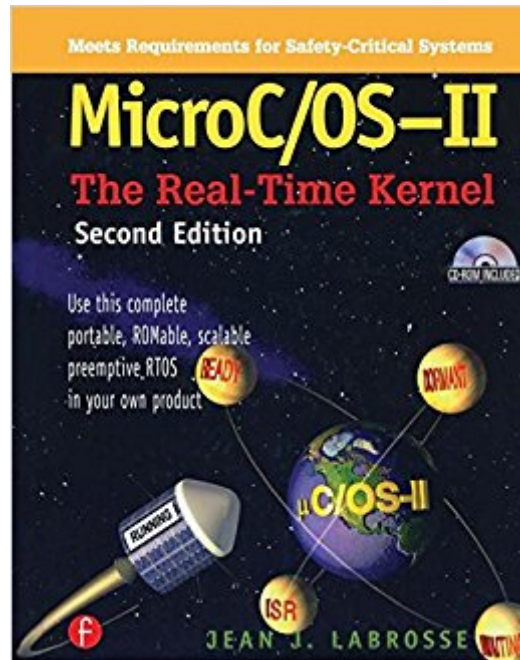# The book was found

MicroC OS II: The Real Time Kernel (With CD-ROM)

Synopsis

MicroC/OS II Second Edition describes the design and implementation of the MicroC/OS-II real-time operating system (RTOS). In addition to its value as a reference to the kernel, it is an extremely detailed and highly readable design study particularly useful to the embedded systems student. While documenting the design and implementation of the kernel, the book also walks the reader through the many related development issues: how to adapt the kernel for a new microprocessor, how to install the kernel, and how to structure the applications that run on the kernel. This edition features documentation for several important new features of the software, including new real-time services, floating points, and coding conventions. The accompanying CDROM includes complete code for the MicroC/OS-II kernel.

Book Information

Hardcover: 648 pages

Publisher: CRC Press; 2 edition (June 15, 2002)

Language: English

ISBN-10: 1578201039

ISBN-13: 978-1578201037

Product Dimensions: 1.5 x 8.2 x 10.8 inches

Shipping Weight: 4 pounds (View shipping rates and policies)

Average Customer Review: 4.5 out of 5 ststarsÂ Â See all reviewsÂ (35 customer reviews)

Best Sellers Rank: #308,564 in Books (See Top 100 in Books)   #35 inÂ Books > Computers & Technology > Hardware & DIY > Microprocessors & System Design > Embedded Systems   #150 inÂ Books > Computers & Technology > Graphics & Design > User Experience & Usability   #188 inÂ Books > Computers & Technology > Computer Science > Robotics

Customer Reviews

Jean LaBrosse is one of those people who has that uncanny knack for taking a complicated issue and making it seem simple. His MicroC/OS is a wonderful example. The book does much more than simply present usable source code (though it does that also). It explains _WHY_ the code is the way it is.MicroC/OS II is very much in today's spirit of open-source software. It is rapidly becoming, for real-time embedded systems, what Linux represents for desktops. I also agree with another reader, who points out that the OS is not just a good OS for real-time systems, but a good, _RELIABLE_, OS in general. If only we could get Microsoft to follow LaBrosse's KISS approaches.

I have been involved with the development of embedded real-time systems since 1969 and have written a good number of kernels in that time. After swapping notes with the author, he convinced me to try uC/OS-II rather than doing another "roll my own". I'm glad he did. The book is extremely well written, clearly explaining the key concepts and his implementation. The software itself is very functional. I plan to use it in several applications I have on the boards. I would highly recommend this book to anyone contemplating an embedded system project.

This is a very valuable book and the RTOS it comes with might be all you ever need. We are using it in an unmanned aerial vehicle project. Knowing what every line of code in the OS is doing is a big advantage for us because of certification issues. We're also not wasting performance on features we don't need. Check it out. It might be the only RTOS you'll ever need.

There is no equivalent to this book. The knowledge offered here, along with the complete source code for the Ã Â C/OS real-time operating system is everything you need to know to understand preemptive priority-based multitasking. Whether you just want to learn what goes on behind the scenes in your commercial RTOS or you're looking for a small, inexpensive RTOS with source code, this is the book for you.

Actually I think that it's a great book only if you have a strong background about how a kernel or RTOS works. I've spent a lot of time reading a theoretical books that teach you what a kernel is about, so for me the reading was easy. The fact that the book includes the source code makes it awesome, 'cause you can see inside a truly real kernel, and eventually write your own (what is I want to do). But if you are an unexperienced user of kernels, or just want to learn something about it, you should start somewhere else. The one a recomend you is that of Andy Tanenbaum, the creator of MINIX (minix is a tiny unix running in an intel 8086 cpu), and this other one that I bought here: Real-Time Concepts for Embedded Systems by Qing Li, Caroline Yao. After you deeply understand what a pipe o scheduler is, among other things, then you should start thinking about coding, because the book has some lack of theory. Anyway I strongly recomend it for experienced users. Have a nice coding =)

I read this book when I needed to understand embedded real-time operating systems. The writing was clear, concise, and well-organized, a rare gem among technical literature of this type. Jean Labrosse used just enough repetition to emphasize important concepts without being anoying.My

project required an RTOS with more features than the MicroC/OS kernel had to offer but, having read this book, I was in a good position to understand the RTOS I did choose. The fundamental concepts were the same and the notation that Jean Labrosse used in his book to describe RTOS components came in handy when designing my own software system architecture.

An excellent and very readable book with a very usable RTOS. The first 75 pages provide a good intro. to operating systems in general and RTOS's in particular. The only downside for PC users is the Borland compiler's used for the examples are hard to come by and not very backward compatible with eachother. I recommend using the protected mode port available on the website with newer compilers. That's the only reason I didn't give it 5 stars.

If you're new to the subject of RTOS: Buy the book.Read it.Try it.Use it.This book is an eye opener.It makes you want to create every project with this RTOS, provided your chip has enough resources.The best thing about it: The price is right and the secrets are out.It's all well documented C-source.You will love it.

MicroC OS II: The Real Time Kernel (With CD-ROM) Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX (4th Edition) (International Computer Science Series) The Real Book of Real Estate: Real Experts. Real Stories. Real Life Real Estate: 25 Best Strategies for Real Estate Investing, Home Buying and Flipping Houses (Real Estate, Real Estate Investing, home buying, flipping houses, ... income, investing, entrepreneurship) Real Estate: 30 Best Strategies to Prosper in Real Estate - Real Estate Investing, Financing & Cash Flow (Real Estate Investing, Flipping Houses, Brokers, Foreclosure) The Mystery at Jamestown (Real Kids, Real Places) (Real Kids! Real Places! (Paperback)) Real Time UML: Advances in the UML for Real-Time Systems (3rd Edition) Real-Time Systems: Design Principles for Distributed Embedded Applications (Real-Time Systems Series) Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications (Real-Time Systems Series) From Kernel to Corn (Start to Finish, Second (Paperback)) LINUX Kernel Internals Understanding the Linux Kernel IA-64 Linux Kernel: Design and Implementation learning by doing: LINUX kernel guide(Chinese Edition) Managing the Linux kernel with AgentX: Design and Implementation Linux Kernel Development Using Linux Kernel Version 2.0 to 2.2 (Keeping Ahead) The Linux Kernel Linux Kernel Architecture (Sams White Book Series) Linux Kernel Architecture