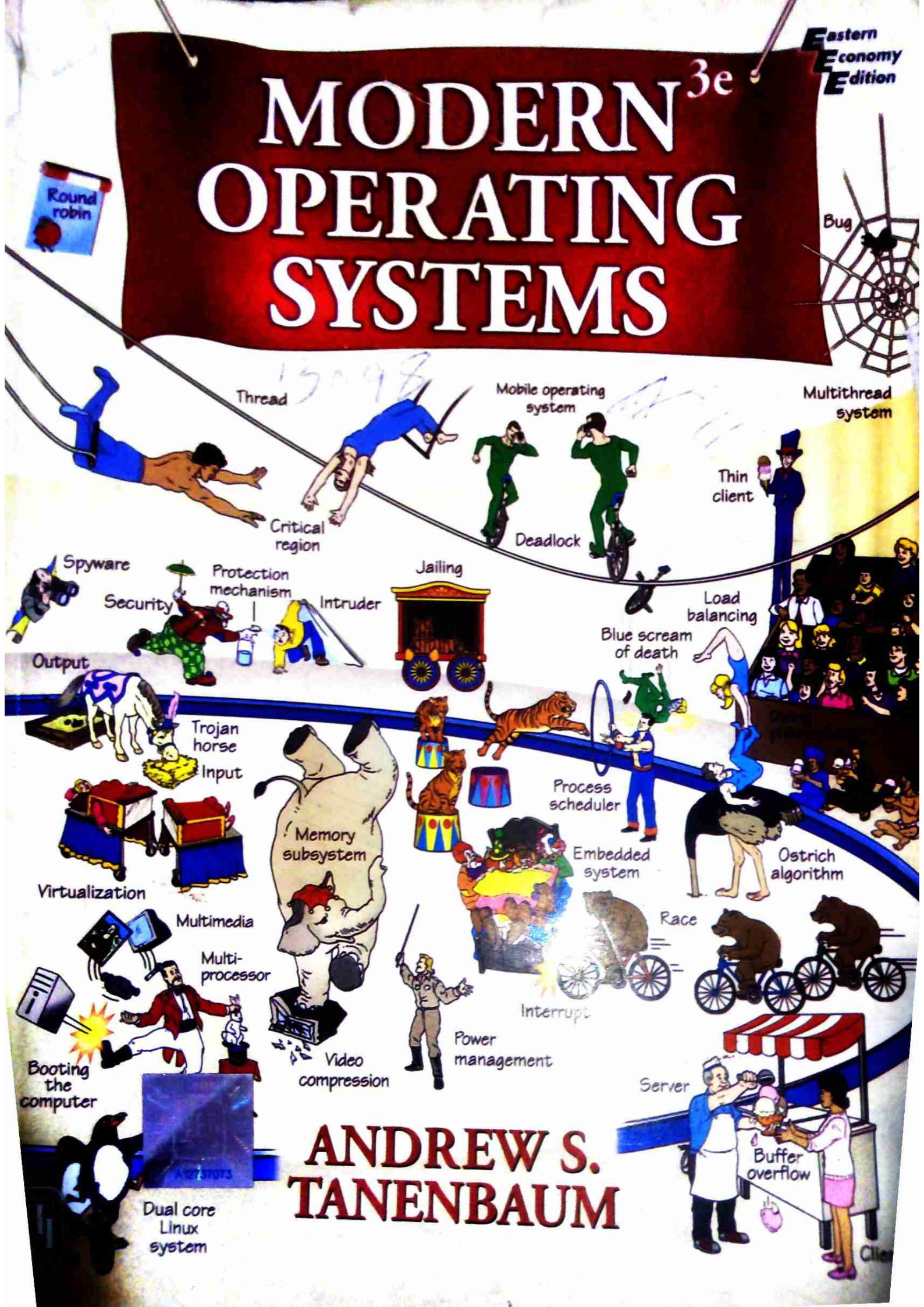


MODERN OPERATING SYSTEMS^{3e}

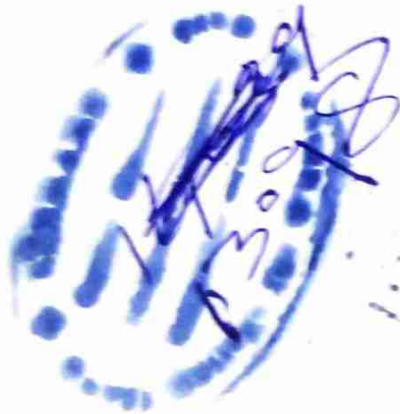


ANDREW S.
TANENBAUM

**MODERN
OPERATING SYSTEMS**

THIRD EDITION

13098



FOR PUBLICITY
New Books 'N' Books
Goal Bagh Near Punjab Bank
Gulgaat, Multan.
061-6511828, 061-6223080

MODERN OPERATING SYSTEMS

THIRD EDITION

This Indian Reprint—~~₹ 10996.00~~
(Original U.S. Edition—₹ 10996.00)

MODERN OPERATING SYSTEMS, 3rd ed.
by Andrew S. Tanenbaum

Original edition, entitled *Modern Operating Systems*, 3rd ed. by Andrew S. Tanenbaum, published by Pearson Education, Inc., publishing as Pearson Prentice Hall.

Copyright © 2008 Pearson Education Inc., Upper Saddle River, New Jersey 07458, U.S.A.

ISBN-978-81-203-3904-0

The author and publisher of this book have used their best efforts in preparing this book. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, expressed or implied, with regard to these programs or the documentation contained in this book. The author and publisher shall not be liable in any event for incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of these programs.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Indian edition published by PHI Learning Private Limited.

This edition is manufactured in India, and is authorized for sale in India, Pakistan, Sri Lanka, Bhutan, Bangladesh, Nepal and the Maldives only.

Published by Asoke K. Ghosh, PHI Learning Private Limited, Rimjhim House, 111, Patparganj Industrial Estate, Delhi-110092 and Printed by Mohan Makhijani at Rekha Printers Private Limited, New Delhi-110020.

CONTENTS

PREFACE

xxiv

1 INTRODUCTION

1

✓ 1.1 WHAT IS AN OPERATING SYSTEM? 3

1.1.1 The Operating System as an Extended Machine 4

