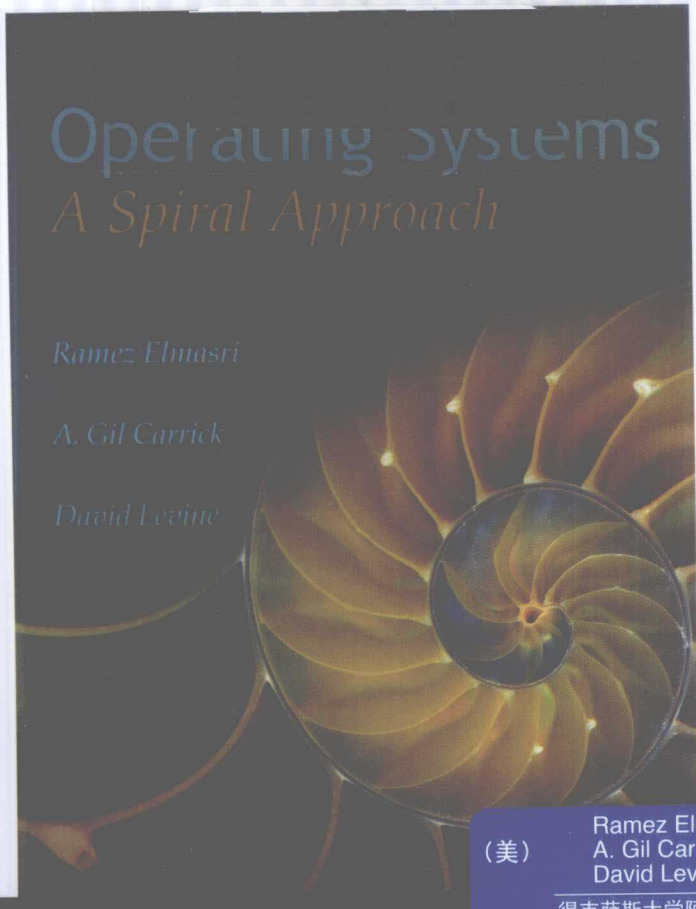
HZ BOOKS 华章教育
Mc Graw Hill Education
经 典 原 版 书 库

操作系统实用教程

螺旋方法

（英文版）



（美） Ramez Elmasri
A. Gil Carrick
David Levine 著
得克萨斯大学阿灵顿分校

机械工业出版社
China Machine Press

经 典 原 版 书 库

操作系统实用教程

螺旋方法

（英文版）

Operating Systems

A Spiral Approach

（美） Ramez Elmasri
A. Gil Carrick
David Levine 著

得克萨斯大学阿灵顿分校

机械工业出版社
China Machine Press

Ramez Elmasri, A. Gil Carrick, David Levine: Operating Systems: A Spiral Approach (ISBN: 978-0-07-244981-5).

Copyright © 2010 by The McGraw-Hill Companies, Inc.

All Rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including without limitation photocopying, recording, taping, or any database, information or retrieval system, without the prior written permission of the publisher.

This authorized Bilingual edition is jointly published by McGraw-Hill Education (Asia) and China Machine Press. This edition is authorized for sale in the People's Republic of China only, excluding Hong Kong, Macao SAR and Taiwan.

Copyright © 2010 by McGraw-Hill Education (Asia), a division of the Singapore Branch of The McGraw-Hill Companies, Inc. and China Machine Press.

版权所有。未经出版人事先书面许可，对本出版物的任何部分不得以任何方式或途径复制或传播，包括但不限于复印、录制、录音，或通过任何数据库、信息或可检索的系统。

本授权双语版由麦格劳-希尔（亚洲）教育出版公司和机械工业出版社合作出版。此版本经授权仅限在中华人民共和国境内（不包括香港、澳门特别行政区及台湾）销售。

版权 © 2010由麦格劳-希尔（亚洲）教育出版公司与机械工业出版社所有。

本书封面贴有McGraw-Hill公司防伪标签，无标签者不得销售。

封底无防伪标均为盗版

版权所有，侵权必究

本书法律顾问　北京市展达律师事务所

本书版权登记号：图字：01-2009-6759

图书在版编目（CIP）数据

操作系统实用教程：螺旋方法（英文版）/（美）埃尔玛斯里（Elmasri, R.），加里克（Carrick, A. G.），莱文（Levine, D.）著. —北京：机械工业出版社，2010.7

