Developing Windows Nt Device Drivers A Programmers Handbook Programming The Win 32 Driver Model

Practical Reverse EngineeringProgramming WindowsWindows InternalsWindows NT File System InternalsWindows Internals, Part 1Programming the Microsoft Windows Driver ModelProgramming the Microsoft Windows Driver ModelWindows InternalsDeveloping Drivers with the Windows Driver FoundationThe Guru's Guide to Transact-SQLWriting Windows WDM Device DriversWindows System ProgrammingWindows NT/2000 Native API ReferenceUndocumented Windows 2000 SecretsWriting Windows WDM Device DriversInside Windows DebuggingInside Windows 2000 ServerWhat Makes It Page?Developing Windows NT Device DriversOld New ThingThe Windows 2000 Device Driver BookWriting Windows VxDs and Device DriversThe Windows NT Device Driver BookShowstopper!Systems Programming for Windows 95FreeBSD Architecture HandbookWindows Kernel ProgrammingProfessional RootkitsWin32 ProgrammingWindows Developer's JournalThe Windows 2000 Device Driver BookWriting MS-DOS Device DriversInside Windows NTInside Windows Server 2003Windows InternalsWindows NT Device Driver DevelopmentWindows 7 Device DriverInside Windows StorageHandbook of Data Communications and NetworksDTrace

Practical Reverse Engineering

An authoritative guide to Windows NT driver development, now completely revised and updated. The CD-ROM includes all source code, plus Microsoft hardware standards documents, demo software, and more.

Programming Windows

Windows NT/2000 Native API Reference is absolutely unique. Currently, documentation on WIndows NT's native APIs can only be found through access to the source code or occasionally Web sites where people have chosen to share bits of insight gained through reverse engineering. This book provides the first complete reference to the API functions native to Windows NT and covers the set of services that are offered by Windows NT to both kernel- and user-mode programs. Ideal for the intermediate and advanced level user- and kernel-mode developers of Windows systems, this books is devoted to the NT native API and consists of documentation of the 210 routines included in the API. Also included are all the functions added in Windows 2000.

Windows Internals

See how the core components of the Windows operating system work behind the scenes—guided by a team of internationally renowned internals experts. Fully updated for Windows Server(R) 2008 and Windows Vista(R), this classic guide delivers key

architectural insights on system design, debugging, performance, and support—along with hands-on experiments to experience Windows internal behavior firsthand. Delve inside Windows architecture and internals: Understand how the core system and management mechanisms work—from the object manager to services to the registry Explore internal system data structures using tools like the kernel debugger Grasp the scheduler's priority and CPU placement algorithms Go inside the Windows security model to see how it authorizes access to data Understand how Windows manages physical and virtual memory Tour the Windows networking stack from top to bottom—including APIs, protocol drivers, and network adapter drivers Troubleshoot file-system access problems and system boot problems Learn how to analyze crashes

Windows NT File System Internals

bull; bull;The data storage market continues to grow even in the current technology downturn. Microsoft is rapidly gaining market share in this area. bull;Other books on storage contain little or no information on Windows. bull;This book appeals both to networking professionals who need to learn about Microsoft as well as Microsoft professionals who need to learn about storage issues.

Windows Internals, Part 1

Programming the Microsoft Windows

Driver Model

There is nothing like the power of the kernel in Windows - but how do you write kernel drivers to take advantage of that power? This book will show you how.The book describes software kernel drivers programming for Windows. These drivers don't deal with hardware, but rather with the system itself: processes, threads, modules, registry and more. Kernel code can be used for monitoring important events, preventing some from occurring if needed. Various filters can be written that can intercept calls that a driver may be interested in.

Programming the Microsoft Windows Driver Model

An in-depth guide of the FreeBSD Operating System Architecture. This manual is available online for free at freebsd.org. This manual is printed in grayscale.

Windows Internals

This "inside account captures the energy—and the madness—of the software giant's race to develop a critical new program. . . . Gripping" (Fortune Magazine). Showstopper is the dramatic, inside story of the creation of Windows NT, told by Wall Street Journal reporter G. Pascal Zachary. Driven by the legendary David Cutler, a picked band of software engineers sacrifices almost everything in their lives to build a new, stable, operating system aimed at giving Microsoft a platform for growth through the next

decade of development in the computing business. Comparable in many ways to the Pulitzer Prize–winning book The Soul of a New Machine by Tracy Kidder, Showstopper gets deep inside the process of software development, the lives and motivations of coders and the pressure to succeed coupled with the drive for originality and perfection that can pull a diverse team together to create a program consisting of many hundreds of thousands of lines of code.

