

Programming With Posix Threads By Butenhof David R Paperback

Particularly helpful for C programmers working with such platforms as UNIX, Windows NT, Windows 95, OS/2, and NextStep, this book has many unique features, including the first detailed look at SMP (symmetrical multiprocessing) and its role in successful parallel processing. Numerous illustrative examples are included throughout.

Mathematics of Computing -- Parallelism.

Book explains how to maximize the benefits of Intel's new dual-core and multi-core processors through a portable C++ library that works on Windows, Linux, Macintosh, and Unix systems.

This is a clear introduction to the basic concepts of multi-threading complemented by a detailed description of the multi-threading facilities available under the UNIX and Windows operating systems. The implementation mechanisms are hidden within C++ classes, which then provide standardized interfaces to the functionality. With traditional single-threaded programming, objects serve as passive repositories of functionality that are invoked by external code multi-threading allows objects to become active entities that independently perform their own processing.

One of the original developers of the NFS and WebNFS offers unique insight into these key technologies, for both programmers creating and debugging NFS-based applications and network engineers creating new implementations.

Readers can gain a deeper understanding of how network file protocols are designed and learn how NFS is implemented on UNIX, Windows NT, Java and web browsers.

Explains how to use Java's portable platforms to program and use threads effectively and efficiently while avoiding common mistakes

The classic guide to UNIX® programming-completely updated! UNIX application programming requires a mastery of system-level services. Making sense of the many functions-more than 1,100 functions in the current UNIX specification-is a daunting task, so for years programmers have turned to Advanced UNIX

Programming for its clear, expert advice on how to use the key functions reliably.

An enormous number of changes have taken place in the UNIX environment since the landmark first edition. In Advanced UNIX Programming, Second Edition, UNIX pioneer Marc J. Rochkind brings the book fully up to date, with all-new, comprehensive coverage including: POSIX Solaris™ Linux® FreeBSD Darwin, the Mac™ OS X kernel And more than 200 new system calls Rochkind's fully updated classic explains all the UNIX system calls you're likely to need, all in a single volume! Interprocess communication, networking (sockets), pseudo terminals, asynchronous I/O, advanced signals, realtime, and threads Covers the system calls you'll actually use-no need to plow through hundreds of improperly implemented, obsolete, and otherwise unnecessary system calls! Thousands of

lines of example code include a Web browser and server, a keystroke recorder/player, and a shell complete with pipelines, redirection, and background processes Emphasis on the practical-ensuring portability, avoiding pitfalls, and much more! Since 1985, the one book to have for mastering UNIX application programming has been Rochkind's Advanced UNIX Programming. Now completely updated, the second edition remains the choice for up-to-the-minute, in-depth coverage of the essential system-level services of the UNIX family of operating systems.

Providing an overview of the Solaris and POSIX multithreading architectures, this book explains threads at a level that is completely accessible to programmers and system architects with no previous knowledge of threads. It covers the business and technical benefits of threaded programs, along with discussions of third party software that is threaded, pointing out the benefits. It also describes the design of the Solaris MT API, with references to distinctions in POSIX, contains a set of example programs which illustrate the usage of the Solaris and POSIX APIs, and explains the use of programming tools: Thread Analyzer, LockLint, LoopTool and Debugger.

In this book, realistic examples show both the situations where threading is valuable and the ways to use threads to improve the modularity and efficiency of a program. The author takes the user behind the scenes to show them how threads work, where to expect problems, and what performance issues exist. Chapters on DCE, real-time, and multiprocessing are included.

Software -- Operating Systems.

If you're looking to take full advantage of multi-core processors with concurrent programming, this practical book provides the knowledge and hands-on experience you need. The Art of Concurrency is one of the few resources to focus on implementing algorithms in the shared-memory model of multi-core processors, rather than just theoretical models or distributed-memory architectures. The book provides detailed explanations and usable samples to help you transform algorithms from serial to parallel code, along with advice and analysis for avoiding mistakes that programmers typically make when first attempting these computations. Written by an Intel engineer with over two decades of parallel and concurrent programming experience, this book will help you: Understand parallelism and concurrency Explore differences between programming for shared-memory and distributed-memory Learn guidelines for designing multithreaded applications, including testing and tuning Discover how to make best use of different threading libraries, including Windows threads, POSIX threads, OpenMP, and Intel Threading Building Blocks Explore how to implement concurrent algorithms that involve sorting, searching, graphs, and other practical computations The Art of Concurrency shows you how to keep algorithms scalable to take advantage of new processors with even more cores. For developing parallel code algorithms for concurrent programming, this book is a must.

