Networking Device Drivers

If you want to study, build, or simply validate your thinking about modern cloud native data center networks, this is your book. Whether you're pursuing a multitenant private cloud, a network for running machine learning, or an enterprise data center, author Dinesh Dutt takes you through the steps necessary to design a data center that's affordable, high capacity, easy to manage, agile, and reliable. Ideal for network architects, data center operators, and network and containerized application developers, this book mixes theory with practice to guide you through the architecture and protocols you need to create and operate a robust, scalable network infrastructure. The book offers a vendor-neutral way to look at network design. For those interested in open networking, this book is chock-full of examples using open source software, from FRR to Ansible. In the context of a cloud native data center, you'll examine: Clos topology Network disaggregation Network operating system choices Routing protocol choices Container networking Network virtualization and EVPN Network automation

Syngress Study Guides guarantee comprehensive coverage of all exam objectives. There are no longer any short cuts or gimmicks that allow candidates to pass Microsoft's new, more rigorous exams. The days of cramming to become a "paper MCSE" are over; candidates must have a full grasp of all core concepts and plenty of hands-on experience to become certified. This book provides complete coverage of Microsoft Exam 70-291 and features one-of-a-kind integration of text, DVD-quality instructor-led training, and Web-based exam simulation and remediation, this study guide and DVD training system gives students 100% coverage of official Microsoft exam objectives plus realistic test prep. The System package consists of: 1) STUDY GUIDE. 800 pages of coverage explicitly organized in the identical structure of Microsoft's exam objectives. Sections are designed to "standalone", allowing readers to focus on those areas in which they are weakest and skim topics they may have already mastered. 2) DVD: A full hour of instructor-led training, complete with on-screen configurations and networking schematics, demystifying the toughest exam topics. 3) ONLINE PRACTICE EXAMS AND E-BOOK. Most exam candidates indicate that PRACTICE EXAMS are their single most valuable exam prep tool. Buyers of our Study Guides have immediate access to our exam simulations located at WWW.SYNGRESS.COM/SOLUTIONS. Syngress practice exams are highly regarded for rigor or the questions, the extensive explanation of the right AND wrong answers, and the direct hyperlinks from the exams to appropriate sections in the e-book for remediation. Readers will be fully prepared to pass the exam based on our 100% Certified guarantee Readers may save thousands of dollars required to purchase alternative methods of exam preparation Because of its breadth of coverage, this book will serve as a post-certification reference for IT professionals

Easy Linux Device Driver : First Step Towards Device Driver Programming Easy Linux Device Driver book is an easy and friendly way of learning device driver programming . Book contains all latest programs along with output screen screenshots. Highlighting important sections and stepwise approach helps for quick understanding of programming . Book contains Linux installation ,Hello world program up to USB 3.0 ,Display Driver ,PCI device driver programming concepts in stepwise approach. Program gives best understanding of theoretical and practical fundamentals of Linux device driver. Beginners should start learning Linux device driver from this book to become device driver expertise. Topics covered: Introduction of Linux Advantages of Linux History of Linux Architecture of Linux Definations Ubuntu installation Ubuntu Installation Steps User Interface Difference About KNOPPIX Important links Terminal: Soul of Linux Creating Root account Terminal Commands Virtual Editor Commands Linux Kernel Linux Kernel Internals Kernel Space and User space Device Driver Place of Driver in System Device Driver working Characteristics of Device Driver Module Commands Hello World Program pre-settings Write Program Printk function Makefile Run program Parameter passing Parameter passing program Parameter Array Process related program Process related program Character Device Driver Major and Minor number API to registers a device Program to show device number Character Driver File Operations File operation program. Include .h header Functions in module.h file Important code snippets Summary of file operations PCI Device Driver Direct Memory Access Module Device Table Code for Basic Device Driver Important code snippets USB Device Driver Fundamentals Architecture of USB device driver USB Device Driver program Structure of USB Device Driver Parts of USB end points Important features USB information Driver USB device Driver File Operations Using URB Simple data transfer Program to read and write Important code snippets Gadget Driver Complete USB Device Driver Program Skeleton Driver Program Special USB 3.0 USB 3.0 Port connection Bulk endpoint streaming Stream ID Device Driver Lock Mutual Exclusion Semaphore Spin Lock Display Device Driver Frame buffer concept Framebuffer Data Structure Check and set Parameter Accelerated Method Display Driver summary Memory Allocation Kmalloc Vmalloc Ioremap Interrupt Handling interrupt registration Proc interface Path of interrupt Programming Tips Softirqs, Tasklets, Work Queues I/O Control Introducing ioctl Prototype Stepwise execution of ioctl Sample Device Driver Complete memory Driver Complete Parallel Port Driver Device Driver Debugging Data Display Debugger Graphical Display Debugger Kernel Graphical Debugger Appendix I Exported Symbols Kobjects, Ksets, and Subsystems DMA I/O