Developing Drivers with the Windows Driver Foundation

This is a book for curious people. It attempts to answer the basic question "how does it work?" As such, it does not explain how to call documented APIs and DDIs to accomplish some specific goal. There is plenty of information available on these subjects, including the MSDN Library, the WDK documentation and several excellent books. Rather, its purpose is to analyze how the Virtual Memory Manager works, simply because it is something worth knowing. With a certain mindset, it might even be something fun to know.Even though this book gives a fairly detailed description of the Virtual Memory Manager, it is not reserved for experienced kernel level programmers. Parts I and II provide information on the x64 processor and enough details on kernel mode code execution to help readers approaching these subjects for the first time.This book describes the Windows 7 x64 implementation of the Virtual Memory Manager. All of the analysis and experiments have been performed

Page 5/27

on this particular version only.

The Guru's Guide to Transact-SQL

Delve inside Windows architecture and internals—and see how core components work behind the scenes. Led by three renowned internals experts, this classic guide is fully updated for Windows 7 and Windows Server 2008 R2—and now presents its coverage in two volumes. As always, you get critical insider perspectives on how Windows operates. And through hands-on experiments, you'll experience its internal behavior firsthand—knowledge you can apply to improve application design, debugging, system performance, and support. In Part 1, you will: Understand how core system and management mechanisms work—including the object manager, synchronization, Wow64, Hyper-V, and the registry Examine the data structures and activities behind processes, threads, and jobs Go inside the Windows security model to see how it manages access, auditing, and authorization Explore the Windows networking stack from top to bottom—including APIs, BranchCache, protocol and NDIS drivers, and layered services Dig into internals hands-on using the kernel debugger, performance monitor, and other tools

Writing Windows WDM Device Drivers

Whether you want to learn how to develop a robust, full-featured rootkit or you're looking for effective ways to prevent one from being installed on your network, this hands-on resource provides you with the

tools you'll need. Expert developer Ric Vieler walks you through all of the capabilities of rootkits, the technology they use, steps for developing and testing them, and the detection methods to impede their distribution. This book provides the detailed, step-by-step instructions and examples required to produce full-featured, robust rootkits. Presented in modular sections, source code from each chapter can be used separately or together to produce highlyspecific functionality. In addition, Vieler details the loading, configuration, and control techniques used to deploy rootkits. All ancillary software is fully detailed with supporting source code and links to the compilers, utilities, and scripts necessary to build and run every example provided. What you will learn from this book Complete coverage of all major rootkit technologies: kernel hooks, process injection, I/O filtering, I/O control, memory management, process synchronization, TDI communication, network filtering, email filtering, key logging, process hiding, device driver hiding, registry key hiding, directory hiding and more Complete coverage of the compilers, kits, utilities, and tools required to develop robust rootkits Techniques for protecting your system by detecting a rootkit before it's installed Ways to create modular, commercial grade software Who this book is for This book is for anyone who is involved in software development or computer security. Wrox Professional guides are planned and written by working programmers to meet the real-world needs of programmers, developers, and IT professionals. Focused and relevant, they address the issues technology professionals face every day. They provide examples, practical solutions, and expert

education in new technologies, all designed to help programmers do a better job.

Windows System Programming

The Oracle Solaris DTrace feature revolutionizes the way you debug operating systems and applications. Using DTrace, you can dynamically instrument software and quickly answer virtually any question about its behavior. Now, for the first time, there's a comprehensive, authoritative guide to making the most of DTrace in any supported UNIX environment--from Oracle Solaris to OpenSolaris, Mac OS X, and FreeBSD. Written by key contributors to the DTrace community, DTrace teaches by example, presenting scores of commands and easy-to-adapt, downloadable D scripts. These concise examples generate answers to real and useful questions, and serve as a starting point for building more complex scripts. Using them, you can start making practical use of DTrace immediately, whether you're an administrator, developer, analyst, architect, or support professional. The authors fully explain the goals, techniques, and output associated with each script or command. Drawing on their extensive experience, they provide strategy suggestions, checklists, and functional diagrams, as well as a chapter of advanced tips and tricks. You'll learn how to Write effective scripts using DTrace's D language Use DTrace to thoroughly understand system performance Expose functional areas of the operating system, including I/O, filesystems, and protocols Use DTrace in the application and database development