In-depth coverage is given of the emerging POSIX Threads library for UNIX and how to code with it. These pages explain the concepts and foundations of threads programming, including real-life constructions. The book compares and contrasts the Pthreads library with those for OS/2 and Windows NT throughout.

Online Library Programming With Posix Threads By Butenhof David R Paperback

"Multithreaded Programming with Java Technology is the first complete guide to multithreaded development with the Java 2 platform. Multithreading experts Bil Lewis and Daniel J. Berg cover the underlying structures upon which threads are built; thread construction; and thread lifecycles, including birth, life, death, and cancellation. Next, using extensive code examples, they cover everything developers need to know to make the most of multithreading."--BOOK JACKET.Title Summary field provided by Blackwell North America, Inc. All Rights Reserved

bull; Learn UNIX essentials with a concentration on communication, concurrency, and multithreading techniques bull; Full of ideas on how to design and implement good software along with unique projects throughout bull; Excellent companion to Stevens' Advanced UNIX System Programming

A practical guide and reference to developing multithreaded programs on UNIX systems written by the foremost experts on the technology. Covers the two main UNIX threads and the UNIX International threads standard. All examples in the book use the POSIX standard.

This is a comprehensive guide to using the complex C++ IOSTREAMS library. Written for experienced C++ programmers, the book focuses on creating software that supports the widest possible range of hardware devices. The book also: * Presents clear, concise explanations of the major components of the IOSTREAMS * Includes solid introductions to the class relationship diagramming technique, the multimedia stream, and the blob stream * Provides object-oriented methods of creating I/O code * Covers VGA and super-VGA monitors, serial and parallel communications ports, modems, printers, and much more

The easiest way for programmers to learn important new multi-threading techniques that are increasingly important in Windows NT/95, UNIX, POSIX, and other application development. Each concept in the book is illustrated with a picture, making this an exceptionally easy-to-understand guide. The book introduces the process model, the thread model, and basic thread management functions. Learn how to synchronize and schedule threads. In a Programming Guidelines chapter, learn the basic do's and don'ts of multithreaded programming. The book includes extensive examples, exercises and references, including manual pages, debugging advice, and a CD-ROM loaded with practical information. This book is an effective introduction to multithreading for both professional programmers and students.

Experienced game developers.

UNIX, UNIX LINUX & UNIX TCL/TK. Write software that makes the most effective use of the Linux system, including the kernel and core system libraries. The majority of both Unix and Linux code is still written at the system level, and this book helps you focus on everything above the kernel, where applications such as Apache, bash, cp, vim, Emacs, gcc, gdb, glibc, ls, mv, and X exist. Written primarily for engineers looking to program at the low level, this updated edition of Linux System Programming gives you an understanding of core internals that makes for better code, no matter where it appears in the stack. -- Provided by publisher.

Based upon the authors' experience in designing and deploying an embedded Linux system with a variety of applications, Embedded Linux System Design and Development contains a full embedded Linux system development roadmap for systems architects and software programmers. Explaining the issues that arise out of

Online Library Programming With Posix Threads By Butenhof David R Paperback

the use of Linux in embedded systems, the book facilitates movement to embedded Linux from traditional real-time operating systems, and describes the system design model containing embedded Linux. This book delivers practical solutions for writing, debugging, and profiling applications and drivers in embedded Linux, and for understanding Linux BSP architecture. It enables you to understand: various drivers such as serial, I2C and USB gadgets; uClinux architecture and its programming model; and the embedded Linux graphics subsystem. The text also promotes learning of methods to reduce system boot time, optimize memory and storage, and find memory leaks and corruption in applications. This volume benefits IT managers in planning to choose an embedded Linux distribution and in creating a roadmap for OS transition. It also describes the application of the Linux licensing model in commercial products. By using this innovative text, students will obtain an understanding of how contemporary operating systems and middleware work, and why they work that way. "This book is organized around three concepts fundamental to OS construction: virtualization (of CPU and memory), concurrency (locks and condition variables), and persistence (disks, RAIDS, and file systems"--Back cover.

