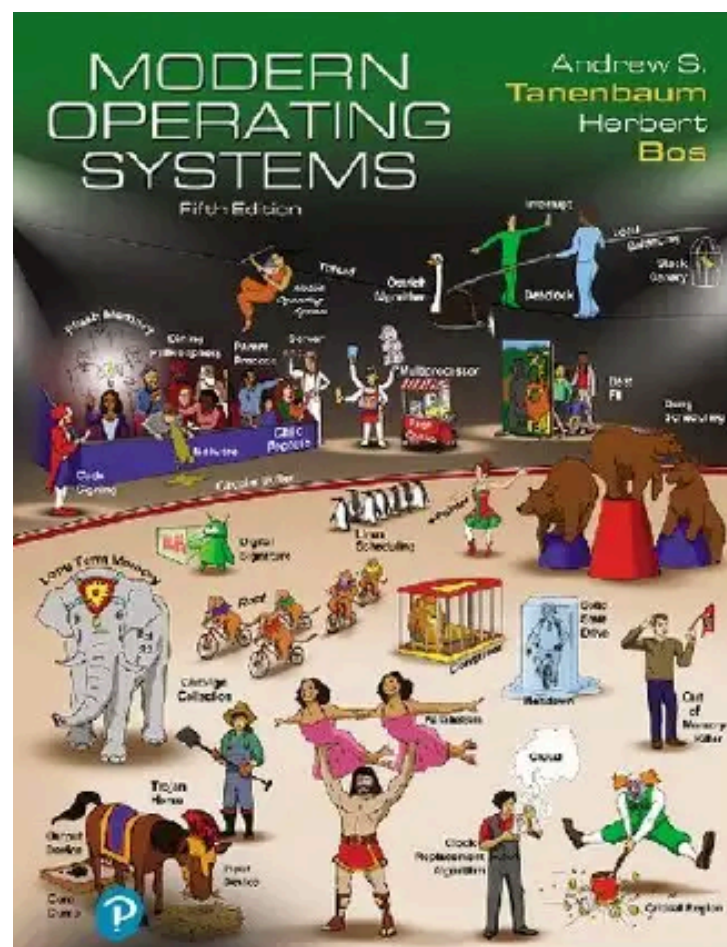




Modern Operating Systems 5th Edition Andrew S. Tanenbaum

<https://scribd.com.co/ebwork-download/modern-operating-systems-5th-edition-andrew-s-tanenbaum>

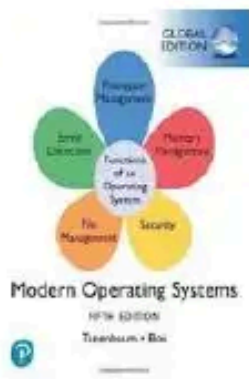


If link not working , Enter text link below to your browser and access it.

ebookwork.com/?p=20572

We provide a comprehensive selection of study tools, including ebooks, test banks, and solution manuals.

Note: Below are some related ebooks you may be interested in. To access them, please click the link below. In some cases, the link may not be clickable, so please open a new browser tab and manually enter the link.



Modern Operating Systems, 5th Global Edition Andrew Tanenbaum

[**EbookWork.com/?p=21060**](http://EbookWork.com/?p=21060)



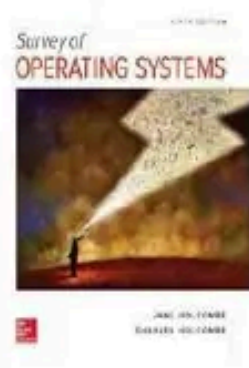
Computer Networks, Global Edition Andrew S. Tanenbaum

[**EbookWork.com/?p=21720**](http://EbookWork.com/?p=21720)



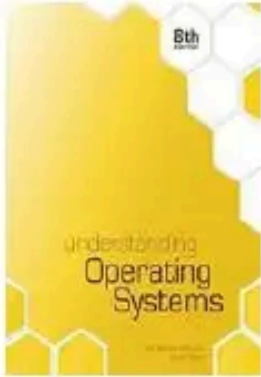
Survey of Operating Systems 5th Edition Jane Holcombe

[**EbookWork.com/?p=53516**](http://EbookWork.com/?p=53516)



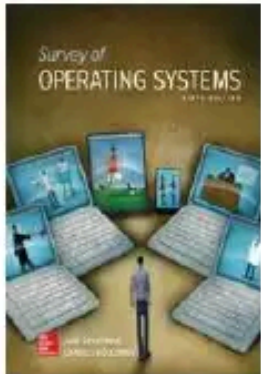
Survey of Operating Systems, 5e 5th Edition u2013 EBook PDF Version

[**EbookWork.com/?p=1192**](http://EbookWork.com/?p=1192)



