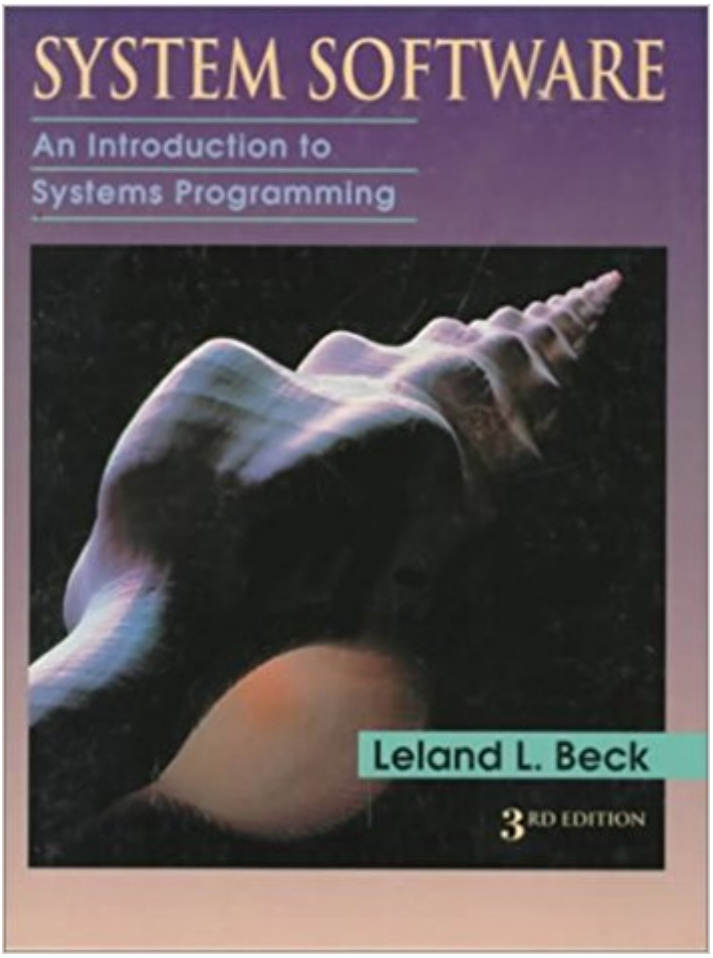
System Software: An Introduction To Systems Programming (3rd Edition)



Synopsis

In this third edition of classic title, Leland Beck provides a complete introduction to the design and implementation of various types of system software. Stressing the relationship between system software and the architecture of the machine it is designed to support, Beck first presents the fundamental concepts and basic design of each type of software in a machine-independent way. He then discusses both machine-dependent and independent extensions to the basic concepts, and gives examples of the actual system software. New FeaturesProvides updated architecture and software examples, including the Intel x86 family (Pentium, P6, etc.), IBM PowerPC, Sun SPARC, and Cray T3E. Includes an introduction to object-oriented programming and design, and illustrates these concepts of object-oriented languages, compilers, and operating systems. Brings the book up-to-speed with industry by including current operating systems topics, such as multiprocessor, distributed, and client/server systems. Contains a wide selection of examples and exercises, providing teaching support as well as flexibility, allowing you to concentrate on the software and architectures that you want to cover.

Book Information

Hardcover: 519 pages
Publisher: Pearson; 3rd Revised edition (August 24, 1996)
Language: English
ISBN-10: 0201423006
ISBN-13: 978-0201423006
Product Dimensions: 7.5 x 1.2 x 9.1 inches
Shipping Weight: 14.4 ounces (View shipping rates and policies)
Average Customer Review: 3.1 out of 5 stars 9 customer reviews
Best Sellers Rank: #75,343 in Books (See Top 100 in Books) #77 inÂ Books > Textbooks > Computer Science > Software Design & Engineering #171 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Development #176 inÂ Books > Computers & Technology > Programming > Introductory & Beginning

Customer Reviews

In this third edition of classic title, Leland Beck provides a complete introduction to the design and implementation of various types of system software. Stressing the relationship between system software and the architecture of the machine it is designed to support, Beck first presents the fundamental concepts and basic design of each type of software in a machine-independent way. He

then discusses both machine-dependent and independent extensions to the basic concepts, and gives examples of the actual system software. New Features Provides updated architecture and software examples, including the Intel x86 family (Pentium, P6, etc.), IBM PowerPC, Sun SPARC, and Cray T3E. Includes an introduction to object-oriented programming and design, and illustrates these concepts of object-oriented languages, compilers, and operating systems. Brings the book up-to-speed with industry by including current operating systems topics, such as multiprocessor, distributed, and client/server systems. Contains a wide selection of examples and exercises, providing teaching support as well as flexibility, allowing you to concentrate on the software and architectures that you want to cover. 0201423006B04062001

