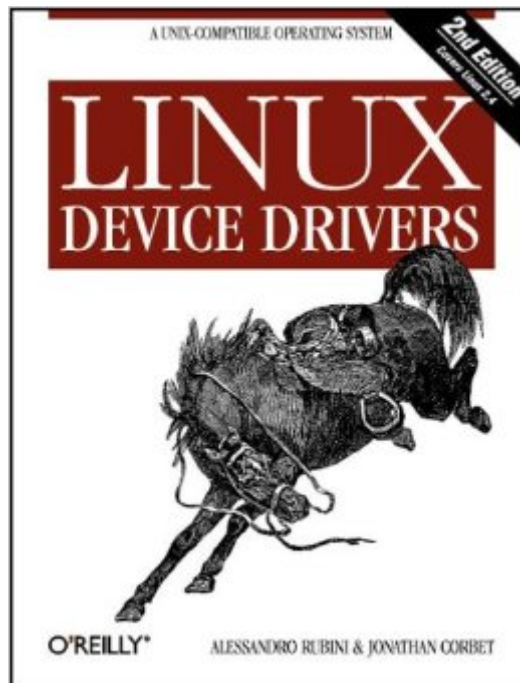**The book was found**

Linux Device Drivers, 2nd Edition

Synopsis

This book is for anyone who wants to support computer peripherals under the Linux operating system or who wants to develop new hardware and run it under Linux. Linux is the fastest-growing segment of the Unix market, is winning over enthusiastic adherents in many application areas, and is being viewed more and more as a good platform for embedded systems. Linux Device Drivers, already a classic in its second edition, reveals information that heretofore has been shared by word of mouth or in cryptic source code comments, on how to write drivers for a wide range of devices.Version 2.4 of the Linux kernel includes significant changes to device drivers, simplifying many activities, but providing subtle new features that can make a driver both more efficient and more flexible. The second edition of this book thoroughly covers these changes, as well as new processors and buses.You don't have to be a kernel hacker to understand and enjoy this book; all you need is an understanding of C and some background in Unix system calls. You'll learn how to write drivers for character devices, block devices, and network interfaces, guided by full-featured examples that you can compile and run without special hardware. Major changes in the second edition include discussions of symmetric multiprocessing (SMP) and locking, new CPUs, and recently supported buses. For those who are curious about how an operating system does its job, this book provides insights into address spaces, asynchronous events, and I/O.Portability is a major concern in the text. The book is centered on version 2.4, but includes information for kernels back to 2.0 where feasible. Linux Device Driver also shows how to maximize portability among hardware platforms; examples were tested on IA32 (PC) and IA64, PowerPC, SPARC and SPARC64, Alpha, ARM, and MIPS.Contents include:Building a driver and loading modulesComplete character, block, and network driversDebugging a driverTimingHandling symmetric multiprocessing (SMP) systemsMemory management and DMAInterruptsPortability issuesPeripheral Component Interconnect (PCI)

Book Information

Paperback: 562 pages

Publisher: O'Reilly Media; 2nd edition (June 2001)

Language: English

ISBN-10: 0596000081

ISBN-13: 978-0596000080

Product Dimensions: 7 x 1.2 x 9.2 inches

Shipping Weight: 2 pounds

Average Customer Review: 4.0 out of 5 stars See all reviews (14 customer reviews)

Best Sellers Rank: #2,720,209 in Books (See Top 100 in Books)   #46 in Books > Computers & Technology > Programming > APIs & Operating Environments > Device Drivers   #441 in Books > Computers & Technology > Hardware & DIY > Peripherals   #443 in Books > Computers & Technology > Operating Systems > Linux > Programming

Customer Reviews

I used this book to write a device driver for my computer engineering senior project. It was very helpfull, but could improve. 2nd edition covers almost everything you'll need for 2.4 kernel drivers. Organization is like a text book that includes reference material, but attempts to be a tutorial. Hopefully the 3rd edition will be better organized. I noticed lots of negative reviews on , but after reading some chapters on safari (the oreilly free book site) I decided to purchase it any ways. If you buy this book and don't have a solid background in operating systems, computer architecture, and microprocessor interfacing you probably won't have an easy time understanding several key topics well enough to write a working driver. This will probably make you mad enough to write another bad review.