Embedded Systems Architecture is a practical and technical guide to understanding the components that make up an embedded system's architecture. This book is perfect for those starting out as technical professionals such as engineers, programmers and designers of embedded systems; and also for students of computer science, computer engineering and electrical engineering. It gives a much-needed 'big picture' for recently graduated engineers grappling with understanding the design of real-world systems for the first time, and provides professionals with a systems-level picture of the key elements that can go into an embedded design, providing a firm foundation on which to build their skills. Real-world approach to the fundamentals, as well as the design and architecture process, makes this book a popular reference for the daunted or the inexperienced: if in doubt, the answer is in here! Fully updated with new coverage of FPGAs, testing, middleware and the latest programming techniques in C, plus complete source code and sample code, reference designs and tools online make this the complete package Visit the companion web site at http://booksite.elsevier.com/9780123821966/ for source code, design examples, data sheets and more A true introductory book, provides a comprehensive get up and running reference for those new to the field, and updating skills: assumes no prior knowledge beyond undergrad level electrical engineering Addresses the needs of practicing engineers, enabling it to get to the point more directly, and cover more ground. Covers hardware, software and middleware in a single volume Includes a library of design examples and design tools, plus a complete set of source code and embedded systems design tutorial materials from companion website

Linux Device Drivers

Linux Kernel in a Nutshell

Exam 70-642

Producing Device Drivers

MCSA/MCSE Implementing, Managing, and Maintaining a Microsoft Windows Server 2003 Network Infrastructure (Exam 70-291)

A Guide for the Intrepid

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.

Newly updated to include new calls and techniques introduced in Versions 2.2 and 2.4 of the Linux kernel, a definitive resource for those who want to support computer peripherals under the Linux operating system explains how to write a driver for a broad spectrum of devices, including character devices, network interfaces, and block devices. Original. (Intermediate)

Wireless home networks are better than ever! The emergence of new industry standards has made them easier, more convenient, less expensive to own and operate. Still, you need to know what to look for (and look out for), and the expert guidance you'll find in Wireless Home Networks For Dummies, 3rd Edition helps you ensure that your wire-free life is also a hassle-free life! This user-friendly, plain-English guide delivers all of the tips, tricks, and knowledge you need to plan your wireless home network, evaluate and select the equipment that will work best for you, install and configure your wireless network, and much more. You'll find out how to share your Internet connection over your network, as well as files, printers, and other peripherals. And, you'll learn how to avoid the "gotchas" that can creep in when you least expect them. Discover how to: Choose the right networking equipment Install and configure your wireless network Integrate Bluetooth into your network Work with servers, gateways, routers, and switches Connect audiovisual equipment to your wireless network Play wireless, multiuser computer games Establish and maintain your network's security Troubleshoot networking problems Improve network performance Understand 802.11n Whether you're working with Windows PCs, Mac OS X machines, or both Wireless Home Networking For Dummies, 3rd Edition, makes it fast and easy to get your wireless network up and running—and keep it that way!

Over 30 recipes to develop custom drivers for your embedded Linux applications. Key Features Use Kernel facilities to develop powerful drivers Via a practical approach, learn core concepts of developing device drivers Program a custom character device to get access to kernel internals Book Description Linux is a unified kernel that is widely used to develop embedded systems. As Linux has turned out to be one of the most popular operating systems used, the interest in developing proprietary device drivers has also increased. Device drivers play a critical role in how the system performs and ensures that the device works in the manner intended. By offering several examples on the development of character devices and how to use other kernel internals, such as interrupts, kernel timers, and wait queue, as well as how to manage a device tree, you will be able to add proper management for custom peripherals to your embedded system. You will begin by installing the Linux kernel and then configuring it. Once you have installed the system, you will learn to use the different kernel features and the character drivers. You will also cover interrupts in-depth and how you can manage them. Later, you will get into the kernel internals required for developing applications. Next, you will implement advanced character drivers and also become an expert in writing important Linux device drivers. By the end of the book, you will be able to easily write a custom character driver and kernel code as per your requirements. What you will learn Become familiar with the latest kernel releases (4.19+/5.x) running on the ESPRESSObin devkit, an ARM 64-bit machine Download, configure, modify, and build kernel sources Add and remove a device driver or a module from the kernel Master kernel programming Understand how to implement character drivers to manage different kinds of computer peripherals Become well versed with kernel helper functions and objects that can be used to build kernel applications Acquire a knowledge of in-depth concepts to manage custom hardware with Linux from both the kernel and user space Who this book is for This book will help anyone who wants to develop their own Linux device drivers for embedded systems. Having basic hand-on with Linux operating system and embedded concepts is necessary.