（经典原版书库）

书名原文：Operating Systems: A Spiral Approach

ISBN 978-7-111-31094-5

Ⅰ. 操…　Ⅱ. ① 埃…　② 加…　③ 莱…　Ⅲ. 操作系统－教材－英文　Ⅳ. TP316

中国版本图书馆CIP数据核字（2010）第118259号

机械工业出版社（北京市西城区百万庄大街22号　邮政编码　100037）

责任编辑：李俊竹

北京京师印务有限公司印刷

2010年8月第1版第1次印刷

150mm × 214mm　·　16.5印张

标准书号：ISBN 978-7-111-31094-5

定价：45.00元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：（010）88378991；88361066

购书热线：（010）68326294；88379649；68995259

投稿热线：（010）88379604

读者信箱：hzjsj@hzbook.com

出版者的话

文艺复兴以降，源远流长的科学精神和逐步形成的学术规范，使西方国家在自然科学的各个领域取得了垄断性的优势；也正是这样的传统，使美国在信息技术发展的六十多年间名家辈出、独领风骚。在商业化的进程中，美国的产业界与教育界越来越紧密地结合，计算机学科中的许多泰山北斗同时身处科研和教学的最前线，由此而产生的经典科学著作，不仅擘划了研究的范畴，还揭示了学术的源变，既遵循学术规范，又自有学者个性，其价值并不会因年月的流逝而减退。

近年，在全球信息化大潮的推动下，我国的计算机产业发展迅猛，对专业人才的需求日益迫切。这对计算机教育界和出版界都既是机遇，也是挑战；而专业教材的建设在教育战略上显得举足轻重。在我国信息技术发展时间较短的现状下，美国等发达国家在其计算机科学发展的几十年间积淀和发展的经典教材仍有许多值得借鉴之处。因此，引进一批国外优秀计算机教材将对我国计算机教育事业的发展起到积极的推动作用，也是与世界接轨、建设真正的世界一流大学的必由之路。

机械工业出版社华章公司较早意识到“出版要为教育服务”。自1998年开始，我们就将工作重点放在了遴选、移译国外优秀教材上。经过多年的不懈努力，我们与Pearson，McGraw-Hill，Elsevier，MIT，John Wiley & Sons，Cengage等世界著名出版公司建立了良好的合作关系，从他们现有的数百种教材中甄选出Andrew S. Tanenbaum，Bjarne Stroustrup，Brain W. Kernighan，Dennis Ritchie，Jim Gray，Afred V. Aho，John E. Hopcroft，Jeffrey D. Ullman，Abraham Silberschatz，William Stallings，Donald E. Knuth，John L. Hennessy，Larry L. Peterson等大师名家的一批经典作品，以“计算机科学丛书”为总称出版，供读者学习、研究及珍藏。大理石纹理的封面，也正体现了这套丛书的品位和格调。

“计算机科学丛书”的出版工作得到了国内外学者的鼎力襄助，国内的专家不仅提供了中肯的选题指导，还不辞劳苦地担任了翻译和审校的工作；而原书的作者也相当关注其作品在中国的传播，有的还专程为其书的

中译本作序。迄今，“计算机科学丛书”已经出版了近两百个品种，这些书籍在读者中树立了良好的口碑，并被许多高校采用为正式教材和参考书籍。其影印版“经典原版书库”作为姊妹篇也被越来越多实施双语教学的学校所采用。

权威的作者、经典的教材、一流的译者、严格的审校、精细的编辑，这些因素使我们的图书有了质量的保证。随着计算机科学与技术专业学科建设的不断完善和教材改革的逐渐深化，教育界对国外计算机教材的需求和应用都将步入一个新的阶段，我们的目标是尽善尽美，而反馈的意见正是我们达到这一终极目标的重要帮助。华章公司欢迎老师和读者对我们的工作提出建议或给予指正，我们的联系方法如下：

华章网站：www.hzbook.com
电子邮件：hzjsj@hzbook.com
联系电话：（010）88379604
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

华章科技图书出版中心

此为试读，需要完整PDF请访问：www.ertongbook.com

Dedication

"To peace, knowledge, and freedom."

