

## Linux Kernel Documentation

The Linux Programming Interface (LPI) 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 the Linux and UNIX programming interface, and a book that's destined to become a new classic.

Whether you're just starting out with Linux or looking to hone your existing skills, this book will provide you with the knowledge you need.

To thoroughly understand what makes Linux tick and why it's so efficient, you need to delve deep into the heart of the operating system--into the Linux kernel itself. The kernel is Linux--in the case of the Linux operating system, it's the only bit of software to which the term "Linux" applies. The kernel handles all the requests or completed I/O operations and determines which programs will share its processing time, and in what order. Responsible for the sophisticated memory management of the whole system, the Linux kernel is the force behind the legendary Linux efficiency. The new edition of *Understanding the Linux Kernel* takes you on a guided tour through the most significant data structures, many algorithms, and programming tricks used in the kernel. Probing beyond the superficial features, the authors offer valuable insights to people who want to know how things really work inside their machine. Relevant segments of code are dissected and discussed line by line. The book covers more than just the functioning of the code, it explains the theoretical underpinnings for why Linux does things the way it does. The new edition of the book has been updated to cover version 2.4 of the kernel, which is quite different from version 2.2: the virtual memory system is entirely new, support for multiprocessor systems is improved, and whole new classes of hardware devices have been added. The authors explore each new feature in detail. Other topics in the book include: Memory management including file buffering, process swapping, and Direct memory Access (DMA) The Virtual Filesystem and the Second Extended Filesystem Process creation and scheduling Signals, interrupts, and the essential interfaces to device drivers Timing Synchronization in the kernel Interprocess Communication (IPC) Program execution *Understanding the Linux Kernel, Second Edition* will acquaint you with all the inner workings of Linux, but is more than just an academic exercise. You'll learn what conditions bring out Linux's best performance, and you'll see how it meets the challenge of providing good system response during process scheduling, file access, and memory management in a wide variety of environments. If knowledge is power, then this book will help you make the most of your Linux system.

Provides information on writing a driver in Linux, covering such topics as character devices, network interfaces, driver debugging, concurrency, and interrupts.

Offers a comprehensive view of the underpinnings of the Linux kernel on the Intel x86 and the Power PC.

This introduction to networking on Linux now covers firewalls, including the use of `ipchains` and `Netfilter`, masquerading, and accounting. Other new topics in this second edition include Novell (NCP/IPX) support and INN (news administration).

Summary: The Linux Kernel Book allows you to delve into the heart of this operating system by means of an in-depth treatment of the internal functioning of the kernel. Each chapter deals in detail with the system components, including: process management, memory management, IPC Systems V, signals, pipes, POSIX tty, file systems, loadable modules, and administration.

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.

For the past 20 years, UNIX insiders have cherished and zealously guarded pirated photocopies of this manuscript, a "hacker trophy" of sorts. Now legal (and legible) copies are available. An international "who's who" of UNIX wizards, including Dennis Ritchie, have contributed essays extolling the merits and importance of this underground classic.

Full coverage of the latest LPI-level 2 exams, with bonus online test bank LPIC-2 is the one-stop preparation resource for the Linux Professional Institute's Advanced Level certification exam. With 100 percent coverage of all exam objectives, this book provides clear and concise coverage of the Linux administration topics you'll need to know for exams 201 and 202. Practical examples highlight the real-world applications of important concepts, and together, the author team provides insights based on almost fifty years in the IT industry. This brand new second edition has been completely revamped to align with the latest versions of the exams, with authoritative coverage of the Linux kernel, system startup, advanced storage, network configuration, system maintenance, web services, security, troubleshooting, and more. You also get access to online learning tools including electronic flashcards, chapter tests, practice exams, and a glossary of critical terms to help you solidify your understanding of upper-level Linux administration topics. The LPI-level 2 certification confirms your advanced Linux skill set, and the demand for qualified professionals continues to grow. This book gives you the conceptual guidance and hands-on practice you need to pass the exam with flying colors. Understand

all of the material for both LPIC-2 exams Gain insight into real-world applications Test your knowledge with chapter tests and practice exams Access online study aids for more thorough preparation Organizations are flocking to the open-source Linux as an excellent, low-cost, secure alternative to expensive operating systems like Microsoft Windows. As the Linux market share continues to climb, organizations are scrambling to find network and server administrators with expert Linux knowledge and highly practical skills. The LPI-level 2 certification makes you the professional they need, and LPIC-2 is your ideal guide to getting there.