Shared Memory Application Programming presents the key concepts and applications of parallel programming, in an accessible and engaging style applicable to developers across many domains. Multithreaded programming is today a core technology, at the basis of all software development projects in any branch of applied computer science. This book guides readers to develop insights about threaded programming and introduces two popular platforms for multicore development: OpenMP and Intel Threading Building Blocks (TBB). Author Victor Alessandrini leverages his rich experience to explain each platform's design strategies, analyzing the focus and strengths underlying their often complementary capabilities, as well as their interoperability. The book is divided into two parts: the first develops the essential concepts of thread management and synchronization, discussing the way they are implemented in native multithreading libraries (Windows threads, Pthreads) as well as in the modern C++11 threads standard. The second provides an in-depth discussion of TBB and OpenMP including the latest features in OpenMP 4.0 extensions to ensure readers' skills are fully up to date. Focus progressively shifts from traditional thread parallelism to modern task parallelism deployed by modern programming environments. Several chapter include examples drawn from a variety of disciplines, including molecular dynamics and image processing, with full source code and a software library incorporating a number of utilities that readers can adapt into their own projects. Designed to introduce threading and multicore programming to teach modern coding strategies for developers in applied computing Leverages author Victor Alessandrini's rich experience to explain each platform's design strategies, analyzing the focus and strengths underlying their often complementary capabilities, as well as their interoperability Includes complete, up-to-date discussions of OpenMP 4.0 and TBB Based on the author's training sessions, including information on source code and software libraries which can be repurposed

The revision of the definitive guide to Unix system programming is now available in a more portable format.

This is the eBook version of the printed book. If the print book includes a CD-ROM, this content is not included within the eBook version. Advanced Linux Programming is divided into two

Online Library Programming With Posix Threads By Butenhof David R Paperback

parts. The first covers generic UNIX system services, but with a particular eye towards Linux specific information. This portion of the book will be of use even to advanced programmers who have worked with other Linux systems since it will cover Linux specific details and differences. For programmers without UNIX experience, it will be even more valuable. The second section covers material that is entirely Linux specific. These are truly advanced topics, and are the techniques that the gurus use to build great applications. While this book will focus mostly on the Application Programming Interface (API) provided by the Linux kernel and the C library, a preliminary introduction to the development tools available will allow all who purchase the book to make immediate use of Linux.

Written in an informal, informative style, this authoritative guide goes way beyond the standard reference manual. It discusses each of the POSIX.4 facilities and what they mean, why and when you would use each of these facilities, and trouble spots you might run into. c.

Learning a language--any language--involves a process wherein you learn to rely less and less on instruction and more increasingly on the aspects of the language you've mastered. Whether you're learning French, Java, or C, at some point you'll set aside the tutorial and attempt to converse on your own. It's not necessary to know every subtle facet of French in order to speak it well, especially if there's a good dictionary available. Likewise, C programmers don't need to memorize every detail of C in order to write good programs. What they need instead is a reliable, comprehensive reference that they can keep nearby. C in a Nutshell is that reference. This long-awaited book is a complete reference to the C programming language and C runtime library. Its purpose is to serve as a convenient, reliable companion in your day-to-day work as a C programmer. C in a Nutshell covers virtually everything you need to program in C, describing all the elements of the language and illustrating their use with numerous examples. The book is divided into three distinct parts. The first part is a fast-paced description, reminiscent of the classic Kernighan & Ritchie text on which many C programmers cut their teeth. It focuses specifically on the C language and preprocessor directives, including extensions introduced to the ANSI standard in 1999. These topics and others are covered: Numeric constants Implicit and explicit type conversions Expressions and operators Functions Fixed-length and variable-length arrays Pointers Dynamic memory management Input and output The second part of the book is a comprehensive reference to the C runtime library; it includes an overview of the contents of the standard headers and a description of each standard library function. Part III provides the necessary knowledge of the C programmer's basic tools: the compiler, the make utility, and the debugger. The tools described here are those in the GNU software collection. C in a Nutshell is the perfect companion to K&R, and destined to be the most reached-for reference on your desk.

