

## <<Windows系统编程>>

### 图书基本信息

书名：<<Windows系统编程>>

13位ISBN编号：9787115103222

10位ISBN编号：7115103224

出版时间：2002-7-1

出版单位：人民邮电出版社

作者：徐丙立,马小亮,周金萍,姜小光

页数：690

字数：1080

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## <<Windows系统编程>>

### 内容概要

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。

## <<Windows系统编程>>

### 书籍目录

#### 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

###### 1.1.2 新一代Windows平台--Windows XP

###### 1.1.3 未来的Windows平台--64位Windows

##### 1.2 集成性开发环境Developer Studio

###### 1.2.1 Microsoft Visual C++和Developer Studio

###### 1.2.2 新一代集成性开发环境Visual Studio.net

##### 1.3 使用编程帮助

###### 1.3.1 为什么需要帮助

###### 1.3.2 如何使用帮助

##### 1.4 自定义Developer Studio

###### 1.4.1 自定义工具条和菜单栏

###### 1.4.2 自定义快捷键

#### 第2章 Win32程序设计

##### 2.1 Windows程序设计的特点

###### 2.1.1 消息驱动机制

###### 2.1.2 图形输出及设备无关性

###### 2.1.3 标准的用户界面对象

###### 2.1.4 Windows资源的共享

##### 2.2 Windows应用程序组成

##### 2.3 用SDK进行Win32程序设计

###### 2.3.1 Win32 API和SDK 27

###### 2.3.2 Win32程序设计的特点

###### 2.3.3 实例--禁止进程的多个实例存在

##### 2.4 初识进程、线程和内存分配

###### 2.4.1 进程与线程的问题

###### 2.4.2 32位应用程序的内存分配

###### 2.4.3 32位应用程序的内存管理模式

##### 2.5 本章实例--消息监视专家Spy

#### 第3章 Windows用户界面

##### 3.1 控件

###### 3.1.1 按钮

###### 3.1.2 组合框

###### 3.1.3 编辑控件

###### 3.1.4 列表框

###### 3.1.5 滚动条

###### 3.1.6 静态控件

##### 3.2 资源

###### 3.2.1 光标

###### 3.2.2 图标

###### 3.2.3 菜单

###### 3.2.4 字符串

##### 3.3 用户输入

###### 3.3.1 通用对话框

## <<Windows系统编程>>

3.3.2 鼠标输入

3.3.3 键盘输入

3.4 窗口

3.5 实例1--迷你视频终端VideoTerminal

3.6 实例2--自定义资源的程序WinMainSample

第4章 Windows程序员基础

4.1 Windows对错误的处理

4.1.1 错误代码表

4.1.2 获取错误信息--GetLastError()

4.1.3 错误代码转换工具

4.1.4 自定义错误代码

4.2 Unicode编程与软件本地化

4.2.1 为什么要选择Unicode

4.2.2 如何编写Unicode源代码

4.2.3 使自己的应用程序符合Unicode规范

4.2.4 如何区分ANSI文本和Unicode文本

4.2.5 在多字节字符与宽字节字符之间转换

4.3 内核对象的概念

4.3.1 什么是内核对象

4.3.2 管理和操作内核对象

4.3.3 进程间共享内核对象

4.4 本章实例--Unicode转换大师UConvert

第5章 进程

5.1 进程的实例句柄

5.2 进程的命令行和环境变量

5.2.1 进程的命令行

5.2.2 进程的环境变量

5.3 进程的当前驱动器和当前目录

5.4 进程的亲缘性

5.5 进程的错误模式

5.6 创建进程与终止进程

5.6.1 创建进程函数CreateProcess()