This book describes the design and implementation of the BSD operating system--previously known as the Berkeley version of UNIX. Today, BSD is found in nearly every variant of UNIX, and is widely used for Internet services and firewalls, timesharing, and multiprocessing systems. Readers involved in technical and sales support can learn the capabilities and limitations of the system; applications developers can learn effectively and efficiently how to interface to the system; systems programmers can learn how to maintain, tune, and extend the system. Written from the unique perspective of the system's architects, this book delivers the most comprehensive, up-to-date, and authoritative technical information on the internal structure of the latest BSD system. As in the previous book on 4.3BSD (with Samuel Leffler), the authors first update the history and goals of the BSD system. Next they provide a coherent overview of its design and implementation. Then, while explaining key design decisions, they detail the concepts, data structures, and algorithms used in implementing the system's facilities. As an in-depth study of a contemporary, portable operating system, or as a practical reference, readers will appreciate the wealth of insight and guidance contained in this book. Highlights of the book: Details major changes in process and memory management Describes the new extensible and stackable filesystem interface Includes an invaluable chapter on the new network filesystem Updates information on networking and interprocess communication

Provides a definitive resource for those who want to support computer peripherals under the Linux operating system, explaining how to write a driver for a broad spectrum of devices, including character devices, network interfaces, and block devices. Original. (Intermediate).

Benvenuti describes the relationship between the Internet's TCP/IP implementation and the Linux Kernel so that programmers and advanced administrators can modify and fine-tune their network environment.

Any UNIX programmer using the latest workstations or super minicomputers from vendors such as Sun, Silicon Graphics (SGI), ATandT, Amdahl, IBM, Apple, Compaq, Mentor Graphics, and Thinking Machines needs this book to optimize his/her job performance. This book teaches how these architectures operate using clear, comprehensible examples to explain the concepts, and provides a good reference for people already familiar with the basic concepts.

This collection of tips, tools, and scripts provides clear, concise, hands-on solutions that can be applied to the challenges facing anyone running a network of Linux servers from small networks to large data centers.

**ALL YOU NEED TO KNOW TO SECURE LINUX SYSTEMS, NETWORKS, APPLICATIONS, AND DATA—IN ONE BOOK**

From the basics to advanced techniques: no Linux security experience necessary Realistic examples & step-by-step activities: practice hands-on without costly equipment The perfect introduction to Linux-based security for all students and IT professionals Linux distributions are widely used to support mission-critical applications and manage crucial data. But safeguarding modern Linux systems is complex, and many Linux books have inadequate or outdated security coverage. Linux Essentials for Cybersecurity is your complete solution. Leading Linux certification and security experts William "Bo" Rothwell and Dr. Denise Kinsey introduce Linux with the primary goal of enforcing and troubleshooting security. Their practical approach will help you protect systems, even if one or more layers are penetrated. First, you'll learn how to install Linux to achieve optimal security upfront, even if you have no Linux experience. Next, you'll master best practices for securely administering accounts, devices, services, processes, data, and networks. Then, you'll master powerful tools and automated scripting techniques for footprinting, penetration testing, threat detection, logging, auditing, software management, and more. To help you earn certification and demonstrate skills, this guide covers many key topics on CompTIA Linux+ and LPIC-1 exams. Everything is organized clearly and logically for easy understanding, effective classroom use, and rapid on-the-job training. **LEARN HOW TO:** Review Linux operating system components from the standpoint of security Master key commands, tools, and skills for securing Linux systems Troubleshoot common Linux security problems, one step at a time Protect user and group accounts with Pluggable Authentication Modules (PAM), SELinux, passwords, and policies Safeguard files and directories with permissions and attributes Create, manage, and protect storage devices: both local and networked Automate system security 24/7 by writing and scheduling scripts Maintain network services, encrypt network connections, and secure network-accessible processes Examine which processes are running—and which may represent a threat Use system logs to pinpoint potential vulnerabilities Keep Linux up-to-date with Red Hat or Debian software management tools Modify boot processes to harden security Master advanced techniques for gathering system information