Architecture, Protocols, and Tools

Beginning Ubuntu Linux

Easy Linux Device Driver, Second Edition

Design and Implementation of Network Protocols in the Linux Kernel

Essential Linux Device Drivers

Where the Kernel Meets the Hardware

Windows Embedded Compact 7 is the natural choice for developing sophisticated, small-footprint devices for both consumers and the enterprise. For this latest version, a number of significant enhancements have been made, most notably the ability to run multi-core processors and address more than the 512 MB of memory constraint in previous versions. Using familiar developer tools, Pro Windows Embedded Compact 7 will take you on a deep-dive into device driver development. You ' ll learn how to set up your working environment, the tools that you ' ll need and how to think about developing for small devices before quickly putting theory into practice and developing your own first driver from the ground up. As you delve deeper into the details of driver development, you ' ll learn how to master hardware details, deal with I/O and interrupts, work with networks, and test and debug your drivers ready for deployment—all in the company of an author who's been working with Windows CE for more than a decade. Packed with code samples, Pro Windows Embedded Compact 7 contains everything you'll need to start developing for small footprint devices with confidence.

" 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

The most complete, authoritative technical guide to the FreeBSD kernel ' s internal structure has now been extensively updated to cover all major improvements between Versions 5 and 11. Approximately one-third of this edition ' s content is completely new, and another one-third has been extensively rewritten. Three long-time FreeBSD project leaders begin with a concise overview of the FreeBSD kernel ' s current design and implementation. Next, they cover the FreeBSD kernel from the system-call level down – from the interface to the kernel to the hardware. Explaining key design decisions, they detail the concepts, data structures, and algorithms used in implementing each significant system facility, including process management, security, virtual memory, the I/O system, filesystems, socket IPC, and networking. This Second Edition · Explains highly scalable and lightweight virtualization using FreeBSD jails, and virtual-machine acceleration with Xen and Virtio device paravirtualization · Describes new security features such as Capsicum sandboxing and GELI cryptographic disk protection · Fully covers NFSv4 and Open Solaris ZFS support · Introduces FreeBSD ' s enhanced volume management and new journaled soft updates · Explains DTrace ' s fine-grained process debugging/profiling · Reflects major improvements to networking, wireless, and USB support Readers can use this guide as both a working reference and an in-depth study of a leading contemporary, portable, open source operating system. Technical and sales support professionals will discover both FreeBSD ' s capabilities and its limitations. Applications developers will learn how to effectively and efficiently interface with it; system administrators will learn how to maintain, tune, and configure it; and systems programmers will learn how to extend, enhance, and interface with it. Marshall Kirk McKusick writes, consults, and teaches classes on UNIX- and BSD-related subjects. While at the University of California, Berkeley, he implemented the 4.2BSD fast filesystem. He was research computer scientist at the Berkeley Computer Systems Research Group (CSRG), overseeing development and release of 4.3BSD and 4.4BSD. He is a FreeBSD Foundation board member and a long-time FreeBSD committer. Twice president of the Usenix Association, he is also a member of ACM, IEEE, and AAAS. George V. Neville-Neil hacks, writes, teaches, and consults on security, networking, and operating systems. A FreeBSD Foundation board member, he served on the FreeBSD Core Team for four years. Since 2004, he has written the " Kode Vicious " column for Queue and Communications of the ACM. He is vice chair of ACM ' s Practitioner Board and a member of Usenix Association, ACM, IEEE, and AAAS. Robert N.M. Watson is a University Lecturer in systems, security, and architecture in the Security Research Group at the University of Cambridge Computer Laboratory. He supervises advanced research in computer architecture, compilers, program analysis, operating systems, networking, and security. A FreeBSD Foundation board member, he served on the Core Team for ten years and has been a committer for fifteen years. He is a member of Usenix Association and ACM.

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 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.