—Ramez Elmasri

"To Judith, whose limited patience was strongly tested."

—Gil Carrick

"To close family and friends."

—David Levine

Preface

WHY WE WROTE YET ANOTHER OPERATING SYSTEMS BOOK

We have long felt that the traditional approach to teaching about Operating Systems (OSs) was not the best approach. The purpose of this book is to support a different approach to this task. When studying any complex domain of knowledge, the order in which one learns the hierarchy of principles, laws, ideas, and concepts can make the process easier or more difficult. The most common technique is to partition the subject into major topics and then study each one in great detail. For OSs, this has traditionally meant that after a brief introduction to some terms and an overview, a student studied isolated topics in depth—processes and process management, then memory management, then file systems, and so on. We can call this a depth-oriented approach or a vertical approach. After learning a great mass of unrelated details in these isolated topic areas, the student then examined case studies, examples of real OSs, and finally saw how the different topics fit together to make a real OS.

We believe that a better model is that followed by children when learning a language: learn a few words, a little grammar, a little sentence structure, and then cycle (or spiral) through; more words, more grammar, more sentence structure. By continuing to spiral through the same sequence, the complexity of the language is mastered. We can call this a breadth-oriented or spiral approach.

We have taken this approach to the subject of OSs. The first few chapters give some basic background and definitions. We then begin to describe a very simple OS in a simple system—early PCs—and evolve toward more complex systems with more features: first limited background tasks (such as simultaneous printing), then multitasking, and so on. In each case we try to show how the increasing requirements caused each system to be designed the way it was. This is not specifically a historical order of OS development. Rather, we choose a representative system at each complexity level in order to see how the different OS components interact with and influence one another. It is our belief that this approach will give the student a greater appreciation of how the various features of each level of OS were put together.

Part of the motivation for this approach has to do with why Computing Science students are told they must study OSs at all. It is highly unlikely that many of these students will work on the development of OSs. However, virtually every system that they do work on will run on top of an OS, though perhaps a very few will work on embedded systems with no OS. For the rest of them, the OS will stand between the applications and the hardware, and failure to thoroughly understand the nature of the OS will mean that these applications will be underperforming at best and hazardous at worst. We believe that our approach will lead students to a better understanding of the entire architecture of modern OSs than does the traditional approach.

THE ORGANIZATION OF THE BOOK

In Part 1 of the book we give some general background information. This information will cover basic principles of OSs and show several different views of an OS. It will also include an overview of typical computer hardware that an OS controls. Another chapter addresses such basic concepts as processes, multiprogramming, time sharing, resource management, and different approaches to OS architecture.

Then in Part 2 of the book, we will cover five types of operating systems in increasing order of complexity, our spiral approach, as follows:

1. A simple single-process OS (CPM)
2. A more complex OS (Palm OS), which allows simple system multitasking
3. An OS with full multitasking for a single user (Apple Mac OS, pre-OS X)
4. An OS that supports multiple users (Linux)
5. A distributed OS (mostly Globus)

In each case we have selected an OS that is typical of the class on which to base the discussion so as to make it more concrete. This selection was made with an eye to practicality. We first discuss simple systems in terms of process, memory, file, and I/O management, and then (slowly) move to more complex systems, gradually introducing such concepts as multitasking, time sharing, networking, security, and other issues. Occasionally we will also mention other well-known OSs that are examples of a class, such as MS-DOS in Chapter 3 and the Symbian OS in Chapter 4.

In Parts 3–5 of the book, we move to an in-depth approach of covering each OS topic in more detail: from processes to memory management to file systems. We also discuss many recent issues in operating systems: threading, object orientation, security, and approaches to parallel and distributed systems. In these chapters we revisit the sample systems discussed in Part 2 and explain the mechanisms in more detail, especially for the modern OSs.

In Part 6 we look more closely at several OSs in what are typically called case studies. Now that we know more about the details, we look at some systems in more depth and see how some features were implemented. In two cases we are revisiting more closely OSs that were covered in Part 2.

