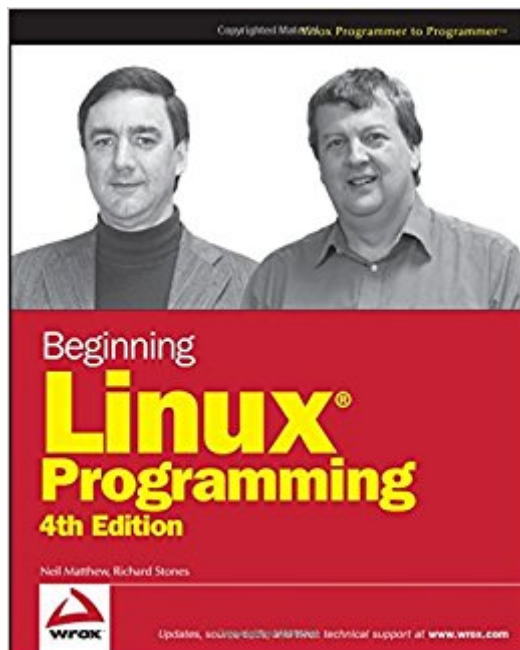The book was found

Beginning Linux Programming

Synopsis

Beginning Linux Programming, Fourth Edition continues its unique approach to teaching UNIX programming in a simple and structured way on the Linux platform. Through the use of detailed and realistic examples, students learn by doing, and are able to move from being a Linux beginner to creating custom applications in Linux. The book introduces fundamental concepts beginning with the basics of writing Unix programs in C, and including material on basic system calls, file I/O, interprocess communication (for getting programs to work together), and shell programming. Parallel to this, the book introduces the toolkits and libraries for working with user interfaces, from simpler terminal mode applications to X and GTK+ for graphical user interfaces. Advanced topics are covered in detail such as processes, pipes, semaphores, socket programming, using MySQL, writing applications for the GNOME or the KDE desktop, writing device drivers, POSIX Threads, and kernel programming for the latest Linux Kernel.

Book Information

Paperback: 816 pages

Publisher: Wrox; 4 edition (November 5, 2007)

Language: English

ISBN-10: 0470147628

ISBN-13: 978-0470147627

Product Dimensions: 7.4 x 1.8 x 9.2 inches

Shipping Weight: 2.6 pounds (View shipping rates and policies)

Average Customer Review: 4.7 out of 5 stars   22 customer reviews

Best Sellers Rank: #126,531 in Books (See Top 100 in Books)   #38 inÃ Â Books > Computers & Technology > Operating Systems > Linux > Programming   #148 inÃ Â Books > Textbooks > Computer Science > Operating Systems   #295 inÃ Â Books > Computers & Technology > Programming > Introductory & Beginning

Customer Reviews

Beginning LinuxÃ Â® Programming 4th Edition Building on the success of its previous editions, this must-have guide continues its popular tutorial approach and brings you a straightforward introduction to developing programs for Linux and other UNIX-style operating systems. The author duo of experienced Linux programmers covers a wide range of topics to help you learn more about what Linux has to offer so you can maximize your programming time and your use of the Linux system. You'll progress from the basics of compiling programs, linking to libraries, and dealing

with terminal input and output to more advanced subjects such as writing applications for the GNOMEÃ Â® and KDEÃ Â® environments, storing data using MySQLÃ Â®, and debugging. As each topic is covered, the authors introduce an appropriate programming theory and then illustrate it with practical examples, clear explanations, and a step-by-step approach with the intent that you will learn by doing. You'll quickly evolve from being a Linux beginner to confidently creating custom applications in Linux. What you will learn from this book How to use the standard Linux C libraries and other facilities Ways to make the most of the standard Linux development tools Tips on basic system calls, file I/O, interprocess communication, and shell programming How to build graphical user interfaces using the GTK+ or Qt toolkits Using sockets to support TCP/IP networking to different machines How to write programs that will work on different distributions of Linux Who this book is for This book is for programmers and developers who want to increase their skill level using Linux. Experience in C and/or C+ + programming is helpful. Wrox Beginning guides are crafted to make learning programming languages and technologies easier than you think, providing a structured, tutorial format that will guide you through all the techniques involved.

Neil Matthew has been interested in and has programmed computers since 1974. A mathematics graduate from the University of Nottingham, Neil is just plain keen on programming languages and likes to explore new ways of solving computing problems. He's written systems to program in BCPL, FP (Functional Programming), Lisp, Prolog, and a structured BASIC. He even wrote a 6502 microprocessor emulator to run BBC microcomputer programs on UNIX systems. In terms of UNIX experience, Neil has used almost every flavor since the late 1970s, including BSD UNIX, AT&T System V, Sun Solaris, IBM AIX, many others, and of course Linux. He can claim to have been using Linux since August 1993 when he acquired a floppy disk distribution of Soft Landing (SLS) from Canada, with kernel version 0.99.11. He's used Linux-based computers for hacking C, C++, Icon, Prolog, Tcl, and Java at home and at work. All of Neil's "home" projects are developed using Linux. He says Linux is much easier because it supports quite a lot of features from other systems, so that both BSD- and System V-targeted programs will generally compile with little or no change. Neil is currently working as an Enterprise Architect specializing in IT strategy at Celesio AG. He has a background in technical consultancy, software development techniques, and quality assurance. Neil has also programmed in C and C++ for real-time embedded systems. Rick Stones started programming at school (more years ago than he cares to remember) on a 6502-powered BBC micro, which, with the help of a few spare parts, continued to function for the next 15 years. He graduated from Nottingham University with a degree in Electronic Engineering, but decided software

was more fun. Over the years he has worked for a variety of companies, from the very small with just a dozen employees, to the very large, including the IT services giant EDS. Along the way he has worked on a range of projects, from real-time communications to accounting systems, to very large help desk systems. He is currently working as an IT architect, acting as a technical authority on various major projects for a large pan-European company. A bit of a programming linguist, he has programmed in various assemblers, a rather neat proprietary telecommunications language called SL-1, some FORTRAN, Pascal, Perl, SQL, and smidgeons of Python and C++, as well as C. (Under duress he even admits that he was once reasonably proficient in Visual Basic, but tries not to advertise this aberration.)

