



Operating Systems (3rd Edition)

By Gary Nutt

Download now

Read Online 

Operating Systems (3rd Edition) By Gary Nutt

Operating Systems provides an understanding of contemporary operating system concepts by integrating the principles behind the design of all operating systems with how they are put into practice in the real world. Throughout, Gary Nutt provides a complete discussion of operating concepts and supplements this with real code examples, algorithms, and discussions about implementation issues. The book also includes many lab exercises that provide students with the opportunity to practice with Linux, UNIX, and Windows.

 [Download Operating Systems \(3rd Edition\) ...pdf](#)

 [Read Online Operating Systems \(3rd Edition\) ...pdf](#)

Operating Systems (3rd Edition)

By Gary Nutt

Operating Systems (3rd Edition) By Gary Nutt

Operating Systems provides an understanding of contemporary operating system concepts by integrating the principles behind the design of all operating systems with how they are put into practice in the real world. Throughout, Gary Nutt provides a complete discussion of operating concepts and supplements this with real code examples, algorithms, and discussions about implementation issues. The book also includes many lab exercises that provide students with the opportunity to practice with Linux, UNIX, and Windows.

Operating Systems (3rd Edition) By Gary Nutt Bibliography

- Sales Rank: #166302 in Books
- Published on: 2003-07-13
- Original language: English
- Number of items: 1
- Dimensions: 9.20" h x 2.00" w x 7.20" l, 3.40 pounds
- Binding: Hardcover
- 894 pages

 [Download Operating Systems \(3rd Edition\) ...pdf](#)

 [Read Online Operating Systems \(3rd Edition\) ...pdf](#)

Editorial Review

From the Author

Undergraduate OS textbooks vary in style from those that focus on a detailed discussion of a single OS to those that provide descriptions of concepts and issues for any OS. OS textbook styles also vary from conversational discussions to detailed discussions of meaty content on the details of an OS. Traditionally, books with meaty content are either very theoretical or mathematical, or address only a single OS; conversely, books that are easy to read tend to be so due to lack of content. Contemporary operating systems are very complex software. A book that lacks content fails to convey the requisite understanding to cope with actual operating systems; a book that focuses on only one OS does not provide the proper perspective on the discipline.

This book was written 6 years after writing a dense and formal book on OS concepts. It preserves the deep concepts, but pushes the formal treatment into later chapters (and even then, it is simplified over the earlier material). The student begins with informal explanations for the initial concepts, then gradually works up to subject matter that is described in more formal and precise terms.

The second edition of the book* also includes an extensive set of exercises for UNIX systems as well as extended discussions and examples from Windows NT, Mach, and other research OSes. This approach was taken to balance the conceptual material with concrete exercises to apply the concepts to UNIX. There are also complementary lab manuals for Windows NT exercises and Linux kernel exercises, if the student wishes to perform more in-depth experiments.

This material is reaching maturity; it has been reviewed by many professional reviewers, used by many instructors and many students. Most "errors" that remain in the code examples are due to misinterpretation of pseudo code rather than faulty algorithms.

----- * Note: All comments dated before January 2000 refer to the first edition of the book.

From the Inside Flap

Operating systems is an exciting software area because the design of an operating system exerts a major influence on the overall function and performance of the entire computer. Today there are basically three types of books that introduce this topic to undergraduates:

The first type provides a detailed description of one or more different operating systems. This approach is particularly useful if you just want to know how a particular operating system works without learning much about how all operating systems work. The second type is an evolution of the first in that it bases all its explanation around a single pedagogical system. It usually provides enough information to derive the underlying principles of good operating systems, but these principles often are obscured by the explanation of the pedagogical system. The third type is written to be independent of any particular operating system. It focuses on the theory behind all operating systems. It tends to leave you without that important link to how an operating system is really built.

I strongly believe that it is important to understand the principles behind the designs of all operating systems and to see how those principles are put into practice in real operating systems. The goal of this book is to provide a complete discussion of theory, along with an extensive set of coding and algorithm examples. The material presented is so that the student can easily differentiate among operating systems fundamentals and the detailed coding examples. Highlighted throughout are issues of performance, since that is a fundamental

factor affecting how an operating system is designed. However, the coverage of performance is not intended to be comprehensive in nature. Additional emphasis is also placed on how various parts of the system are related to real-world demand and hardware constraints. I decided to forego extensive coverage of analysis and performance theory in favor of explanation of performance issues. Students should have plenty of time to study analysis techniques in a graduate-level operating system course.

