Advanced Unix Programming By Marc Rochkind

Getting the books **Advanced Unix Programming By Marc Rochkind** now is not type of challenging means. You could not by yourself going later ebook amassing or library or borrowing from your links to log on them. This is an entirely simple means to specifically get guide by on-line. This online revelation Advanced Unix Programming By Marc Rochkind can be one of the options to accompany you like having other time.

It will not waste your time. receive me, the e-book will completely look you further event to read. Just invest tiny grow old to way in this on-line message **Advanced Unix Programming By Marc Rochkind** as without difficulty as review them wherever you are now.

Systems Programming in Unix/Linux - K.C. Wang 2018-08-27 Covering all the essential components of Unix/Linux, including process management, concurrent programming, timer and time service, file systems and network programming, this textbook emphasizes programming practice in the Unix/Linux environment. Systems Programming in Unix/Linux is intended as a textbook for systems programming courses in technically-oriented Computer Science/Engineering curricula that emphasize both theory and programming practice. The book contains many detailed working example programs with complete source code. It is also suitable for self-study by advanced programmers and computer enthusiasts. Systems programming is an indispensable part of Computer Science/Engineering education. After taking an introductory programming course, this book is meant to further knowledge by detailing how dynamic data structures are used in practice, using programming exercises and programming projects on such topics as C structures, pointers, link lists and trees. This book provides a wide range of knowledge about computer systemsoftware and advanced programming skills, allowing readers to interface with operatingsystem kernel, make efficient use of system resources and develop application software.It also prepares readers with the needed background to pursue advanced studies inComputer Science/Engineering, such as operating systems, embedded systems, databasesystems, data mining, artificial intelligence, computer networks, network security,distributed and parallel computing.

Advanced UNIX Programming - Marc J. Rochkind 2004-04-29 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.

Dive Into Systems - Suzanne J. Matthews 2022-09-20

Dive into Systems is a vivid introduction to computer organization, architecture, and operating systems that is already being used as a classroom textbook at more than 25 universities. This textbook is a crash course in the major hardware and software components of a modern computer system. Designed for use in a wide range of introductory-level computer science classes, it guides readers through the vertical slice of a computer so they can develop an understanding of the machine at various layers of abstraction. Early chapters begin with the basics of the C programming language often used in systems programming. Other topics explore the architecture of modern computers, the inner workings of operating systems, and the assembly languages that translate human-readable instructions into a binary representation that the computer understands. Later chapters explain how to optimize code for various architectures, how to implement parallel computing with shared memory, and how memory management works in multi-core CPUs. Accessible and easy to follow, the book uses images and hands-on exercise to break down complicated topics, including code examples that can be modified and executed.

Linux Internals - Moshe Bar 2000-01-01

Furnishing in-depth coverage of Linux source-code internals, this high-level handbook explains how the Linux system operating system works and how to use it with various programming applications, discussing the various Linux versions, performance and tuning issues, kernel programming, troubleshooting details, and other important topics. Original. (Intermediate)

C Interfaces and Implementations - David R. Hanson 1997

C Interfaces and Implementations describes how to use interface-based design in the C programming language, and it illustrates this approach by describing 24 interfaces and their implementations in detail. The source code in the book is interleaved with its explanation in an order that best suits understanding the code.

C++ Gotchas - Stephen C. Dewhurst 2003

Corpus linguistics is a research approach to investigate the patterns of language use empirically, based on analysis of large collections of natural texts. While corpus-based analysis has had relatively little influence on theoretical linguistics, it has revolutionized the study of language variation and use: what speakers and writers actually do with the lexical and grammatical resources of a language. Corpus-based research employs the research methods of quantitative and qualitative social science to investigate language use patterns empirically. This four-volume collection is organized around linguistic research questions that can be investigated from a corpus perspective and includes amongst others studies of individual words, comparisons of supposedly synonymous words, studies of grammatical variation, and sociolinguistic studies of dialects, registers, styles, and world varieties. Corpus-based analysis has also proven to be important for the study of historical change.

