 则不是。

操作系统通常有如下几种分类方法。

- 实时操作系统：一种多任务操作系统，用于执行实时应用程序。实时操作系统通常使用专门的调度算法，从而对各种事件做出快速响应。可以采用事件驱动或共享时间的设计。事件驱动系统根据任务的优先级切换任务；而共享时间操作系统则根据时钟中断切换任务。
- 多用户和单用户操作系统：多用户操作系统允许多个用户并发地访问计算机系统。共享时间操作系统通常是多用户操作系统，因为它允许多个用户通过共享时间片来访问计算机。UNIX 允许两个以上用户同时登录，因此，它属于多用户操作系统。尽管 Windows 系统允许创建多个账户，但它还是单用户操作系统，因为同时登录的账户中，只有网络管理员才是真正的用户。
- 多任务和单任务操作系统：单任务操作系统同时只能运行一个程序，而多任务操作系统则可以同时运行多个程序。多任务可以分为两种类型，即先占式多任务和合作式多任务。先占式多任务操作系统将 CPU 时间分成片，然后分给每个程序。UNIX 系列的操作系统（比如 Solaris 和 Linux）支持先占式多任务。在合作式多任务操作系统中，所有正在运行的应用程序必须一起工作在共享系统资源上，每个进程按定义的方式为其他进程提供时间。Windows 95 之前的 Windows 只支持合作式多任务。
- 分布式操作系统：可以对一组独立的计算机进行管理，使它们看起来像一台计算机。使用分布式操作系统的网络计算机可以相互连接和通信，并进行分布式计算。
- 嵌入式系统：为嵌入式计算机系统设计的操作系统被称为嵌入式操作系统，它需要在有限的资源下操作，通常嵌入式系统的设计是非常紧凑和高效的。Windows CE 和 Minix 3 都是嵌入式操作系统。

1.1.2 Windows 的发展过程

Windows 是微软推出的系列操作系统软件，最早以 Windows 命名的操作环境于 1985 年 11 月 20 日作为 MS-DOS 的插件推出，其后逐渐取代 Mac OS 统治了个人计算机市场。据 2009 年 10 月的统计，Windows 大约占据 Internet 上客户端操作系统的 91%的市场份额。在编者编写此书时，最新的客户端版本是 Windows 7，最新的服务器版本是 Windows Server 2008 R2，最新的移动操作

系统版本是 Windows Phone 7。

Microsoft Windows 最初于 1981 年在 IBM 的 PC 上作为 MS-DOS 的插件出现，1990 年发布的 Windows 3.0 和 1992 年发布的 Windows 3.1 在以下几个方面改进了设计。

- 虚拟内存：为多任务内核设计的内存管理技术，它对计算机体系结构中的各种硬件设备（例如，RAM 模块和磁盘存储设备）进行虚拟化。在虚拟内存技术中应用程序认为其拥有连续的可用的内存（连续完整的地址空间），而实际上，它通常被分隔成多个物理内存碎片。
- 虚拟设备驱动器（VxD）：Windows 提供的与外设进行数据交换的核心。用户对外设的操作请求由 Windows 的虚拟机管理器(VMM)传递到 VxD，然后由 VxD 去操作硬件。
- 保护模式：也称保护虚拟地址模式，是 x86 兼容 CPU 的可选模式。它允许系统软件使用虚拟内存、分页和安全多任务等特性，还包括为提高操作系统对应用软件控制而设计的其他特性。

Windows 95 于 1995 年 8 月发布，它具有全新的用户界面，支持多达 255 个字符的长文件名；支持即插即用技术，可以自动检测和配置已安装的硬件。它可以运行 32 位应用程序，并且通过技术改进使其稳定性高于 Windows 3.1。

