

<<深入理解Linux内核>>

图书基本信息

书名：<<深入理解Linux内核>>

13位ISBN编号：9787564102760

10位ISBN编号：7564102764

出版时间：2006-4-1

出版时间：东南大学出版社

作者：Daniel P.Bovet,Marco Cesati

页数：923

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>

<<深入理解Linux内核>>

内容概要

为了彻底理解是什么使得Linux能正常运行以及其为何能在各种不同的系统中运行良好，你需要深入研究内核最本质的部分。

内核处理CPU与外界间的所有交互，并且决定哪些程序将以什么顺序共享处理器时间。

它如此有效地管理有限的内存，以至成百上千的进程能高效地共享系统。

它熟练地统筹数据传输，这样CPU不用为等待速度相对较慢的硬盘而消耗比正常耗时更长的时间。

《深入理解Linux内核，第三版》指导你对内核中使用的最重要的数据结构、算法和程序设计诀窍进行一次遍历。

通过对表面特性的探究，作者给那些想知道自己机器工作原理的人提供了颇有价值的见解。

书中讨论了Intel特有的重要性质。

相关的代码片段被逐行剖析。

然而，本书涵盖的不仅仅是代码的功能，它解释了Linux以自己的方式工作的理论基础。

本书将使你了解Linux的所有内部工作，它不仅仅是一个理论上的练习。

你将学习到哪些情况下Linux性能最佳，并且你将看到，在大量的不同环境里进行进程调度、文件存取和内存管理时，它如何满足提供良好的系统响应的需要。

这本书将帮助你充分利用Linux系统。

<<深入理解Linux内核>>

作者简介

作者：(意)Daniel P.Bovet (意)Marco Cesati

书籍目录

Preface The Audience for This Book Organization of the Material Level of Description Overview of the Book
Background Information Conventions in This Book How to Contact Us Safari? Enabled Acknowledgments
Chapter 1. Introduction Section 1.1. Linux Versus Other Unix-Like Kernels Section 1.2. Hardware Dependency
Section 1.3. Linux Versions Section 1.4. Basic Operating System Concepts Section 1.5. An Overview of the Unix
Filesystem Section 1.6. An Overview of Unix Kernels Chapter 2. Memory Addressing Section 2.1. Memory
Addresses Section 2.2. Segmentation in Hardware Section 2.3. Segmentation in Linux Section 2.4. Paging in
Hardware Section 2.5. Paging in Linux Chapter 3. Processes Section 3.1. Processes, Lightweight Processes, and
Threads Section 3.2. Process Descriptor Section 3.3. Process Switch Section 3.4. Creating Processes Section 3.5.
Destroying Processes Chapter 4. Interrupts and Exceptions Section 4.1. The Role of Interrupt Signals Section 4.2.
Interrupts and Exceptions Section 4.3. Nested Execution of Exception and Interrupt Handlers Section 4.4.
Initializing the Interrupt Descriptor Table Section 4.5. Exception Handling Section 4.6. Interrupt Handling
Section 4.7. Softirqs and Tasklets Section 4.8. Work Queues Section 4.9. Returning from Interrupts and
Exceptions Chapter 5. Kernel Synchronization Section 5.1. How the Kernel Services Requests Section 5.2.
Synchronization Primitives Section 5.3. Synchronizing Accesses to Kernel Data Structures Section 5.4. Examples
of Race Condition Prevention Chapter 6. Timing Measurements Section 6.1. Clock and Timer Circuits Section
6.2. The Linux Timekeeping Architecture Section 6.3. Updating the Time and Date Section 6.4. Updating System
Statistics Section 6.5. Software Timers and Delay Functions Section 6.6. System Calls Related to Timing
Measurements Chapter 7. Process Scheduling Section 7.1. Scheduling Policy Section 7.2. The Scheduling
Algorithm Section 7.3. Data Structures Used by the Scheduler Section 7.4. Functions Used by the Scheduler
Section 7.5. Runqueue Balancing in Multiprocessor Systems Section 7.6. System Calls Related to Scheduling
Chapter 8. Memory Management Section 8.1. Page Frame Management Section 8.2. Memory Area Management
Section 8.3. Noncontiguous Memory Area Management Chapter 9. Process Address Space Section 9.1. The
Process's Address Space Section 9.2. The Memory Descriptor Section 9.3. Memory Regions Section 9.4. Page
Fault Exception Handler Section 9.5. Creating and Deleting a Process Address Space Section 9.6. Managing the
Heap Chapter 10. System Calls Section 10.1. POSIX APIs and System Calls Section 10.2. System Call Handler
and Service Routines Section 10.3. Entering and Exiting a System Call Section 10.4. Parameter Passing Section
10.5. Kernel Wrapper Routines Chapter 11. Signals Section 11.1. The Role of Signals Section 11.2. Generating a
Signal Section 11.3. Delivering a Signal Section 11.4. System Calls Related to Signal Handling Chapter 12. The
Virtual Filesystem Section 12.1. The Role of the Virtual Filesystem (VFS) Section 12.2. VFS Data Structures
Section 12.3. Filesystem Types Section 12.4. Filesystem Handling Section 12.5. Pathname Lookup Section 12.6.
Implementations of VFS System Calls Section 12.7. File Locking Chapter 13. I/O Architecture and Device Drivers
Section 13.1. I/O Architecture Section 13.2. The Device Driver Model Section 13.3. Device Files Section 13.4.
Device Drivers Section 13.5. Character Device Drivers Chapter 14. Block Device Drivers Section 14.1. Block
Devices Handling Section 14.2. The Generic Block Layer Section 14.3. The I/O Scheduler Section 14.4. Block
Device Drivers Section 14.5. Opening a Block Device File Chapter 15. The Page Cache Section 15.1. The Page
Cache Section 15.2. Storing Blocks in the Page Cache Section 15.3. Writing Dirty Pages to Disk Section 15.4. The
sync(), fsync(), and fdatasync() System Calls Chapter 16. Accessing Files Section 16.1. Reading and Writing a
File Section 16.2. Memory Mapping Section 16.3. Direct I/O Transfers Section 16.4. Asynchronous I/O Chapter
17. Page Frame Reclaiming Section 17.1. The Page Frame Reclaiming Algorithm Section 17.2. Reverse Mapping
Section 17.3. Implementing the PFRA Section 17.4. Swapping Chapter 18. The Ext2 and Ext3 Filesystems Section
18.1. General Characteristics of Ext2 Section 18.2. Ext2 Disk Data Structures Section 18.3. Ext2 Memory Data
Structures Section 18.4. Creating the Ext2 Filesystem Section 18.5. Ext2 Methods Section 18.6. Managing Ext2
Disk Space Section 18.7. The Ext3 Filesystem Chapter 19. Process Communication Section 19.1. Pipes Section
19.2. FIFOs Section 19.3. System V IPC Section 19.4. POSIX Message Queues Chapter 20. Program Execution
Section 20.1. Executable Files Section 20.2. Executable Formats Section 20.3. Execution Domains Section 20.4.