The UNIX System V Environment - S. R. Bourne 1987

Software -- Operating Systems.

Programming with POSIX Threads - David R. Butenhof 1993-05-15

With this practical book, you will attain a solid understanding of threads and will discover how to put this powerful mode of programming to work in real-world applications. The primary advantage of threaded programming is that it enables your applications to accomplish more than one task at the same time by using the number-crunching power of multiprocessor parallelism and by automatically exploiting I/O concurrency in your code, even on a single processor machine. The result: applications that are faster, more responsive to users, and often easier to maintain. Threaded programming is particularly well suited to network programming where it helps alleviate the bottleneck of slow network I/O. This book offers an in-depth description of the IEEE operating system interface standard, POSIXAE (Portable Operating System Interface) threads, commonly called Pthreads. Written for experienced C programmers, but assuming no previous knowledge of threads, the book explains basic concepts such as asynchronous programming, the lifecycle of a thread, and synchronization. You then move to more advanced topics such as attributes objects, thread-specific data, and realtime scheduling. An entire chapter is devoted to "real code," with a look at barriers, read/write locks, the work queue manager, and how to utilize existing libraries. In addition, the book tackles one of the thorniest problems faced by thread programmers-debugging-with valuable suggestions on how to avoid code errors and performance problems from the outset. Numerous annotated examples are used to illustrate real-world concepts. A Pthreads mini-reference and a look at future standardization are also included.

Advanced UNIX Programming - Marc J. Rochkind 1985

This book covers how to program UNIX clearly and systematically at the system call level while providing the seasoned programmer with practical advice for using I/O on files and terminals, multitasking, signals and system administration.

The Art of UNIX Programming - Eric S. Raymond 2003-09-23

The Art of UNIX Programming poses the belief that understanding the unwritten UNIX engineering tradition and mastering its design patterns will help programmers of all stripes to become better programmers. This book attempts to capture the engineering wisdom and design philosophy of the UNIX, Linux, and Open Source software development community as it has evolved over the past three decades, and as it is applied today by the most experienced programmers. Eric Raymond offers the next generation of "hackers" the unique opportunity to learn the connection between UNIX philosophy and practice through careful case studies of the very best UNIX/Linux programs.

Learning the bash Shell - Cameron Newham 2005-03-29

O'Reilly's bestselling book on Linux's bash shell is at it again. Now that Linux is an established player both as a server and on the desktop Learning the bash Shell has been updated and refreshed to account for all the latest changes. Indeed, this third edition serves as the most valuable guide yet to the bash shell.As any good programmer knows, the first thing users of the Linux operating system come face to face with is the shell the UNIX term for a user interface to the system. In other words, it's what lets you communicate with the computer via the keyboard and display. Mastering the bash shell might sound fairly simple but it isn't. In truth, there are many complexities that need careful explanation, which is just what Learning the bash Shell provides.If you are new to shell programming, the book provides an excellent introduction, covering everything from the most basic to the most advanced features. And if you've been writing shell scripts for years, it offers a great way to find out what the new shell offers. Learning the bash Shell is also full of practical examples of shell commands and programs that will make everyday use of Linux that much easier. With this book, programmers will learn: How to install bash as your login shell The basics of interactive shell use, including UNIX file and directory structures, standard I/O, and background jobs Command line editing, history substitution, and key bindings How to customize your shell environment without programming The nuts and bolts of basic shell programming, flow control structures, command-line options and typed variables Process handling, from job control to processes, coroutines and subshells Debugging techniques, such as trace and verbose modes Techniques for implementing system-wide shell customization and features related to system security

Effective STL - Scott Meyers 2001

C++'s Standard Template Library is revolutionary, but learning to use it well has always been a challenge for students. In Effective STL, best-selling author Scott Meyers (Effective C++, More Effective C++) reveals the critical rules of thumb employed by the experts -- the things they almost always do or almost always avoid doing -- to get the most out of the library. This book offers clear, concise, and concrete guidelines to C++ programmers. While other books describe what's in the STL, Effective STL shows the student how to use it. Each of the book's 50 guidelines is backed by Meyers' legendary analysis and incisive examples, so the student will learn not only what to do, but also when to do it - and why.

