HZ BOOKS

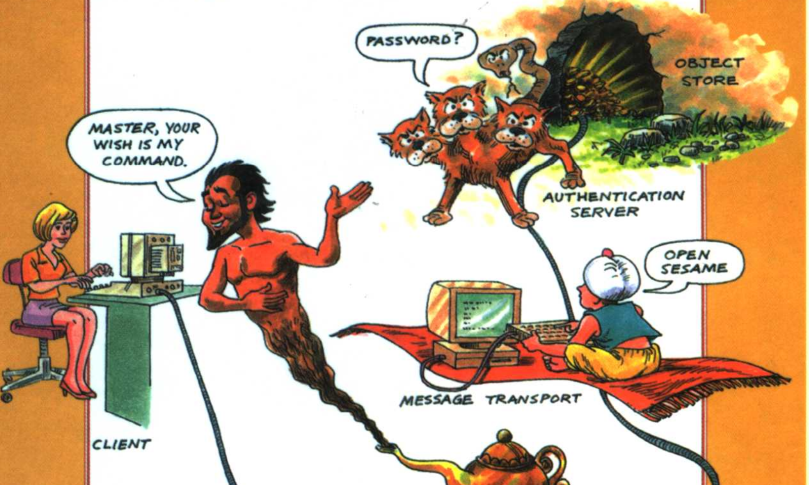
经　典　原　版　书　库

PEARSON Prentice Hall

分布式操作系统

(英文版)

Distributed Operating Systems



Andrew S. Tanenbaum

(荷) Andrew S. Tanenbaum 著

机械工业出版社
China Machine Press

TP316.4
Y3

经 典 原 版 书 库

分布式操作系统

(英文版)

Distributed Operating Systems

江苏工业学院图书馆
藏 书 章

(荷) Andrew S. Tanenbaum 著

机械工业出版社
China Machine Press

English reprint edition copyright © 2006 by Pearson Education Asia Limited and China Machine Press.

Original English language title: *Distributed Operating Systems* (ISBN 0-13-219908-4) by Andrew S. Tanenbaum, Copyright © 1995 by Prentice Hall, Inc.

All rights reserved.

Published by arrangement with the original publisher, Pearson Education, Inc., publishing as Prentice Hall.

For sale and distribution in the People's Republic of China exclusively (except Taiwan, Hong Kong SAR and Macau SAR).

本书英文影印版由Pearson Education Asia Ltd.授权机械工业出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

仅限于中华人民共和国境内（不包括中国香港、澳门特别行政区和中国台湾地区）销售发行。

本书封面贴有Pearson Education（培生教育出版集团）激光防伪标签，无标签者不得销售。

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：01-2006-3114

图书在版编目（CIP）数据

分布式操作系统（英文版）/（荷）塔嫩鲍姆（Tanenbaum, A. S.）著. –北京：机械工业出版社，2006.7

（经典原版书库）

书名原文：Distributed Operating Systems

ISBN 7-111-19347-4

Ⅰ. 分…　Ⅱ. 塔…　Ⅲ. 分布式操作系统 – 英文　Ⅳ. TP316.4

中国版本图书馆CIP数据核字（2006）第062555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：迟振春

三河市明辉印装有限公司 印刷 · 新华书店北京发行所发行

2006年7月第1版第1次印刷

170mm × 242mm · 39.5印张

定价：69.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

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年开始，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、研究及庋藏。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：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。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们服务，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。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。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、有的历经三十年而不衰、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，我们的联系方法如下:

电子邮件: hzjsj@hzbook.com
联系电话:（010）68995264
联系地址: 北京市西城区百万庄南街1号
邮政编码: 100037

试读结束：需要全本请在线购买： www.ertongbook.com

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

With the publication of *Distributed Operating Systems* I have now completed my trilogy on operating systems. The three volumes of this trilogy are:

- *Operating Systems: Design and Implementation*
- *Distributed Operating Systems*
- *Modern Operating Systems*

The three volumes are not completely independent, however. For schools having a two-course sequence in operating systems (or an undergraduate course plus a graduate course), one possible choice is to use *Operating Systems: Design and Implementation* in the first course and *Distributed Operating Systems* in the second one.

The former book treats the standard principles of single-processor systems, including processes, synchronization, I/O, deadlocks, memory management, file systems, security, and so on. It also illustrates these principles in great detail through the use of MINIX, a UNIX-clone whose source listing is given in an appendix. MINIX is available on diskette from Prentice Hall for the IBM PC (8088 and up), Atari, Amiga, Macintosh, and SPARC processors.