I was just beginning in Linux programming when I bought this book. It was fantastic. The authors provide clear yet concise explanations of basic Linux system calls and library functions, as well as tons of background information concerning the inner workings of Linux and basic knowledge that one coming from a Microsoft background may not have considered. The book is very well-written and is easy to follow, with some humor along the way. All of my questions of the specific subject material seemed to be answered very soon after they popped in my head.One word of caution, though. I was proficient in C# on Microsoft platforms prior to starting this book, so it was quite a shock going to pure C in Linux. I would strongly recommend a basic knowledge of C and how it differs from C++ and C# beforehand. Be sure to have a thorough understanding of pointers, pointers to arrays, pointers to multi-dimensional arrays, and anything else concerning pointers. The authors seem to expect it.All in all, though, I give it an A+!

I own a lot of IT books. This is by far the best book I have ever bought. This book doesn't teach you C or C++, but it does show you how to get started programming in a Linux Environment.I like to learn by deconstructing simple examples. I have some "headfirst" books, but they are a little too nuts for me. Maybe that series attempts to teach artists or left brained people how to think like programmers. Perhaps the 1 star reviewers should have gotten one of those instead.If your analytical and right brained, and if you like to learn by doing, look no further.Ignore all the 1 star or negative reviews, 70 (5 star) reviews cant be wrong. You need to pickup a good C or C++ programming book as a companion to this, such as C++ primer plus.This book is responsible for taking my career to the next level. Since this book I have moved on to other classics such as "linkers and loaders" and the art of GDB Debugging among many others.Believe me, if you want to get started programming in Linux, you need to start here.Finally, I would like to thank the authors for

this book because they literally hold nothing back. They give it to you straight and provide concrete code examples on which you can build your understanding through hard work and experimentation.

Great book and very useful.

Very clear for linux beginners ! I use Mac OS X and some Linux clusters in my work so I need a lot of command line and system call knowledge. This book almost answer all the questions in my mind! I would recommend some basic knowledge in C before reading this book.

Needed to get more familiar with Linux development for work. This book had good reviews so thought I'd check it out. Starts with bash and then moves to c. The application you develop the throughout the book is helpful.

Easy to understand, helps newbies get started and experienced programmers get better. One of the better programming books out there. Written by nerds for non-nerds

This book is an excellent place to start with programming in the Linux environment. Its a good overview of all bases in the Linux environment, it takes every major topic in Linux programming and gives the reader a good foundation and gives enough information to help the reader know where to go from there. If you are new to Linux programming and not sure where to start, THIS IS THE BOOK FOR YOU.

I used an earlier edition of this book as a textbook for a Linux programmingclass - it is that good. This book presumes a good knowledge of C language,but only to understand the existing code, which is available for downloadfrom the publisher. The breadth and precision of presentation, from shellbasics to introductory SQL for MySQL is impressive.

Download to continue reading...

The Linux Programming Interface: A Linux and UNIX System Programming Handbook CompTIA Linux+ Powered by Linux Professional Institute Study Guide: Exam LX0-103 and Exam LX0-104 (Comptia Linux + Study Guide) Python Programming: Python Programming for Beginners, Python Programming for Intermediates, Python Programming for Advanced C++: The Ultimate Crash Course to Learning the Basics of C++ (C programming, C++ in easy steps, C++ programming, Start coding today) (CSS,C Programming, ... Programming,PHP, Coding, Java Book 1) Beginning Linux

Programming Easy Linux For Beginners: A Complete Introduction To Linux Operating System & Command Line Fast! CompTIA Linux+ Guide to Linux Certification C++ and Python Programming: 2 Manuscript Bundle: Introductory Beginners Guide to Learn C++ Programming and Python Programming C++ and Python Programming 2 Bundle Manuscript. Introductory Beginners Guide to Learn C++ Programming and Python Programming Python Programming: The Complete Step By Step Guide to Master Python Programming and Start Coding Today! (Computer Programming Book 4) A Practical Guide to Linux Commands, Editors, and Shell Programming (3rd Edition) Linux System Programming: Talking Directly to the Kernel and C Library Assembly Language Step-by-Step: Programming with Linux Hacking University: Learn Python Computer Programming from Scratch & Precisely Learn How the Linux Operating Command Line Works: 2 Manuscript Bundle Python Programming Advanced: A Complete Guide on Python Programming for Advanced Users PYTHON: LEARN PYTHON in A Day and MASTER IT WELL. The Only Essential Book You Need To Start Programming in Python Now. Hands On Challenges INCLUDED! (Programming for Beginners, Python) Python Programming Guide + SQL Guide - Learn to be an EXPERT in a DAY!: Box Set Guide (Python Programming, SQL) The Complete Software Developer's Career Guide: How to Learn Your Next Programming Language, Ace Your Programming Interview, and Land The Coding Job Of Your Dreams Programming with MicroPython: Embedded Programming with Microcontrollers and Python CNC 50 Hour Programming Course: For lathes, ISO Standard functions, Siemens fixed cycles, parametric programming, methods of use