process Identify and fix security problems with DTrace Analyze the operating system kernel Integrate DTrace into source code Extend DTrace with other tools This book will help you make the most of DTrace to solve problems more quickly and efficiently, and build systems that work faster and more reliably.

Windows NT/2000 Native API Reference

This is a guide book with software for programmers writing device drivers for Windows NT. This is the only book and sample software available on Device Drivers--NT.

Undocumented Windows 2000 Secrets

This superb introduction to device drivers describes what device drivers do, how they interface with DOS, and provides examples and techniques for building a collection of device drivers that can be customized for individual use.

Writing Windows WDM Device Drivers

Explaining how and why developers can combine various low-level system calls to accomplish high-end results, this book emphasizes low-level solutions using C and C++. The CD contains sample code so programmers can work with it online.

Inside Windows Debugging

Inside Windows 2000 Server

Delve inside Windows architecture and internals - and see how core components work behind the scenes. This classic guide has been fully updated for Windows 8.1 and Windows Server 2012 R2, and now presents its coverage in three volumes: Book 1, User Mode; Book 2, Kernel Mode; Book 3, Device Driver Models. In Book 1, you'll plumb Windows fundamentals, independent of platform - server, desktop, tablet, phone, Xbox. Coverage focuses on high-level functional descriptions of the various Windows components and features that interact with, or are manipulated by, user mode programs, or applications. You'll also examine management mechanisms and operating system components that are implemented in user mode, such as service processes. As always, you get critical insider perspectives on how Windows operates. And through hands-on experiments, you'll experience its internal behavior firsthand - knowledge you can apply to improve application design, debugging, system performance, and support. Planned chapters: Concepts & Tools; System Architecture; Windows Application Support; Windows Store Apps; Graphics & the Desktop; Management Mechanisms; User Mode Memory Management; Security; Storage; Networking; Hyper-V.

What Makes It Page?

"Windows NT File System Internals" examines the NT/IO Manager, the Cache Manager, and the Memory Manager from the perspective of a software developer

writing a file system driver or implementing a kernel-mode filter driver. The book provides numerous code examples, as well as the source for a complete, usable filter driver.

Developing Windows NT Device Drivers

Analyzing how hacks are done, so as to stop them in thefuture Reverse engineering is the process of analyzing hardware orsoftware and understanding it, without having access to the sourcecode or design documents. Hackers are able to reverse engineersystems and exploit what they find with scary results. Now the goodguys can use the same tools to thwart these threats. PracticalReverse Engineering goes under the hood of reverse engineeringfor security analysts, security engineers, and system programmers,so they can learn how to use these same processes to stop hackersin their tracks. The book covers x86, x64, and ARM (the first book to cover allthree); Windows kernel-mode code rootkits and drivers; virtualmachine protection techniques; and much more. Best of all, itoffers a systematic approach to the material, with plenty ofhands-on exercises and real-world examples. Offers a systematic approach to understanding reverseengineering, with hands-on exercises and real-world examples Covers x86, x64, and advanced RISC machine (ARM) architecturesas well as deobfuscation and virtual machine protectiontechniques Provides special coverage of Windows kernel-mode code(rootkits/drivers), a topic not often covered elsewhere, andexplains how to analyze drivers step

Page 11/27

by step Demystifies topics that have a steep learning curve Includes a bonus chapter on reverse engineering tools Practical Reverse Engineering: Using x86, x64, ARM, WindowsKernel, and Reversing Tools provides crucial, up-to-dateguidance for a broad range of IT professionals.