The latter book (this one), covers distributed operating systems in detail, including communication, synchronization, processes, file systems, and memory management, but this time in the context of distributed systems. Four examples of distributed systems are given in great detail: Amoeba, Mach, Chorus, and DCE. Amoeba is available for free to universities for educational use. It runs on the Intel 386/486, SPARC, and Sun 3 processors. For information on how to obtain Amoeba please FTP the file *amoeba/Intro.ps.Z* from *ftp.cs.vu.nl* or contact the author by electronic mail at *ast@cs.vu.nl.* Potential users should be forewarned that Amoeba is considerably more complex than MINIX: the documentation alone (available by FTP), runs to well over 1000 pages and the system requires at least five large machines and an Ethernet to run well.

By studying these two books in sequence and using both MINIX and Amoeba, students will obtain a thorough knowledge of the principles and practice of both single-processor and distributed operating systems. Now that the trilogy is completed, I plan to revise MINIX and the book describing it.

For universities or computer professionals with less time available, *Modern Operating Systems* can be thought of as a condensed version of the other two books. It provides an introduction to the principles of both single-processor and distributed operating systems, but without the detailed example of MINIX. It also omits many of the advanced topics present in this book, including an introduction to ATM, fault-tolerant distributed systems, real time distributed systems, distributed shared memory, Chorus, DCE, and other topics. In all, about 230 pages of material on distributed systems present in this book have been omitted from *Modern Operating Systems.*

Many people have helped me with this book. I would especially like to thank the following people for reading portions of the manuscript and giving me many useful suggestions for improvement: Irina Athanasiu, Henri Bal, Saniya Ben Hassen, David Black, John Carter, Randall Dean, Wiebren de Jonge, John Dugas, Dick Grune, Anoop Gupta, Frans Kaashoek, Marcus Koebler, Hermann Kopetz, Ed Lazowska, Dan Lenoski, Kai Li, Marc Maathuis, David Mosberger, Douglas Orr, Craig Partridge, Carlton Pu, Marc Rozier, Rich Salz, Mike Schroeder, Karsten Schwan, Greg Sharp, Dennis Shasha, Sol Shatz, Jennifer Steiner, Chuck Thacker, John Turek, Walt Tuvell, Leendert van Doorn, Robbert van Renesse, Kees Verstoep, Ellen Zegura, Willy Zwaenpoel, and the anonymous reviewers. My editor, Bill Zobrist, put up with my attempts to get everything perfect with nary a whimper.

Despite all this help, no doubt some errors remain. That seems to be inevitable, no matter how many people read the manuscript. People who wish to report errors should contact me by electronic mail.

Finally, I would like to thank Suzanne again. After eight books, she knows the implications of another one, but her patience and love are boundless. I also want to thank Barbara and Marvin for using their computers and leaving mine alone (except for the printer). Teaching them how to use PC word processing programs has made me appreciate *troff* more than ever. Finally, I would like to thank Little Bram for being quiet while I was writing.

Andrew S. Tanenbaum

Contents

PREFACE **vii**

1 INTRODUCTION TO DISTRIBUTED SYSTEMS **1**

1.1 WHAT IS A DISTRIBUTED SYSTEM? 2

1.2 GOALS 3

- 1.2.1 Advantages of Distributed Systems over Centralized Systems 3
- 1.2.2 Advantages of Distributed Systems over Independent PCs 6
- 1.2.3 Disadvantages of Distributed Systems 6

1.3 HARDWARE CONCEPTS 8

- 1.3.1 Bus-Based Multiprocessors 10
- 1.3.2 Switched Multiprocessors 12
- 1.3.3 Bus-Based Multicomputers 13
- 1.3.4 Switched Multicomputers 14

1.4 SOFTWARE CONCEPTS 15

- 1.4.1 Network Operating Systems 16
- 1.4.2 True Distributed Systems 18
- 1.4.3 Multiprocessor Timesharing Systems 20

1.5 DESIGN ISSUES 22

- 1.5.1 Transparency 22
- 1.5.2 Flexibility 25
- 1.5.3 Reliability 27
- 1.5.4 Performance 28
- 1.5.5 Scalability 29

