

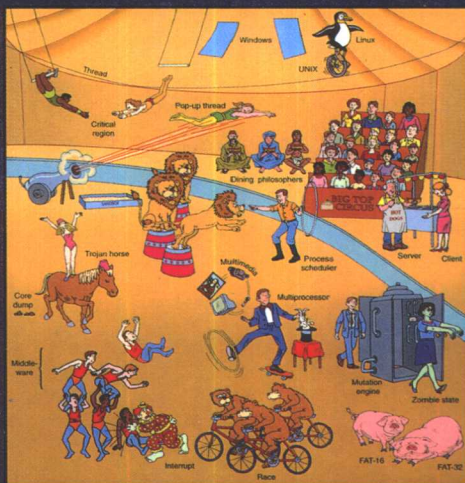
经 典 原 版 书 库

# 现代操作系统

(英文版·第2版)

## MODERN OPERATING SYSTEMS

SECOND EDITION



Andrew S. Tanenbaum

(荷) Andrew S. Tanenbaum 著



机械工业出版社  
China Machine Press

Prentice Hall

经典原版书库

(英文版·第2版)

# 现代操作系统

Modern Operating Systems  
(Second Edition)

(荷) Andrew S. Tanenbaum 著



机械工业出版社  
China Machine Press

English reprint edition copyright © 2002 by PEARSON  
EDUCATION ASIA LIMITED and CHINA MACHINE PRESS.

Modern Operating Systems, Second Edition by Andrew S.  
Tanenbaum, Copyright © 2001. All rights reserved. Published by  
arrangement with Pearson Education, Inc.

本书英文影印版由美国Prentice Hall公司授权机械工业出版社  
在中国大陆境内独家出版发行，未经出版者许可，不得以任何方  
式抄袭、复制或节录本书中的任何部分。

版权所有，侵权必究。

本书版权登记号：图字：01-2001-3762

图书在版编目(CIP)数据

M/S 311/06

现代操作系统(英文版·第2版)/(荷)特纳鲍姆  
(Tanenbaum, A. S.)著.-北京:机械工业出版社, 2002.1

(经典原版书库)

ISBN 7-111-09156-6

I. 现… II. 特… III. 操作系统 IV. TP316

中国版本图书馆CIP数据核字(2001)第051445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:华章

北京昌平奔腾印刷厂印刷·新华书店北京发行所发行

2002年1月第1版第1次印刷

850mm×1168mm 1/32·30.875印张

印数:0 001-4 000册

定价:48.00元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换

# 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，还揭橥了学术的源变，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、从业人员较少的现状下，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图文信息有限公司较早意识到“出版要为教育服务”。自1998年始，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall, Addison-Wesley, McGraw-Hill, Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum, Stroustrup, Kernighan, Jim Gray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、研究及收藏。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，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，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，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步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步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：针对本科生的核心课程，剔抉外版菁华而成“国外经典教材”系列；对影印版的教材，则单独开辟出“经典原版书库”；定位在高级教程和专业参考的“计算机科学丛书”还将保持原来的风格，继续出版新的品种。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

“经典原版书库”是响应教育部提出的使用原版国外教材的号召，为国内高校的计算机教学度身订造的。在广泛地征求并听取丛书的“专家指导委员会”的意见后，我们最终选定了这30多种篇幅内容适度、讲解鞭辟入里的教材，其中的大部分已经被M.I.T.、Stanford、U.C. Berkley、C.M.U.等世界名牌大学采用。丛书不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

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，但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。教材的出版只是我们的后续服务的起点。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，我们的联系方式如下：

电子邮件：[hzedu@hzbook.com](mailto:hzedu@hzbook.com)

联系电话：(010) 68995265

联系地址：北京市西城区百万庄南街1号

邮政编码：100037

# 专家指导委员会

(按姓氏笔画顺序)

尤晋元	王 珊	冯博琴	史忠植	史美林
石教英	吕 建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周克定	周傲英	孟小峰	岳丽华	范 明
郑国梁	施伯乐	钟玉琢	唐世渭	袁崇义
高传善	梅 宏	程 旭	程时端	谢希仁
裘宗燕	戴 葵			

# PREFACE

The world has changed a great deal since the first edition of this book appeared in 1992. Computer networks and distributed systems of all kinds have become very common. Small children now roam the Internet, where previously only computer professionals went. As a consequence, this book has changed a great deal, too.