An appendix covers basic computer hardware architecture for those institutions that do not require such a course as a prerequisite for an Operating Systems course. It can also be used as a reference for those who need to review a specific topic.

THE STYLE OF THE BOOK

- We use a conversational style to avoid boring the students with excessive pedantry.
- We avoid the use of excessive formalisms. A more formal presentation is provided where needed. This choice stems from our belief that most students will not develop OSs, but rather will use them to support applications.

- We use the normal, accepted terms but also discuss alternative terms when no accepted standard terminology exists or where other terms were used historically.
- We discuss algorithmic solutions as opposed to listing actual code since students at different schools will have been exposed to different languages.
- For each OS that is treated separately, whether in the spiral section or in the case studies, we include some history of the industry at the time, and sometimes the key companies or individuals involved. This follows from our basic belief that a student can understand OSs better if they are placed in a meaningful context.
- We cover modern OSs found in devices not conventionally regarded as computers since the students use these devices every day and have an operational familiarity with them.
- Frequent figures are incorporated as an aid to those who learn best visually rather than by reading sequences of words.
- Each chapter ends with a set of questions that a student can use to assess the level of understanding of the material in the chapter.
- Projects are outlined for many chapters, which can be used by the instructor to ground the students' understanding in reality.

HOW TO USE THIS BOOK—FOR INSTRUCTORS

This text is intended to be used for a one-semester undergraduate course in Operating Systems, probably in the junior or senior year. The first part of the book is designed to consolidate basic background information necessary for the following chapters. Chapter 1 sets the discussion and gives some historical perspective. The instructor can skim this chapter and decide what to include. The appendix is a brief look at fairly modern hardware architectures. If a course in hardware is not a prerequisite for this course, then this appendix could be included. Chapter 2 defines some simple terms used in OSs and offers some more perspective on the larger topic of OS design. Again, an instructor can review this chapter and select different parts to include or exclude.

Part 2 begins the spiral approach. We believe this is a significant portion of the book. Here the student is gradually introduced to a series of OSs with more complex goals. These increasingly more complex goals lead to increasingly more complex OSs. Only two of these chapters are not normal topics in OS texts—Chapter 4 on a single-user multitasking operating system and Chapter 7 on a distributed operating system. They could be left out at the instructor's discretion, but more and more students will be working in such environments as users and as programmers.

Part 3 begins the in-depth chapters. Each chapter is fairly independent of the others, though Chapters 12 and 13 are strongly related. Beginning with Chapter 14 the individual chapters can probably be left out if the topic is the major subject of another course that the students will be required to take.

Notes about the bibliographies: The chapters in Part 3 all include a bibliography section. The reference papers that are cited are widely regarded as being seminal papers or good summaries. They may cover material that is not covered in the text. If an instructor or a student is looking for material to provide a better understanding of a given topic, then they are suggested reading.

HOW TO USE THIS BOOK—FOR STUDENTS

For students the most important thing about using this text is to understand how one learns best. There are many pathways to get information into the brain. The book itself directly addresses two of these pathways. There is obviously the text for those who learn best through reading the words and the illustrations for those who are more visually oriented. When you attend the lectures you will hear the instructor talk about the material. This is for those who learn best through hearing words. At the same time, the instructor will probably use visual aids such as the PowerPoint slides that are available for the text. Again, this is to the benefit of those who learn best by reading the words and seeing the illustrations. Some students learn best from mechanical skills, so the process of outlining the material or making study notes works well for those students.

Also presented in the book at the end of each chapter are questions about the material. These questions are designed such that a student who has a reasonable grasp of the material should be able to answer the question.