Trust in Cyberspace - National Research Council 1999-02-08

Whether or not you use a computer, you probably use a telephone, electric power, and a bank. Although you may not be aware of their presence, networked computer systems are increasingly becoming an integral part of your daily life. Yet, if such systems perform poorly or don't work at all, then they can put life, liberty, and property at tremendous risk. Is the trust that weâ€"as individuals and as a societyâ€"are placing in networked computer systems justified? And if it isn't, what can we do to make such systems more trustworthy? This book provides an assessment of the current state of the art procedures for building trustworthy networked information systems. It proposes directions for research in computer and network security, software technology, and system architecture. In addition, the book assesses current technical and market trends in order to better inform public policy as to where progress is likely and where incentives could help. Trust in Cyberspace offers insights into: The strengths and vulnerabilities of the telephone network and Internet, the two likely building blocks of any networked information system. The interplay between various dimensions of trustworthiness: environmental disruption, operator error, "buggy" software, and hostile attack. The implications for trustworthiness of anticipated developments in hardware and software technology, including the consequences of mobile code. The shifts in security technology and research resulting from replacing centralized mainframes with networks of computers. The heightened concern for integrity and availability where once only secrecy mattered. The way in which federal research funding levels and practices have affected the evolution and current state of the science and technology base in this area. You will want to read this book if your life is touched in any way by computers or telecommunications. But then, whose life isn't?

Software Engineering for Internet Applications - Eve Astrid Andersson 2006

After completing this self-contained course on server-based Internet applications software that grew out of an MIT course, students who start with only the knowledge of how to write and debug a computer program will have learned how to build sophisticated Web-based applications.

Computer Systems - Randal E.. Bryant 2013-07-23

For Computer Systems, Computer Organization and Architecture courses in CS, EE, and ECE departments. Few students studying computer science or computer engineering will ever have the opportunity to build a computer system. On the other hand, most students will be required to use and program computers on a near daily basis. Computer Systems: A Programmer's Perspective introduces the important and enduring concepts that underlie computer systems by showing how these ideas affect the correctness, performance, and utility of application programs. The text's hands-on approach (including a comprehensive set of labs) helps students understand the under-the-hood operation of a modern computer system and prepares them for future courses in systems topics such as compilers, computer architecture, operating systems, and networking.

Advanced UNIX Programming - Marc J. Rochkind 1985

Readers can learn to program UNIX clearly and systematically at the system call level with this book that provides the seasoned programmer with practical advice for using I/O on files and terminals, multitasking, signals and system administration.

POSIX.4 Programmers Guide - Bill Gallmeister 1995

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.

Programming Chrome Apps - Marc Rochkind 2014-12-15

Put your web app design skills to work by learning how to create powerful and portable Chrome Apps. With this practical book, you'll learn how to build Google's unique apps to behave just like native apps so they can interact with hardware devices, access external files, and send notifications. Chrome Apps run on any platform that supports the Chrome browser--including OS X, Windows, Linux, as well as Android

and iOS. If you know how to work with HTML, CSS, JavaScript, and the DOM, you're ready to get started.
Advanced C Programming for Displays - Marc J. Rochkind 1988