The exec Functions Appendix A. System Startup Section A.1. Prehistoric Age: the BIOS Section A.2. Ancient Age: the Boot Loader Section A.3. Middle Ages: the setup() Function Section A.4. Renaissance: the startup_32() Functions Section A.5. Modern Age: the start_kernel() Function Appendix B. Modules Section B.1. To Be (a Module) or Not to Be? Section B.2. Module Implementation Section B.3. Linking and Unlinking Modules Section B.4. Linking Modules on Demand Bibliography Books on Unix Kernels Books on the Linux Kernel Books on PC Architecture and Technical Manuals on Intel Microprocessors Other Online Documentation Sources Research Papers Related to Linux Development About the Authors Colophon Index

<<深入理解Linux内核>>

编辑推荐

为了彻底理解是什么使得Linux能正常运行以及其为何能在各种不同的系统中运行良好，你需要深入研究内核最本质的部分。

内核处理CPU与外界间的所有交互，并且决定哪些程序将以什么顺序共享处理器时间。

它如此有效地管理有限的内存，以至成百上千的进程能高效地共享系统。

它熟练地统筹数据传输，这样CPU不用为等待速度相对较慢的硬盘而消耗比正常耗时更长的时间。

《深入理解Linux内核，第三版》指导你对内核中使用的最重要的数据结构、算法和程序设计诀窍进行一次遍历。

通过对表面特性的探究，作者给那些想知道自己机器工作原理的人提供了颇有价值的见解。

书中讨论了Intel特有的重要性质。

相关的代码片段被逐行剖析。

然而，本书涵盖的不仅仅是代码的功能，它解释了Linux以自己的方式工作的理论基础。

本书将使你了解Linux的所有内部工作，它不仅仅是一个理论上的练习。

你将学习到哪些情况下Linux性能最佳，并且你将看到，在大量的不同环境里进行进程调度、文件存取和内存管理时，它如何满足提供良好的系统响应的需要。

这本书将帮助你充分利用Linux系统。

<<深入理解Linux内核>>

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>