Old New Thing

Use Windows debuggers throughout the development cycle—and build better software Rethink your use of Windows debugging and tracing tools—and learn how to make them a key part of test-driven software development. Led by a member of the Windows Fundamentals Team at Microsoft, you'll apply expert debugging and tracing techniques—and sharpen your C++ and C# code analysis skills—through practical examples and common scenarios. Learn why experienced developers use debuggers in every step of the development process, and not just when bugs appear. Discover how to: Go behind the scenes to examine how powerful Windows debuggers work Catch bugs early in the development cycle with static and runtime analysis tools Gain practical strategies to tackle the most common code defects Apply expert tricks to handle user-mode and kernel-mode debugging tasks Implement postmortem techniques such as JIT and dump debugging Debug the concurrency and security aspects of your software Use debuggers to analyze interactions between your code and the operating system Analyze software behavior with Xperf and the Event Tracing for Windows (ETW) framework

The Windows 2000 Device Driver Book

A guide to Windows 2000 Server technology offers IT professionals solutions and strategies for managing installation, account administration, file and print configuration, security, maintenance, back-up, and troubleshooting

Writing Windows VxDs and Device Drivers

"Raymond Chen is the original raconteur of Windows." --Scott Hanselman, ComputerZen.com "Raymond has been at Microsoft for many years and has seen many nuances of Windows that others could only ever hope to get a glimpse of. With this book, Raymond shares his knowledge, experience, and anecdotal stories, allowing all of us to get a better understanding of the operating system that affects millions of people every day. This book has something for everyone, is a casual read, and I highly recommend it!" --Jeffrey Richter, Author/Consultant, Cofounder of Wintellect "Very interesting read. Raymond tells the inside story of why Windows is the way it is." --Eric Gunnerson, Program Manager, Microsoft Corporation "Absolutely essential reading for understanding the history of Windows, its intricacies and quirks, and why they came about." --Matt Pietrek, MSDN Magazine's Under the Hood Columnist "Raymond Chen has become something of a legend in the software industry, and in this book you'll discover why. From his high-level reminiscences on the design of the Windows Start button to his low-level discussions of GlobalAlloc that

only your inner-geek could love, The Old New Thing is a captivating collection of anecdotes that will help you to truly appreciate the difficulty inherent in designing and writing quality software." --Stephen Toub, Technical Editor, MSDN Magazine Why does Windows work the way it does? Why is Shut Down on the Start menu? (And why is there a Start button, anyway?) How can I tap into the dialog loop? Why does the GetWindowText function behave so strangely? Why are registry files called "hives"? Many of Windows' quirks have perfectly logical explanations, rooted in history. Understand them, and you'll be more productive and a lot less frustrated. Raymond Chen--who's spent more than a decade on Microsoft's Windows development team--reveals the "hidden Windows" you need to know. Chen's engaging style, deep insight, and thoughtful humor have made him one of the world's premier technology bloggers. Here he brings together behind-the-scenes explanations, invaluable technical advice, and illuminating anecdotes that bring Windows to life--and help you make the most of it. A few of the things you'll find inside: What vending machines can teach you about effective user interfaces A deeper understanding of window and dialog management Why performance optimization can be so counterintuitive A peek at the underbelly of COM objects and the Visual C++ compiler Key details about backwards compatibility--what Windows does and why Windows program security holes most developers don't know about How to make your program a better Windows citizen

The Windows NT Device Driver Book

The Definitive Guide to Windows API Programming, Fully Updated for Windows 7, Windows Server 2008, and Windows Vista Windows System Programming, Fourth Edition, now contains extensive new coverage of 64-bit programming, parallelism, multicore systems, and many other crucial topics. Johnson Hart's robust code examples have been updated and streamlined throughout. They have been debugged and tested in both 32-bit and 64-bit versions, on single and multiprocessor systems, and under Windows 7, Vista, Server 2008, and Windows XP. To clarify program operation, sample programs are now illustrated with dozens of screenshots. Hart systematically covers Windows externals at the API level, presenting practical coverage of all the services Windows programmers need, and emphasizing how Windows functions actually behave and interact in real-world applications. Hart begins with features used in single-process applications and gradually progresses to more sophisticated functions and multithreaded environments. Topics covered include file systems, memory management, exceptions, processes, threads, synchronization, interprocess communication, Windows services, and security. New coverage in this edition includes Leveraging parallelism and maximizing performance in multicore systems Promoting source code portability and application interoperability across Windows, Linux, and UNIX Using 64-bit address spaces and ensuring 64-bit/32-bit portability Improving performance and scalability using threads, thread pools, and

completion ports Techniques to improve program reliability and performance in all systems Windows performance-enhancing API features available starting with Windows Vista, such as slim reader/writer locks and condition variables A companion Web site, jmhartsoftware.com, contains all sample code, Visual Studio projects, additional examples, errata, reader comments, and Windows commentary and discussion.