Dorothea Lange: A Life Beyond Limits - Linda Gordon 2010-10-11
Winner of the 2010 Bancroft Prize and finalist for the 2009 Los Angeles Times Book Prize in Biography: The definitive biography of a heroic chronicler of America's Depression and one of the twentieth century's greatest photographers. We all know Dorothea Lange's iconic photos—the Migrant Mother holding her child, the shoeless children of the Dust Bowl—but now renowned American historian Linda Gordon brings them to three-dimensional life in this groundbreaking exploration of Lange's transformation into a documentarist. Using Lange's life to anchor a moving social history of twentieth-century America, Gordon masterfully re-creates bohemian San Francisco, the Depression, and the Japanese-American internment camps. Accompanied by more than one hundred images—many of them previously unseen and some formerly suppressed—Gordon has written a sparkling, fast-moving story that testifies to her status as one of the most gifted historians of our time. Finalist for the Los Angeles Times Book Prize; a New York Times Notable Book; New Yorker's A Year's Reading; and San Francisco Chronicle Best Book.
Radical Innovations of Software and Systems Engineering in the Future - Martin Wirsing 2004-03-03
This volume contains the papers from the workshop "Radical Innovations of Software and Systems Engineering in the Future." This workshop was the ninth in the series of Monterey Software Engineering workshops for formulating and advancing software engineering models and techniques, with the fundamental theme of increasing the practical impact of formal methods. During the last decade object orientation was the driving factor for new system solutions in many areas ranging from e-commerce to embedded systems. New modeling languages such as UML and new programming languages such as Java and CASE tools have considerably in?uenced the system development techniques of today and will remain key techniques for the near future. However, actual practice shows many de?ciencies of these new approaches: – there is no proof and no evidence that software productivity has increased with the new methods; – UML has no clean scienti?c foundations, which inhibits the construction of powerful analysis and development tools; – support for mobile distributed system development is missing; – formanyapplications,object-orienteddesignisnotsuitedtoproducingclean well-structured code, as many applications show.
Effective C++ - Scott Meyers 1998
Effective C++ has been updated to reflect the latest ANSI/ISO standards. The author, a recognised authority on C++, shows readers fifty ways to improve their programs and designs.
The Man in the Box - Marc J. Rochkind 2018-12-23
The Man in the Box is the story of two mis-matched twenty-somethings in New York who get swept into solving a very old mystery. There are lots of surprises, including almost everything that goes on between them.
AUUGN - 1991-10

Unix Shell Programming - Stephen G. Kochan 2003-02-27
Unix Shell Programming is a tutorial aimed at helping Unix and Linux users get optimal performance out of their operating out of their operating system. It shows them how to take control of their systems and work efficiently by harnessing the power of the shell to solve common problems. The reader learns everything he or she needs to know to customize the way a Unix system responds. The vast majority of Unix users utilize the Korn shell or some variant of the Bourne shell, such as bash. Three are covered in the third edition of Unix Shell Programming. It begins with a generalized tutorial of Unix and tools and then moves into detailed coverage of shell programming. Topics covered include: regular expressions, the kernel and the utilities, command files, parameters, manipulating text filters, understanding and debugging shell scripts, creating and utilizing variables, tools, processes, and customizing the shell.
Advanced Programming in the UNIX Environment - W. Richard Stevens

2008-01-01
The revision of the definitive guide to Unix system programming is now available in a more portable format.

Learning GNU Emacs - Debra Cameron 1996
Carries readers from the beginning through the proficient stages of learning the GNU Emacs editor, covering everything from simple text editing to moderately complicated customization and programming. Original. (Advanced).

Linux System Programming - Robert Love 2013-05-14
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.

Operating Systems - Thomas Anderson 2014
Over the past two decades, there has been a huge amount of innovation in both the principles and practice of operating systems Over the same period, the core ideas in a modern operating system - protection, concurrency, virtualization, resource allocation, and reliable storage - have become widely applied throughout computer science. Whether you get a job at Facebook, Google, Microsoft, or any other leading-edge technology company, it is impossible to build resilient, secure, and flexible computer systems without the ability to apply operating systems concepts in a variety of settings. This book examines the both the principles and practice of modern operating systems, taking important, high-level concepts all the way down to the level of working code. Because operating systems concepts are among the most difficult in computer science, this top to bottom approach is the only way to really understand and master this important material.