Before starting let me assure you that I had written SCSI drivers for SVR4 and 4.2. I had also done some proprietary drivers for *BSDs earlier during undergrad/grad years.After reading all the reviews and all the ravings about this book, I made the mistake of spending my money on it! Alas! What a waste. Its like one gigantic piece of mish-mash-mess. All the information might be in there (who knows). But, finding out the correct info is almost impossible! The scull driver is a joke.Another irritating thing about the book is that it attempts to throw all sorts of info at you, all at the same time! Guys, take a break. Present information, one at a time, and only when needed.Essentially, I was trying to write a driver to read information off my Nikon F100 onto my TP600 running Mandrake 8.0. And life was miserable till I finally started digging through some other driver code that you get with the kernel sources. Gosh, that made life so simple.But, when I first read this book, I was almost ready to give up writing Linux drivers. I didn't know where to start and where I would end up! I was lost, confused, and felt utterly hopeless.Also, I wondered if the other guys who had raved about this book were:1. the authors and their friends?2. guys who had no clue what they were talking/writing about?3. guys who were happy to see that there was a book on Linux drivers, but neither bought nor read it!So, don't trust these reviews. This is one heck of a horrible book that you should most definitely stay away from. Cause if you read, you would never be able to write any useful

drivers.Read the source.

This is a good book - and I joined the Linux kernel immortals by writing a driver for a previously unsupported device based on it. But it does have a few flaws.Firstly, it is pretty much Intel (ix86) specific - other processors are mentioned in passing.Secondly it doesn't deal with the real world of Linux devices (ie where the manufacturer has told you little or nothing).And thirdly, it never quite goes far enough in exploring the difficult issues - I thought this was the case with IRQ handling especially.But, all in all, the best on offer and well worth buying.

This work is a great introduction to device drivers. Unfortunately, it was obsolete before it go to press... For beginners, check out Xinu, by Doug Comer. For veterans, check online for the latest... and always "use the source".

If you're looking for a good overview/reference for making device drivers happen on Linux, this book is a must-have. It covers most of the pertinent topics (incl. handling cross-platform support details) and covers them with a view on the 2.4.X kernels- whlie explaining the old functionalities all the way back to the 2.0.X kernels to enable you to write backwards compatible drivers if you so desire.

I can't understand the 5-star reviews of this book. I wonder if anyone who gave the book 5 stars ever tried to sit down and actually write a device driver. I doubt it.The book suffers from two main problems:1. Presentation is disorganized. The book reads as if the authors sat down and planned out what chapters to cover. That part is good. It does NOT seem like they planned further than that. The text within each chapter seems haphazard. Disorganized. Thrown together. The authors have no concept of when to start a new paragraph, so topics are strung together piecemeal. The whole book is confusing, making the reading of this book very frustrating.2. The material is presented at a frenetic pace. As I was reading the book, it felt like there was a conveyor belt feeding me information, getting faster and faster without a break. The authors feel like they can throw everything but the kitchen sink at you in a very short time. Information, minute details, big ideas, analogies, and code get thrown at you fast and furious. It starts at chapter one and just gets worse from there, making the reading of this book downright difficult.Difficult and confusing. A good characterization of the book.In summary, this book is NOT a tutorial. It is NOT a guide. Don't make the mistake of buying this book expecting a gentle (or even a not-so-gentle) introduction to writing device drivers.IMHO, the market is still open for a good book on the subject of writing device drivers for

Linux.

Linux Device Drivers, 2nd Edition Linux: Linux Command Line - A Complete Introduction To The Linux Operating System And Command Line (With Pics) (Unix, Linux kemel, Linux command line, ... CSS, C++, Java, PHP, Excel, code) (Volume 1) Linux Device Drivers, 3rd Edition Writing Linux Device Drivers: a guide with exercises Writing Linux Device Drivers: Lab Solutions: a guide with exercises Practical Linux Programming: Device Drivers, Embedded systems, and the Internet (with CD- ROM) (Programming Series) LINUX: Easy Linux For Beginners, Your Step-By-Step Guide To Learning The Linux Operating System And Command Line (Linux Series) Writing Windows Device Drivers Writing Device Drivers Writing OpenVMS Alpha Device Drivers in C: Developer's Guide and Reference Manual Writing Device Drivers: Tutorial and Reference Writing Windows Device Drivers Course Notes... Writing DOS Device Drivers in C Writing Os/2 Device Drivers Writing Device Drivers for Sco Unix: A Practical Approach Linux For Beginners: The Ultimate Guide To The Linux Operating System & Linux Linux Administration: The Linux Operating System and Command Line Guide for Linux Administrators CompTIA Linux+ Powered by Linux Professional Institute Study Guide: Exam LX0-103 and Exam LX0-104 (Comptia Linux + Study Guide) Introduction to 64 Bit Assembly Programming for Linux and OS X: Third Edition - for Linux and OS X Python para administracion de sistemas Unix y Linux/ Pythons for Management of Unix and Linux Sistems (Spanish Edition)