1.1.2 The Operating System as a Resource Manager 6

1.2 HISTORY OF OPERATING SYSTEMS 7

1.2.1 The First Generation (1945–55) Vacuum Tubes 7

1.2.2 The Second Generation (1955–65) Transistors and Batch Systems 8

1.2.3 The Third Generation (1965–1980) ICs and Multiprogramming 10

1.2.4 The Fourth Generation (1980–Present) Personal Computers 15

1.3 COMPUTER HARDWARE REVIEW 19

1.3.1 Processors 19

1.3.2 Memory 23

1.3.3 Disks 26

1.3.4 Tapes 27

1.3.5 I/O Devices 27

1.3.6 Buses 30

1.3.7 Booting the Computer 33

1.4	THE OPERATING SYSTEM ZOO	33
1.4.1	Mainframe Operating Systems	34
1.4.2	Server Operating Systems	34
1.4.3	Multiprocessor Operating Systems	34
1.4.4	Personal Computer Operating Systems	35
1.4.5	Handheld Computer Operating Systems	35
1.4.6	Embedded Operating Systems	35
1.4.7	Sensor Node Operating Systems	36
1.4.8	Real-Time Operating Systems	36
1.4.9	Smart Card Operating Systems	37
1.5	OPERATING SYSTEM CONCEPTS	37
1.5.1	Processes	38
1.5.2	Address Spaces	40
1.5.3	Files	40
1.5.4	Input/Output	43
1.5.5	Protection	44
1.5.6	The Shell	44
1.5.7	Ontogeny Recapitulates Phylogeny	46
1.6	SYSTEM CALLS	49
1.6.1	System Calls for Process Management	52
1.6.2	System Calls for File Management	56
1.6.3	System Calls for Directory Management	57
1.6.4	Miscellaneous System Calls	58
1.6.5	The Windows Win32 API	59
1.7	OPERATING SYSTEM STRUCTURE	62
1.7.1	Monolithic Systems	62
1.7.2	Layered Systems	63
1.7.3	Microkernels	64
1.7.4	Client-Server Model	67
1.7.5	Virtual Machines	67
1.7.6	Exokernels	71
1.8	THE WORLD ACCORDING TO C	72
1.8.1	The C Language	72
1.8.2	Header Files	73
1.8.3	Large Programming Projects	74
1.8.4	The Model of Run Time	75
1.9	RESEARCH ON OPERATING SYSTEMS	76