The Linux Programming Interface (TLPI) is the definitive guide to the Linux and UNIX programming interface—the interface employed by nearly every application that runs on a Linux or UNIX system. In this authoritative work, Linux programming expert Michael Kerrisk provides detailed descriptions of the system calls and library functions that you need in order to master the craft of system programming, and accompanies his explanations with clear, complete example programs. You'll find descriptions of over 500 system calls and library functions, and more than 200 example programs, 88 tables, and 115 diagrams. You'll learn how to: –Read and write files efficiently –Use signals, clocks, and timers –Create processes and execute programs –Write secure programs –Write multithreaded programs using POSIX threads –Build and use shared libraries –Perform interprocess communication using pipes, message queues, shared memory, and semaphores –Write network applications with the sockets API While The Linux Programming Interface covers a wealth of Linux-specific features, including epoll, inotify, and the /proc file system, its emphasis on UNIX standards (POSIX.1-2001/SUSv3 and POSIX.1-2008/SUSv4) makes it equally valuable to programmers working on other UNIX platforms. The Linux Programming Interface is the most comprehensive single-volume work on

Online Library Programming With Posix Threads By Butenhof David R Paperback

the Linux and UNIX programming interface, and a book that's destined to become a new classic.

Beginning and experienced programmers will use this comprehensive guide to persistent memory programming. You will understand how persistent memory brings together several new software/hardware requirements, and offers great promise for better performance and faster application startup times—a huge leap forward in byte-addressable capacity compared with current DRAM offerings. This revolutionary new technology gives applications significant performance and capacity improvements over existing technologies. It requires a new way of thinking and developing, which makes this highly disruptive to the IT/computing industry. The full spectrum of industry sectors that will benefit from this technology include, but are not limited to, in-memory and traditional databases, AI, analytics, HPC, virtualization, and big data. Programming Persistent Memory describes the technology and why it is exciting the industry. It covers the operating system and hardware requirements as well as how to create development environments using emulated or real persistent memory hardware. The book explains fundamental concepts; provides an introduction to persistent memory programming APIs for C, C++, JavaScript, and other languages; discusses RMDA with persistent memory; reviews security features; and presents many examples. Source code and examples that you can run on your own systems are included. What You'll Learn Understand what persistent memory is, what it does, and the value it brings to the industry Become familiar with the operating system and hardware requirements to use persistent memory Know the fundamentals of persistent memory programming: why it is different from current programming methods, and what developers need to keep in mind when programming for persistence Look at persistent memory application development by example using the Persistent Memory Development Kit (PMDK) Design and optimize data structures for persistent memory Study how real-world applications are modified to leverage persistent memory Utilize the tools available for persistent memory programming, application performance profiling, and debugging Who This Book Is For C, C++, Java, and Python developers, but will also be useful to software, cloud, and hardware architects across a broad spectrum of sectors, including cloud service providers, independent software vendors, high performance compute, artificial intelligence, data analytics, big data, etc.

Master multithreading and concurrent processing with C++ About This Book Delve into the fundamentals of multithreading and concurrency and find out how to implement them Explore atomic operations to optimize code performance Apply concurrency to both distributed computing and GPGPU processing Who This Book Is For This book is for intermediate C++ developers who wish to extend their knowledge of multithreading and concurrent processing. You should have basic experience with multithreading and be comfortable using C++ development toolchains on the command line. What You Will Learn Deep dive into the details of the how various operating systems currently implement multithreading Choose the best multithreading APIs when designing a new application Explore the use of mutexes, spin-locks, and other synchronization concepts and see how to safely pass data between threads Understand the level of API support provided by various C++ toolchains Resolve common issues in multithreaded code and recognize common pitfalls using tools such as Memcheck, CacheGrind, DRD, Helgrind, and more Discover the nature of atomic operations and understand how they can be useful in optimizing code Implement a multithreaded application in a distributed computing environment Design a C++-based GPGPU application that employs multithreading In Detail Multithreaded applications execute multiple threads in a single processor environment, allowing developers achieve concurrency. This book will teach you the finer points of multithreading and concurrency concepts and how to apply them efficiently in C++. Divided into three modules, we start with a brief introduction to the fundamentals of multithreading and concurrency concepts. We then take an in-depth look at how these

Online Library Programming With Posix Threads By Butenhof David R Paperback

concepts work at the hardware-level as well as how both operating systems and frameworks use these low-level functions. In the next module, you will learn about the native multithreading and concurrency support available in C++ since the 2011 revision, synchronization and communication between threads, debugging concurrent C++ applications, and the best programming practices in C++. In the final module, you will learn about atomic operations before moving on to apply concurrency to distributed and GPGPU-based processing. The comprehensive coverage of essential multithreading concepts means you will be able to efficiently apply multithreading concepts while coding in C++. Style and approach This book is filled with examples that will help you become a master at writing robust concurrent and parallel applications in C++.

