<<深入解析Windows操作系统>>

图书基本信息

书名：<<深入解析Windows操作系统>>

13位ISBN编号：9787115290908

10位ISBN编号：7115290903

出版时间：2012-9

出版时间：人民邮电出版社

作者：[美]Mark Russinovich　David Solomon　[加]Alex Ionescu 著

页数：726

字数：744000

<<深入解析Windows操作系统>>

内容概要

《深入解析Windows操作系统，卷1》是操作系统内核专家Russinovich等人的Windows操作系统原理的最新版著作，针对Windows
7和Windows Server 2008
R2进行了全面的更新，主要讲述Windows的底层关键机制、Windows的核心组件(包括进程/线程/作业、安全性、I/O系统、存储管理、内存管理、缓存管理、文件系统和网络)，并分析了启动进程、关机进程以及缓存转储。
书中提供了许多实例，读者可以借此更好地理解Windows
的内部行为。

《深入解析Windows操作系统，卷1》内容丰富，信息全面，适合众多Windows平台开发人员、系统管理员阅读。

作者简介

作者:(美)Mark Russinovich，(美)David Solomon，(加)Alex Ionescu

<<深入解析Windows操作系统>>

书籍目录

Chapter 1 Concepts and Tools
Windows Operating System Versions
Foundation Concepts and Terms
Windows API
Services, Functions, and Routines
Processes, Threads, and Jobs
Virtual Memory
Kernel Mode vs. User Mode
Terminal Services and Multiple Sessions
Objects and Handles
Security
Registry
Unicode
Digging into Windows Internals
Performance Monitor
Kernel Debugging
Windows Software Development Kit
Windows Driver Kit
Sysinternals Tools
Conclusion
Chapter 2 System Architecture
Requirements and Design Goals
Operating System Model
Architecture Overview
Portability
Symmetric Multiprocessing
Scalability
Differences Between Client and Server Versions
Checked Build
Key System Components
Environment Subsystems and Subsystem DLLs
Ntdll.dll
Executive
Kernel
Hardware Abstraction Layer
Device Drivers
System Processes
Conclusion
Chapter 3 System Mechanisms
Trap Dispatching
Interrupt Dispatching
Timer Processing
Exception Dispatching
System Service Dispatching
Object Manager

Executive Objects
Object Structure
Synchronization
High-IRQL Synchronization
Low-IRQL Synchronization
System Worker Threads
Windows Global Flags
Advanced Local Procedure Call
Connection Model
Message Model
Asynchronous Operation
Views, Regions, and Sections
Attributes
Blobs, Handles, and Resources
Security
Performance
Debugging and Tracing
Kernel Event Tracing
Wow64
Wow64 Process Address Space Layout
System Calls
Exception Dispatching
User APC Dispatching
Console Support
User Callbacks
File System Redirection
Registry Redirection
I/O Control Requests
16-Bit Installer Applications
Printing
Restrictions
User-Mode Debugging
Kernel Support
Native Support
Windows Subsystem Support
Image Loader
Early Process Initialization
DLL Name Resolution and Redirection
Loaded Module Database
Import Parsing
Post-Import Process Initialization
SwitchBack
API Sets
Hypervisor (Hyper-V)
Partitions
Parent Partition
Child Partitions

Hardware Emulation and Support
Kernel Transaction Manager
Hotpatch Support
Kernel Patch Protection
Code Integrity
Conclusion
Chapter 4 Management Mechanisms
The Registry
Viewing and Changing the Registry
Registry Usage
Registry Data Types
Registry Logical Structure
Transactional Registry (TxR)
Monitoring Registry Activity
Process Monitor Internals
Registry Internals
Services
Service Applications
The Service Control Manager
Service Startup
Startup Errors
Accepting the Boot and Last Known Good
Service Failures
Service Shutdown
Shared Service Processes
Service Tags
Unified Background Process Manager
Initialization
UBPM API
Provider Registration
Consumer Registration
Task Host
Service Control Programs
Windows Management Instrumentation
Providers
The Common Information Model and the Managed Object Format Language
Class Association
WMI Implementation
WMI Security
Windows Diagnostic Infrastructure
WDI Instrumentation
Diagnostic Policy Service
Diagnostic Functionality
Conclusion
Chapter 5 Processes, Threads, and Jobs
Process Internals

