The Design And Implementation Of The FreeBSD Operating System (2nd Edition)
The most complete, authoritative technical guide to the FreeBSD kernel’s internal structure has now been extensively updated to cover all major improvements between Versions 5 and 11. Approximately one-third of this edition’s content is completely new, and another one-third has been extensively rewritten. Three long-time FreeBSD project leaders begin with a concise overview of the FreeBSD kernel’s current design and implementation. Next, they cover the FreeBSD kernel from the system-call level down—from the interface to the kernel to the hardware. Explaining key design decisions, they detail the concepts, data structures, and algorithms used in implementing each significant system facility, including process management, security, virtual memory, the I/O system, filesystems, socket IPC, and networking. This Second Edition Explains highly scalable and lightweight virtualization using FreeBSD jails, and virtual-machine acceleration with Xen and Virtio device paravirtualization. Describes new security features such as Capsicum sandboxing and GELI cryptographic disk protection. Fully covers NFSv4 and Open Solaris ZFS support. Introduces FreeBSD’s enhanced volume management and new journaled soft updates. Explains DTrace’s fine-grained process debugging/profiling. Reflects major improvements to networking, wireless, and USB support. Readers can use this guide as both a working reference and an in-depth study of a leading contemporary, portable, open source operating system. Technical and sales support professionals will discover both FreeBSD’s capabilities and its limitations. Applications developers will learn how to effectively and efficiently interface with it; system administrators will learn how to maintain, tune, and configure it; and systems programmers will learn how to extend, enhance, and interface with it. Marshall Kirk McKusick writes, consults, and teaches classes on UNIX- and BSD-related subjects. While at the University of California, Berkeley, he implemented the 4.2BSD fast filesystem. He was research computer scientist at the Berkeley Computer Systems Research Group (CSRG), overseeing development and release of 4.3BSD and 4.4BSD. He is a FreeBSD Foundation board member and a long-time FreeBSD committer. Twice president of the Usenix Association, he is also a member of ACM, IEEE, and AAAS. George V. Neville-Neil hacks, writes, teaches, and consults on security, networking, and operating systems. A FreeBSD Foundation board member, he served on the FreeBSD Core Team for four years. Since 2004, he has written the Kode Vicious column for Queue and Communications of the ACM. He is vice chair of ACM’s Practitioner Board and a member of Usenix Association, ACM, IEEE, and AAAS. Robert N.M. Watson is a University Lecturer in systems, security, and architecture in the Security Research Group at the University of Cambridge Computer Laboratory. He supervises advanced research in
computer architecture, compilers, program analysis, operating systems, networking, and security. A FreeBSD Foundation board member, he served on the Core Team for ten years and has been a committer for fifteen years. He is a member of Usenix Association and ACM.

Book Information

Hardcover: 928 pages
Publisher: Addison-Wesley Professional; 2 edition (September 15, 2014)
Language: English
ISBN-10: 0321968972
Product Dimensions: 6.4 x 1.4 x 9.3 inches
Shipping Weight: 2.7 pounds (View shipping rates and policies)
Average Customer Review: 5.0 out of 5 stars See all reviews (23 customer reviews)
Best Sellers Rank: #95,253 in Books (See Top 100 in Books) #1 in Books > Computers & Technology > Operating Systems > BSD #15 in Books > Computers & Technology > Programming > APIs & Operating Environments > Operating Systems Theory #33 in Books > Computers & Technology > Operating Systems > Unix

Customer Reviews

This book is beautifully done! Bravo! The way I see it, the kernel has two faces- one close to the hardware, and the other close to the person writing an application using system calls. And a big chunk in between. Here’s the ? If FreeBSD and its beautiful sister PC-BSD boot up into zfs, zfs is in the kernel, it’s the mounted filesystem, why only 30-something pages in this edition on zfs ? This is a kernel book. If someone is writing system calls to work on or with the file system, they’re working on the face of the kernel close to them, isn’t that the face zfs presents itself via system calls to a programmer? Or is zfs on the face closest to the hardware? Not clear on this myself. I am still trying to understand what is being said in Chapter 7, but do see that excellent kernel diagram at the start of Chapter 7! I was hoping there would be more of a description of how zfs actually stitched into the kernel, on both faces and in the big chunk in the middle. It’s probably closer to the hardware than I imagine. As I posted on the PC-BSD forums, the on-disk specification of zfs is complex. I don’t see how the system calls you make have anything to do with zfs, unless I guess you are writing extensions to it. I would appreciate someone clarifying this issue for me, someone that does that kind of programming. ***It doesn’t seem as if the entire chapter, Chapter 9 on the Fast File System, is applicable any more to the two current BSD’s. Perhaps historically. Too bad they didn’t add those
Just as a postscript, the only two UNIX systems that ship with zfs in the kernel that I know of are FreeBSD/PCBSD and Oracle Solaris/OpenIndie. I know you can build it from openzfs source, into a Steve, Linus, or probably even Bill machine.

Download to continue reading...