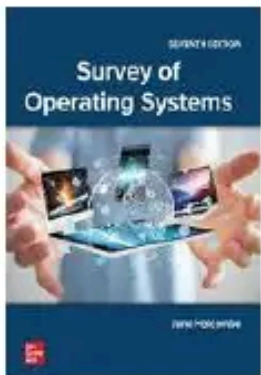
Understanding Operating Systems 8th Edition

EbookWork.com/?p=1306



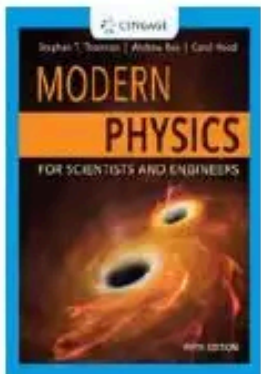
Survey of Operating Systems 6th Edition Jane Holcombe

EbookWork.com/?p=54707



Survey of Operating Systems, 7th Edition Jane Holcombe

EbookWork.com/?p=50774



Modern Physics for Scientists and Engineers 5th Edition Stephen Thornton & Andrew Rex

EbookWork.com/?p=38803



Understanding Operating Systems, 8th ed. 8th Edition Ann Mchoes

EbookWork.com/?p=37570

Andrew S.
Tanenbaum
Herbert
Bos

Interrupt

Load Balancing

Stack
Canary

Thread

Ostrich Algorithm

Deadlock

Flash Memory

Dining Philosophers

Parent Process

Server

Child Process

Multiprocessor

Page Cache

**Best
Fit**

Gang Scheduling

Code Signing

Circular Buffer

Linux Scheduling

pointer

**Digital
Signature**

Race

Linux Scheduling

Solid State Drive

Out of Memory Killer

Meltdown

Cloud

Clock Replacement Algorithm

Critical Region

Output Device

Input
Device

Core
Dump

Long Term Memory

Fat
32

Garbage Collection

Trojan Horse

Parallelism

**MODERN
OPERATING SYSTEMS**

FIFTH EDITION

This page intentionally left blank

**MODERN
OPERATING SYSTEMS**

FIFTH EDITION

**ANDREW S. TANENBAUM
HERBERT BOS**

Vrije Universiteit
Amsterdam, The Netherlands



Content Management: Tracy Johnson, Erin Sullivan
Content Production: Carole Snyder, Pallavi Pandit
Product Management: Holly Stark
Product Marketing: Krista Clark
Rights and Permissions: Anjali Singh

Please contact <https://support.pearson.com/getsupport/s/> with any queries on this content.

Cover Image by Jason Consalvo.

Microsoft and/or its respective suppliers make no representations about the suitability of the information contained in the documents and related graphics published as part of the services for any purpose. All such documents and related graphics are provided “as is” without warranty of any kind. Microsoft and/or its respective suppliers hereby disclaim all warranties and conditions with regard to this information, including all warranties and conditions of merchantability, whether express, implied or statutory, fitness for a particular purpose, title and non-infringement. In no event shall Microsoft and/or its respective suppliers be liable for any special, indirect or consequential damages or any damages whatsoever resulting from loss of use, data or profits, whether in an action of contract, negligence or other tortious action, arising out of or in connection with the use or performance of information available from the services.

The documents and related graphics contained herein could include technical inaccuracies or typographical errors. Changes are periodically added to the information herein. Microsoft and/or its respective suppliers may make improvements and/or changes in the product(s) and/or the program(s) described herein at any time. Partial screen shots may be viewed in full within the software version specified.

Microsoft® and Windows® are registered trademarks of the Microsoft Corporation in the U.S.A. and other countries. This book is not sponsored or endorsed by or affiliated with the Microsoft Corporation.

Copyright © 2023, 2014, 2008 by Pearson Education, Inc. or its affiliates, 221 River Street, Hoboken, NJ 07030. All Rights Reserved. Manufactured in the United States of America. This publication is protected by copyright, and permission should be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights and Permissions department, please visit www.pearsoned.com/permissions/.

Acknowledgments of third-party content appear on the appropriate page within the text.

PEARSON is an exclusive trademark owned by Pearson Education, Inc. or its affiliates in the U.S. and/or other countries.

Unless otherwise indicated herein, any third-party trademarks, logos, or icons that may appear in this work are the property of their respective owners, and any references to third-party trademarks, logos, icons, or other trade dress are for demonstrative or descriptive purposes only. Such references are not intended to imply any sponsorship, endorsement, authorization, or promotion of Pearson’s products by the owners of such marks, or any relationship between the owner and Pearson Education, Inc., or its affiliates, authors, licensees, or distributors.

