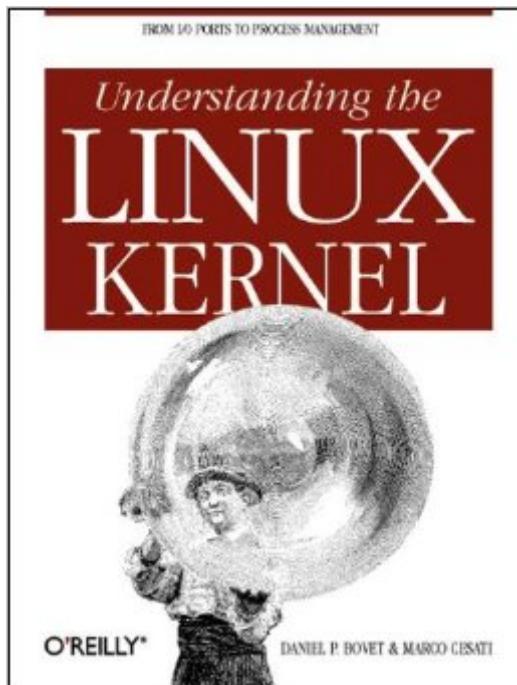


The book was found

Understanding The LINUX Kernel: From I/O Ports To Process Management



Synopsis

Why is Linux so efficient? Is it the right operating system for a particular application? What can be learned from looking at the kernel source code? These are the kinds of questions that Understanding the Linux Kernel takes in stride in this guided tour of the code that forms the core of all Linux operating systems. Linux is presented too often as a casual hacker experiment. It has increasingly become not only a mission-critical part of many organizations, but a sophisticated display of programming skill. It incorporates many advanced operating system concepts and has proven itself extremely robust and efficient for a wide range of uses. Understanding the Linux Kernel helps readers understand how Linux performs best and how it meets the challenge of different environments. The authors introduce each topic by explaining its importance, and show how kernel operations relate to the utilities that are familiar to Unix programmers and users. Major topics include:

- Memory management, including file buffering, process swapping, and Direct Memory Access (DMA)
- The Virtual File System and the Second Extended File System
- Process creation and scheduling
- Signals, interrupts, and the essential interfaces to device drivers
- Timing and Synchronization in the kernel
- Inter-Process Communication (IPC)
- Program execution

Book Information

Paperback: 685 pages

Publisher: O'Reilly Media; 1 edition (October 11, 2000)

Language: English

ISBN-10: 0596000022

ISBN-13: 978-0596000028

Product Dimensions: 7 x 1.3 x 9.2 inches

Shipping Weight: 2.2 pounds

Average Customer Review: 4.1 out of 5 stars [See all reviews](#) (18 customer reviews)

Best Sellers Rank: #3,227,024 in Books (See Top 100 in Books) #8 in Books > Computers & Technology > Operating Systems > Linux > Kernel & Peripherals #505 in Books > Computers & Technology > Operating Systems > Linux > Programming #910 in Books > Computers & Technology > Operating Systems > Unix

Customer Reviews

Can a professional OS book feature both easy to read and detailed/accurate in technical level? The answer is 'yes' for this book. I have been looking for this kind of book for years. Now I have it. This book delayed publishing for 3 months, the waiting is worthwhile. I spent 3 weeks to read this book

very carefully. I got tons of my uncertain question marks about kernel in mind answered. I have explored the kernel for years. I should say, I have browsed almost all documents/books I could find from websites and bookstores. This book is the best one (not one of the best) in this topic I could find on this planet. Besides the 'traditional' subjects, like, process, memory, interrupt, signals, IPC, it also depicts 'file system' in very details which is rarely done by other kernel books. The most outstanding trait is that it is so 'user friendly' for you to read. Its style always remind me of my own jotting in my note book, but of course, it is much more complete, accurate and super well organised. Its narration is brief and simple enough for you to understand and remember; meanwhile, it never loses the technical details and accuracy for your further adventure. If there is any 'complaints', I should say, I like to see some general I/O subjects, like, tty console, display, network and etc, to be discussed, maybe in another book. However, I'd like to say that this book is a great contribution to linux kernel community. Thanks authors, this book deserves a 6 stars mark!

A pretty handy book - goes well with the O'Reilly book on linux device drivers. Covers memory management well, although it can be a bit confusing on the difference between Intel memory segmentation and kernel Paging. Good attention paid to forward looking differences between the 2.2 and 2.4 kernels. It would be nice to see the next revision deal exclusively with 2.4 (lets hope :)). All in all, a worthwhile book to have on your shelf!