Embedded Systems Architecture

Wireless Home Networking For Dummies

Embedded Linux System Design and Development

Quality of Service in Multiservice IP Networks

The Design and Implementation of the FreeBSD Operating System

Networks

Device drivers literally drive everything you're interested in--disks, monitors, keyboards, modems--everything outside the computer chip and memory. And writing device drivers is one of the few areas of programming for the Linux operating system that calls for unique, Linux-specific knowledge. For years now, programmers have relied on the classic Linux Device Drivers from O'Reilly to master this critical subject. Now in its third edition, this bestselling guide provides all the information you'll need to write drivers for a wide range of devices.Over the years the book has helped countless programmers learn: how to support computer peripherals under the Linux operating system how to develop and write software for new hardware under Linux the basics of Linux operation even if they are not expecting to write a driver The new edition of Linux Device Drivers is better than ever. The book covers all the significant changes to Version 2.6 of the Linux kernel, which simplifies many activities, and contains subtle new features that can make a driver both more efficient and more flexible. Readers will find new chapters on important types of drivers not covered previously, such as consoles, USB drivers, and more.Best of all, you don't have to be a kernel hacker to understand and enjoy this book. All you need is an understanding of the C programming language and some background in Unix system calls. And for maximum ease-of-use, the book uses full-featured examples that you can compile and run without special hardware.Today Linux holds fast as the most rapidly growing segment of the computer market and continues to win over enthusiastic adherents in many application areas. With this increasing support, Linux is now absolutely mainstream, and viewed as a solid platform for embedded systems. If you're writing device drivers, you'll want this book. In fact, you'll wonder how drivers are ever written without it.

Included are numerous Challenge Exercises, which allow students to gain hands-on experience with networking related tools and utilities, and Challenge Scenarios.

No previous knowledge of data communications and related fields is required for understanding this text. It begins with the basic components of telephone and computer networks and their interaction, centralized and distributive processing networks, Local Area Networks (LANs), Metropolitan Area Networks (MANs), Wide Area Networks (WANs), the International Standards Organization (OSI) Management Model, network devices that operate at different layers of the OSI model, and the IEEE 802 Standards. This text also introduces several protocols including X.25, TCP/IP, IPX/SPX, NetBEUI, AppleTalk, and DNA. The physical topologies, bus, star, ring, and mesh are discussed, and the ARCNet, Ethernet, Token Ring, and Fiber Distributed Data Interface (FDDI) are described in detail. Wiring types and network adapters are well covered, and a detailed discussion on wired and wireless transmissions including Bluetooth and Wi-Fi is included. An entire chapter is devoted to the various types of networks that one can select and use for his needs, the hardware and software required, and tasks such as security and safeguarding data from internal and external disasters that the network administrator must perform to maintain the network(s) he is responsible for. Two chapters serve as introductions to the Simple Network Management Protocol (SNMP) and Remote Monitoring (RMON). This text includes also five appendices with very useful information on how computers use numbers to condition and distribute data from source to destination, and a design example to find the optimum path for connecting distant facilities. Each chapter includes True-False, Multiple-Choice, and problems to test the reader's understanding. Answers are also provided.

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.

Design and Management

Linux Networking Cookbook

Linux Device Drivers Development

Networking Device Drivers

How to Build a Digital Library

Implementation and Theory

Beginning Ubuntu Linux, Fourth Edition is the update to the bestselling book on Ubuntu, today's hottest Linux distribution. Targeting newcomers to Linux and to the Ubuntu distribution alike, readers are presented with an introduction to the world of Linux and open source community, followed by a detailed overview of Ubuntu's installation and configuration process. From there readers learn how to wield total control over their newly installed operating system, and are guided through common tasks such as writing documents, listening to audio CDs and MP3s, watching movies, using VoIP and chat, and of course general system maintenance matters. Additionally, there's a series of comprehensive tutorials on Linux internals and the command-line prompt—essential for any Linux user—and the book includes special sections on optimization, security, and system maintenance. The book comes with a DVD containing the complete Ubuntu Linux distribution. All you need to do is insert the DVD and follow the instructions in the book to install this distribution. The ultimate guide to Ubuntu, the hottest Linux distribution on the planet. Forgoes introductions to esoteric Linux topics so commonly found in other books and instead focuses on everyday tasks for everyday users: printer and file sharing configuration, office document management, and listening to MP3s and watching movies among them.

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

This unique Linux networking tutorial reference provides students with a practical overview and understanding of the implementation of networking protocols in the Linux kernel. By gaining a familiarity with the Linux kernel architecture, students can modify and enhance the functionality of protocol instances. -- Provided by publisher.