Leland L. Beck is a Professor of Mathematical Sciences and the Coordinator of the Computer Science Program at San Diego State University. Dr. Beck received his Bachelor of Arts in Mathematics/Physics from Rice University, and his Masters and Doctorate degrees in Systems Engineering and Computer Science from Southern Methodist University. His research interests include computer security, operating systems and software engineering. Professor Beck is also the author of System Software: An Introduction to Systems Programming. 0201423006AB04062001

Good

This book, while old, is a good introduction to the SIC/XE assembler. I used it as a reference for my Systems Software class, and it looks like i'll be able to get some use out of it when I take my compiler class as well. I must say, the book could use more reference on how to assemble complete instructions, our professor went over this in class but I found myself wishing that the text had more information on it. All in all not a bad textbook!

This book tells you the fundamentals about systems programming.At first it starts off really strong and informative, giving you details on parts of the computer that does the work while you do more than one task at a time. However, some of the info can be quite lengthy and too detailed it can get dry from time to time.Only get this book if you really need it for your class.

Update image to show first edition

After working on C, C++ and a little bit of Assembly languages, you feel excited about the

knowledge & underlying mechanisms of assemblers, loaders and linkers etc. I have gone through this book and found it extremely helpful in getting the complex concepts of system softwares. I agree with the idea of author to give the concepts on the basis of a hypothetical machine, rather than going into the complex details of any real system. This provides a big scope of the subject, givingyou the ability to feel at ease with any system.What I like most in this book is that, its not so big and with just eight chapters, you feel comfortable about the idea of reading the book.I think, its a must to read book for computer science students.

To my opinion, the merit of this book is that it gives good basic knowledge about assembler and loader. Especially, the chapter about loader is very clearly written. I think that it's better than Levin's book (Linkers and Loaders). However, the level of information is too low. If you are under-graduate student and don't have many experience about the assembly programming, this book is helpful to raise your level in system programming. But to experienced engineer, I think it will be a disappointing book. I'd like to recommend this book as the introductory book or the summary of assembler, link and loader, compiler, os, software engineering, and so forth.

I found this book to be a pretty decent guide to understanding basic compilers structures. The reason I give it a four is because it's verbiage was a bit difficult to understand at times. The author begins by using a very novel example of a simple CPU and explains how to write an assembler for it and then a compiler. When you are done reading it you should be able to go off an write a pretty simple assembler, compiler and interpreter. (I did!)

This is an excellent book. However it is not a beginners book. It is primarily a concept orientated tome, and will take a bit of work. While there is many booksthat claim to show you how computer progams run on a system, those books are mainly concerned with the hardware aspects of systems. This book completely ignores hardware and focuses exclusively on the software, which makes it a rarity. It is definetely worth 5 stars however a 4th edition is certainly due.

Innovations in Software Engineering and Software Development Series) Assessment, Evaluation, and Programming System for Infants and Children (AEPSÂ®), Second Edition, Curriculum for Three to Six Years (AEPS: Assessment, Evalutaion, and Programming System (Unnumbered)) 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 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) The Linux Programming Interface: A Linux and UNIX System Programming Handbook IEC 61511-1 Ed. 1.0 b:2003, Functional safety - Safety instrumented systems for the process industry sector - Part 1: Framework, definitions, system, hardware and software requirements Rapid Prototyping Software for Avionics Systems: Model-oriented Approaches for Complex Systems Certification (Iste) Introduction to Programming with Greenfoot: Object-Oriented Programming in Java with Games and Simulations (2nd Edition) The Software Requirements Memory Jogger: A Pocket Guide to Help Software And Business Teams Develop And Manage Requirements (Memory Jogger) Head First Software Development: A Learner's Companion to Software Development Agile Project Management: Agile Revolution, Beyond Software Limits: A Practical Guide to Implementing Agile Outside Software Development (Agile Business Leadership, Book 4) Don't Buy Software For Your Small Business Until You Read This Book: A guide to choosing the right software for your SME & achieving a rapid return on your investment Software Agreements Line by Line, 2nd ed.: A Detailed Look at Software Agreements and How to Draft Them to Meet Your Needs IEC 62304 Ed. 1.0 b:2006, Medical device software - Software life cycle processes Agile Software Development with Scrum (Series in Agile Software Development)