Showstopper!

A complete reference on using and programming the Win32 Driver Model describes how it communicates with PC peripherals, as well as its efficiency benefits in device support and development, and features a CD-ROM with sample code and portions of the WDM Device Driver Kit. Original. (Advanced).

Systems Programming for Windows 95

The Microsoft® Windows® driver model (WDM) supports Plug and Play, provides power management capabilities, and expands on the driver/minidriver approach. Written by long-time device-driver expert Walter Oney in cooperation with the Windows kernel team, this book provides extensive practical examples, illustrations, advice, and line-by-line analysis of code samples to clarify real-world driver-programming issues. And it's been updated with the latest details about the driver technologies in Windows XP and Windows 2000, plus more information about how to debug drivers. Topics

covered include: Beginning a driver project and the structure of a WDM driver; NEW: Minidrivers and class drivers, driver taxonomy, the WDM development environment and tools, management checklist, driver selection and loading, approved API calls, and driver stacks Basic programming techniques; NEW: Safe string functions, memory limits, the Driver Verifier scheme and tags, the kernel handle flag, and the Windows 98 floating-point problem Synchronization; NEW: Details about the interrupt request level (IRQL) scheme, along with Windows 98 and Windows Me compatibility The I/O request packet (IRP) and I/O control operations; NEW: How to send control operations to other drivers, custom queue implementations, and how to handle and safely cancel IRPs Plug and Play for function drivers; NEW: Controller and multifunction devices, monitoring device removal in user mode, Human Interface Devices (HID), including joysticks and other game controllers, minidrivers for non-HID devices, and feature reports Reading and writing data, power management, and Windows Management Instrumentation (WMI) NEW: System wakeup, the WMI control for idle detection, and using WMIMOFCK Specialized topics and distributing drivers; NEW: USB 2.0, selective suspend, Windows Hardware Quality Lab (WHQL) certification, driver selection and loading, officially approved API calls, and driver stacks COVERS WINDOWS 98, WINDOWS ME, WINDOWS 2000, AND WINDOWS XP! CD-ROM FEATURES: A fully searchable electronic copy of the book Sample code in Microsoft Visual C++® A Note Regarding the CD or DVD The print version of this book ships with a CD or DVD. For those customers purchasing one of the

digital formats in which this book is available, we are pleased to offer the CD/DVD content as a free download via O'Reilly Media's Digital Distribution services. To download this content, please visit O'Reilly's web site, search for the title of this book to find its catalog page, and click on the link below the cover image (Examples, Companion Content, or Practice Files). Note that while we provide as much of the media content as we are able via free download, we are sometimes limited by licensing restrictions. Please direct any questions or concerns to booktech@oreilly.com.

FreeBSD Architecture Handbook

Windows Kernel Programming

The object of this book is to cover most of the currently relevant areas of data communications and networks. These include: Communications protocols (especially TCP/IP) Networking (especially in Ethernet, Fast Ethernet, FDDI and ATM) Networking operating systems (especially in Windows NT, Novell NetWare and UNIX) Communications programs (especially in serial communications, parallel communications and TCP/IP) Computer hardware (especially in PC hardware, serial communications and parallel communication) The book thus splits into 15 different areas, these are: General data compression (Chapters 2 and 3) Video, images and sound (Chapters 4-11) Error coding and encryption (Chapters 12-17) TCP/IP, WWW, Internets and Intranets (Chapters 18-20 and

23) Electronic Mail (Chapter 21) HTML (Chapters 25 and 26) Java (Chapters 27-29) Communication Programs (Chapters 20, 29 and 49) Network Operating Systems (Chapters 31-34) LANs/WANs (Chapters 35, 38-46) Serial Communications (Chapters 47 and 48) Parallel Communications (Chapters 50-52) Local Communications (Chapters 53-57) Routing and Protocols (Chapters 36 and 37) Cables and connectors (Chapters 58--60) Many handbooks and reference guides on the market contain endless tables and mathematics, or are dry to read and contain very little insight in their subject area. I have tried to make this book readable, but also contain key information which can be used by professionals.