5.6.2 终止进程

5.7 子进程

5.8 本章实例1--进程查看器ProcessView

5.9 本章实例2--事件调试浏览器Debug Event Browser

第6章 进程的作业

6.1 对作业进程的限制

6.2 将进程放入作业和终止作业

6.2.1 将进程放入作业

6.2.2 终止作业中的进程

6.3 查询作业信息

6.4 作业通知信息

第7章 线程基础

7.1 由进程到线程

7.2 线程的使用条件

7.2.1 何时能够使用线程

## <<Windows系统编程>>

### 7.2.2 何时不能使用线程

### 7.3 线程的创建与终止

#### 7.3.1 线程函数的编写

#### 7.3.2 线程的创建

#### 7.3.3 线程的终止

#### 7.3.4 深入了解线程本质

### 7.4 C/C++运行时库与线程

#### 7.4.1 C/C++运行时库的问题

#### 7.4.2 C/C++运行时库函数与局部数据块tiddata

#### 7.4.3 为什么不调用CreateThread()创建线程

#### 7.4.4 不应该调用的C/C++运行时库函数

### 7.5 线程在系统中的ID

#### 7.5.1 通过ID操作线程

#### 7.5.2 将伪句柄转换为实句柄

### 7.6 线程分类

#### 7.6.1 工作线程

#### 7.6.2 用户界面线程

### 7.7 本章实例--文件比较工具WinDiff

## 第8章 线程的调度、优先级和亲缘性

### 8.1 线程的调度

#### 8.1.1 系统对线程的调度过程

#### 8.1.2 暂停和恢复线程的运行

#### 8.1.3 睡眠方式

#### 8.1.4 转换到另一个线程

#### 8.1.5 线程的运行时间

#### 8.1.6 CONTEXT结构

### 8.2 优先级

#### 8.2.1 线程的优先级

#### 8.2.2 优先级的抽象理解

#### 8.2.3 使用优先级编程

### 8.3 亲缘性

#### 8.3.1 软亲缘性和硬亲缘性

#### 8.3.2 进程的亲缘性屏蔽

#### 8.3.3 进程中线程的亲缘性屏蔽

### 8.4 本章实例--一个多线程程序MThread

## 第9章 线程的同步

### 9.1 用户模式中的线程同步

#### 9.1.1 原子访问与互锁函数

#### 9.1.2 高级线程同步

#### 9.1.3 高速缓存行

#### 9.1.4 临界代码区

### 9.2 线程与内核对象的同步

#### 9.2.1 已通知状态与未通知状态

#### 9.2.2 等待函数

#### 9.2.3 事件 ( Event )

#### 9.2.4 等待定时器 ( WaitableTimer )

#### 9.2.5 信号量 ( Semaphore )

## <<Windows系统编程>>

- 9.2.6 互斥对象 ( Mutex )
- 9.2.7 线程同步对象速查表
- 9.2.8 其他线程同步函数
- 9.3 线程池
- 9.4 本章实例--声音的获取与回放 ( AudioLoop )
- 第10章 线程的堆栈与纤程
  - 10.1 线程的堆栈
    - 10.1.1 Windows 2000下的线程堆栈
    - 10.1.2 Windows 98下的线程堆栈
  - 10.2 纤程
    - 10.2.1 纤程的意义
    - 10.2.2 纤程的使用
  - 10.3 本章实例--基于纤程的文件拷贝器Fibers
- 第11章 系统信息与注册表
  - 11.1 系统信息
    - 11.1.1 硬件配置
    - 11.1.2 操作系统版本
    - 11.1.3 计算机名
    - 11.1.4 操作系统配置
    - 11.1.5 系统参数
    - 11.1.6 系统尺寸
  - 11.2 注册表
    - 11.2.1 注册表结构
    - 11.2.2 注册表存储空间
    - 11.2.3 预定义的关键字
    - 11.2.4 数据分类
    - 11.2.5 关键字的打开、创建与关闭
    - 11.2.6 注册表数据的添加和删除
    - 11.2.7 注册表关键字的安全属性与访问权限
    - 11.2.8 从注册表中检索数据
    - 11.2.9 注册表文件
    - 11.2.10 注册表的使用
  - 11.3 本章实例--注册表读取专家Registry
- 第12章 虚拟内存
  - 12.1 进程的虚拟地址空间
    - 12.1.1 虚拟地址空间的分区
    - 12.1.2 地址空间中区域的管理
    - 12.1.3 地址空间区域中物理内存的占用
    - 12.1.4 实例--创建保护页面
    - 12.1.5 CPU的数据对齐特性
  - 12.2 虚拟内存的状态
    - 12.2.1 内存的使用状态
    - 12.2.2 虚拟地址空间的状态
  - 12.3 地址窗口扩展
  - 12.4 本章实例--虚拟内存的管理员Walker
- 第13章 文件映射
  - 13.1 关于文件映射

## <<Windows系统编程>>

- 13.1.1 文件视图的数据一致性
- 13.1.2 文件映射的优势
- 13.2 使用文件映射
  - 13.2.1 创建或打开文件内核对象
  - 13.2.2 创建文件映射内核对象
  - 13.2.3 创建文件视图
  - 13.2.4 撤销文件视图
  - 13.2.5 关闭文件映射对象和文件对象
  - 13.2.6 两个文件映射的例子
- 13.3 几个不同类型文件的内存映射
  - 13.3.1 EXE文件和DLL文件的映射
  - 13.3.2 数据文件的映射
- 13.4 共享文件和内存
- 13.5 本章实例--实现命名共享内存的程序Memory
- 第14章 内存堆栈
  - 14.1 关于内存堆栈
    - 14.1.1 堆栈的概念
    - 14.1.2 进程的默认堆栈
  - 14.2 创建与使用内存堆栈
    - 14.2.1 为什么要创建内存堆栈
    - 14.2.2 如何使用内存堆栈
    - 14.2.3 C++程序中如何使用堆栈
  - 14.3 其他堆栈函数的使用
  - 14.4 本章实例--多线程的堆栈管理器MpHeap
- 第15章 动态链接库
  - 15.1 为什么要使用DLL
  - 15.2 DLL在进程的地址空间
  - 15.3 DLL的隐式链接
    - 15.3.1 DLL模块的创建
    - 15.3.2 EXE模块的创建
    - 15.3.3 EXE模块的运行
  - 15.4 DLL的显式链接
    - 15.4.1 显式加载DLL模块
    - 15.4.2 显式卸载DLL模块
    - 15.4.3 DLL的使用计数
    - 15.4.4 获得输出符号地址
  - 15.5 DLL的进入点函数
    - 15.5.1DllMain()函数
    - 15.5.2 C/C++运行时库的情况
  - 15.6 DLL的高级使用
    - 15.6.1 延迟加载DLL
    - 15.6.2 操作系统的DLL
    - 15.6.3 DLL的转移
    - 15.6.4 模块的移位
    - 15.6.5 模块的绑定
    - 15.6.6 线程本地存储器
  - 15.7 本章实例1--一个简单的动态链接库程序dll

## <<Windows系统编程>>

15.8 本章实例2--标准DLL的创建实例SpinCube

第16章 结构化异常处理

16.1 关于结构化异常处理

16.1.1 异常处理

16.1.2 基于帧的异常处理

16.1.3 结束处理

16.1.4 处理器的文法结构

16.2 结构化异常处理的使用

16.2.1 使用异常处理器

16.2.2 使用结束处理器

16.3 结构化异常与C++异常的对比

16.4 本章实例--模式匹配查找工具Asyncio 654



#### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>