The most obvious change is that the first edition was about half on single-processor operating systems and half on distributed systems. I chose that format in 1991 because few universities then had courses on distributed systems and whatever students learned about distributed systems had to be put into the operating systems course, for which this book was intended. Now most universities have a separate course on distributed systems, so it is not necessary to try to combine the two subjects into one course and one book. This book is intended for a first course on operating systems, and as such focuses mostly on traditional single-processor systems.

I have coauthored two other books on operating systems. This leads to two possible course sequences.

Practically-oriented sequence:

1. Operating Systems Design and Implementation by Tanenbaum and Woodhull
2. Distributed Systems by Tanenbaum and Van Steen

Traditional sequence:

1. Modern Operating Systems by Tanenbaum
2. Distributed Systems by Tanenbaum and Van Steen

The former sequence uses MINIX and the students are expected to experiment with MINIX in an accompanying laboratory supplementing the first course. The latter sequence does not use MINIX. Instead, some small simulators are available that can be used for student exercises during a first course using this book. These simulators can be found starting on the author's Web page: [www.cs.vu.nl/~ast/](http://www.cs.vu.nl/~ast/) by clicking on [Software and supplementary material for my books](#).

In addition to the major change of switching the emphasis to single-processor operating systems in this book, other major changes include the addition of entire chapters on computer security, multimedia operating systems, and Windows 2000, all important and timely topics. In addition, a new and unique chapter on operating system design has been added.

Another new feature is that many chapters now have a section on research about the topic of the chapter. This is intended to introduce the reader to modern work in processes, memory management, and so on. These sections have numerous references to the current research literature for the interested reader. In addition, Chapter 13 has many introductory and tutorial references.

Finally, numerous topics have been added to this book or heavily revised. These topics include: graphical user interfaces, multiprocessor operating systems, power management for laptops, trusted systems, viruses, network terminals, CD-ROM file systems, mutexes, RAID, soft timers, stable storage, fair-share scheduling, and new paging algorithms. Many new problems have been added and old ones updated. The total number of problems now exceeds 450. A solutions manual is available to professors using this book in a course. They can obtain a copy from their local Prentice Hall representative. In addition, over 250 new references to the current literature have been added to bring the book up to date.

Despite the removal of more than 400 pages of old material, the book has increased in size due to the large amount of new material added. While the book is still suitable for a one-semester or two-quarter course, it is probably too long for a one-quarter or one-trimester course at most universities. For this reason, the book has been designed in a modular way. Any course on operating systems should cover chapters 1 through 6. This is basic material that every student should know.

If additional time is available, additional chapters can be covered. Each of them assumes the reader has finished chapters 1 through 6, but Chaps. 7 through 12 are each self contained, so any desired subset can be used and in any order, depending on the interests of the instructor. In the author's opinion, Chaps. 7 through 12 are much more interesting than the earlier ones. Instructors should tell their students that they have to eat their broccoli before they can have the double chocolate fudge cake dessert.

I would like to thank the following people for their help in reviewing parts of the manuscript: Rida Bazzi, Riccardo Bettati, Felipe Cabrera, Richard Chapman, John Connely, John Dickinson, John Elliott, Deborah Frincke, Chandana Gamage, Robbert Geist, David Golds, Jim Griffioen, Gary Harkin, Frans Kaashoek, Muk-



kaj Krishnamoorthy, Monica Lam, Jussi Leiwo, Herb Mayer, Kirk McKusick, Evi Nemeth, Bill Potvin, Prasant Shenoy, Thomas Skinner, Xian-He Sun, William Terry, Robbert Van Renesse, and Maarten van Steen. Jamie Hanrahan, Mark Russinovich, and Dave Solomon were enormously knowledgeable about Windows 2000 and very helpful. Special thanks go to Al Woodhull for valuable reviews and thinking of many new end-of-chapter problems.

My students were also helpful with comments and feedback, especially Staas de Jong, Jan de Vos, Niels Drost, David Fokkema, Auke Folkerts, Peter Groenewegen, Wilco Ibes, Stefan Jansen, Jeroen Ketema, Joeri Mulder, Irwin Oppenheim, Stef Post, Umar Rehman, Daniel Rijkhof, Maarten Sander, Maurits van der Schee, Rik van der Stoel, Mark van Driel, Dennis van Veen, and Thomas Zeeman.

Barbara and Marvin are still wonderful, as usual, each in a unique way. Finally, last but not least, I would like to thank Suzanne for her love and patience, not to mention all the *druiven* and *kersen*, which have replaced the *sinasappelsap* in recent times.