Praise for the first edition of Building Storage Networks: "This book is the Bible of storage networking" --Dave Hill, Senior Storage Analyst, the Aberdeen Group Now more than ever, especially in the age of e-commerce, data must be available and accessible 24x7 on a network. This easy-to-understand book clearly explains all the latest methods of storing data on a network, including updated coverage of Internet storage service providers.

Cloud Services, Networking, and Management

Windows NT Device Driver Development

Write custom device drivers to support computer peripherals in Linux operating systems

From Asterisk to Zebra with Easy-to-Use Recipes

Official Gazette of the United States Patent and Trademark Office

Demystifying Embedded Systems Middleware

Learn to develop customized device drivers for your embedded Linux system About This Book Learn to develop customized Linux device drivers Learn the core concepts of device drivers such as memory management, kernel caching, advanced IRQ management, and so on. Practical experience on the embedded side of Linux Who This Book Is For This book will help anyone who wants to get started with developing their own Linux device drivers for embedded systems. Embedded Linux users will benefit highly from this book. This book covers all about device driver development, from char drivers to network device drivers to memory management. What You Will Learn Use kernel facilities to develop powerful drivers Develop drivers for widely used I2C and SPI devices and use the regmap API Write and support devicetree from within your drivers Program advanced drivers for network and frame buffer devices Delve into the Linux irqdomain API and write interrupt controller drivers Enhance your skills with regulator and PWM frameworks Develop measurement system drivers with IIO framework Get the best from memory management and the DMA subsystem Access and manage GPIO subsystems and develop GPIO controller drivers In Detail Linux kernel is a complex, portable, modular and widely used piece of software, running on around 80% of servers and embedded systems in more than half of devices throughout the World. Device drivers play a critical role in how well a Linux system performs. As Linux has turned out to be one of the most popular operating systems used, the interest in developing proprietary device drivers is also increasing steadily. This book will initially help you understand the basics of drivers as well as prepare for the long journey through the Linux Kernel. This book then covers drivers development based on various Linux subsystems such as memory management, PWM, RTC, IIO, IRQ management, and so on. The book also offers a practical approach on direct memory access and network device drivers. By the end of this book, you will be comfortable with the concept of device driver development and will be in a position to write any device driver from scratch using the latest kernel version (v4.13 at the time of writing this book). Style and approach A set of engaging examples to develop Linux device drivers

Master the art of developing customized device drivers for your embedded Linux systems Key FeaturesStay up to date with the Linux PCI, ASoC, and V4L2 subsystems and write device drivers for themGet to grips with the Linux kernel power management infrastructureAdopt a practical approach to customizing your Linux environment using best practicesBook Description Linux is one of the fastest-growing operating systems around the world, and in the last few years, the Linux kernel has evolved significantly to support a wide variety of embedded devices with its improved subsystems and a range of new features. With this book, you'll find out how you can enhance your skills to write custom device drivers for your Linux operating system. Mastering Linux Device Driver Development provides complete coverage of kernel topics, including video and audio frameworks, that usually go unaddressed. You'll work with some of the most complex and impactful Linux kernel frameworks, such as PCI, ALSA for SoC, and Video4Linux2, and discover expert tips and best practices along the way. In addition to this, you'll understand how to make the most of frameworks such as NVMEM and Watchdog. Once you've got to grips with Linux kernel helpers, you'll advance to working with special device types such as Multi-Function Devices (MFD) followed by video and audio device drivers. By the end of this book, you'll be able to write feature-rich device drivers and integrate them with some of the most complex Linux kernel frameworks, including V4L2 and ALSA for SoC. What you will learnExplore and adopt Linux kernel helpers for locking, work deferral, and interrupt managementUnderstand the Regmap subsystem to manage memory accesses and work with the IRQ subsystemGet to grips with the PCI subsystem and write reliable drivers for PCI devicesWrite full multimedia device drivers using the kernel power management frameworkFind out how to get the most out of miscellaneous kernel subsystems such as NVMEM and WatchdogWho this book is for This book is for embedded developers, Linux system engineers, and system programmers who want to explore Linux kernel frameworks and subsystems. C programming skills and a basic understanding of driver development are necessary to get started with this book.