Windows 98 于 1998 年 6 月发布，它对 Internet 提供了很多支持，集成了 Internet Explorer 4.01、Outlook Express、Microsoft NetMeeting、Microsoft Chat、Personal Web Server、NetMeeting 和 NetShow Player（Windows Media Player 的前身）等众多 Internet 应用程序，并且对用户界面做了很多改进。1999 年 5 月微软发布了 Windows 98 第二版（Windows 98 SE）。Windows 98 深受当时用户的欢迎，微软原计划到 2004 年 11 月就不再对 Windows 98 提供技术支持，但考虑到当时依然有众多用户在使用 Windows 98，微软最终决定将技术支持维持到 2006 年 11 月。

Windows Me 是 Windows 98 的升级版本，也是 Windows 9x 系列的最后一个产品。它于 2000 年 9 月发布，目标用户是家庭 PC 用户。Windows Me 似乎一直生活在它的前辈 Windows 98 和后代 Windows XP 的阴影中，很多用户是直接从 Windows 98 升级到 Windows XP 的，因此忽略了 Windows Me 的存在。

从 1993 年开始，微软陆续发布了 Windows NT 家族，并在商业应用领域获得了很大的成功。最先发布的版本是 NT 3.1，其后又发布了 NT 3.5（1994）、NT 3.51（1995）、NT 4.0（1996）。Windows 2000 于 2000 年 2 月发布，它是 Windows NT 4.0 的升级版本，包括 Professional、Server、Advanced Server 和 Datacenter Server 等版本，主要用于商业领域。

2001 年 8 月，微软推出了集合家用和商用需求的操作系统 Windows XP，XP 是 eXPerience 的缩写，取“给用户带来全新的使用体验”之意。它采用 Windows NT 内核和体系结构，是最受欢迎的 Windows 版本。

Windows Server 2003 是微软服务器操作系统，发布于 2003 年 4 月，多安装于商用服务器计算机上。它在稳定性、可扩展性和性能方面均优于 Windows Server 2000，因此在商用领域获得了成功。Windows Server 2008 是 Windows Server 2003 的升级产品，发布于 2008 年 2 月。这是到目前为止最新的 Windows Server 操作系统产品。

Windows Vista 是 Windows XP 的升级产品，发布于 2007 年 1 月。尽管 Windows Vista 在用户界面和使用便捷性等多方面做了很大改进，但它与之前开发的很多应用程序的兼容性并不好，因此影响了它的普及，很多用户选择继续使用 Windows XP。

Windows 7 是到目前为止最新的 Windows 操作系统产品，发布于 2009 年 7 月。它在软硬件兼容性方面优于 Windows Vista。它包括 Home Basic（家庭普通版）、Home Premium（家庭高级版）、Professional（专业版）、Enterprise（企业版）和 Windows 7 Ultimate（旗舰版）等版本。

1.2 Windows 体系结构和 Windows 系统开发

了解 Windows 的体系结构可以使读者对 Windows 操作系统形成整体认识。掌握 Windows 的内部设计结构，是学习 Windows 系统编程的基础。

1.2.1 Windows 2000/XP 体系结构概述

因为从 Windows XP 开始，所有现用版本 Windows 都基于 Windows NT 内核，所以它们的体系结构大多相近。本节将以 Windows 2000/XP 为例介绍 Windows 操作系统的体系结构。Windows 2000/XP 操作系统的体系结构如图 1.2 所示。