Andrew S. Tanenbaum

# CONTENTS

## PREFACE

## 1 INTRODUCTION

1

- 1.1. WHAT IS AN OPERATING SYSTEM? 3
  - 1.1.1. The Operating System as an Extended Machine 3
  - 1.1.2. The Operating System as a Resource Manager 5
- 1.2. HISTORY OF OPERATING SYSTEMS 6
  - 1.2.1. The First Generation (1945-55) 6
  - 1.2.2. The Second Generation (1955-65) 7
  - 1.2.3. The Third Generation (1965-1980) 9
  - 1.2.4. The Fourth Generation (1980-Present) 13
  - 1.2.5. Ontogeny Recapitulates Phylogeny 16
- 1.3. THE OPERATING SYSTEM ZOO 18
  - 1.3.1. Mainframe Operating Systems 18
  - 1.3.2. Server Operating Systems 19
  - 1.3.3. Multiprocessor Operating Systems 19
  - 1.3.4. Personal Computer Operating Systems 19
  - 1.3.5. Real-Time Operating Systems 19
  - 1.3.6. Embedded Operating Systems 20
  - 1.3.7. Smart Card Operating Systems 20

- 1.4. COMPUTER HARDWARE REVIEW 20
  - 1.4.1. Processors 21
  - 1.4.2. Memory 23
  - 1.4.3. I/O Devices 28
  - 1.4.4. Buses 31
  
- 1.5. OPERATING SYSTEM CONCEPTS 34
  - 1.5.1. Processes 34
  - 1.5.2. Deadlocks 36
  - 1.5.3. Memory Management 37
  - 1.5.4. Input/Output 38
  - 1.5.5. Files 38
  - 1.5.6. Security 41
  - 1.5.7. The Shell 41
  - 1.5.8. Recycling of Concepts 43
  
- 1.6. SYSTEM CALLS 44
  - 1.6.1. System Calls for Process Management 48
  - 1.6.2. System Calls for File Management 50
  - 1.6.3. System Calls for Directory Management 51
  - 1.6.4. Miscellaneous System Calls 53
  - 1.6.5. The Windows Win32 API 53
  
- 1.7. OPERATING SYSTEM STRUCTURE 56
  - 1.7.1. Monolithic Systems 56
  - 1.7.2. Layered Systems 57
  - 1.7.3. Virtual Machines 59
  - 1.7.4. Exokernels 61
  - 1.7.5. Client-Server Model 61
  
- 1.8. RESEARCH ON OPERATING SYSTEMS 63
  
- 1.9. OUTLINE OF THE REST OF THIS BOOK 65
  
- 1.10. METRIC UNITS 66
  
- 1.11. SUMMARY 67

**2 PROCESSES AND THREADS****71**

- 2.1. PROCESSES 71
  - 2.1.1. The Process Model 72
  - 2.1.2. Process Creation 73
  - 2.1.3. Process Termination 75
  - 2.1.4. Process Hierarchies 76
  - 2.1.5. Process States 77
  - 2.1.6. Implementation of Processes 79
  
- 2.2. THREADS 81
  - 2.2.1. The Thread Model 81
  - 2.2.2. Thread Usage 85
  - 2.2.3. Implementing Threads in User Space 90
  - 2.2.4. Implementing Threads in the Kernel 93
  - 2.2.5. Hybrid Implementations 94
  - 2.2.6. Scheduler Activations 94
  - 2.2.7. Pop-Up Threads 96
  - 2.2.8. Making Single-Threaded Code Multithreaded 97
  
- 2.3. INTERPROCESS COMMUNICATION 100
  - 2.3.1. Race Conditions 100
  - 2.3.2. Critical Regions 102
  - 2.3.3. Mutual Exclusion with Busy Waiting 103
  - 2.3.4. Sleep and Wakeup 108
  - 2.3.5. Semaphores 110
  - 2.3.6. Mutexes 113
  - 2.3.7. Monitors 115
  - 2.3.8. Message Passing 119
  - 2.3.9. Barriers 123
  
- 2.4. CLASSICAL IPC PROBLEMS 124
  - 2.4.1. The Dining Philosophers Problem 125
  - 2.4.2. The Readers and Writers Problem 128
  - 2.4.3. The Sleeping Barber Problem 129
  
- 2.5. SCHEDULING 132
  - 2.5.1. Introduction to Scheduling 132
  - 2.5.2. Scheduling in Batch Systems 138
  - 2.5.3. Scheduling in Interactive Systems 142
  - 2.5.4. Scheduling in Real-Time Systems 148
  - 2.5.5. Policy versus Mechanism 149
  - 2.5.6. Thread Scheduling 150