Summary This bestseller has been updated and revised to cover all the latest changes to C++ 14 and 17! C++ Concurrency in Action, Second Edition teaches you everything you need to write robust and elegant multithreaded applications in C++17. Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About the Technology You choose C++ when your applications need to run fast. Well-designed concurrency makes them go even faster. C++ 17 delivers strong support for the multithreaded, multiprocessor programming required for fast graphic processing, machine learning, and other performance-sensitive tasks. This exceptional book unpacks the features, patterns, and best practices of production-grade C++ concurrency. About the Book C++ Concurrency in Action, Second Edition is the definitive guide to writing elegant multithreaded applications in C++. Updated for C++ 17, it carefully addresses every aspect of concurrent development, from starting new threads to designing fully functional multithreaded algorithms and data structures. Concurrency master Anthony Williams presents examples and practical tasks in every chapter, including insights that will delight even the most experienced developer. What's inside Full coverage of new C++ 17 features Starting and managing threads Synchronizing concurrent operations Designing concurrent code Debugging multithreaded applications About the Reader Written for intermediate C and C++ developers. No prior experience with concurrency required. About the Author Anthony Williams has been an active member of the BSI C++ Panel since 2001 and is the developer of the just::thread Pro extensions to the C++ 11 thread library. Table of Contents Hello, world of concurrency in C++! Managing threads Sharing data between threads Synchronizing concurrent operations The C++ memory model and operations on atomic types Designing lock-based concurrent data structures Designing lock-free concurrent data structures Designing concurrent code Advanced thread management Parallel algorithms Testing and debugging multithreaded applications

Master the essentials of concurrent programming, including testing and debugging This textbook examines languages and libraries for multithreaded programming. Readers learn how to create threads in Java and C++, and develop essential concurrent programming and problem-solving skills. Moreover, the textbook sets itself apart from other comparable works by helping readers to become proficient in key testing and debugging techniques. Among the topics covered, readers are introduced to the relevant aspects of Java, the POSIX Pthreads library, and the Windows Win32 Applications Programming Interface. The authors have developed and fine-tuned this book through the concurrent programming courses they have taught for the past twenty years. The material, which emphasizes practical tools and techniques to solve concurrent programming problems, includes original results from the authors' research. Chapters include: * Introduction to concurrent programming * The critical section problem * Semaphores and locks * Monitors * Message-passing * Message-passing in distributed programs * Testing and debugging concurrent programs As an aid to both students and instructors, class libraries have been implemented to provide working examples of all the material that is covered. These libraries and the testing techniques they support can be used to assess student-written programs. Each chapter includes exercises that build skills in

Online Library Programming With Posix Threads By Butenhof David R Paperback

programwriting and help ensure that readers have mastered the chapter's key concepts. The source code for all the listings in the text and for the synchronization libraries is also provided, as well as startup files and test cases for the exercises. This textbook is designed for upper-level undergraduates and graduate students in computer science. With its abundance of practical material and inclusion of working code, coupled with an emphasis on testing and debugging, it is also a highly useful reference for practicing programmers.