Up-to-the-Minute, Complete Guidance for Developing Embedded Solutions with Linux Linux has emerged as today's #1 operating system for embedded products. Christopher Hallinan's Embedded Linux Primer has proven itself as the definitive real-world guide to building efficient, high-value, embedded systems with Linux. Now, Hallinan has thoroughly updated this highly praised book for the newest Linux kernels, capabilities, tools, and hardware support, including advanced multicore processors. Drawing on more than a decade of embedded Linux experience, Hallinan helps you rapidly climb the learning curve, whether you're moving from legacy environments or you're new to embedded programming. Hallinan addresses today's most important development challenges and demonstrates how to solve the problems you're most likely to encounter. You'll learn how to build a modern, efficient embedded Linux development environment, and then utilize it as productively as possible. Hallinan offers up-to-date guidance on everything from kernel configuration and initialization to bootloaders, device drivers to file systems, and BusyBox utilities to real-time configuration and system analysis. This edition adds entirely new chapters on UDEV, USB, and open source build systems. Tour the typical embedded system and development environment and understand its concepts and components. Understand the Linux kernel and userspace initialization processes. Preview bootloaders, with specific emphasis on U-Boot.

Configure the Memory Technology Devices (MTD) subsystem to interface with flash (and other) memory devices. Make the most of BusyBox and latest open source development tools. Learn from expanded and updated coverage of kernel debugging. Build and analyze real-time systems with Linux. Learn to configure device files and driver loading with UDEV. Walk through detailed coverage of the USB subsystem. Introduces the latest open source embedded Linux build systems. Reference appendices include U-Boot and BusyBox commands.

This book contains the practical labs corresponding to the "Linux Kernel and Driver Development: Training Handouts" book from Bootlin. Get your hands on an embedded board based on an ARM processor (the Beagle Bone Black board), and apply what you learned: write a Device Tree to declare devices connected to your board, configure pin multiplexing, and implement drivers for I2C and serial devices. You will learn how to manage multiple devices with the same driver, to access and write hardware registers, to allocate memory, to register and manage interrupts, as well as how to debug your code and interpret the kernel error messages. You will also keep an eye on the board and CPU datasheets so that you will always understand the values that you feed to the kernel.

Provides help texts that contain useful information for the configuration of Linux kernel versions 1.2 and 1.3. Includes instructions on how to install the new configure script for kernel versions 1.2.x, and other kernel versions. Links to Linux Documentation Project Home Page and the home page of the author.

"Probably the most wide ranging and complete Linux device driver book I've read." --Alan Cox, Linux Guru and Key Kernel Developer "Very comprehensive and detailed, covering almost every single Linux device driver type." --Theodore Ts'o, First Linux Kernel Developer in North America and Chief Platform Strategist of the Linux Foundation  
**The Most Practical Guide to Writing Linux Device Drivers** Linux now offers an exceptionally robust environment for driver development: with today's kernels, what once required years of development time can be accomplished in days. In this practical, example-driven book, one of the world's most experienced Linux driver developers systematically demonstrates how to develop reliable Linux drivers for virtually any device. **Essential Linux Device Drivers** is for any programmer with a working knowledge of operating systems and C, including programmers who have never written drivers before. Sreekrishnan Venkateswaran focuses on the essentials, bringing together all the concepts and techniques you need, while avoiding topics that only matter in highly specialized situations. Venkateswaran begins by reviewing the Linux 2.6 kernel capabilities that are most relevant to driver developers. He introduces simple device classes; then turns to serial buses such as I2C and SPI; external buses such as PCMCIA, PCI, and USB; video, audio, block, network, and wireless device drivers; user-space drivers; and drivers for embedded Linux—one of today's fastest growing areas of Linux development. For each, Venkateswaran explains the technology, inspects relevant kernel source files, and walks through developing a complete example. • Addresses drivers discussed in no other book, including drivers for I2C, video, sound, PCMCIA, and different types of flash memory • Demystifies essential kernel services and facilities, including kernel threads and helper interfaces • Teaches polling, asynchronous notification, and I/O control • Introduces the Inter-Integrated Circuit Protocol for embedded Linux drivers • Covers multimedia device drivers using the Linux-Video subsystem and Linux-Audio framework • Shows how Linux implements support for wireless technologies such as Bluetooth, Infrared, WiFi, and cellular networking • Describes the entire driver development lifecycle, through debugging and maintenance • Includes reference appendixes covering Linux assembly, BIOS calls, and Seq files