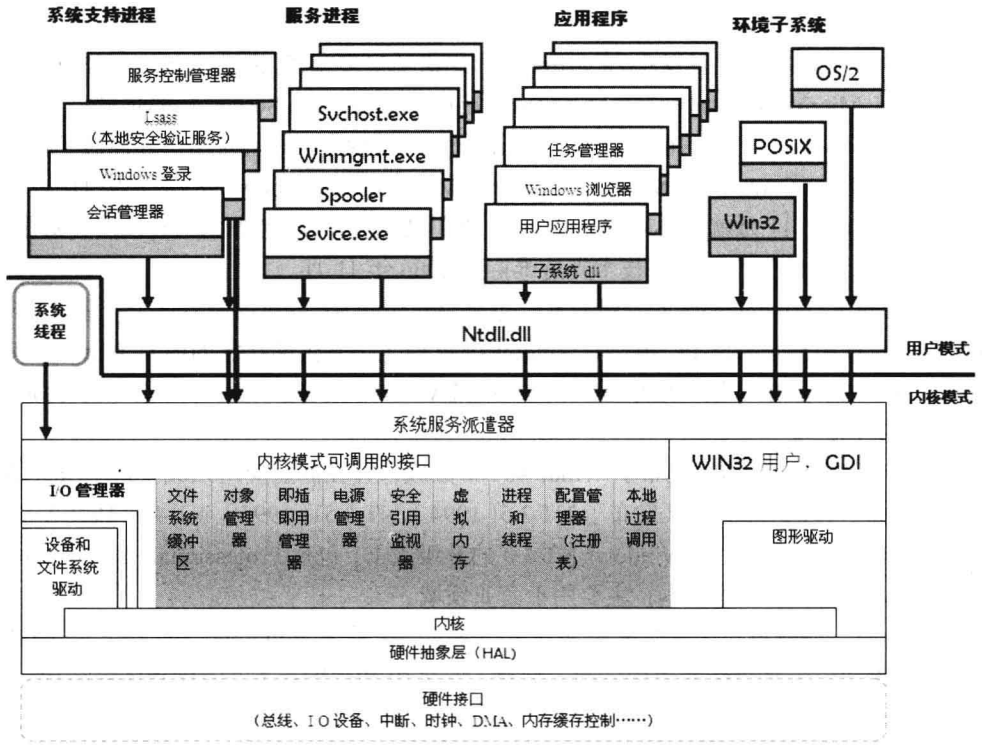


图 1.2 Windows 2000/XP 的体系结构

可以看到，Windows 操作系统由用户模式（User mode，也称用户态）和内核模式（Kernel mode，也称内核态）组成。

1. 用户模式

在用户模式下，程序可以受限制地访问系统资源，而且应用程序的运行优先级比内核模式进程低。从图 1.1 中可以看到，用户模式由如下几个部分组成。

- 系统支持进程：启动 Windows 时必须运行的系统进程，包括服务控制管理器、本地安全认证服务、Windows 登录和会话管理器等。只有运行这些进程，才能在启动 Windows 时自动启动必要的 Windows 服务，为用户提供登录和身份认证服务。

- 服务进程：Windows 服务是长时间运行的、完成特定功能的、不需要用户参与的程序。它可以在 Windows 启动后一直在后台运行，可以设置为当 Windows 启动时自动运行。许多服务出现在 Windows 任务管理器的进程列表中，但不能被用户手动结束进程。在 C/S 结构中的服务程序通常可以 Windows 服务的形式运行，本书第 9 章将介绍 Windows 服务编程的方法。

- 应用程序：Windows 任务管理器、浏览器和用户应用程序都可以看作是应用程序。用户应用程序可以调用动态链接库（DLL）中定义的函数。在程序设计时，可以将功能拆分成小块，并封装在不同的 DLL 项目中实现这些子功能，这是非常好的习惯。这样做不但可以使应用程序的模块化更清晰，而且便于项目组中的不同程序员分工合作，在出现问题时能够更方便地定位问题和调试程序。本书第 10 章将介绍 Windows DLL 编程的方法。

- 环境子系统：在用户应用程序和 Windows 之间扮演中介，使不同操作系统下的应用程序都可以在 Windows 下运行。Windows 支持 Win32、POSIX 和 OS/2 等 3 种环境子系统。每个子系统都独立地运行，应用程序只能在相应的子系统下运行。 Win32 是 Windows 中最主要的子系统，它的基础是 Win32 API 集；POSIX 是由 IEEE 组织发布的 UNIX 系统的标准；OS/2 子系统使 Windows 支持 OS/2 应用程序，但不能运行 OS/2 2.x 图形应用程序。