Master the techniques needed to build great, efficient embedded devices on Linux About This Book Discover how to build and configure reliable embedded Linux devices This book has been updated to include Linux 4.9 and Yocto Project 2.2 (Morty) This comprehensive guide covers the remote update of devices in the field and power management Who This Book Is For If you are an engineer who wishes to understand and use Linux in embedded devices, this book is for you. It is also for Linux developers and system programmers who are familiar with embedded systems and want to learn and program the best in class devices. It is appropriate for students studying embedded techniques, for developers implementing embedded Linux devices, and engineers supporting existing Linux devices. What You Will Learn Evaluate the Board Support Packages offered by most manufacturers of a system on chip or embedded module Use Buildroot and the Yocto Project to create embedded Linux systems quickly and efficiently Update IoT devices in the field without compromising security Reduce the power budget of devices to make batteries last longer Interact with the hardware without having to write kernel device drivers Debug devices remotely using GDB, and see how to measure the performance of the systems using powerful tools such as `perf`, `ftrace`, and `valgrind` Find out how to configure Linux as a real-time operating system In Detail Embedded Linux runs many of the devices we use every day, from smart TVs to WiFi routers, test equipment to industrial controllers - all of them have Linux at their heart. Linux is a core technology in the implementation of the inter-connected world of the Internet of Things. The comprehensive guide shows you the technologies and techniques required to build Linux into embedded systems. You will begin by learning about the fundamental elements that underpin all embedded Linux projects: the toolchain, the bootloader, the kernel, and the root filesystem. You'll see how to create each of these elements from scratch, and how to automate the process using Buildroot and the Yocto Project. Moving on, you'll find out how to implement an effective storage strategy for flash memory chips, and how to install updates to the device remotely once it is deployed. You'll also get to know the key aspects of writing code for embedded Linux, such as how to access hardware from applications, the implications of writing multi-threaded code, and techniques to manage memory in an efficient way. The final chapters show you how to debug your code, both in applications and in the Linux kernel, and how to profile the system so that you can look out for performance bottlenecks. By the end of the book, you will have a complete overview of the steps required to create a successful embedded Linux system. Style and approach This book is an easy-to-follow and pragmatic guide with in-depth analysis of the implementation of embedded devices. It follows the life cycle of a project from inception through to completion, at each stage giving both the theory that underlies the topic and practical step-by-step walkthroughs of an example implementation. Learn to write advanced C programs that are strongly type-checked, compact, and easy to maintain. This book focuses on real-life applications and problem solving in networking, database development, compilers, operating systems, and CAD.

If you want to develop efficient, smooth-running applications, controlling concurrency and memory are vital. Automatic Reference Counting is Apple's game-changing memory management system, new to Xcode 4.2. Pro Multithreading and Memory Management for iOS and OS X shows you how ARC works and how best to incorporate it into your applications. Grand Central Dispatch (GCD) and blocks are key to developing great apps, allowing you to control threads for maximum performance. If for you, multithreading is an unsolved mystery

Online Library Programming With Posix Threads By Butenhof David R Paperback

and ARC is unexplored territory, then this is the book you'll need to make these concepts clear and send you on your way to becoming a master iOS and OS X developer. What are blocks? How are they used with GCD? Multithreading with GCD Managing objects with ARC This book assumes familiarity with threads (in a language such as Ada, C#, or Java) and introduces the entity-life modeling (ELM) design approach for certain kinds of multithreaded software. ELM focuses on "reactive systems," which continuously interact with the problem environment. These "reactive systems" include embedded systems, as well as such interactive systems as cruise controllers and automated teller machines. Part I covers two fundamentals: program-language thread support and state diagramming. These are necessary for understanding ELM and are provided primarily for reference. Part II covers ELM from different angles. Part III positions ELM relative to other design approaches.

An Introduction to Parallel Programming, Second Edition presents a tried-and-true tutorial approach that shows students how to develop effective parallel programs with MPI, Pthreads and OpenMP. As the first undergraduate text to directly address compiling and running parallel programs on multi-core and cluster architecture, this second edition carries forward its clear explanations for designing, debugging and evaluating the performance of distributed and shared-memory programs while adding coverage of accelerators via new content on GPU programming and heterogeneous programming. New and improved user-friendly exercises teach students how to compile, run and modify example programs. Takes a tutorial approach, starting with small programming examples and building progressively to more challenging examples Explains how to develop parallel programs using MPI, Pthreads and OpenMP programming models A robust package of online ancillaries for instructors and students includes lecture slides, solutions manual, downloadable source code, and an image bank New to this edition: New chapters on GPU programming and heterogeneous programming New examples and exercises related to parallel algorithms

Programming with POSIX Threads Addison-Wesley Professional

Twenty five years ago, as often happens in our industry, pundits laughed at and called Linux a joke. To say that view has changed is a massive understatement. This book will cement for you both the conceptual 'why' and the practical 'how' of systems programming on Linux, and covers Linux systems programming on the latest 4.x kernels.

Here is a programmer's guide to using and programming POSIX threads, commonly known as Pthreads. A "coder's book", this title tells how to use Pthreads in the real world, making efficient and portable applications. Pthreads are an important set of current tools programmers need to have in today's network-intensive climate.

[Copyright: 37ae18f405ad4e63361581fdbdd02ecf](#)