Learn how to write high-quality kernel module code, solve common Linux kernel programming issues, and understand the fundamentals of Linux kernel internals  
**Key Features** Discover how to write kernel code using the Loadable Kernel Module framework Explore industry-grade techniques to perform efficient memory allocation and data synchronization within the kernel Understand the essentials of key internals topics such as kernel architecture, memory management, CPU scheduling, and kernel synchronization  
**Book Description** **Linux Kernel Programming** is a comprehensive introduction for those new to Linux kernel and module development. This easy-to-follow guide will have you up and running with writing kernel code in next-to-no time. This book uses the latest 5.4 Long-Term Support (LTS) Linux kernel, which will be maintained from November 2019 through to December 2025. By working with the 5.4 LTS kernel throughout the book, you can be confident that your knowledge will continue to be valid for years to come. This Linux book begins by showing you how to build the kernel from the source. Next, you'll learn how to write your first kernel module using the powerful Loadable Kernel Module (LKM) framework. The book then covers key kernel internals topics including Linux kernel architecture, memory management, and CPU scheduling. Next, you'll delve into the fairly complex topic of concurrency within the kernel, understand the issues it can cause, and learn how they can be addressed with various locking technologies (mutexes, spinlocks, atomic, and refcount operators). You'll also benefit from more advanced material on cache effects, a primer on lock-free techniques within the kernel, deadlock avoidance (with lockdep), and kernel lock debugging techniques. By the end of this kernel book, you'll have a detailed understanding of the fundamentals of writing Linux kernel module code for real-world projects and products. What you will learn Write high-quality modular kernel code (LKM framework) for 5.x kernels Configure and build a kernel from source Explore the Linux kernel architecture Get to grips with key internals regarding memory management within the kernel Understand and work with various dynamic kernel memory alloc/dealloc APIs Discover key internals aspects regarding CPU scheduling within the kernel Gain an understanding of kernel concurrency issues Find out how to work with key kernel synchronization primitives Who this book is for This book is for Linux programmers beginning to find their way with Linux kernel development. Linux kernel and driver developers looking to overcome frequent and common kernel development issues, as well as understand kernel internals, will benefit from this book. A basic understanding of Linux CLI and C programming is required.

This book will get you up to speed quickly on Fedora Linux, a securely-designed Linux distribution that includes a massive selection of free software packages. Fedora is hardened out-of-the-box, it's easy to install, and extensively customizable - and this book shows you how to make Fedora work for you.--[from publisher's description]

This book follows on from **Linux Kernel Programming**, helping you explore the Linux character device driver framework and enables you to write 'misc' class drivers. You'll learn how to efficiently interface with user apps, perform I/O on hardware memory, handle hardware interrupts, and leverage kernel delays, timers, kthreads, and workqueues.

**Linux Kernel Networking** takes you on a guided in-depth tour of the current Linux networking implementation and the theory behind it. Linux kernel networking is a complex topic, so the book won't burden you with topics not directly related to networking. This book will also not overload you with cumbersome line-by-line code walkthroughs not directly related to what you're searching for; you'll find just what you need, with in-depth explanations in each chapter and a quick reference at the end of each chapter. **Linux Kernel Networking** is the only up-to-date reference guide to understanding how networking is implemented, and it will be indispensable in years to come since so many devices now use Linux or operating systems based on Linux, like Android, and since Linux is so prevalent in the data center arena, including Linux-based virtualization technologies like Xen and KVM.