- Ntdll.dll：是 Windows NT 平台操作系统的重要模块，很多基本的 Windows API 函数都在其中定义。系统支持进程、服务进程、应用程序和环境子系统都可以通过 Ntdll.dll 访问 Windows 内核。Ntdll.dll 丢失或损坏都可能导致系统崩溃。

2. 内核模式

顾名思义，内核模式就是 Windows 内核运行的模式。在内核模式下，程序对所有资源都拥有完全访问的权限，并且运行在系统的保护内存区域。

之所以要将 Windows 划分成用户模式和内核模式，是因为如果所有程序都运行在内核模式下，那么它们都可以改写其他程序的内存区域或系统内存，这样当该程序崩溃时，就可能导致其他程序，甚至整个系统崩溃。

内核模式由执行体服务、内核和硬件抽象层等几个部分组成。

执行体服务由低级内核模式部分组成，包含在 ntkernel.exe 中。它可以处理 I/O、对象管理、安全和进程管理，主要由下面几个部分组成。

- 系统服务派遣器：可以将来自 Ntdll.dll 或其他用户模式模块对系统内核的调用分派到内核模式的可调用接口。

- Win32 用户，GDI：GDI（Graphics Device Interface，图形设备接口）是微软的 Windows 应用程序编程接口，也是 Windows 操作系统的核心组件，它可以将图形对象传送到输出设备（例如显示器或打印机）。GDI 可以实现画线、画正方形、画圆、设置字体、处理调色板等功能，但不包括画窗口和菜单的功能。后者在用户子系统中实现，在 use32.dll 中定义。GDI 基于图形驱动工作。

- I/O 管理器：计算机由多个输入/输出设备组成，例如键盘、鼠标、声音控制器、视频控制器、磁盘驱动器、网络端口等。设备驱动可以控制设备的功能，为设备和操作系统之间提供软件连接。I/O 管理器管理应用程序和设备驱动提供的接口之间的通信。

- 文件系统缓存：文件系统是存储和管理计算机文件和数据的方法。文件系统用于数据存储设备（例如硬盘或 CD-ROM）。

- 对象管理器：是 Windows 内核中用于管理 Windows 资源的子系统。每个资源都对应一个对象。资源可以是物理设备、文件、目录或正在运行的进程。

- 即插即用管理器： 是 Windows 在硬件配置改变时自动重新配置的重要组件。即插即用需

要在硬件驱动程序中提供支持。

- 电源管理器：用于监测设备的使用状态，如果设备不再被使用，则减小该设备的电源状态。同样，这需要硬件支持。
- 安全引用监视器：对象管理器使用安全引用监视器，在访问对象之前进行必要的认证和检查。
- 虚拟内存：当一个程序启动时，就会运行一个对应的进程，每个进程都包含自己的虚拟内存空间和资源。之所以称之为“虚拟”内存，是因为进程认为的内存地址 0x12345678 可能实际是在物理内存的地址 0x65f7a678 上。两个不同的进程可能在同一个虚拟内存地址上存储不同的数据。虚拟内存是通过把内存分页实现的。在 x86 系统上一页的大小是 4KB，每个页都有自己的属性（只读或可读写等）。CPU 支持一种透明的机制将虚拟内存转换为物理内存。使用虚拟内存的主要原因是不能访问其他进程的内存空间。
- 进程和线程：进程是操作系统中的重要概念，它代表一个正在运行的程序，是操作系统动态执行的基本单元。每个进程都有自己的地址空间，可以和其他进程并发地、独立地运行；而线程则是操作系统可以调度的最小的执行单元，一个运行的程序（进程）可以分成多个并发执行的线程。同一进程中的线程可以共享进程的内存。在应用程序中使用多线程编程可以提高应用程序的并发性和处理速度，使后台计算不影响前台界面和用户的交互。
- 配置管理器（注册表）：注册表是用来保存 Windows 操作系统配置选项的层次数据库。系统内核、服务、设备驱动和用户应用程序都可以使用注册表保存配置选项。本书将在第 6 章介绍 Windows 注册表编程的方法。
- 本地过程调用（LPC）：是为子系统通信特别设计的，它的基础是远程过程调用（RPC），RPC 是 UNIX 上用于运行在两台机器上的进程间通信的标准。LPC 为在同一台机器上运行的进程间的通信进行了优化。客户线程当需要子系统的服务时就会调用 LPC。LPC 将调用的参数传递给服务器进程。服务器进程执行服务并使用 LPC 将结果传回给客户端线程。