The main part of the text presents operating systems fundamentals, along with general examples. It also includes three types of examples: IN THE COCKPIT examples illustrate how to use concepts being explained in the chapter. IN THE HANGAR examples focus on how these concepts can be implemented and performance issues are highlighted in PERFORMANCE TUNING boxes. If desired, the reader could ignore these examples and focus only on operating systems theory. Or one could browse the main text but focus primarily on these examples for a course that emphasizes a hands-on approach to operating systems.

Code Examples. The intent of including a variety of code examples is to describe operating system more concretely and provide samples of the implementation of OS theory. Sometimes these examples include descriptions at the level of programs and algorithms. A few of these code examples are complete programs that have been compiled and executed. Most examples, however, are simply descriptions of algorithms or techniques using the C programming language. These descriptions deliberately omit detail that would be necessary in an actual implementation but that do not contribute to the understanding of the algorithm. The context in which the code appears should make clear when the code is an actual implementation; otherwise it should always be assumed to be a description of an algorithm or technique. I believe that detailed descriptions are mandatory for describing operating systems. I also believe that full implementations contain so many code-specific details that the essence of the ideas being illustrated are difficult to abstract and separate from the details of the language implementation. I have experimented with using pseudo code languages for these descriptions, but students and reviewers have consistently preferred the use of C. Be careful not to interpret the descriptions in C as complete implementations

Topic Order

The order of presentation is based on my experience teaching operating systems courses with input from many experienced OS instructors. This organization thus reflects the combined knowledge and experience of many different teachers and I believe the result is logical, conducive to learning, and generally accepted by most operating system instructors.

Each chapter begins with a transition from the previous chapter and a preview of what is covered in the current chapter. Students can look at this material as well as the summary at the end of the chapter to get a quick idea what a chapter is about. Chapters 1-4 consist of important introductory material that provides a solid foundation for the study of operating systems. Teachers may decide to go over this material rather quickly, perhaps assigning it as outside reading material, especially if this was covered in other courses. However, understanding this material is critical before one dives into the further study of meat of operating systems starting in Chapter 5.

Chapter 1 shows how operating system; fix into software technology, in earlier drafts, an historical perspective had been included. Instructors tend to like a little history and context but many students think it is boring, so we have dispensed with history.

Chapter 2 is unique among operating system books in that it considers how to use an operating system, particularly how to use multiple processes. This chapter was added because my experience with computer science juniors and seniors is that they may have written considerable single-threaded code but are far less likely to have written or studied multithreaded software. This chapter offers an immediate opportunity to

learn this new material.

Chapter 3 describes the fundamental organization of operating systems, including implementation strategies.

Chapter 4 finishes the preliminaries for studying operating systems-computer organization. For students who have already taken a computer organization class, the first half of Chapter 4 will be elementary. The second half describes interrupts, emphasizing the aspects that are critical to operating systems.

Chapter 5 describes device management, specifically general techniques, buffering, and device drivers. It is tempting to become completely immersed in Linux device drivers. However, I resisted this temptation to focus instead on a macro-level view of the purpose and general organization of interrupt-driven I/O. Included are extensive boxed sections on device drivers, but these stop short of providing an actual Linux driver. The chapter examines devices before considering processes because devices provide an elementary case in which physical concurrency exists in the computer and the software must be carefully designed to control that concurrency. This provides a natural introduction to process management.

Chapters 6-10 are devoted to process management. They start from the basic tasks and organization of process and resource managers (Chapter 6) and move to scheduling (Chapter 7), synchronization (Chapters 8 and 9), and deadlock (Chapter 10).

Chapter 11 deals with traditional issues in memory management, while Chapter 12 covers the contemporary approach to memory manager's using virtual memory. Because of the popularity of paging, most of the discussion is directed at this technology. However, with the current trends in memory technology, it would be a mistake to ignore segmentation. Thus part of this discussion deals with segmentation. Unfortunately, the best example of a robust segmentation system is still the (now obsolete) Multics system.

Chapter 13 describes file management. Less space is devoted to file management than is customary in operating systems books because it is not as difficult to understand as process management and memory management. This discussion is augmented in Chapter 16, which deals with remote files.