Mastering Regular Expressions - Jeffrey E.F. Friedl 2006-08-08
Regular expressions are an extremely powerful tool for manipulating text and data. They are now standard features in a wide range of languages and popular tools, including Perl, Python, Ruby, Java, VB.NET and C# (and any language using the .NET Framework), PHP, and MySQL. If you don't use regular expressions yet, you will discover in this book a whole new world of mastery over your data. If you already use them, you'll appreciate this book's unprecedented detail and breadth of coverage. If you think you know all you need to know about regularexpressions, this book is a stunning eye-opener. As this book shows, a command of regular expressions is an invaluable skill. Regular expressions allow you to code complex and subtle text processing that you never imagined could be automated. Regular expressions can save you time and aggravation. They can be used to craft elegant solutions to a wide range of problems. Once you've mastered regular expressions, they'll become an invaluable part of your toolkit. You will wonder how you ever got by without them. Yet despite their wide availability, flexibility, and unparalleled power, regular expressions are frequently underutilized. Yet what is power in the hands of an expert can be fraught with peril for the unwary. Mastering Regular Expressions will help you navigate the minefield to becoming an expert and help you optimize your use of regular expressions. Mastering Regular Expressions, Third Edition, now includes a full chapter devoted to PHP and its powerful and expressive suite of regular expression functions, in addition to enhanced PHP coverage in the central "core" chapters. Furthermore, this edition has been updated throughout to reflect advances in other languages, including expanded in-depth coverage of Sun's java.util.regex package, which has emerged as the standard Java regex implementation.Topics include: A comparison of features among different versions of many languages and tools How the regular expression engine works Optimization (major savings available here!) Matching just what you want, but not what you don't want Sections and chapters on individual languages Written in the lucid, entertaining tone that makes a complex, dry topic become crystal-clear to programmers, and sprinkled with solutions to complex real-world

problems, Mastering Regular Expressions, Third Edition offers a wealth information that you can put to immediateuse. Reviews of this new edition and the second edition: "There isn't a better (or more useful) book available on regular expressions." --Zak Greant, Managing Director, eZ Systems "A real tour-de-force of a book which not only covers the mechanics of regexes in extraordinary detail but also talks about efficiency and the use of regexes in Perl, Java, and .NET...If you use regular expressions as part of your professional work (even if you already have a good book on whatever language you're programming in) I would strongly recommend this book to you." --Dr. Chris Brown, Linux Format "The author does an outstanding job leading the reader from regexnovice to master. The book is extremely easy to read and chock full ofuseful and relevant examples...Regular expressions are valuable toolsthat every developer should have in their toolbox. Mastering RegularExpressions is the definitive guide to the subject, and an outstandingresource that belongs on every programmer's bookshelf. Ten out of TenHorseshoes." --Jason Menard, Java Ranch

UNIX System Programming Using C++ - Terrence Chan 1997
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.

Modern Operating Systems - Andrew S. Tanenbaum 2014-03-10
Modern Operating Systems, Fourth Edition, is intended for introductory courses in Operating Systems in Computer Science, Computer Engineering, and Electrical Engineering programs. It also serves as a useful reference for OS professionals ¿ The widely anticipated revision of this worldwide best-seller incorporates the latest developments in operating systems (OS) technologies. The Fourth Edition includes up-to-date materials on relevant¿OS. Tanenbaum also provides information on current research based on his experience as an operating systems researcher. ¿ Modern Operating Systems, Third Editionwas the recipient of the 2010 McGuffey Longevity Award. The McGuffey Longevity Award recognizes textbooks whose excellence has been demonstrated over time.¿http://taaonline.net/index.html ¿¿ Teaching and Learning Experience This program will provide a better teaching and learning experience–for you and your students. It will help: ¿ Provide Practical Detail on the Big Picture Concepts: A clear and entertaining writing style outlines the concepts every OS designer needs to master. Keep Your Course Current: This edition includes information on the latest OS technologies and developments Enhance Learning with Student and Instructor Resources: Students will gain hands-on experience using the simulation exercises and lab experiments.