- 1.10 OUTLINE OF THE REST OF THIS BOOK 77
- 1.11 METRIC UNITS 78
- 1.12 SUMMARY 79

2 PROCESSES AND THREADS

83

- 2.1 PROCESSES 83
 - 2.1.1 The Process Model 84
 - 2.1.2 Process Creation 86
 - 2.1.3 Process Termination 88
 - 2.1.4 Process Hierarchies 89
 - 2.1.5 Process States 90
 - 2.1.6 Implementation of Processes 91
 - 2.1.7 Modeling Multiprogramming 93
- 2.2 THREADS 95
 - 2.2.1 Thread Usage 95
 - 2.2.2 The Classical Thread Model 100
 - 2.2.3 POSIX Threads 104
 - 2.2.4 Implementing Threads in User Space 106
 - 2.2.5 Implementing Threads in the Kernel 109
 - 2.2.6 Hybrid Implementations 110
 - 2.2.7 Scheduler Activations 111
 - 2.2.8 Pop-Up Threads 112
 - 2.2.9 Making Single-Threaded Code Multithreaded 114
- 2.3 INTERPROCESS COMMUNICATION 117
 - 2.3.1 Race Conditions 117
 - 2.3.2 Critical Regions 119
 - 2.3.3 Mutual Exclusion with Busy Waiting 120
 - 2.3.4 Sleep and Wakeup 125
 - 2.3.5 Semaphores 128
 - 2.3.6 Mutexes 130
 - 2.3.7 Monitors 134
 - 2.3.8 Message Passing 140
 - 2.3.9 Barriers 144