This practical technical guide to embedded middleware implementation offers a coherent framework that guides readers through all the key concepts necessary to gain an understanding of this broad topic. It integrates big picture theoretical discussion with down-to-earth advice on successful real-world use via step-by-step examples of each type of middleware implementation. It demystifies core middleware, such as networking protocols, file systems, virtual machines, and databases; more complex middleware that builds upon generic pieces, such as MOM, ORB, and RPC; and integrated middleware software packages, such as embedded JVMs, .NET, and CORBA packages. Technically detailed case studies bring it all together, by providing insight into typical engineering situations readers are likely to encounter. * The only complete guide to middleware, one of the most important AND most widely misunderstood aspects of embedded systems - hundreds of devices, from digital TVs to smart phones, can't function without it! * Offers thorough middleware coverage, including basic theory and core middleware, as well as complex implementations and integrated packages * Detailed case studies, real-world examples, hundreds of diagrams, and a free CD-ROM provide context and aid understanding of embedded middleware

Learn to develop customized device drivers for your embedded Linux systemAbout This Book* Learn to develop customized Linux device drivers* Learn the core concepts of device drivers such as memory management, kernel caching, advanced IRQ management, and so on.* Practical experience on the embedded side of LinuxWho This Book Is ForThis book will help anyone who wants to get started with developing their own Linux device drivers for embedded systems. Embedded Linux users will benefit highly from this book.This book covers all about device driver development, from char drivers to network device drivers to memory management.What You Will Learn* Use kernel facilities to develop powerful drivers* Develop drivers for widely used I2C and SPI devices and use the regmap API* Write and support devicetree from within your drivers* Program advanced drivers for network and frame buffer devices* Delve into the Linux irqdomain API and write interrupt controller drivers* Enhance your skills with regulator and PWM frameworks* Develop measurement system drivers with IIO framework* Get the best from memory management and the DMA subsystem* Access and manage GPIO subsystems and develop GPIO controller driversIn DetailLinux kernel is a complex, portable, modular and widely used piece of software, running on around 80% of servers and embedded systems in more than half of devices throughout the World. Device drivers play a critical role in how well a Linux system performs. As Linux has turned out to be one of the most popular operating systems used, the interest in developing proprietary device drivers is also increasing steadily.This book will initially help you understand the basics of drivers as well as prepare for the long journey through the Linux Kernel. This book then covers drivers development based on various Linux subsystems such as memory management, PWM, RTC, IIO, IRQ management, and so on. The book also offers a practical approach on direct memory access and network device drivers.By the end of this book, you will be comfortable with the concept of device driver development and will be in a position to write any device driver from scratch using the latest kernel version (v4.13 at the time of writing this book).Style and approachA set of engaging examples to develop Linux device drivers

Building Storage Networks

Linux Device Driver Development Cookbook

Understanding Linux Network Internals

The Linux Networking Architecture

First Step Towards Device Driver Programming

FreeBSD Device Drivers

An exhaustive technical manual outlines the Windows NT concepts related to drivers; shows how to develop the best drivers for particular applications; covers the I/O Subsystem and implementation of standard kernel mode drivers; and more. Original. (Intermediate).

Efficiently prepare for the Windows Server 2008 certification exam with MCTS: Windows Server 2008 Network Infrastructure Configuration: Exam 70-642, a comprehensive study tool that will guide you through everything you need to know for the test. This study guide includes 100% coverage of the exam objectives, real world scenarios, hands-on exercises, and challenging review questions, both in the book and on the CD. With Microsoft's release of Windows Server 2008 and a new generation of certification exams, you have more reason than ever to certify your expertise. For Instructors: Teaching supplements are available for this title.

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).

More and more, technology professionals are relying on the Web, online help, and other online information sources to solve their tough problems. Now, with O'Reilly's "Networking CD Bookshelf, Version 2.0, you can have the same convenient online access to your favorite O'Reilly books--all from your CD-ROM drive. We've packed seven of our best-selling guides onto this CD-ROM, giving you 4,016 pages of O'Reilly references and tutorials --fully searchable and cross-referenced, so you can search either the individual index for each book or the master index for the entire collection. Included are the complete, unabridged versions of these popular titles: "TCP/IP Network Administration, 3rd Edition DNS & Bind, 4th Edition Building Internet Firewalls, 2nd Edition SSH, The Secure Shell: The Definitive Guide Network Troubleshooting Tools Managing NFS & NIS, 2nd Edition Essential SNMP As a bonus, you also get the new paperback version of "TCP/IP Network Administration, 3rd Edition. Now it's easier than ever to find what you need to know about managing, administering, and protecting networks. This unique CD-ROM is a dream come true for network and system administrators--potent combination of books that offers unprecedented power and flexibility in this ever-expanding field. Formatted in HTML, "The Networking CD Bookshelf, Version 2.0, can be accessed with any web browser, so you have a complete library of technical books that you can carry with you anywhere you need it. No other resource makes so much valuable information so easy to find and so convenient to use.