1.6 SUMMARY 31

2 COMMUNICATION IN DISTRIBUTED SYSTEMS 34

2.1 LAYERED PROTOCOLS 35
2.1.1 The Physical Layer 38
2.1.2 The Data Link Layer 38
2.1.3 The Network Layer 40
2.1.4 The Transport Layer 40
2.1.5 The Session Layer 41
2.1.6 The Presentation Layer 41
2.1.7 The Application Layer 42
2.2 ASYNCHRONOUS TRANSFER MODE NETWORKS 42
2.2.1 What Is Asynchronous Transfer Mode? 42
2.2.2 The ATM Physical Layer 44
2.2.3 The ATM Layer 45
2.2.4 The ATM Adaptation Layer 46
2.2.5 ATM Switching 47
2.2.6 Some Implications of ATM for Distributed Systems 49
2.3 THE CLIENT-SERVER MODEL 50
2.3.1 Clients and Servers 51
2.3.2 An Example Client and Server 52
2.3.3 Addressing 56
2.3.4 Blocking versus Nonblocking Primitives 58
2.3.5 Buffered versus Unbuffered Primitives 61
2.3.6 Reliable versus Unreliable Primitives 63
2.3.7 Implementing the Client-Server Model 65
2.4 REMOTE PROCEDURE CALL 68
2.4.1 Basic RPC Operation 68
2.4.2 Parameter Passing 72
2.4.3 Dynamic Binding 77
2.4.4 RPC Semantics in the Presence of Failures 80
2.4.5 Implementation Issues 84
2.4.6 Problem Areas 95
2.5 GROUP COMMUNICATION 99
2.5.1 Introduction to Group Communication 99
2.5.2 Design Issues 101
2.5.3 Group Communication in ISIS 110
2.6 SUMMARY 114

3 SYNCHRONIZATION IN DISTRIBUTED SYSTEMS 118

3.1 CLOCK SYNCHRONIZATION 119
3.1.1 Logical Clocks 120
3.1.2 Physical Clocks 124
3.1.3 Clock Synchronization Algorithms 127
3.1.4 Use of Synchronized Clocks 132
3.2 MUTUAL EXCLUSION 134
3.2.1 A Centralized Algorithm 134
3.2.2 A Distributed Algorithm 135
3.2.3 A Token Ring Algorithm 138
3.2.4 A Comparison of the Three Algorithms 139
3.3 ELECTION ALGORITHMS 140
3.3.1 The Bully Algorithm 141
3.3.2 A Ring Algorithm 143
3.4 ATOMIC TRANSACTIONS 144
3.4.1 Introduction to Atomic Transactions 144
3.4.2 The Transaction Model 145
3.4.3 Implementation 150
3.4.4 Concurrency Control 154
3.5 DEADLOCKS IN DISTRIBUTED SYSTEMS 158
3.5.1 Distributed Deadlock Detection 159
3.5.2 Distributed Deadlock Prevention 163
3.6 SUMMARY 165

4 PROCESSES AND PROCESSORS IN DISTRIBUTED SYSTEMS 169

4.1 THREADS 169
4.1.1 Introduction to Threads 170
4.1.2 Thread Usage 171
4.1.3 Design Issues for Threads Packages 174
4.1.4 Implementing a Threads Package 178
4.1.5 Threads and RPC 184
4.2 SYSTEM MODELS 186
4.2.1 The Workstation Model 186
4.2.2 Using Idle Workstations 189
4.2.3 The Processor Pool Model 193
4.2.4 A Hybrid Model 197
4.3 PROCESSOR ALLOCATION 197
4.3.1 Allocation Models 197

试读结束：需要全本请在线购买： www.ertongbook.com

4.3.2 Design Issues for Processor Allocation Algorithms 199
4.3.3 Implementation Issues for Processor Allocation Algorithms 201
4.3.4 Example Processor Allocation Algorithms 203
4.4 SCHEDULING IN DISTRIBUTED SYSTEMS 210
4.5 FAULT TOLERANCE 212
4.5.1 Component Faults 212
4.5.2 System Failures 213
4.5.3 Synchronous versus Asynchronous Systems 214
4.5.4 Use of Redundancy 214
4.5.5 Fault Tolerance Using Active Replication 215
4.5.6 Fault Tolerance Using Primary-Backup 217
4.5.7 Agreement in Faulty Systems 219
4.6 REAL-TIME DISTRIBUTED SYSTEMS 223
4.6.1 What Is a Real-Time System? 223
4.6.2 Design Issues 226
4.6.3 Real-Time Communication 230
4.6.4 Real-Time Scheduling 234
4.7 SUMMARY 240