CONTENTS

2.4 SCHEDULING 145

2.4.1 Introduction to Scheduling 145

2.4.2 Scheduling in Batch Systems 152

2.4.3 Scheduling in Interactive Systems 154

2.4.4 Scheduling in Real-Time Systems 160

2.4.5 Policy versus Mechanism 161

2.4.6 Thread Scheduling 162

2.5 CLASSICAL IPC PROBLEMS 163

2.5.1 The Dining Philosophers Problem 164

2.5.2 The Readers and Writers Problem 167

2.6 RESEARCH ON PROCESSES AND THREADS 168

2.7 SUMMARY 169

175

3 MEMORY MANAGEMENT

3.1 NO MEMORY ABSTRACTION 176

3.2 A MEMORY ABSTRACTION: ADDRESS SPACES 179

3.2.1 The Notion of an Address Space 180

3.2.2 Swapping 181

3.2.3 Managing Free Memory 184

3.3 VIRTUAL MEMORY 188

3.3.1 Paging 189

3.3.2 Page Tables 193

3.3.3 Speeding Up Paging 194

3.3.4 Page Tables for Large Memories 198

3.4 PAGE REPLACEMENT ALGORITHMS 201

3.4.1 The Optimal Page Replacement Algorithm 202

3.4.2 The Not Recently Used Page Replacement Algorithm 203

3.4.3 The First-In, First-Out (FIFO) Page Replacement Algorithm 204

3.4.4 The Second-Chance Page Replacement Algorithm 204

3.4.5 The Clock Page Replacement Algorithm 205

3.4.6 The Least Recently Used (LRU) Page Replacement Algorithm 206

3.4.7 Simulating LRU in Software 207

3.4.8 The Working Set Page Replacement Algorithm 209

CONTENTS

3.4.9	The WSClock Page Replacement Algorithm	213
3.4.10	Summary of Page Replacement Algorithms	215
3.5	DESIGN ISSUES FOR PAGING SYSTEMS	216
3.5.1	Local versus Global Allocation Policies	216
3.5.2	Load Control	218
3.5.3	Page Size	219
3.5.4	Separate Instruction and Data Spaces	221
3.5.5	Shared Pages	221
3.5.6	Shared Libraries	223
3.5.7	Mapped Files	225
3.5.8	Cleaning Policy	226
3.5.9	Virtual Memory Interface	226
3.6	IMPLEMENTATION ISSUES	227
3.6.1	Operating System Involvement with Paging	227
3.6.2	Page Fault Handling	228
3.6.3	Instruction Backup	229
3.6.4	Locking Pages in Memory	230
3.6.5	Backing Store	231
3.6.6	Separation of Policy and Mechanism	233
3.7	SEGMENTATION	234
3.7.1	Implementation of Pure Segmentation	237
3.7.2	Segmentation with Paging: MULTICS	238
3.7.3	Segmentation with Paging: The Intel Pentium	242
3.8	RESEARCH ON MEMORY MANAGEMENT	247
3.9	SUMMARY	248

4 FILE SYSTEMS

255

4.1	FILES	257
4.1.1	File Naming	257
4.1.2	File Structure	259
4.1.3	File Types	260
4.1.4	File Access	262
4.1.5	File Attributes	263

CONTENTS

4.1.6	File Operations	264
4.1.7	An Example Program Using File System Calls	265
4.2	DIRECTORIES	268
4.2.1	Single-Level Directory Systems	268
4.2.2	Hierarchical Directory Systems	268
4.2.3	Path Names	269
4.2.4	Directory Operations	272
4.3	FILE SYSTEM IMPLEMENTATION	273
4.3.1	File System Layout	273
4.3.2	Implementing Files	274
4.3.3	Implementing Directories	280
4.3.4	Shared Files	283
4.3.5	Log-Structured File Systems	285
4.3.6	Journaling File Systems	287
4.3.7	Virtual File Systems	288
4.4	FILE SYSTEM MANAGEMENT AND OPTIMIZATION	292
4.4.1	Disk Space Management	292
4.4.2	File System Backups	298
4.4.3	File System Consistency	304
4.4.4	File System Performance	307
4.4.5	Defragmenting Disks	311
4.5	EXAMPLE FILE SYSTEMS	312
4.5.1	CD-ROM File Systems	312
4.5.2	The MS-DOS File System	318
4.5.3	The UNIX V7 File System	321
4.6	RESEARCH ON FILE SYSTEMS	324
4.7	SUMMARY	324

5 INPUT/OUTPUT

5.1	PRINCIPLES OF I/O HARDWARE	329
5.1.1	I/O Devices	330
5.1.2	Device Controllers	331

- 5.1.3 Memory-Mapped I/O 332
- 5.1.4 Direct Memory Access (DMA) 336
- 5.1.5 Interrupts Revisited 339
- 5.2 PRINCIPLES OF I/O SOFTWARE 343
 - 5.2.1 Goals of the I/O Software 343
 - 5.2.2 Programmed I/O 344
 - 5.2.3 Interrupt-Driven I/O 346
 - 5.2.4 I/O Using DMA 347
- 5.3 I/O SOFTWARE LAYERS 348
 - 5.3.1 Interrupt Handlers 348
 - 5.3.2 Device Drivers 349
 - 5.3.3 Device-Independent I/O Software 353
 - 5.3.4 User-Space I/O Software 359
- ✓ 5.4 DISKS 360
 - 5.4.1 Disk Hardware 361
 - 5.4.2 Disk Formatting 376
 - 5.4.3 Disk Arm Scheduling Algorithms 379
 - 5.4.4 Error Handling 382
 - 5.4.5 Stable Storage 385
- 5.5 CLOCKS 388
 - 5.5.1 Clock Hardware 388
 - 5.5.2 Clock Software 390
 - 5.5.3 Soft Timers 393
- 5.6 USER INTERFACES: KEYBOARD, MOUSE, MONITOR 394
 - 5.6.1 Input Software 394
 - 5.6.2 Output Software 399
- 5.7 THIN CLIENTS 415
- 5.8 POWER MANAGEMENT 417
 - 5.8.1 Hardware Issues 418
 - 5.8.2 Operating System Issues 419
 - 5.8.3 Application Program Issues 424
- 5.9 RESEARCH ON INPUT/OUTPUT 425
- 5.10 SUMMARY 426