Understanding Unix/Linux Programming - Bruce Molay 2003
This book explains in a clear and coherent manner how Unix works, how to understand existing Unix programs, and how to design and create new Unix programs. The book is organized by subsystem, each presented in visual terms and explained using vivid metaphors. It breaks the information into manageable parts that can be presented, explained, and mastered.By using case studies and an extremely reader-friendly manner to illustrate complex ideas and concepts, the book covers the basics of systems programming, users, files and manuals, how to read a directory, using 1S, writing PWD, studying STTY, writing a video game, studying SH, environment and shell variables, I/O redirection and pipes, servers and sockets, writing a web server, license servers, and concurrent functions.For Unix system administrators and programmers, network programmers, and others who have used other operating systems and need to learn Unix programming to expand their skill sets.

Expert PHP and MySQL - Marc Rochkind 2013-09-30
Expert PHP and MySQL takes you beyond learning syntax to showing you how to apply proven software development methods to building commerce-grade PHP and MySQL projects that will stand the test of time and reliably deliver on customer needs. Developers of real-world applications face numerous problems that seem trivial on the surface, but really do take some skill to get right. Error handling is about more than just the mechanics in the PHP syntax, but also about handling MySQL errors, logging those errors, and about hiding information about application internals that error messages sometimes can expose. Meet

these challenges and more head-on! Author Marc Rochkind shows how to begin a project right, with a clear contract and set of written requirements. You'll learn about project organization, setting up a solid development environment, connecting with client personnel. Database design is essential, and Expert PHP and MySQL has you covered with guidance on creating a sound model and database, and on pushing functionality into the database as appropriate; not everything should be done in PHP. Error handling is covered at both the PHP and MySQL levels. Application structure is covered. Guidance is provided on reporting. And finally there is conversion. In Expert PHP and MySQL you'll explore the following: The popular and widely used combination of PHP and MySQL Commercial-grade application of language and database features Human factors such as planning and organization Organizing a project to meet requirements and satisfy the customer Structuring an application for efficient development and future modification Coding PHP for productivity, reliability, security Generating online, downloadable, and printed reports Converting existing data to the new application

UNIX Systems Programming - Kay A. Robbins 2003

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

Large-Scale C++ Volume I - John Lakos 2019-12-02

Writing reliable and maintainable C++ software is hard. Designing such software at scale adds a new set of challenges. Creating large-scale systems requires a practical understanding of logical design – beyond the theoretical concepts addressed in most popular texts. To be successful on an enterprise scale, developers must also address physical design, a dimension of software engineering that may be unfamiliar even to expert developers. Drawing on over 30 years of hands-on experience building massive, mission-critical enterprise systems, John Lakos shows how to create and grow Software Capital. This groundbreaking volume lays the foundation for projects of all sizes and demonstrates the processes, methods, techniques, and tools needed for successful real-world, large-scale development. Up to date and with a solid engineering focus, Large-Scale C++, Volume I: Process and Architecture, demonstrates fundamental design concepts with concrete examples. Professional developers of all experience levels will gain insights that transform their approach to design and development by understanding how to Raise productivity by leveraging differences between infrastructure and application development Achieve exponential productivity gains through feedback and hierarchical reuse Embrace the component's role as the fundamental unit of both logical and physical design Analyze how fundamental properties of compiling and linking affect component design Discover effective partitioning of logical content in appropriately sized physical aggregates Internalize the important differences among sufficient, complete, minimal, and primitive software Deliver solutions that simultaneously optimize encapsulation, stability, and performance Exploit the nine established levelization techniques to avoid cyclic physical dependencies Use lateral designs judiciously to avoid the "heaviness" of conventional layered architectures Employ appropriate architectural insulation techniques for eliminating compile-time coupling Master the multidimensional process of designing large systems using component-based methods This is the first of John Lakos's three authoritative volumes on developing large-scale systems using C++. This book, written for fellow software practitioners, uses familiar C++ constructs to solve real-world problems while identifying (and motivating) modern C++ alternatives. Together with the forthcoming Volume II: Design and Implementation and Volume III: Verification and Testing, Large-Scale C++ offers comprehensive guidance for all aspects of large-scale C++ software development. If you are an architect or project leader, this book will empower you to solve critically important problems right now – and serve as your go-to reference for years to come. Register your book for convenient access to downloads, updates, and/or corrections as they become available. See inside book for details.