- 2.6. RESEARCH ON PROCESSES AND THREADS 151
- 2.7. SUMMARY 152

## **3 DEADLOCKS**

**159**

- 3.1. RESOURCES 160
  - 3.1.1. Preemptable and Nonpreemptable Resources 160
  - 3.1.2. Resource Acquisition 161
- 3.2. INTRODUCTION TO DEADLOCKS 163
  - 3.2.1. Conditions for Deadlock 164
  - 3.2.2. Deadlock Modeling 164
- 3.3. THE OSTRICH ALGORITHM 167
- 3.4. DEADLOCK DETECTION AND RECOVERY 168
  - 3.4.1. Deadlock Detection with One Resource of Each Type 168
  - 3.4.2. Deadlock Detection with Multiple Resource of Each Type 171
  - 3.4.3. Recovery from Deadlock 173
- 3.5. DEADLOCK AVOIDANCE 175
  - 3.5.1. Resource Trajectories 175
  - 3.5.2. Safe and Unsafe States 176
  - 3.5.3. The Banker's Algorithm for a Single Resource 178
  - 3.5.4. The Banker's Algorithm for Multiple Resources 179
- 3.6. DEADLOCK PREVENTION 180
  - 3.6.1. Attacking the Mutual Exclusion Condition 180
  - 3.6.2. Attacking the Hold and Wait Condition 181
  - 3.6.3. Attacking the No Preemption Condition 182
  - 3.6.4. Attacking the Circular Wait Condition 182
- 3.7. OTHER ISSUES 183
  - 3.7.1. Two-Phase Locking 183
  - 3.7.2. Nonresource Deadlocks 184
  - 3.7.3. Starvation 184
- 3.8. RESEARCH ON DEADLOCKS 185
- 3.9. SUMMARY 185

**4 MEMORY MANAGEMENT****189**

- 4.1. BASIC MEMORY MANAGEMENT 190
  - 4.1.1. Monoprogramming without Swapping or Paging 190
  - 4.1.2. Multiprogramming with Fixed Partitions 191
  - 4.1.3. Modeling Multiprogramming 192
  - 4.1.4. Analysis of Multiprogramming System Performance 194
  - 4.1.5. Relocation and Protection 194
  
- 4.2. SWAPPING 196
  - 4.2.1. Memory Management with Bitmaps 199
  - 4.2.2. Memory Management with Linked Lists 200
  
- 4.3. VIRTUAL MEMORY 202
  - 4.3.1. Paging 202
  - 4.3.2. Page Tables 205
  - 4.3.3. TLBs—Translation Lookaside Buffers 211
  - 4.3.4. Inverted Page Tables 213
  
- 4.4. PAGE REPLACEMENT ALGORITHMS 214
  - 4.4.1. The Optimal Page Replacement Algorithm 215
  - 4.4.2. The Not Recently Used Page Replacement Algorithm 216
  - 4.4.3. The First-In, First-Out 217
  - 4.4.4. The Second Chance Page Replacement Algorithm 217
  - 4.4.5. The Clock Page Replacement Algorithm 218
  - 4.4.6. The Least Recently Used 218
  - 4.4.7. Simulating LRU in Software 220
  - 4.4.8. The Working Set Page Replacement Algorithm 222
  - 4.4.9. The WSClock Page Replacement Algorithm 225
  - 4.4.10. Summary of Page Replacement Algorithms 227
  
- 4.5. MODELING PAGE REPLACEMENT ALGORITHMS 228
  - 4.5.1. Belady's Anomaly 229
  - 4.5.2. Stack Algorithms 229
  - 4.5.3. The Distance String 232
  - 4.5.4. Predicting Page Fault Rates 233
  
- 4.6. DESIGN ISSUES FOR PAGING SYSTEMS 234
  - 4.6.1. Local versus Global Allocation Policies 234
  - 4.6.2. Load Control 236
  - 4.6.3. Page Size 237
  - 4.6.4. Separate Instruction and Data Spaces 239