Data Structures
Protected Processes
Flow of CreateProcess
Stage 1： Converting and Validating Parameters and Flags
Stage 2： Opening the Image to Be Executed
Stage 3： Creating the Windows Executive Process Object (PspAllocateProcess)
Stage 4： Creating the Initial Thread and Its Stack and Context
Stage 5： Performing Windows Subsystem-Specific Post-Initialization
Stage 6： Starting Execution of the Initial Thread
Stage 7： Performing Process Initialization in the Context of the New Process
Thread Internals
Data Structures
Birth of a Thread
Examining Thread Activity
Limitations on Protected Process Threads
Worker Factories (Thread Pools)
Thread Scheduling
Overview of Windows Scheduling
Priority Levels
Thread States
Dispatcher Database
Quantum
Priority Boosts
Context Switching
Scheduling Scenarios
Idle Threads
Thread Selection
Multiprocessor Systems
Thread Selection on Multiprocessor Systems
Processor Selection
Processor Share-Based Scheduling
Distributed Fair Share Scheduling
CPU Rate Limits
Dynamic Processor Addition and Replacement
Job Objects
Job Limits
Job Sets
Conclusion
Chapter 6 Security
Security Ratings
Trusted Computer System Evaluation Criteria
The Common Criteria
Security System Components

Protecting Objects
Access Checks
Security Identifiers
Virtual Service Accounts
Security Descriptors and Access Control
The AuthZ API
Account Rights and Privileges
Account Rights
Privileges
Super Privileges
Access Tokens of Processes and Threads
Security Auditing
Object Access Auditing
Global Audit Policy
Advanced Audit Policy Settings
Logon
Winlogon Initialization
User Logon Steps
Assured Authentication
Biometric Framework for User Authentication
User Account Control and Virtualization
File System and Registry Virtualization
Elevation
Application Identi cation (AppID)
AppLocker
Software Restriction Policies
Conclusion
Chapter 7 Networking
Windows Networking Architecture
The OSI Reference Model
Windows Networking Components
Networking APIs
Windows Sockets
Winsock Kernel
Remote Procedure Call
Web Access APIs
Named Pipes and Mailslots
NetBIOS
Other Networking APIs
Multiple Redirector Support
Multiple Provider Router
Multiple UNC Provider
Surrogate Providers
Redirector
Mini-Redirectors
Server Message Block and Sub-Redirectors
Distributed File System Namespace

Distributed File System Replication
Offline Files
Caching Modes
Ghosts
Data Security
Cache Structure
BranchCache
Caching Modes
BranchCache Optimized Application Retrieval：SMB Sequence
BranchCache Optimized Application Retrieval：HTTP Sequence
Name Resolution
Domain Name System
Peer Name Resolution Protocol
Location and Topology
Network Location Awareness
Network Connectivity Status Indicator
Link-Layer Topology Discovery
Protocol Drivers
Windows Filtering Platform
NDIS Drivers
Variations on the NDIS Miniport
Connection-Oriented NDIS
Remote NDIS
QoS
Binding
Layered Network Services
Remote Access
Active Directory
Network Load Balancing
Network Access Protection
Direct Access
Conclusion
Index

章节摘录

版权页： 插图： This logical behavior (which helps ensure that administrators will always have full control of the running code on the system) clashes with the system behavior for digital rights management require-ments imposed by the media industry on computer operating systems that need to support playback of advanced, high-quality digital content such as Blu-ray and HD-DVD media. To support reliable and protected playback of such content, Windows uses protected processes. These processes exist along-side normal Windows processes, but they add significant constraints to the access rights that other processes on the system (even when running with administrative privileges) can request. Protected processes can be created by any application; however, the operating system will allow a process to be protected only if the image file has been digitally signed with a special Windows Media Certificate. The Protected Media Path (PMP) in Windows makes use of protected processes to provide protection for high-value media, and developers of applications such as DVD players can make use of protected processes by using the Media Foundation API. The Audio Device Graph process (Audiodg.exe) is a protected process because protected music content can be decoded through it. Similarly, the Windows Error Reporting (or WER, discussed in Chapter 3) client process (Werfault.exe) can also run protected because it needs to have access to protected processes in case one of them crashes. Finally, the System process itself is protected because some of the decryption information is generated by the Ksecdd.sys driver and stored in its user-mode memory. The System process is also protected to protect the integrity of all kernel handles (because the System process' handle table contains all the kernel handles on the system).

编辑推荐

名人推荐

“ 在微软。
我们一直用本书培训新员工……本书是深入理解Windows的绝佳入门书。
”——Windows之父 Jim Allchin“ 每一位操作系统开发人员都应该拥有本书。
”——微软技术院士、Windows NT首席设计师 David Cutler“ 我想不出还有哪一本书比本书更具权威性。
”——微软公司副总裁 Ben Fathi

版权说明