Chapter 14 provides a general discussion of protection mechanisms and security policies. It might be argued that this section belongs in the process management discussion, although much of the technology is just as closely related to files, memory, and other resources. It is much easier for someone to appreciate the need for protection and security after they have seen the process, memory, and file managers.

Chapters 15-17 introduce operating system technology to support distributed computing. Distributed computing is a dominant aspect of modern operating systems and I feel strongly that coverage of this important issue belongs in all introductory texts on operating systems.

The study of operating systems has traditionally been one of the most challenging and exciting software disciplines in computer science. I hope this book makes complex aspects of operating systems easy to understand and avoids making simple aspects boring. Good luck in your study of operating systems; I hope you enjoy it as much as I do!

To the Instructor

Thank you for choosing this book as your aide in teaching undergraduates about operating systems. Operating systems continues to be an

From the Back Cover

presenting core operating system concepts illustrated with real-world examples that provide a practical

perspective on the material. This book features a complete discussion of operating system principles, supplemented with code, algorithms, implementation issues, and lab exercises to provide a comprehensive understanding of contemporary operating system practice.

Features

Discusses the basic principles behind the design of all operating systems In the Hangar examples show how the principles are applied in practice using Linux/UNIX and the Windows NT operating systems
Performance Tuning discussions explain how system designers have exploited the basic principles to achieve higher performance

Lab Exercises allow students to gain hands-on experience with the details of how to use Linux/UNIX

0201612518B04062001

Users Review

From reader reviews:

Rebecca Clark:

Book is to be different for each and every grade. Book for children until adult are different content. As we know that book is very important normally. The book Operating Systems (3rd Edition) seemed to be making you to know about other understanding and of course you can take more information. It is rather advantages for you. The guide Operating Systems (3rd Edition) is not only giving you much more new information but also to be your friend when you experience bored. You can spend your own personal spend time to read your reserve. Try to make relationship together with the book Operating Systems (3rd Edition). You never experience lose out for everything should you read some books.

Helen Sullivan:

Do you one among people who can't read enjoyable if the sentence chained inside the straightway, hold on guys that aren't like that. This Operating Systems (3rd Edition) book is readable by you who hate those perfect word style. You will find the data here are arrange for enjoyable studying experience without leaving perhaps decrease the knowledge that want to offer to you. The writer associated with Operating Systems (3rd Edition) content conveys objective easily to understand by many individuals. The printed and e-book are not different in the articles but it just different available as it. So , do you continue to thinking Operating Systems (3rd Edition) is not loveable to be your top collection reading book?

Sandra Bryson:

Is it you who having spare time then spend it whole day by simply watching television programs or just resting on the bed? Do you need something totally new? This Operating Systems (3rd Edition) can be the reply, oh how comes? It's a book you know. You are consequently out of date, spending your extra time by reading in this fresh era is common not a nerd activity. So what these ebooks have than the others?

Martin Williams:

Some people said that they feel weary when they reading a e-book. They are directly felt it when they get a half portions of the book. You can choose the particular book Operating Systems (3rd Edition) to make your own reading is interesting. Your own skill of reading talent is developing when you such as reading. Try to choose easy book to make you enjoy to read it and mingle the sensation about book and reading through especially. It is to be initial opinion for you to like to open a book and go through it. Beside that the book Operating Systems (3rd Edition) can to be your brand-new friend when you're really feel alone and confuse with the information must you're doing of their time.

**Download and Read Online Operating Systems (3rd Edition) By
Gary Nutt #YOJZXWQ1LVS**

Read Operating Systems (3rd Edition) By Gary Nutt for online ebook

Operating Systems (3rd Edition) By Gary Nutt Free PDF d0wnl0ad, audio books, books to read, good books to read, cheap books, good books, online books, books online, book reviews epub, read books online, books to read online, online library, greatbooks to read, PDF best books to read, top books to read Operating Systems (3rd Edition) By Gary Nutt books to read online.

Online Operating Systems (3rd Edition) By Gary Nutt ebook PDF download

Operating Systems (3rd Edition) By Gary Nutt Doc

Operating Systems (3rd Edition) By Gary Nutt Mobipocket

Operating Systems (3rd Edition) By Gary Nutt EPub