The Networking CD Bookshelf

Pro Windows Embedded Compact 7

Understanding File Systems, Databases, Virtual Machines, Networking and More!

Computer Networking Illuminated

Third International Workshop, QoS-IP 2005, Catania, Italy, February 2-4, 2005

A Distribution-Neutral Guide for Servers and Desktops

Cloud Services, Networking and Management provides a comprehensive overview of the cloud infrastructure and services, as well as their underlying management mechanisms, including data center virtualization and networking, cloud security and reliability, big data analytics, scientific and commercial applications. Special features of the book include: State-of-the-art content Self-contained chapters for readers with specific interests Includes commercial applications on Cloud (video services and games)

Guide to Linux Networking and Security is a hands-on, practical guide that can be used to master Linux networking and security, in preparation for the Linux certification exams from SAIR/GNU and LPI. This book begins by introducing networking technologies and protocols, then moves into configuring a Linux network using a variety of command line and graphical utilities. Specific protocols and applications are covered in the networking chapters, including the r-utilities, NFS, Samba, and FTP, plus business-critical services such as e-mail, Web, and DNS. The second half of this book includes a discussion of security in the context of protecting business assets and user privacy, with emphasis on system administrator ethics. Cryptography and encrypted protocols lay a foundation for discussion of specific Linux security tools, including PAM, sudo, and GPG. User, file, and network security are covered. The network security discussion includes firewalls, VPNs, and utilities such as nmap, ethereal, and the SAINT profiling tool. Throughout, the book provides examples of sample commands and output, plus screen shots of related graphical utilities.

This book constitutes the refereed proceedings of the Third International Workshop on Quality of Service in Multiservice IP Networks, QoS-IP 2005, held in Catania, Italy in February 2005. The 50 revised full papers presented were carefully reviewed and selected from around 100 submissions. The papers are organized in topical sections on analytical models, traffic characterization, MPLS failure and restoration, network planning and dimensioning, DiffServ and IntServ, routing, software routers, network architectures for QoS provisioning, multiservice in wireless networks, TCP in special environments, and scheduling.

Device drivers make it possible for your software to communicate with your hardware, and because every operating system has specific requirements, driver writing is nontrivial. When developing for FreeBSD, you've probably had to scour the Internet and dig through the kernel sources to figure out how to write the drivers you need. Thankfully, that stops now. In FreeBSD Device Drivers, Joseph Kong will teach you how to master everything from the basics of building and running loadable kernel modules to more complicated topics like thread synchronization. After a crash course in the different FreeBSD driver frameworks, extensive tutorial sections dissect real-world drivers like the parallel port printer driver. You'll learn: –All about Newbus, the infrastructure used by FreeBSD to manage the hardware devices on your system –How to work with ISA, PCI, USB, and other buses –The best ways to control and communicate with the hardware devices from user space –How to use Direct Memory Access (DMA) for maximum system performance –The inner workings of the virtual null modem terminal driver, the USB printer driver, the Intel PCI Gigabit Ethernet adapter driver, and other important

drivers –How to use Common Access Method (CAM) to manage host bus adapters (HBAs) Concise descriptions and extensive annotations walk you through the many code examples. Don't waste time searching man pages or digging through the kernel sources to figure out how to make that arcane bit of hardware work with your system. FreeBSD Device Drivers gives you the framework that you need to write any driver you want, now.
MCTS Windows Server 2008 Network Infrastructure Configuration Study Guide
Patents
Develop customized drivers for embedded Linux
Running Linux
Cloud Native Data Center Networking
Linux Kernel Networking

This soup-to-nuts collection of recipes covers everything you need to know to perform your job as a Linux network administrator, whether you're new to the job or have years of experience. With Linux Networking Cookbook, you'll dive straight into the gnarly hands-on work of building and maintaining a computer network. Running a network doesn't mean you have all the answers. Networking is a complex subject with reams of reference material that's difficult to keep straight, much less remember. If you want a book that lays out the steps for specific tasks, that clearly explains the commands and configurations, and does not tax your patience with endless ramblings and meanderings into theory and obscure RFCs, this is the book for you. You will find recipes for: Building a gateway, firewall, and wireless access point on a Linux network Building a VoIP server with Asterisk Secure remote administration with SSH Building secure VPNs with OpenVPN, and a Linux PPTP VPN server Single sign-on with Samba for mixed Linux/Windows LANs Centralized network directory with OpenLDAP Network monitoring with Nagios or MRTG Getting acquainted with IPv6 Setting up hands-free networks installations of new systems Linux system administration via serial console And a lot more. Each recipe includes a clear, hands-on solution with tested code, plus a discussion on why it works. When you need to solve a network problem without delay, and don't have the time or patience to comb through reference books or the Web for answers, Linux Networking Cookbook gives you exactly what you need.