- 6.1 RESOURCES 434
 - 6.1.1 Preemptable and Nonpreemptable Resources 434
 - 6.1.2 Resource Acquisition 435
- 6.2 INTRODUCTION TO DEADLOCKS 437
 - 6.2.1 Conditions for Resource Deadlocks 438
 - 6.2.2 Deadlock Modeling 438
- 6.3 THE OSTRICH ALGORITHM 441
- 6.4 DEADLOCK DETECTION AND RECOVERY 442
 - 6.4.1 Deadlock Detection with One Resource of Each Type 442
 - 6.4.2 Deadlock Detection with Multiple Resources of Each Type 444
 - 6.4.3 Recovery from Deadlock 447
- 6.5 DEADLOCK AVOIDANCE 448
 - 6.5.1 Resource Trajectories 449
 - 6.5.2 Safe and Unsafe States 450
 - 6.5.3 The Banker's Algorithm for a Single Resource 451
 - 6.5.4 The Banker's Algorithm for Multiple Resources 452
- 6.6 DEADLOCK PREVENTION 454
 - 6.6.1 Attacking the Mutual Exclusion Condition 454
 - 6.6.2 Attacking the Hold and Wait Condition 455
 - 6.6.3 Attacking the No Preemption Condition 455
 - 6.6.4 Attacking the Circular Wait Condition 456
- 6.7 OTHER ISSUES 457
 - 6.7.1 Two-Phase Locking 457
 - 6.7.2 Communication Deadlocks 458
 - 6.7.3 Livelock 459
 - 6.7.4 Starvation 461
- 6.8 RESEARCH ON DEADLOCKS 461
- 6.9 SUMMARY 462

7.2 MULTIMEDIA OPERATING SYSTEMS 467

- 7.1 INTRODUCTION TO MULTIMEDIA 468
- 7.2 MULTIMEDIA FILES 472
 - 7.2.1 Video Encoding 473
 - 7.2.2 Audio Encoding 476
- 7.3 VIDEO COMPRESSION 478
 - 7.3.1 The JPEG Standard 478
 - 7.3.2 The MPEG Standard 481
- 7.4 AUDIO COMPRESSION 484
- 7.5 MULTIMEDIA PROCESS SCHEDULING 487
 - 7.5.1 Scheduling Homogeneous Processes 488
 - 7.5.2 General Real-Time Scheduling 488
 - 7.5.3 Rate Monotonic Scheduling 490
 - 7.5.4 Earliest Deadline First Scheduling 491
- 7.6 MULTIMEDIA FILE SYSTEM PARADIGMS 493
 - 7.6.1 VCR Control Functions 494
 - 7.6.2 Near Video on Demand 496
 - 7.6.3 Near Video on Demand with VCR Functions 498
- 7.7 FILE PLACEMENT 499
 - 7.7.1 Placing a File on a Single Disk 500
 - 7.7.2 Two Alternative File Organization Strategies 501
 - 7.7.3 Placing Files for Near Video on Demand 504
 - 7.7.4 Placing Multiple Files on a Single Disk 506
 - 7.7.5 Placing Files on Multiple Disks 508
- 7.8 CACHING 510
 - 7.8.1 Block Caching 511
 - 7.8.2 File Caching 512
- 7.9 DISK SCHEDULING FOR MULTIMEDIA 513
 - 7.9.1 Static Disk Scheduling 513
 - 7.9.2 Dynamic Disk Scheduling 515
- 7.10 RESEARCH ON MULTIMEDIA 516
- 7.11 SUMMARY 517

8 MULTIPLE PROCESSOR SYSTEMS 523

- 8.1 MULTIPROCESSORS 526
 - 8.1.1 Multiprocessor Hardware 526
 - 8.1.2 Multiprocessor Operating System Types 534
 - 8.1.3 Multiprocessor Synchronization 538
 - 8.1.4 Multiprocessor Scheduling 542