5 DISTRIBUTED FILE SYSTEMS 245

5.1 DISTRIBUTED FILE SYSTEM DESIGN 246
5.1.1 The File Service Interface 246
5.1.2 The Directory Server Interface 248
5.1.3 Semantics of File Sharing 253
5.2 DISTRIBUTED FILE SYSTEM IMPLEMENTATION 256
5.2.1 File Usage 256
5.2.2 System Structure 258
5.2.3 Caching 262
5.2.4 Replication 268
5.2.5 An Example: Sun's Network File System 272
5.2.6 Lessons Learned 278
5.3 TRENDS IN DISTRIBUTED FILE SYSTEMS 279
5.3.1 New Hardware 280
5.3.2 Scalability 282
5.3.3 Wide Area Networking 283
5.3.4 Mobile Users 284
5.3.5 Fault Tolerance 284
5.3.6 Multimedia 285
5.4 SUMMARY 285

6 DISTRIBUTED SHARED MEMORY 289

6.1 INTRODUCTION 290
6.2 WHAT IS SHARED MEMORY? 292
6.2.1 On-Chip Memory 293
6.2.2 Bus-Based Multiprocessors 293
6.2.3 Ring-Based Multiprocessors 298
6.2.4 Switched Multiprocessors 301
6.2.5 NUMA Multiprocessors 308
6.2.6 Comparison of Shared Memory Systems 312
6.3 CONSISTENCY MODELS 315
6.3.1 Strict Consistency 315
6.3.2 Sequential Consistency 317
6.3.3 Causal Consistency 321
6.3.4 PRAM Consistency and Processor Consistency 322
6.3.5 Weak Consistency 325
6.3.6 Release Consistency 327
6.3.7 Entry Consistency 330
6.3.8 Summary of Consistency Models 331
6.4 PAGE-BASED DISTRIBUTED SHARED MEMORY 333
6.4.1 Basic Design 334
6.4.2 Replication 334
6.4.3 Granularity 336
6.4.4 Achieving Sequential Consistency 337
6.4.5 Finding the Owner 339
6.4.6 Finding the Copies 342
6.4.7 Page Replacement 343
6.4.8 Synchronization 344
6.5 SHARED-VARIABLE DISTRIBUTED SHARED MEMORY 345
6.5.1 Munin 346
6.5.2 Midway 353
6.6 OBJECT-BASED DISTRIBUTED SHARED MEMORY 356
6.6.1 Objects 356
6.6.2 Linda 358
6.6.3 Orca 365
6.7 COMPARISON 371
6.8 SUMMARY 372

7 CASE STUDY 1: AMOEBA 376

7.1 INTRODUCTION TO AMOEBA 376
7.1.1 History of Amoeba 376
7.1.2 Research Goals 377
7.1.3 The Amoeba System Architecture 378
7.1.4 The Amoeba Microkernel 380
7.1.5 The Amoeba Servers 382
7.2 OBJECTS AND CAPABILITIES IN AMOEBA 384
7.2.1 Capabilities 384
7.2.2 Object Protection 385
7.2.3 Standard Operations 387
7.3 PROCESS MANAGEMENT IN AMOEBA 388
7.3.1 Processes 388
7.3.2 Threads 391
7.4 MEMORY MANAGEMENT IN AMOEBA 392
7.4.1 Segments 392
7.4.2 Mapped Segments 393
7.5 COMMUNICATION IN AMOEBA 393
7.5.1 Remote Procedure Call 394
7.5.2 Group Communication in Amoeba 398
7.5.3 The Fast Local Internet Protocol 407
7.6 THE AMOEBA SERVERS 415
7.6.1 The Bullet Server 415
7.6.2 The Directory Server 420
7.6.3 The Replication Server 425
7.6.4 The Run Server 425
7.6.5 The Boot Server 427
7.6.6 The TCP/IP Server 427
7.6.7 Other Servers 428
7.7 SUMMARY 428

8 CASE STUDY 2: MACH 431

8.1 INTRODUCTION TO MACH 431
8.1.1 History of Mach 431
8.1.2 Goals of Mach 433
8.1.3 The Mach Microkernel 433
8.1.4 The Mach BSD UNIX Server 435

