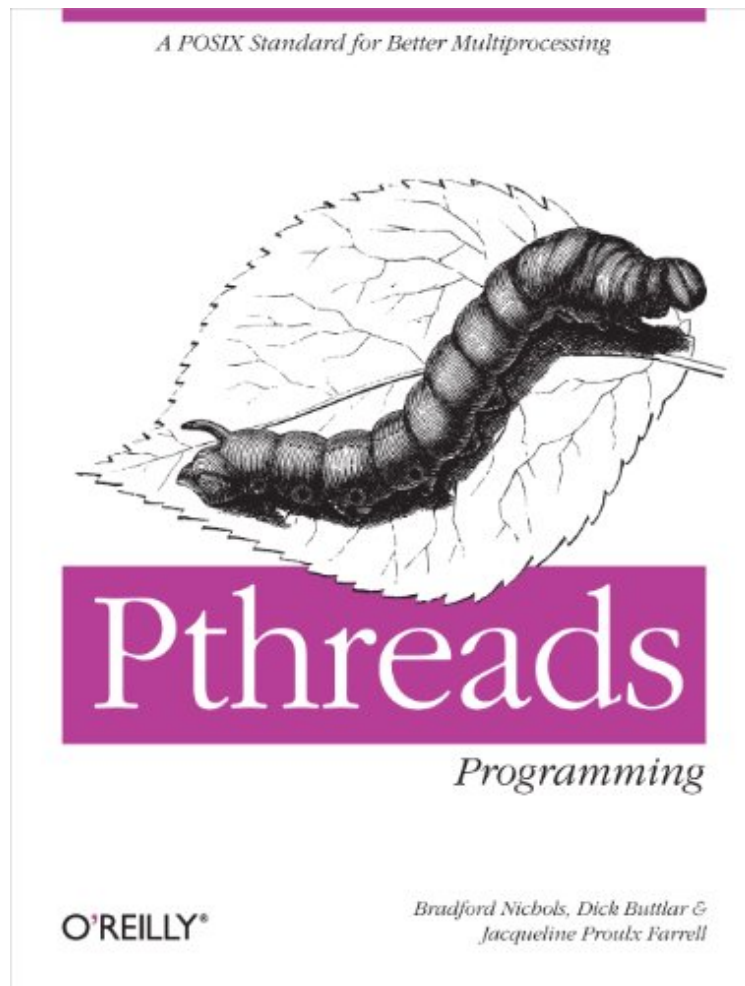


[Download free pdf] PThreads Programming: A POSIX Standard for Better Multiprocessing (A Nutshell handbook)

PThreads Programming: A POSIX Standard for Better Multiprocessing (A Nutshell handbook)

Von Dick Buttlar, Jacqueline Farrell, Bradford Nichols
DOC | *audiobook | ebooks | Download PDF | ePub



DOWNLOAD



READ ONLINE

Produktinformation -Verkaufsrang: #845626 in eBooksVerffentlicht am: 1996-09-01Erscheinungsdatum:
2013-01-24File Name: B00B5KBVHK | File size: 75.Mb

Von Dick Buttlar, Jacqueline Farrell, Bradford Nichols : PThreads Programming: A POSIX Standard for Better Multiprocessing (A Nutshell handbook) before purchasing it in order to gage whether or not it would be worth my time, and all praised PThreads Programming: A POSIX Standard for Better Multiprocessing (A Nutshell handbook):

KundenrezensionenHilfreichste Kundenrezensionen4 von 4 Kunden fanden die folgende Rezension hilfreich. Okay for concepts, but no help in codingVon Ein KundeThe strength of this book is it's brevity: 233 pp of text plus appendices. But the code samples are incomplete (fragments). You'll be able to get an idea of how pthreads work and the methods available, but you'll have a very hard time if you need to actually write code. There is an error on p.126. If you want to

write code, get "Programming with POSIX Threads" by David Butenhof. It has complete code examples and is not that much longer: 305 pp of text plus appendices. But I did find this Nichols book helpful when I was curious about pthreads. I commend O'Reilly for the nice illustrations in this book -- above average. They helped convey concepts.0 von 0 Kunden fanden die folgende Rezension hilfreich. Walk before you run Von Bernie With all the sophisticated tools available today such as OpenMP, this book may seem quaint. However before grasping at concepts or just using tools with out any understanding or their make up it would be wise to add this to your learning curve. This book gives a good basic understanding of Pthreads. Of course you will later have to apply it to the real world but as a learning tool this is pretty darn good. The website or path on the sight has changed since publication; but the examples are still there. Because I am using AIX it does take a little time to convert from a gcc format to an AIX format. Then a little more time to apply AIX specific advantages. As you go from the front of the book to the end the samples get added to and new concepts become available. There are plenty of diagrams for the visual learner. I personally found the signal handling of great use. Any way this book is not the end-all, be-all, of threads but it sure cleared a lot of concepts up for me. 1. Why Pthreads 2. Designing threaded programs 3. Synchronizing Pthreads 4. Managing Pthreads 5. Pthreads and UNIX 6. practical Considerations Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation) 1 von 1 Kunden fanden die folgende Rezension hilfreich. Second best book on the subject Von Ein Kunde As usual, O'Reilly have produced an excellent reference book. In a few respects, this book is preferable to Scott Norton's "Thread Time" - it has better examples, for one. But this book is not as detailed or complete as Norton, so I don't recommend it as the best choice on the subject.

Kurzbeschreibung Computers are just as busy as the rest of us nowadays. They have lots of tasks to do at once, and need some cleverness to get them all done at the same time. That's why threads are seen more and more often as a new model for programming. Threads have been available for some time. The Mach operating system, the Distributed Computer Environment (DCE), and Windows NT all feature threads. One advantage of most UNIX implementations, as well as DCE, is that they conform to a recently ratified POSIX standard (originally 1003.4a, now 1003.1c), which allows your programs to be portable between them. POSIX threads are commonly known as pthreads, after the word that starts all the names of the function calls. The standard is supported by Solaris, OSF/1, AIX, and several other UNIX-based operating systems. The idea behind threads programming is to have multiple tasks running concurrently within the same program. They can share a single CPU as processes do, or take advantage of multiple CPUs when available. In either case, they provide a clean way to divide the tasks of a program while sharing data. A window interface can read input on dozens of different buttons, each responsible for a separate task. A network server has to accept simultaneous calls from many clients, providing each with reasonable response time. A multiprocessor runs a number-crunching program on several CPUs at once, combining the results when all are done. All these kinds of applications can benefit from threads. In this book you will learn not only what the pthread calls are, but when it is a good idea to use threads and how to make them efficient (which is the whole reason for using threads in the first place). The authors delves into performance issues, comparing threads to processes, contrasting kernel threads to user threads, and showing how to measure speed. He also describes in a simple, clear manner what all the advanced features are for, and how threads interact with the rest of the UNIX system. Topics include: Basic design techniques Mutexes, conditions, and specialized synchronization techniques Scheduling, priorities, and other real-time issues Cancellation UNIX libraries and re-entrant routines Signals Debugging tips Measuring performance Special considerations for the Distributed Computing Environment (DCE)