- 8.2 MULTICOMPUTERS 548
 - 8.2.1 Multicomputer Hardware 549
 - 8.2.2 Low-Level Communication Software 553
 - 8.2.3 User-Level Communication Software 555
 - 8.2.4 Remote Procedure Call 558
 - 8.2.5 Distributed Shared Memory 560
 - 8.2.6 Multicomputer Scheduling 565
 - 8.2.7 Load Balancing 565

- 8.3 VIRTUALIZATION 568
 - 8.3.1 Requirements for Virtualization 570
 - 8.3.2 Type 1 Hypervisors 571
 - 8.3.3 Type 2 Hypervisors 572
 - 8.3.4 Paravirtualization 574
 - 8.3.5 Memory Virtualization 576
 - 8.3.6 I/O Virtualization 578
 - 8.3.7 Virtual Appliances 579
 - 8.3.8 Virtual Machines on Multicore CPUs 579
 - 8.3.9 Licensing Issues 580

- 8.4 DISTRIBUTED SYSTEMS 580
 - 8.4.1 Network Hardware 583
 - 8.4.2 Network Services and Protocols 586
 - 8.4.3 Document-Based Middleware 590
 - 8.4.4 File-System-Based Middleware 591
 - 8.4.5 Object-Based Middleware 596
 - 8.4.6 Coordination-Based Middleware 598
 - 8.4.7 Grids 603

- 8.5 RESEARCH ON MULTIPLE PROCESSOR SYSTEMS 604

- 8.6 SUMMARY 605

9.1	THE SECURITY ENVIRONMENT	613
9.1.1	Threats	613
9.1.2	Intruders	615
9.1.3	Accidental Data Loss	616
9.2	BASICS OF CRYPTOGRAPHY	616
9.2.1	Secret-Key Cryptography	617
9.2.2	Public-Key Cryptography	618
9.2.3	One-Way Functions	619
9.2.4	Digital Signatures	619
9.2.5	Trusted Platform Module	621
9.3	PROTECTION MECHANISMS	622
9.3.1	Protection Domains	622
9.3.2	Access Control Lists	624
9.3.3	Capabilities	627
9.3.4	Trusted Systems	630
9.3.5	Trusted Computing Base	631
9.3.6	Formal Models of Secure Systems	632
9.3.7	Multilevel Security	634
9.3.8	Covert Channels	637
9.4	AUTHENTICATION	641
9.4.1	Authentication Using Passwords	642
9.4.2	Authentication Using a Physical Object	651
9.4.3	Authentication Using Biometrics	653
9.5	INSIDER ATTACKS	656
9.5.1	Logic Bombs	656
9.5.2	Trap Doors	657
9.5.3	Login Spoofing	658
9.6	EXPLOITING CODE BUGS	659
9.6.1	Buffer Overflow Attacks	660
9.6.2	Format String Attacks	662
9.6.3	Return to libc Attacks	664
9.6.4	Integer Overflow Attacks	665
9.6.5	Code Injection Attacks	666
9.6.6	Privilege Escalation Attacks	667

CONTENTS

- 9.7 MALWARE 667
 - 9.7.1 Trojan Horses 670
 - 9.7.2 Viruses 672
 - 9.7.3 Worms 682
 - 9.7.4 Spyware 684
 - 9.7.5 Rootkits 688
- 9.8 DEFENSES 692
 - 9.8.1 Firewalls 693
 - 9.8.2 Antivirus and Anti-Antivirus Techniques 695
 - 9.8.3 Code Signing 701
 - 9.8.4 Jailing 702
 - 9.8.5 Model-Based Intrusion Detection 703
 - 9.8.6 Encapsulating Mobile Code 705
 - 9.8.7 Java Security 709
- 9.9 RESEARCH ON SECURITY 711
- 9.10 SUMMARY 712

10 CASE STUDY 1: LINUX

719

- 10.1 HISTORY OF UNIX AND LINUX 720
 - 10.1.1 UNICS 720
 - 10.1.2 PDP-11 UNIX 721
 - 10.1.3 Portable UNIX 722
 - 10.1.4 Berkeley UNIX 723
 - 10.1.5 Standard UNIX 724
 - 10.1.6 MINIX 725
 - 10.1.7 Linux 726
- 10.2 OVERVIEW OF LINUX 728
 - 10.2.1 Linux Goals 729
 - 10.2.2 Interfaces to Linux 730
 - 10.2.3 The Shell 731
 - 10.2.4 Linux Utility Programs 734
 - 10.2.5 Kernel Structure 736