Advanced CORBA® Programming with C++ - Michi Henning 1999-02-17

Here is the CORBA book that every C++ software engineer has been waiting for. Advanced CORBA® Programming with C++ provides designers and developers with the tools required to understand CORBA technology at the architectural, design, and source code levels. This book offers hands-on explanations for building efficient applications, as well as lucid examples that provide practical advice on avoiding costly mistakes. With this book as a guide, programmers will find the support they need to successfully undertake industrial-strength CORBA development projects. The content is systematically arranged and presented so the book may be used as both a tutorial and a reference. The rich example programs in this definitive text show CORBA developers how to write clearer code that is more maintainable, portable, and efficient. The authors' detailed coverage of the IDL-to-C++ mapping moves beyond the mechanics of the APIs to discuss topics such as potential pitfalls and efficiency. An in-depth presentation of the new Portable Object Adapter (POA) explains how to take advantage of its numerous features to create scalable and high-performance servers. In addition, detailed discussion of advanced topics, such as garbage collection and multithreading, provides developers with the knowledge they need to write commercial applications. Other highlights In-depth coverage of IDL, including common idioms and design trade-offs Complete and detailed explanations of the Life Cycle, Naming, Trading, and Event Services Discussion of IIOP and implementation repositories Insight into the dynamic aspects of CORBA, such as dynamic typing and the new DynAny interfaces Advice on selecting appropriate application architectures and designs Detailed, portable, and vendor-independent source code

Software Evolution - Tom Mens 2008-01-25

This book focuses on novel trends in software evolution research and its relations with other emerging disciplines. Mens and Demeyer, both authorities in the field of software evolution, do not restrict themselves to the evolution of source code but also address the evolution of other, equally important software artifacts. This book is the indispensable source for researchers and professionals looking for an introduction and comprehensive overview of the state-of-the-art.

Understanding Software Dynamics - Richard Sites 2021-10-15

Troubleshoot and Optimize Complex, Time-Constrained Software From mobile and cloud apps to video games to driverless vehicle control, more and more software is time-constrained: It must deliver reliable results seamlessly, consistently, and virtually instantaneously. If it doesn't, customers are unhappy--and sometimes lives are put at risk. When time-constrained software underperforms or fails, software professionals must quickly identify and address the root causes. This is difficult and, historically, few tools have been available to help. In Understanding Software Dynamics, performance expert Richard L. Sites tackles the problem head on, offering expert methods and advanced tools for understanding complex time-constrained software dynamics, improving reliability, and troubleshooting challenging performance problems. Sites draws on several decades of experience pioneering software performance optimization, as well as extensive experience teaching graduate-level developers. He introduces principles and techniques for use in any environment, from embedded devices to datacenters, illuminating them with examples based on x86 or ARM processors running Linux and linked by Ethernet. He also guides readers through building and applying a powerful, new, extremely low-overhead open-source software tool, KUtrace, to precisely trace executions on every CPU core. Using insights gleaned from this tool, readers can apply nuanced solutions--not merely brute-force techniques such as turning off caches or cores. Measure and address issues associated with CPUs, memory, disk/SSD, networks, and their interactions Fix programs that are always too slow, and those that sometimes lag for no apparent reason Design useful observability, logging, and time-stamping capabilities into your code Reason more effectively about performance data to see why reality differs from expectations Identify problems such as excess execution, slow instruction execution, waiting for resources, and software locks Understanding Software Dynamics will be valuable to experienced software professionals, including application and OS developers, hardware and system architects, real-time system designers, and game developers, as well as advanced students. Register your book for convenient access to downloads, updates, and/or corrections as they become available. See inside book for details.

Programming with Curses - John Strang 1986
Understanding windows; Terminal independence; The curses library; Sample program; Quick reference.