硬件抽象层是物理硬件与计算机之间的一层，设计这一层的目的是隐藏硬件中的不同，从而为操作系统内核提供一个一致的运行平台，这样 Windows 就可以在不同的母版上运行而不需要修改内核。

内核位于硬件抽象层和执行层之间，提供多处理器同步、线程和中断的调度和派发以及错误处理和异常派发等功能。它还负责在系统启动时初始化设备驱动。

3. 用户模式和内核模式之间的切换

当 Windows 被加载时，将启动 Windows 内核，然后在内核模式下运行并建立分页和虚拟内存，然后创建一些系统进程（就是图 1.1 中所示的系统支持进程），并在用户模式下运行它们。那么，CPU 又是怎样切换回内核模式呢？这并不是 CPU 自动完成的。CPU 经常被特定的事件（例如时钟、键盘、硬盘 I/O）打断，这就是所谓的中断。内核必须首先建立中断处理器来处理这些事件。当中断发生时，CPU 会停止执行正在运行的程序，立即切换到内核模式，执行该事件的中断处理器。处理器保存 CPU 的状态，执行与该事件相关的一些处理，然后恢复 CPU 的状态（可能的话切换到用户模式下），这样 CPU 就可以恢复执行应用程序了。

当程序需要调用 Windows API 函数时，将触发一个中断，并导致 CPU 切换到内核模式，开始执行指定的 API 函数。当 API 函数执行完成后，将切换回用户模式，继续执行程序。之所以要这样做，是因为像 ReadProcessMemory 这样的 API 函数不能在用户模式下工作。在用户模式下，程序不能访问其他程序的内存；但是，在内核模式下，API 函数可以无限制地访问任何内存区域。

下面来看一下在用户模式下，程序之间是如何切换的。事实上，大多数时间正在运行的程序都处于等待状态（等待用户输入或硬盘操作等），允许操作系统切换到其他程序，此时内核可以执

行上下文切换，运行其他程序，操作如下。

- 保存当前程序的状态（包括注册表）；
- 配置要运行的程序；
- 恢复要运行程序的状态。

如果程序（准确地说应该是线程）运行了一段时间（线程时限），操作系统将切换上下文到其他程序（线程）。在用户看来，Windows 就可以同时运行多个应用程序了。

1.2.2　进行 Windows 系统开发时应考虑的因素

Windows 是个人计算机上最流行的操作系统，在国内外拥有众多家庭和商务用户，因此 Windows 系统编程是很多程序员的必修课。在学习 Windows 系统编程之前，需要考虑如下因素。

1. 选择开发语言

可以使用很多开发语言进行 Windows 系统开发。例如 C++、Visual Basic、C#、Delphi 等。因为 Windows 的很多部分是使用 C 语言开发的，而且微软也为 Visual C++提供了包含丰富 Windows API 的开发包，所以本书选择 Visual C++来介绍 Windows 系统开发的方法。

2. 选择开发环境

每个版本的 Windows 都对外公布大量 API（应用程序编程接口），使开发人员可以方便地管理和控制系统资源。那么，面对这么多版本的 Windows，开发人员应该选择哪个作为开发环境呢？应该参考如下因素决定。

（1）对应用程序的用户群进行分析，有些行业的用户至今还在使用 Windows 98 系统。如果要为他们开发应用程序，就要考虑使用 Windows 98 作为开发环境。而如果应用程序的用户是普通的家庭用户，则要考虑使用 Windows XP 作为开发环境。