10.3 PROCESSES IN LINUX	739
10.3.1 Fundamental Concepts	739
10.3.2 Process Management System Calls in Linux	741
10.3.3 Implementation of Processes and Threads in Linux	745
10.3.4 Scheduling in Linux	752
10.3.5 Booting Linux	755
10.4 MEMORY MANAGEMENT IN LINUX	758
10.4.1 Fundamental Concepts	758
10.4.2 Memory Management System Calls in Linux	761
10.4.3 Implementation of Memory Management in Linux	762
10.4.4 Paging in Linux	768
10.5 INPUT/OUTPUT IN LINUX	771
10.5.1 Fundamental Concepts	772
10.5.2 Networking	773
10.5.3 Input/Output System Calls in Linux	775
10.5.4 Implementation of Input/Output in Linux	775
10.5.5 Modules in Linux	779
10.6 THE LINUX FILE SYSTEM	779
10.6.1 Fundamental Concepts	780
10.6.2 File System Calls in Linux	785
10.6.3 Implementation of the Linux File System	788
10.6.4 NFS: The Network File System	796
10.7 SECURITY IN LINUX	803
10.7.1 Fundamental Concepts	803
10.7.2 Security System Calls in Linux	805
10.7.3 Implementation of Security in Linux	806
10.8 SUMMARY	806

11 CASE STUDY 2: WINDOWS VISTA 813

11.1 HISTORY OF WINDOWS VISTA	813
11.1.1 1980s: MS-DOS	814
11.1.2 1990s: MS-DOS-based Windows	815
11.1.3 2000s: NT-based Windows	815
11.1.4 Windows Vista	818

11.2	PROGRAMMING WINDOWS VISTA	819
11.2.1	The Native NT Application Programming Interface	822
11.2.2	The Win32 Application Programming Interface	825
11.2.3	The Windows Registry	829
11.3	SYSTEM STRUCTURE	831
11.3.1	Operating System Structure	832
11.3.2	Booting Windows Vista	847
11.3.3	Implementation of the Object Manager	848
11.3.4	Subsystems, DLLs, and User-Mode Services	858
11.4	PROCESSES AND THREADS IN WINDOWS VISTA	861
11.4.1	Fundamental Concepts	861
11.4.2	Job, Process, Thread, and Fiber Management API Calls	866
11.4.3	Implementation of Processes and Threads	871
11.5	MEMORY MANAGEMENT	879
11.5.1	Fundamental Concepts	879
11.5.2	Memory Management System Calls	884
11.5.3	Implementation of Memory Management	885
11.6	CACHING IN WINDOWS VISTA	894
11.7	INPUT/OUTPUT IN WINDOWS VISTA	896
11.7.1	Fundamental Concepts	897
11.7.2	Input/Output API Calls	898
11.7.3	Implementation of I/O	901
11.8	THE WINDOWS NT FILE SYSTEM	906
11.8.1	Fundamental Concepts	907
11.8.2	Implementation of the NT File System	908
11.9	SECURITY IN WINDOWS VISTA	918
11.9.1	Fundamental Concepts	919
11.9.2	Security API Calls	921
11.9.3	Implementation of Security	922
11.10	SUMMARY	924