Professional Rootkits

Win32 Programming

Since its introduction over a decade ago, the Microsoft SQL Server query language, Transact-SQL, has become increasingly popular and more powerful. The current version sports such advanced features as OLE Automation support, cross-platform querying facilities, and full-text search management. This book is the consummate guide to Microsoft Transact-SQL. From data type nuances to complex statistical computations to the bevy of undocumented features in the language, The Guru's Guide to Transact-SQL imparts the knowledge you need to become a virtuoso of the language as quickly as possible. In this

book, you will find the information, explanations, and advice you need to master Transact-SQL and develop the best possible Transact-SQL code. Some 600 code examples not only illustrate important concepts and best practices, but also provide working Transact-SQL code that can be incorporated into your own real-world DBMS applications. Your journey begins with an introduction explaining language fundamentals such as database and table creation, inserting and updating data, queries, joins, data presentation, and managing transactions. Moving on to more advanced topics, the journey continues with in-depth coverage of: Transact-SQL performance tuning using tools such as Query Analyzer and Performance Monitor Nuances of the various T-SQL data types Complex statistical calculations such as medians, modes, and sliding aggregates Run, sequence, and series identification and interrogation Advanced Data Definition Language (DDL) and Data Management Language (DML) techniques Stored procedure and trigger best practices and coding methods Transaction management Optimal cursor use and caveats to look out for Full-text search Hierarchies and arrays Administrative Transact-SQL OLE Automation More than 100 undocumented commands and language features, including numerous unpublished DBCC command verbs, trace flags, stored procedures, and functions Comprehensive, written in understandable terms, and full of practical information and examples, The Guru's Guide to Transact-SQL is an indispensable reference for anyone working with this database development language. The accompanying CD-ROM includes the complete set of code examples found in the book as well as a SQL programming environment

that will speed the development of your own top-notch Transact-SQL code.

Windows Developer's Journal

PLEASE PROVIDE DESCRIPTION

The Windows 2000 Device Driver Book

The definitive guide–fully updated for Windows 10 and Windows Server 2016 Delve inside Windows architecture and internals, and see how core components work behind the scenes. Led by a team of internals experts, this classic guide has been fully updated for Windows 10 and Windows Server 2016. Whether you are a developer or an IT professional, you'll get critical, insider perspectives on how Windows operates. And through hands-on experiments, you'll experience its internal behavior firsthand–knowledge you can apply to improve application design, debugging, system performance, and support. This book will help you: · Understand the Window system architecture and its most important entities, such as processes and threads · Examine how processes manage resources and threads scheduled for execution inside processes · Observe how Windows manages virtual and physical memory · Dig into the Windows I/O system and see how device drivers work and integrate with the rest of the system · Go inside the Windows security model to see how it manages access, auditing, and authorization, and learn about the new mechanisms in Windows 10 and Server 2016

Writing MS-DOS Device Drivers

Developing Windows NT Device Drivers: A Programmer's Handbookoffers programmers a comprehensive and in-depth guide to building device drivers for Windows NT. Written by two experienced driver developers, Edward N. Dekker and Joseph M. Newcomer, this book provides detailed coverage of techniques, tools, methods, and pitfalls to help make the often complex and byzantine "black art" of driver development straightforward and accessible. This book is designed for anyone involved in the development of Windows NT Device Drivers, particularly those working on drivers for nonstandard devices that Microsoft has not specifically supported. Because Windows NT does not permit an application program to directly manipulate hardware, a customized kernel mode device driver must be created for these nonstandard devices. And since experience has clearly shown that superficial knowledge can be hazardous when developing device drivers, the authors have taken care to explore each relevant topic in depth. This book's coverage focuses on drivers for polled, programmed I/O, interrupt-driven, and DMA devices. The authors discuss the components of a kernel mode device driver for Windows NT, including background on the two primary bus interfaces used in today's computers: the ISA and PCI buses. Developers will learn the mechanics of compilation and linking, how the drivers register themselves with the system, experience-based techniques for debugging, and how to build robust, portable, multithread- and multiprocessor-safe

device drivers that work as intended and won't crash the system. The authors also show how to call the Windows NT kernel for the many services required to support a device driver and demonstrate some specialized techniques, such as mapping device memory or kernel memory into user space. Thus developers will not only learn the specific mechanics of high-quality device driver development for Windows NT, but will gain a deeper understanding of the foundations of device driver design.