- 4.6.5. Shared Pages 239
- 4.6.6. Cleaning Policy 241
- 4.6.7. Virtual Memory Interface 241
- 4.7. IMPLEMENTATION ISSUES 242
  - 4.7.1. Operating System Involvement with Paging 242
  - 4.7.2. Page Fault Handling 243
  - 4.7.3. Instruction Backup 244
  - 4.7.4. Locking Pages in Memory 246
  - 4.7.5. Backing Store 246
  - 4.7.6. Separation of Policy and Mechanism 247
- 4.8. SEGMENTATION 249
  - 4.8.1. Implementation of Pure Segmentation 253
  - 4.8.2. Segmentation with Paging: MULTICS 254
  - 4.8.3. Segmentation with Paging: The Intel Pentium 257
- 4.9. RESEARCH ON MEMORY MANAGEMENT 262
- 4.10. SUMMARY 262

## **5 INPUT/OUTPUT**

**269**

- 5.1. PRINCIPLES OF I/O HARDWARE 269
  - 5.1.1. I/O Devices 270
  - 5.1.2. Device Controllers 271
  - 5.1.3. Memory-Mapped I/O 272
  - 5.1.4. Direct Memory Access 276
  - 5.1.5. Interrupts Revisited 279
- 5.2. PRINCIPLES OF I/O SOFTWARE 282
  - 5.2.1. Goals of the I/O Software 283
  - 5.2.2. Programmed I/O 284
  - 5.2.3. Interrupt-Driven I/O 286
  - 5.2.4. I/O Using DMA 287
- 5.3. I/O SOFTWARE LAYERS 287
  - 5.3.1. Interrupt Handlers 287
  - 5.3.2. Device Drivers 289

- 5.3.3. Device-Independent I/O Software 292
- 5.3.4. User-Space I/O Software 298
- 5.4. DISKS 300
  - 5.4.1. Disk Hardware 300
  - 5.4.2. Disk Formatting 315
  - 5.4.3. Disk Arm Scheduling Algorithms 318
  - 5.4.4. Error Handling 322
  - 5.4.5. Stable Storage 324
- 5.5. CLOCKS 327
  - 5.5.1. Clock Hardware 328
  - 5.5.2. Clock Software 329
  - 5.5.3. Soft Timers 332
- 5.6. CHARACTER-ORIENTED TERMINALS 333
  - 5.6.1. RS-232 Terminal Hardware 334
  - 5.6.2. Input Software 336
  - 5.6.3. Output Software 341
- 5.7. GRAPHICAL USER INTERFACES 342
  - 5.7.1. Personal Computer Keyboard, Mouse, and Display Hardware 343
  - 5.7.2. Input Software 347
  - 5.7.3. Output Software for Windows 347
- 5.8. NETWORK TERMINALS 355
  - 5.8.1. The X Window System 356
  - 5.8.2. The SLIM Network Terminal 360
- 5.9. POWER MANAGEMENT 363
  - 5.9.1. Hardware Issues 364
  - 5.9.2. Operating System Issues 365
  - 5.9.3. Degraded Operation 370
- 5.10. RESEARCH ON INPUT/OUTPUT 371
- 5.11. SUMMARY 372



**6 FILE SYSTEMS**

- 6.1. FILES 380
  - 6.1.1. File Naming 380
  - 6.1.2. File Structure 382
  - 6.1.3. File Types 383
  - 6.1.4. File Access 385
  - 6.1.5. File Attributes 386
  - 6.1.6. File Operations 387
  - 6.1.7. An Example Program Using File System Calls 389
  - 6.1.8. Memory-Mapped Files 391
  
- 6.2. DIRECTORIES 393
  - 6.2.1. Single-Level Directory Systems 393
  - 6.2.2. Two-level Directory Systems 394
  - 6.2.3. Hierarchical Directory Systems 395
  - 6.2.4. Path Names 395
  - 6.2.5. Directory Operations 398
  
- 6.3. FILE SYSTEM IMPLEMENTATION 399
  - 6.3.1. File System Layout 399
  - 6.3.2. Implementing Files 400
  - 6.3.3. Implementing Directories 405
  - 6.3.4. Shared Files 408
  - 6.3.5. Disk Space Management 410
  - 6.3.6. File System Reliability 416
  - 6.3.7. File System Performance 424
  - 6.3.8. Log-Structured File Systems 428
  
- 6.4. EXAMPLE FILE SYSTEMS 430
  - 6.4.1. CD-ROM File Systems 430
  - 6.4.2. The CP/M File System 435
  - 6.4.3. The MS-DOS File System 438
  - 6.4.4. The Windows 98 File System 442
  - 6.4.5. The UNIX V7 File System 445
  
- 6.5. RESEARCH ON FILE SYSTEMS 448
  
- 6.6. SUMMARY 448