**Ubuntu Unleashed 2010 Edition** presents comprehensive coverage of the popular Ubuntu Linux distribution. Windows users, Mac users, and Linux enthusiasts have been increasingly turning to Ubuntu for a user-friendly, easy-to-use Linux distribution. This book provides detailed information on installing, using, and administering Ubuntu. You will learn how to set up a workstation or a server, and you will find complete details on Ubuntu's easy-to-use desktop and productivity software. **Ubuntu Unleashed 2010 Edition** includes a range of coverage: From the software you need in your everyday work, such as the OpenOffice.org productivity suite, to how to configure your Linux desktop to run smoothly using multiple printers, shell scripts, and more. For the hardcore Linux enthusiast, there is complete coverage of the X Window

system, Linux programming, web server administration, and network administration. Install and configure Ubuntu Get all your system's devices and peripherals up and running Configure and use the X Window System Manage Linux services and users Run a printer server Connect to a local network and the Internet Set up and administer a web server with Apache Secure your machine and your network from intruders Learn shell scripting Share files with Windows users using Samba Get productive with OpenOffice.org Play games on Linux Use Linux multimedia programs Create and maintain a MySQL database Configure a firewall Set up an FTP server Use Ubuntu's development and programming tools Tune your Ubuntu system for maximum performance Learn to manage and compile the kernel and modules

C++ is a powerful, highly flexible, and adaptable programming language that allows software engineers to organize and process information quickly and effectively. But this high-level language is relatively difficult to master, even if you already know the C programming language. The new second edition of "Practical C++ Programming is a complete introduction to the C++ language for programmers who are learning C++. Reflecting the latest changes to the C++ standard, this new edition takes a useful down-to-earth approach, placing a strong emphasis on how to design clean, elegant code. In short, to-the-point chapters, all aspects of programming are covered including style, software engineering, programming design, object-oriented design, and debugging. It also covers common mistakes and how to find (and avoid) them. End of chapter exercises help you ensure you've mastered the material. Steve Oualline's clear, easy-going writing style and hands-on approach to learning make "Practical C++ Programming a nearly painless way to master this complex but powerful programming language.

The authoritative guide to the latest Linux kernel: fully updated, with an all-new chapter on kernel data structures. \*

\*Authored by a well-known member of the Linux kernel development team with a reputation for clarity, readability, and insight. \*Covers all major subsystems and features of the latest version of the Linux 2.6.xx kernel. \*Provides examples based on real kernel code: samples that developers can use to modify and improve the Linux kernel on their own. Linux Kernel Development, 3/e, is a start-to-finish guide to the design and implementation of the latest Linux 2.6.xx kernel, written specifically for programmers who want to understand the existing kernel, write new kernel code, and write software that relies on the kernel's behavior. Author Robert Love is respected worldwide for his contributions to the Linux kernel: contributions that have improved everything from Linux preemption and process scheduling to virtual memory. In this book, he illuminates every major subsystem and feature of the current Linux kernel: their purpose, goals, design, implementation, and programming interfaces. He covers the kernel both from a theoretical and applied standpoint, helping programmers gain deep insights into operating system design as they master the skills of writing Linux kernel code. Love covers all important algorithms, relevant subsystems, process management, scheduling, time management and timers, system call interface, memory addressing, memory management, paging strategies, caching layers, VFS, kernel synchronization, signals, and more. This edition has been updated throughout to reflect changes since the original Linux kernel 2.6 was released. It also contains an entirely new chapter on kernel data structures.

RTFM Do not be afraid to ask, but first read the documentation, otherwise you might annoy others and not get help.

Micro-course discusses possible ways to obtain help from Linux: bash help, manuals and system documentation. It also explains where to find Linux documentation on the Internet.