（2）在经典的 C/S 结构应用程序中，服务器程序通常运行在 Windows Server 操作系统上，而客户端程序通常运行在 Windows XP 和 Windows 7 等兼顾家用和商用的操作系统上。

（3）从 Windows XP 开始，所有现用版本 Windows 都基于 Windows NT 内核，具体如表 1.1 所示。因此，Windows 系列产品具有很好的向下兼容性。也就是说在低版本 Windows 上开发的应用程序通常可以稳定地运行在高版本 Windows，而反之则不一定。因此应尽可能选择满足客户需求的低版本 Windows 作为开发环境。

表 1.1　　Windows 产品的 Windows NT 内核版本号

Windows 产品	Windows NT 内核版本号
Windows 2000	Windows NT 5.0
Windows XP	Windows NT 5.1
Windows Server 2003	Windows NT 5.2
Windows XP SP2	Windows NT 5.2
Windows Vista	Windows NT 6.0
Windows Server 2008	Windows NT 6.0（与 Windows Vista 基于相同版本的 Windows NT 内核）
Windows 7	Windows NT 6.1

基于以上因素，本书所有实例都在 Windows XP、Windows 2003 和 Windows 7 下调试成功，应该能够稳定地运行在目前流行的 Windows 操作系统中。

3. 了解 Windows 系统开发涉及的主要技术与本书内容编排

大多数 Windows 应用程序都在 Windows 系统的用户模式下运行，它们需要通过 Windows 内

核模式的各模块来访问和控制系统的各种资源。Windows 系统开发涉及的技术很多，从 Windows 体系结构中内核模式的视角来总结，Windows 系统编程技术主要包含如下要点。

- 用户界面设计：Windows 应用程序以友好的用户界面而著称，绝大多数 Windows 应用程序都可以通过简单地使用控件来设计用户界面，具体方法将在第 3 章中介绍。有些比较专业的软件可能需要使用绘图技术，例如，在网络管理软件中需要绘制网络拓扑图。在 Windows 应用程序中绘图通常要通过 Windows 内核模式中的 GDI 编程接口，具体方法将在第 4 章中介绍。
- 数据存储：大多数情况下，Windows 应用程序通过 Windows 内核模式中的文件系统来存储简单数据，本书将在第 5 章介绍文件系统编程的基本方法。如果数据量很大或数据结构很复杂，则可以使用专业的数据库来存储数据。由于篇幅所限，而且涉及数据库软件的使用和管理，本书没有对数据库编程技术进行介绍。
- 配置管理：Windows 系统和很多用户应用程序都使用注册表来保存配置信息。应用程序可以通过 Windows 内核模式中的配置管理器来访问和管理注册表。本书将在第 6 章介绍 Windows 注册表编程的方法。
- 应用程序的运行方式：进程和线程是 Windows 内核运行应用程序的基本方式，服务是 Windows 应用程序运行的另一种方式，很多不需要用户交互、只在后台运行的应用程序都以 Windows 服务的形式运行。本书将在第 7 章和第 8 章中介绍进程编程和多线程编程的方法，在第 9 章介绍 Windows 服务编程的方法。
- DLL 编程：DLL 也是一种 Windows 应用程序，只是它不能直接运行，而是被其他应用程序调用。本书将在第 10 章介绍 Windows DLL 编程的方法。
- Windows 内核相关编程：最经典的内核编程就是驱动程序编程。事实上，驱动程序就是操作系统的组成模块，它可以对所有应用程序起作用。那么，如果可以编写驱动程序，不就可以实现 Windows 内核编程了吗？让自己的程序成为 Windows 内核的一部分，这貌似很神秘的事情也并非遥不可及。在杀毒软件、防火墙、U 盘加密等应用程序中都用到了 Windows 驱动编程技术。由于篇幅所限，本书不介绍 Windows 驱动编程技术。驱动程序是运行于 Windows 内核的应用程序，而使用 Windows 钩子的应用程序则是运行于用户模式，拦截 Windows 消息和 API 调用。这样就可以应用程序中影响其他应用程序的运行结果，因此看起来使用 Windows 钩子的应用程序好像运行在 Windows 内核模式中一样。本书将在第 11 章介绍 Windows 钩子编程技术。