12 CASE STUDY 3: SYMBIAN OS

xvi

929

- 12.1 THE HISTORY OF SYMBIAN OS 930**
 - 12.1.1 Symbian OS Roots: Psion and EPOC 930
 - 12.1.2 Symbian OS Version 6 931
 - 12.1.3 Symbian OS Version 7 932
 - 12.1.4 Symbian OS Today 932
- 12.2 AN OVERVIEW OF SYMBIAN OS 932**
 - 12.2.1 Object Orientation 933
 - 12.2.2 Microkernel Design 934
 - 12.2.3 The Symbian OS Nanokernel 935
 - 12.2.4 Client/Server Resource Access 935
 - 12.2.5 Features of a Larger Operating System 936
 - 12.2.6 Communication and Multimedia 937
- 12.3 PROCESSES AND THREADS IN SYMBIAN OS 937**
 - 12.3.1 Threads and Nanothreads 938
 - 12.3.2 Processes 939
 - 12.3.3 Active Objects 939
 - 12.3.4 Interprocess Communication 940
- 12.4 MEMORY MANAGEMENT 941**
 - 12.4.1 Systems with No Virtual Memory 941
 - 12.4.2 How Symbian OS Addresses Memory 943
- 12.5 INPUT AND OUTPUT 945**
 - 12.5.1 Device Drivers 945
 - 12.5.2 Kernel Extensions 946
 - 12.5.3 Direct Memory Access 946
 - 12.5.4 Special Case: Storage Media 947
 - 12.5.5 Blocking I/O 947
 - 12.5.6 Removable Media 948
- 12.6 STORAGE SYSTEMS 948**
 - 12.6.1 File Systems for Mobile Devices 948
 - 12.6.2 Symbian OS File Systems 949
 - 12.6.3 File System Security and Protection 949
- 12.7 SECURITY IN SYMBIAN OS 950**

CONTENTS

12.8	COMMUNICATION IN SYMBIAN OS	953
12.8.1	Basic Infrastructure	953
12.8.2	A Closer Look at the Infrastructure	954
12.9	SUMMARY	957

13 OPERATING SYSTEM DESIGN

959

13.1	THE NATURE OF THE DESIGN PROBLEM	960
13.1.1	Goals	960
13.1.2	Why Is It Hard to Design an Operating System?	961
13.2	INTERFACE DESIGN	963
13.2.1	Guiding Principles	963
13.2.2	Paradigms	965
13.2.3	The System Call Interface	968
13.3	IMPLEMENTATION	971
13.3.1	System Structure	971
13.3.2	Mechanism versus Policy	975
13.3.3	Orthogonality	976
13.3.4	Naming	977
13.3.5	Binding Time	978
13.3.6	Static versus Dynamic Structures	979
13.3.7	Top-Down versus Bottom-Up Implementation	980
13.3.8	Useful Techniques	981
13.4	PERFORMANCE	987
13.4.1	Why Are Operating Systems Slow?	987
13.4.2	What Should Be Optimized?	988
13.4.3	Space-Time Trade-offs	988
13.4.4	Caching	991
13.4.5	Hints	992
13.4.6	Exploiting Locality	993
13.4.7	Optimize the Common Case	993
13.5	PROJECT MANAGEMENT	994
13.5.1	The Mythical Man Month	994
13.5.2	Team Structure	995

CONTENTS

xxiii

- 13.5.3 The Role of Experience 997
- 13.5.4 No Silver Bullet 998
- 13.6 TRENDS IN OPERATING SYSTEM DESIGN 998
 - 13.6.1 Virtualization 999
 - 13.6.2 Multicore Chips 999
 - 13.6.3 Large Address Space Operating Systems 1000
 - 13.6.4 Networking 1000
 - 13.6.5 Parallel and Distributed Systems 1001
 - 13.6.6 Multimedia 1001
 - 13.6.7 Battery-Powered Computers 1002
 - 13.6.8 Embedded Systems 1002
 - 13.6.9 Sensor Nodes 1003
- 13.7 SUMMARY 1003

14 READING LIST AND BIBLIOGRAPHY

1007

- 14.1 SUGGESTIONS FOR FURTHER READING 1007
 - 14.1.1 Introduction and General Works 1008
 - 14.1.2 Processes and Threads 1008
 - 14.1.3 Memory Management 1009
 - 14.1.4 Input/Output 1009
 - 14.1.5 File Systems 1010
 - 14.1.6 Deadlocks 1010
 - 14.1.7 Multimedia Operating Systems 1010
 - 14.1.8 Multiple Processor Systems 1011
 - 14.1.9 Security 1012
 - 14.1.10 Linux 1014
 - 14.1.11 Windows Vista 1014
 - 14.1.12 The Symbian OS 1015
 - 14.1.13 Design Principles 1015

- 14.2 ALPHABETICAL BIBLIOGRAPHY 1016

INDEX

1049