Since the introduction of Linux version 1.2 in March 1995, a worldwide community has evolved from programmers who were attracted by the reliability and flexibility of this completely free operating system. Now at version 2.0, Linux is no longer simply the operating system of choice for hackers, but is being successfully employed in commercial software development, by Internet providers and in research and teaching. This book is written for anybody who wants to learn more about Linux. It explains the inner mechanisms of Linux from process scheduling to memory management and file systems, and will tell you all you need to know about the structure of the kernel, the heart of the Linux operating system. This New Edition: has been thoroughly updated throughout to cover Linux 2.0 shows you how the Linux operating system actually works so that you can start to program the Linux kernel for yourself introduces the kernel sources and describes basic algorithms and data structures, such as scheduling and task structure helps you to understand file systems, networking, and how systems boot The accompanying CD-ROM contains Slackware distribution 3.1 together with its complete source code, the Linux kernel sources up to version 2.0.27, the PC speaker driver, and a wealth of documentation. 0201331438B04062001

Linux Kernel Module Programming Guide is for people who want to write kernel modules. It takes a hands-on approach starting with writing a small "hello, world" program, and quickly moves from there. Far from a boring text on programming, Linux Kernel Module Programming Guide has a lively style that entertains while it educates. An excellent guide for anyone wishing to get started on kernel module programming. \*\*\* Money raised from the sale of this book supports the development of free software and documentation.

Linux Device Drivers "O'Reilly Media, Inc."

Introduces the features of the C programming language, discusses data types, variables, operators, control flow, functions, pointers, arrays, and structures, and looks at the UNIX system interface

The FreeBSD Handbook is a comprehensive FreeBSD tutorial and reference. It covers installation, day-to-day use of FreeBSD, and much more, such as the Ports collection, creating a custom kernel, security topics, the X Window System, how to use FreeBSD's Linux binary compatibility, and how to upgrade your system from source using the 'make world' command, to name a few.

This is an expert guide to the 2.6 Linux Kernel's most important component: the Virtual Memory Manager.

With Kernel Projects for Linux, Professor Gary Nutt provides a series of 12 lab exercises that illustrate how to implement core operating system concepts in the increasingly popular Linux environment. The makeup of the manual allows readers to learn concepts on a modern operating system—Linux—while at the same time viewing the source code. This hands-on

manual complements any core OS book by demonstrating how theoretical concepts are realized in Linux. Part I presents an overview of the Linux design, offering some insight into such topics as runtime organization and process, file, and device management. Part II consists of a graduated set of exercises where readers move from inspecting various aspects of the operating systems's internals to developing their own functions and data structures for the Linux kernel. This book is designed for programmers who need to learn the fundamentals of operating systems on a modern OS. The progressively harder exercises allow them to learn concepts in a hands-on setting.

Presents an overview of kernel configuration and building for version 2.6 of the Linux kernel.

Briefly, a boot loader is the first software program that runs when a computer starts. It is responsible for loading and transferring control to an operating system kernel software (such as Linux or GNU Mach). The kernel, in turn, initializes the rest of the operating system (e.g. a GNU system). GNU GRUB is a very powerful boot loader, which can load a wide variety of free operating systems, as well as proprietary operating systems with chain-loading. GRUB is designed to address the complexity of booting a personal computer; both the program and this manual are tightly bound to that computer platform, although porting to other platforms may be addressed in the future. One of the important features in GRUB is flexibility; GRUB understands filesystems and kernel executable formats, so you can load an arbitrary operating system the way you like, without recording the physical position of your kernel on the disk. Thus you can load the kernel just by specifying its file name and the drive and partition where the kernel resides. This manual is available online for free at gnu.org. This manual is printed in grayscale.

The Bash Guide for Beginners (Second Edition) discusses concepts useful in the daily life of the serious Bash user.

While a basic knowledge of shell usage is required, it starts with a discussion of shell building blocks and common practices. Then it presents the grep, awk and sed tools that will later be used to create more interesting examples. The second half of the course is about shell constructs such as loops, conditional tests, functions and traps, and a number of ways to make interactive scripts. All chapters come with examples and exercises that will help you become familiar with the theory.

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.

[Copyright: ce3aaf25b8f2a24794ae7917b90205f0](#)