另外，WMI 也是 Windows 系统的一个重要插件，在应用程序中可以使用 WMI 获取 Windows 的各种信息，本书将在第 12 章中介绍 WMI 编程技术。

习　　题

一、选择题

1. 下面不是操作系统的是（　　）。

   A. IBM　　B. UNIX

   C. Windows　　D. Mac OS

2. 下面不属于基于 UNIX 的商业化产品的是（　　）。

   A. HP-UNIX　　B. AIX

   C. Solaris　　D. Mac OS

二、填空题

1. 操作系统是一种软件，由____【1】____和____【2】____组成。
2. 最早以 Windows 命名的操作环境是作为____【3】____的插件推出的。
3. 从体系结构上看，Windows 操作系统由____【4】____和____【5】____两种模式组成。
4. Windows 支持____【6】____、____【7】____和____【8】____等 3 种环境子系统。
5. Windows 内核模式由____【9】____、____【10】____和____【11】____等几个部分组成。

三、简答题

1. 简述开发人员在选择哪个版本的 Windows 作为开发环境时的原则。
2. 简述 Windows 体系结构中用户模式由哪几个部分组成。
3. 简述 Windows 体系结构中硬件抽象层的作用。

第2章 使用 Visual Studio 2008 开发 Visual C++应用程序

本书中的程序实例都是使用 Visual Studio 2008 开发的 Visual C++应用程序，本章将对这种开发模式的基础知识进行介绍。由于篇幅所限，本章将不介绍 C++的基本语法，需要了解的读者可以查阅相关书籍和资料。

2.1 Visual Studio 2008 开发环境

Visual Studio 是一套完整的开发工具集，可以用于生成 ASP.NET Web 应用程序、桌面应用程序、移动应用程序等。它集成 Visual C++、Visual C#、Visual Basic、Visual J#等多种开发语言，并全面支持 Microsoft .NET Framework。本节通过一个小例子来介绍 Visual Studio 2008 的开发环境，使读者对其有一个初步的了解。

2.1.1 启动 Visual Studio 2008

在“开始”菜单中依次选择“程序”/“Microsoft Visual Studio 2008”/“Microsoft Visual Studio 2008”，启动 Microsoft Visual Studio 2008 开发环境窗口，如图 2.1 所示。

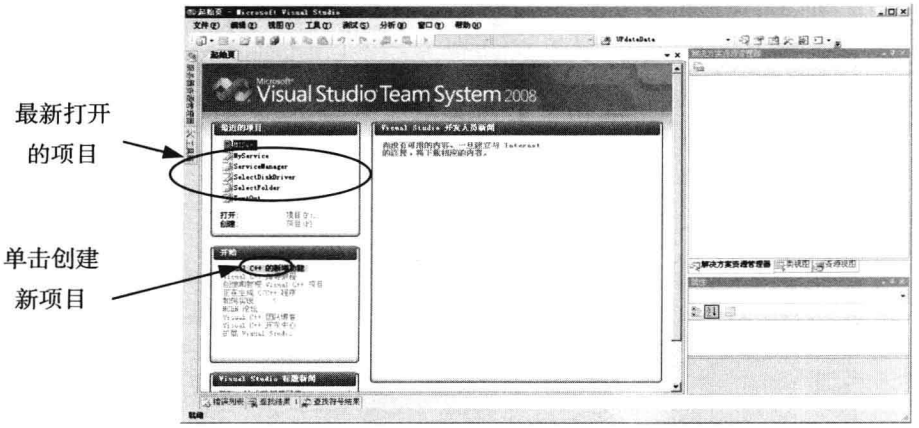


图 2.1 Visual Studio 2008 开发环境窗口