Inside Windows NT

The first authoritative guide to programming Windows 7 device drivers: save time, save money, and write more reliable drivers • •Shows experienced programmers how to make the most of Microsoft's latest and most powerful models and tools for Windows 7 driver development, including C# and Visual Studio 2010. •Thoroughly covers Microsoft's Windows Driver Foundation (WDF) Architecture. •There are no other books, and little information anywhere, about Windows 7 device drivers. This is the only comprehensive, state-of-the-art guide to writing Windows 7 device drivers. Written by Ronald Reeves, one of the field's leading experts, Windows 7 Device Driver Book helps experienced developers make the most of the powerful new tools and models Microsoft has made available for driver development. Reeves provides an extensive collection of sample code on CDROM, as well as best-practice guidance for maximizing reliability and performance. Reeves shows how C# and Visual Studio 2010 can be used to

develop device drivers more rapidly, and debug them more effectively. He covers a broad array of topics, including both kernel- and user-mode driver development; Windows Driver Foundation (WDF) architecture, and much more. From start to finish, this book is designed to significantly reduce the time it takes for device driver programmers to find the information they need -- and then apply that information in reliable, production code.

Inside Windows Server 2003

Master the new Windows Driver Model (WDM) common to Windows 98 and Windows 2000. You get theory, instruction and practice in driver development, installation and debugging. Addresses hardware and software interface issues, driver types, and a description of the new 'layer' model of WDM. ;

Windows Internals

An exhaustive technical manual outlines the Windows NT concepts related to drivers; shows how to develop the best drivers for particular applications; covers the I/O Subsystem and implementation of standard kernel mode drivers; and more. Original. (Intermediate).

Windows NT Device Driver Development

An authoritative guide to Windows NT driver development, now completely revised and updated. The CD-ROM includes all source code, plus Microsoft hardware standards documents, demo software, and

more.

Windows 7 Device Driver

Master the new Windows Driver Model (WDM) common to Windows 98 and Windows 2000. You get theory, instruction and practice in driver development, installation and debugging. Addresses hardware and software interface issues, driver types, and a description of the new 'layer' model of WDM. ;

Inside Windows Storage

A comprehensive manual for deploying and administering Windows .NET Server 2003 furnishes detailed coverage of all aspects of .NET Server, including its more than two hundred new features, along with thousands of tips and recommendations, real-world solutions and guidance, and tips on design, installation, configuration, and more. Original. (Advanced)

Handbook of Data Communications and Networks

Software developer and author Karen Hazzah expands her original treatise on device drivers in the second edition of Writing Windows VxDs and Device Drivers. The book and companion disk include the author's library of wrapper functions that allow the progr

DTrace

Start developing robust drivers with expert guidance from the teams who developed Windows Driver Foundation. This comprehensive book gets you up to speed quickly and goes beyond the fundamentals to help you extend your Windows development skills. You get best practices, technical guidance, and extensive code samples to help you master the intricacies of the next-generation driver model—and simplify driver development. Discover how to: Use the Windows Driver Foundation to develop kernel-mode or user-mode drivers Create drivers that support Plug and Play and power management—with minimal code Implement robust I/O handling code Effectively manage synchronization and concurrency in driver code Develop user-mode drivers for protocol-based and serial-bus-based devices Use USB-specific features of the frameworks to quickly develop drivers for USB devices Design and implement kernel-mode drivers for DMA devices Evaluate your drivers with source code analysis and static verification tools Apply best practices to test, debug, and install drivers PLUS—Get driver code samples on the Web

ROMANCE ACTION & ADVENTURE MYSTERY & THRILLER BIOGRAPHIES & HISTORY CHILDREN'S YOUNG ADULT FANTASY HISTORICAL FICTION HORROR LITERARY FICTION NON-FICTION SCIENCE FICTION