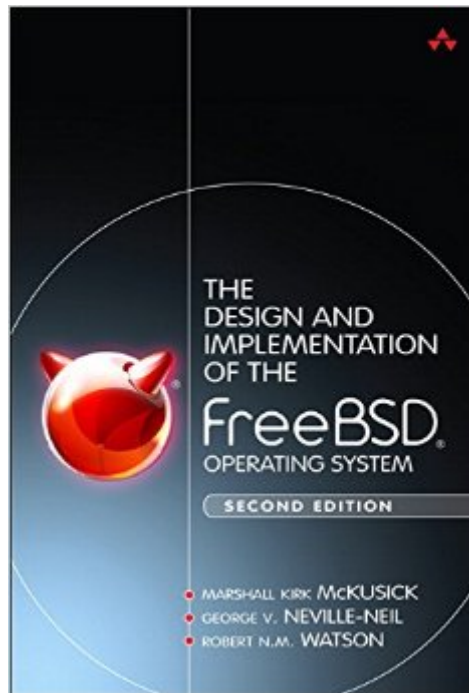# The Design And Implementation Of The FreeBSD Operating System (2nd Edition)

Synopsis

The most complete, authoritative technical guide to the FreeBSD kernelâ ™s internal structure has now been extensively updated to cover all major improvements between Versions 5 and 11. Approximately one-third of this editionâ ™s content is completely new, and another one-third has been extensively rewritten. Â Three long-time FreeBSD project leaders begin with a concise overview of the FreeBSD kernelâ ™s current design and implementation. Next, they cover the FreeBSD kernel from the system-call level downâ "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. Â This Second Edition Â â ¢ Explains highly scalable and lightweight virtualization using FreeBSD jails, and virtual-machine acceleration with Xen and Virtio device paravirtualization Â â ¢ Describes new security features such as Capsicum sandboxing and GELI cryptographic disk protection Â â ¢ Fully covers NFSv4 and Open Solaris ZFS support Â â ¢ Introduces FreeBSDâ ™s enhanced volume management and new journaled soft updates Â â ¢ Explains DTraceâ ™s fine-grained process debugging/profiling Â â ¢ Reflects major improvements to networking, wireless, and USB support Â 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 FreeBSDâ ™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. Â 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. Â 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 â œKode Viciousâ • column for Queue and Communications of the ACM. He is vice chair of ACMâ ™s Practitioner Board and a member of Usenix Association, ACM, IEEE, and AAAS. Â 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

ISBN-13: 978-0321968975

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 starsÂ Â See all reviewsÂ (23 customer reviews)

Best Sellers Rank: #95,253 in Books (See Top 100 in Books)   #1 inÂ Books > Computers & Technology > Operating Systems > BSD   #15 inÂ Books > Computers & Technology > Programming > APIs & Operating Environments > Operating Systems Theory   #33 inÂ 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 it's 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

pages to zfs explication.***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.

The Design and Implementation of the FreeBSD Operating System (2nd Edition) Absolute FreeBSD: The Complete Guide to FreeBSD Linux: Linux Mastery. The Ultimate Linux Operating System and Command Line Mastery (Operating System, Linux) The Design and Implementation of the 4.4 BSD Operating System (Addison-Wesley UNIX and Open Systems Series) Operating Systems Design and Implementation (3rd Edition) Create Your Own Operating System: Build, deploy, and test your very own operating systems for the Internet of Things and other devices Greenberg's Repair and Operating Manual for Lionel Trains, 1945-1969: 1945-1969 (Greenberg's Repair and Operating Manuals) Gilbert American Flyer S Gauge Operating & Repair Guide: Volume 2 (Gilbert American Flyer S Gauge Operating and Repair Guide) Instrumentation for the Operating Room: A Photographic Manual, 6e (Instrumentation for the Operating Room (Brooks-T)) Model Predictive Control System Design and Implementation Using MATLABÃ Â® (Advances in Industrial Control) Windows 8 Tips for Beginners 2nd Edition: A Simple, Easy, and Efficient Guide to a Complex System of Windows 8! (Windows 8, Operating Systems, Windows ... Networking, Computers, Technology) Operating System Concepts Essentials, 2nd Edition USB: The Universal Serial Bus (FYSOS: Operating System Design Book 8) The Design of the UNIX Operating System FreeBSD: Servidores de Alta Performance (Portuguese Edition) Part18: Using bhyve on FreeBSD 10 How to implement a hypervisor (Japanese Edition) Der eigene Server mit FreeBSD 9: Konfiguration, Sicherheit und Pflege (German Edition) DTrace: Dynamic Tracing in Oracle Solaris, Mac OS X and FreeBSD Mastering FreeBSD and OpenBSD Security DTrace: Dynamic Tracing in Oracle Solaris, Mac OS X, and FreeBSD (Oracle Solaris Series)