Ubuntu Linux is the fastest growing Linux-based operating system, and Beginning Ubuntu Linux, Fifth Edition teaches all of us—including those who have never used Linux—how to use it productively, whether you come from Windows or the Mac or the world of open source. Beginning Ubuntu Linux, Fifth Edition shows you how to take advantage of Lucid Lynx. Based on the best-selling previous edition, Emilio Raggi maintains a fine balance between teaching Ubuntu and introducing new features. Whether you aim to use it in the home or in the office, you'll be introduced to the world of Ubuntu Linux, from simple word processing to using cloud services. You'll learn how to control the Ubuntu system, which you just installed from the book's DVD, as you are guided through common tasks such as configuring the system's graphical user interface (GUI), listening to audio CDs and MP3s, producing documents, using VoIP and chat, and of course, general system maintenance. This book also supplies a series of comprehensive tutorials on Ubuntu administration and security—essential for any Ubuntu user—while not neglecting matters pertaining to office applications and the cloud.

The only book available on networking device drivers, this book describes the various network device driver architectures and covers the most common ones in great detail--including NDIS, 3COM and Microsoft; ODI from Novell; Packet Driver from Ftp Software; and DLPI from USL, Inc. Popular network operating systems are also covered from the device driver standpoint.

You may be contemplating your first Linux installation. Or you may have been using Linux for years and need to know more about adding a network printer or setting up an FTP server. Running Linux, now in its fifth edition, is the book you'll want on hand in either case. Widely recognized in the Linux community as the ultimate getting-started and problem-solving book, it answers the questions and tackles the configuration issues that frequently plague users, but are seldom addressed in other books. This fifth edition of Running Linux is greatly expanded, reflecting the maturity of the operating system and the teeming wealth of software available for it. Hot consumer topics suchas audio and video playback applications, groupware functionality, and spam filtering are covered, along with the basics in configuration and management that always have made the book popular. Running Linux covers basic communications such as mail, web surfing, and instant messaging, but also delves into the subtleties of network configuration--including dial-up, ADSL, and cable modems--in case you need to set up your network manually. The book canmake you proficient on office suites and personal productivity applications--and also tells you what programming tools are available if you're interested in contributing to these applications. Other new topics in the fifth edition include encrypted email and filesystems, advanced shell techniques, and remote login applications. Classic discussions on booting, package management, kernel recompilation, and X configuration have also been updated. The authors of Running Linux have anticipated problem areas, selected stable and popular solutions, and provided clear instructions to ensure that you'll have a satisfying experience using Linux. The discussion is direct and complete enough to guide novice users, while still providing the additional information experienced users will need to progress in their mastery of Linux. Whether you're using Linux on a home workstation or maintaining a network server, Running Linux will provide expert advice just when you need it.

Develop custom drivers for your embedded Linux applications
Study Guide and DVD Training System
Mastering Linux Device Driver Development
Linux Network Administrator's Guide
A Comprehensive Guide for Engineers and Programmers
Guide to Linux Networking and Security

How to Build a Digital Library reviews knowledge and tools to construct and maintain a digital library, regardless of the size or purpose. A resource for individuals, agencies, and institutions wishing to put this powerful tool to work in their burgeoning information treasuries. The Second Edition reflects developments in the field as well as in the Greenstone Digital Library open source software. In Part I, the authors have added an entire new chapter on user groups, user support, collaborative browsing, user contributions, and so on. There is also new material on content-based queries, map-based queries, cross-media queries. There is an increased emphasis placed on multimedia by adding a "digitizing" section to each major media type. A new chapter has also been added on "internationalization," which will address Unicode standards, multi-language interfaces and collections, and issues with non-European languages (Chinese, Hindi, etc.). Part II, the software tools section, has been completely rewritten to reflect the new developments in Greenstone Digital Library Software, an internationally popular open source software tool with a comprehensive graphical facility for creating and maintaining digital libraries. Outlines the history of libraries on both traditional and digital Written for both technical and non-technical audiences and covers the entire spectrum of media, including text, images, audio, video, and related XML standards Web-enhanced with software documentation, color illustrations, full-text index, source code, and more
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).
Presents an overview of kernel configuration and building for version 2.6 of the Linux kernel.