8.2 PROCESS MANAGEMENT IN MACH 436
- 8.2.1 Processes 436
- 8.2.2 Threads 439
- 8.2.3 Scheduling 442

8.3 MEMORY MANAGEMENT IN MACH 445
- 8.3.1 Virtual Memory 446
- 8.3.2 Memory Sharing 449
- 8.3.3 External Memory Managers 452
- 8.3.4 Distributed Shared Memory in Mach 456

8.4 COMMUNICATION IN MACH 457
- 8.4.1 Ports 457
- 8.4.2 Sending and Receiving Messages 464
- 8.4.3 The Network Message Server 469

8.5 UNIX EMULATION IN MACH 471

8.6 SUMMARY 472

9 CASE STUDY 3: CHORUS 475

9.1 INTRODUCTION TO CHORUS 475
- 9.1.1 History of Chorus 476
- 9.1.2 Goals of Chorus 477
- 9.1.3 System Structure 478
- 9.1.4 Kernel Abstractions 479
- 9.1.5 Kernel Structure 481
- 9.1.6 The UNIX Subsystem 483
- 9.1.7 The Object-Oriented Subsystem 483

9.2 PROCESS MANAGEMENT IN CHORUS 483
- 9.2.1 Processes 484
- 9.2.2 Threads 485
- 9.2.3 Scheduling 486
- 9.2.4 Traps, Exceptions, and Interrupts 487
- 9.2.5 Kernel Calls for Process Management 488

9.3 MEMORY MANAGEMENT IN CHORUS 490
- 9.3.1 Regions and Segments 490
- 9.3.2 Mappers 491
- 9.3.3 Distributed Shared Memory 492
- 9.3.4 Kernel Calls for Memory Management 493

9.4 COMMUNICATON IN CHORUS 495
9.4.1 Messages 495
9.4.2 Ports 495
9.4.3 Communication Operations 496
9.4.4 Kernel Calls for Communication 498
9.5 UNIX EMULATION IN CHORUS 499
9.5.1 Structure of a UNIX Process 500
9.5.2 Extensions to UNIX 500
9.5.3 Implementation of UNIX on Chorus 501
9.6 COOL: AN OBJECT-ORIENTED SUBSYSTEM 507
9.6.1 The COOL Architecture 507
9.6.2 The COOL Base Layer 507
9.6.3 The COOL Generic Runtime System 509
9.6.4 The Language Runtime System 509
9.6.5 Implementation of COOL 510
9.7 COMPARISON OF AMOEBA, MACH, AND CHORUS 510
9.7.1 Philosophy 511
9.7.2 Objects 512
9.7.3 Processes 513
9.7.4 Memory Model 514
9.7.5 Communication 515
9.7.6 Servers 516
9.8 SUMMARY 517

10 CASE STUDY 4: DCE **520**
10.1 INTRODUCTION TO DCE 520
10.1.1 History of DCE 520
10.1.2 Goals of DCE 521
10.1.3 DCE Components 522
10.1.4 Cells 525
10.2 THREADS 527
10.2.1 Introduction to DCE Threads 527
10.2.2 Scheduling 529
10.2.3 Synchronization 530
10.2.4 Thread Calls 531
10.3 REMOTE PROCEDURE CALL 535
10.3.1 Goals of DCE RPC 535
10.3.2 Writing a Client and a Server 536
10.3.3 Binding a Client to a Server 538
10.3.4 Performing an RPC 539

10.4 TIME SERVICE 540
10.4.1 DTS Time Model 541
10.4.2 DTS Implementation 543
10.5 DIRECTORY SERVICE 544
10.5.1 Names 546
10.5.2 The Cell Directory Service 547
10.5.3 The Global Directory Service 549
10.6 SECURITY SERVICE 554
10.6.1 Security Model 555
10.6.2 Security Components 557
10.6.3 Tickets and Authenticators 558
10.6.4 Authenticated RPC 559
10.6.5 ACLs 562
10.7 DISTRIBUTED FILE SYSTEM 564
10.7.1 DFS Interface 565
10.7.2 DFS Components in the Server Kernel 566
10.7.3 DFS Components in the Client Kernel 569
10.7.4 DFS Components in User Space 571
10.8 SUMMARY 573

11 BIBLIOGRAPHY AND SUGGESTED READINGS 577

11.1 SUGGESTED READINGS 577
11.2 ALPHABETICAL BIBLIOGRAPHY 584

INDEX 603

试读结束：需要全本请在线购买： www.ertongbook.com