As new information is presented to the brain it takes a certain amount of time to link with other information already there. But the brain gets much information during the day that is not significant and therefore it does not retain it. Only when presented with the same or similar material again a short time later will the brain retain a significant amount of the information. The more different mechanisms that are used and the more times the information is repeated, the stronger the retention of the material. So the best method is to use all these methods combined, focusing on what works best for you. What we have found works well for most students is the following sequence:

- Print the slides to be covered in the next section, with several slides per page.
- Read the assigned material in the text. Note questions on the slide printouts.
- Come to class and listen to the instructor, amplifying any notes, especially things the instructor says that are not in the text. (Those points are favorite issues for the instructor and they tend to show up on exams.)
- Ask questions about things that are unclear.
- When it is time to review the material for an exam, go over the slides. If there are points that are unclear, go back to the text to fill them in. If any questions remain, then contact the instructor or teaching assistants.
- The review questions can be studied at any time the student finds convenient.

AVAILABLE RESOURCES FOR INSTRUCTORS

The text is supported by a website with separate sections for instructors and students.

- Supplements to the text will be made from time to time as the need presents itself.
- A set of suggested projects will be available for instructors. Most of these projects will have been used by the authors. They should be sufficiently rich and OS independent that they can be readily adapted to fit any situation. They are not based on any specific package that the instructor, students, or assistants will have to master in order to work the labs.
- PowerPoint slides are provided for the students to use, as described earlier. Instructors are encouraged to modify these presentations to fit their needs. Acknowledgement of their source is requested.
- Review question answers are provided for the instructors in order that they not be embarrassed by not knowing some arcane point the authors thought was important.
- A current list of errata will be maintained on the website.
- Reference to web resources are provided for many chapters, but the web is very volatile. The website for the book will contain an up-to-date set of web references.

ACKNOWLEDGMENTS

This text has actually been developing for longer than we would like to remember. The people at McGraw-Hill have been exceptionally patient with us. In particular, we would like to thank the following folks with McGraw-Hill: Melinda Bilecki, Kay Brimeyer, Brenda Rolwes, Kara Kudronowicz, Faye Schilling, and Raghu Srinivasan. We would also like to thank Alan Apt and Emily Lupash, who were our editors when we started working on the book. Finally, we also thank Erika Jordan and Laura Patchkofsky with Pine Tree Composition.

The chapter on Windows Vista was reviewed by Dave Probert of Microsoft. He provided valuable feedback on some items we had only been able to speculate on and brought several problems to our attention. His participation was arranged by Arkady Retik, also with Microsoft Corporation. Two chapters were reviewed by our fellow faculty members at University of Texas, Arlington. These included Yonghe Liu who reviewed the chapter on networking and Matthew Wright who reviewed the chapter on protection and security. Another faculty member, Bahram Khalili, used drafts of the text in his OS class. Naturally any remaining problems are our responsibility and not theirs.

We have used drafts of these materials in our teaching for several years and we wish to thank all our students for their feedback. In particular we wish to thank the following students: Zaher Naarane, Phil Renner, William Peacock, Wes Parish, Kyle D. Witt, David M. Connelly, and Scott Purdy.

此为试读,需要完整PDF请访问: www.ertongbook.com

REMAINING ERRORS

One difficulty with working on a project with multiple authors is that with the best of intentions, one of the writers can alter a bit of text that he himself did not write, thinking that he is clearing up some minor point, but actually altering the meaning in some subtle but important way. Accordingly, you may find minor errors in the text. Naturally these errors were not the fault of the original author, who doubtless wrote the original text correctly, but were introduced by another well-meaning author who was not as familiar with the material.

Still, such errors may be present, and we must deal with them. So, if you do find errors, we would be very happy to know about them. We will publish any errata, fix them in the next edition, determine who is to blame, and deal with the offending authors appropriately.

Table of Contents

Part 1

Operating Systems Overview and Background

In this part:

本书的第一部分包含两章。第 1 章主要介绍操作系统是什么，包括操作系统如何为用户和程序员提供服务，这些服务使得我们不需要知道底层神秘的细节就可以使用计算机，从而能够将精力集中于关注上层的问题求解。这些问题可能是任意的，不仅包括我们通常考虑的计算活动，也包括玩游戏、动态生成艺术作品以及检测汽车引擎性能的活动等。

第 2 章，关于操作系统概念、构件和体系结构提供一个初步的概观，并介绍一些通用的术语，为读者学习本书第二部分更复杂的操作系统的相关知识打下铺垫。

此为试读，需要完整PDF请访问：www.ertongbook.com