This book is probably the best book around, but it's still not what it should be. The authors emphasize a lot on assembly language but never bother to explain the big picture. What might have helped is a short chapter on C and assembly language interfacing, and just a road map of function calls with a detailed description of data structures, locking, and deadlock conditions. More emphasis should have been given to tricky things like bh, tasklets, and softirqs (which is absent). At many places the book is ambiguous. For example, if process A is running when an interrupt comes that will eventually wake up process B -- which kernel stack does this interrupt use? A or B. Well not too difficult to figure out, but the book should point these little things out rather than making general statements like "the IF flags are saved on the stack" -- everyone knows its saved on the stack, but which one? There is no shortcut to reading the source code so there is no point in explaining one zillion times that mov a, b will move a to b. BTW the author never explains various things that gcc and ld implicitly do to the final image (e.g., how is the function table for do_initcalls created and populated and why does the order of linking change the initialization process. etc etc) I regret buying the book and I wish I had spent my time and money on grepping and buying coffee. Read the

DJASM guide to gcc and assembly and use any source navigator to browse through the source. Its far simpler that way -- and you are up to date with the kernel releases. happy hacking!

I just recently started working on the Linux kernel and this book helped get me off the ground and going. I do wish it had more depth and detail in some areas, but for the price I don't think you can go wrong. For example, I needed much more detail in the area of multiprocessing than this book provided. An excellent companion book is "UNIX Systems for Modern Architectures: Symmetric Multiprocessing and Caching for Kernel Programmers" by Curt Schimmel. This latter book provided the extra background and depth for me to finally understand how all the pieces fit together in this area.

In reference to some comment made by another reader I just want to point out that kernel stack switch happens in the middle of `_switch_to` macro as described in the para -4 of chapter-3(Pg-87). I don't want to start a war here but I would like to say to reviewers to kindly read carefully before blaming a book. I guess the right way to understand the kernel is download the intel programmer's guide (80386) and the kernel source code for linux version 2.2.14. Read the gates and segments section from intel, you may skip the task switching part for linux does not use the task switch of intel. All task switch is done in software. Dig in the source code along with the book. But don't confuse between versions. Because the way fast and slow intrs were handled before are different than that in 2.2.*. So in a way the device driver book by Rubini's interrupt handling section is not in accordance with this book. But this book's way is the one as per linux version 2.2 which is later than what Rubini's book is based on 2.0. I owe many thanks to the writers of this book who made an excellent job.

[Download to continue reading...](#)

LINUX: Linux Command Line, Cover all essential Linux commands. A complete introduction to Linux Operating System, Linux Kernel, For Beginners, Learn Linux in easy steps, Fast! A Beginner's Guide Understanding the LINUX Kernel: From I/O Ports to Process Management Linux: Linux Guide for Beginners: Command Line, System and Operation (Linux Guide, Linux System, Beginners Operation Guide, Learn Linux Step-by-Step) Learning Linux Kernel - Process management and scheduling (Japanese Edition) Linux: Linux Mastery. The Ultimate Linux Operating System and Command Line Mastery (Operating System, Linux) uC/OS-III, The Real-Time Kernel, or a High Performance, Scalable, ROMable, Preemptive, Multitasking Kernel for Microprocessors, Microcontrollers & DSPs (Board NOT Included) Kernel of the Kernel (Suny Series in Islam)

Understanding the Linux Kernel, Third Edition Linux Kernel Programming (3rd Edition) Professional Linux Kernel Architecture Linux Kernel Internals (2nd Edition) Linux Kernel Development (2nd Edition) The Linux Kernel Book Linux System Programming: Talking Directly to the Kernel and C Library Linux Kernel Development (3rd Edition) National Geographic Traveler: The Mediterranean: Ports of Call and Beyond Cruising the Mediterranean: A Guide to the Ports of Call (Cruising the Mediterranean) Cruising the Southern and Western Caribbean: A Guide to the Ships & the Ports of Call (Cruising the Southern and Western Caribbean) (Cruising the Southern & Western Caribbean) Understanding Bergson, Understanding Modernism (Understanding Philosophy, Understanding Modernism) Linux: Linux Bash Scripting - Learn Bash Scripting In 24 hours or less

[Dmca](#)