Library of Congress Cataloging-in-Publication Data

MODERN OPERATING SYSTEMS

Library of Congress Control Number: 2022902564

ScoutAutomatedPrintCode



Rental
ISBN-10: 0-13-761887-5
ISBN-13: 978-0-13-761887-3

To Suzanne, Barbara, Aron, Nathan, Marvin, Matilde, Olivia and Mirte. (AST)

To Marieke, Duko, Jip, Loeka and Spot. (HB)

Pearson's Commitment to Diversity, Equity, and Inclusion



Pearson is dedicated to creating bias-free content that reflects the diversity, depth, and breadth of all learners' lived experiences.

We embrace the many dimensions of diversity, including but not limited to race, ethnicity, gender, sex, sexual orientation, socioeconomic status, ability, age, and religious or political beliefs.

Education is a powerful force for equity and change in our world. It has the potential to deliver opportunities that improve lives and enable economic mobility. As we work with authors to create content for every product and service, we acknowledge our responsibility to demonstrate inclusivity and incorporate diverse scholarship so that everyone can achieve their potential through learning. As the world's leading learning company, we have a duty to help drive change and live up to our purpose to help more people create a better life for themselves and to create a better world.

Our ambition is to purposefully contribute to a world where:

- Everyone has an equitable and lifelong opportunity to succeed through learning.
- Our educational content accurately reflects the histories and lived experiences of the learners we serve.
- Our educational products and services are inclusive and represent the rich diversity of learners.
- Our educational content prompts deeper discussions with students and motivates them to expand their own learning (and worldview).

Accessibility

We are also committed to providing products that are fully accessible to all learners. As per Pearson's guidelines for accessible educational Web media, we test and retest the capabilities of our products against the highest standards for every release, following the WCAG guidelines in developing new products for copyright year 2022 and beyond.



You can learn more about Pearson's commitment to accessibility at <https://www.pearson.com/us/accessibility.html>

Contact Us

While we work hard to present unbiased, fully accessible content, we want to hear from you about any concerns or needs with this Pearson product so that we can investigate and address them.



Please contact us with concerns about any potential bias at <https://www.pearson.com/report-bias.html>



For accessibility-related issues, such as using assistive technology with Pearson products, alternative text requests, or accessibility documentation, email the Pearson Disability Support team at disability.support@pearson.com

CONTENTS

PREFACE

xxiii

1 INTRODUCTION

1

- 1.1 WHAT IS AN OPERATING SYSTEM? 4
 - 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 (1945–1955): Vacuum Tubes 8
 - 1.2.2 The Second Generation (1955–1965): 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
 - 1.2.5 The Fifth Generation (1990–Present): Mobile Computers 19
- 1.3 COMPUTER HARDWARE REVIEW 20
 - 1.3.1 Processors 20
 - 1.3.2 Memory 24
 - 1.3.3 Nonvolatile Storage 28
 - 1.3.4 I/O Devices 29
 - 1.3.5 Buses 33
 - 1.3.6 Booting the Computer 34

1.4	THE OPERATING SYSTEM ZOO	36
1.4.1	Mainframe Operating Systems	36
1.4.2	Server Operating Systems	37
1.4.3	Personal Computer Operating Systems	37
1.4.4	Smartphone and Handheld Computer Operating Systems	37
1.4.5	The Internet of Things and Embedded Operating Systems	37
1.4.6	Real-Time Operating Systems	38
1.4.7	Smart Card Operating Systems	39
1.5	OPERATING SYSTEM CONCEPTS	39
1.5.1	Processes	39
1.5.2	Address Spaces	41
1.5.3	Files	42
1.5.4	Input/Output	45
1.5.5	Protection	46
1.5.6	The Shell	46
1.5.7	Ontogeny Recapitulates Phylogeny	47
1.6	SYSTEM CALLS	50
1.6.1	System Calls for Process Management	53
1.6.2	System Calls for File Management	57
1.6.3	System Calls for Directory Management	58
1.6.4	Miscellaneous System Calls	60
1.6.5	The Windows API	60
1.7	OPERATING SYSTEM STRUCTURE	63
1.7.1	Monolithic Systems	63
1.7.2	Layered Systems	64
1.7.3	Microkernels	66
1.7.4	Client-Server Model	68
1.7.5	Virtual Machines	69
1.7.6	Exokernels and Unikernels	73
1.8	THE WORLD ACCORDING TO C	74
1.8.1	The C Language	74
1.8.2	Header Files	75
1.8.3	Large Programming Projects	76
1.8.4	The Model of Run Time	78
1.9	RESEARCH ON OPERATING SYSTEMS	78
1.10	OUTLINE OF THE REST OF THIS BOOK	79

1.11 METRIC UNITS 80

1.12 SUMMARY 81

2 PROCESSES AND THREADS

85

2.1 PROCESSES 85

2.1.1 The Process Model 86

2.1.2 Process Creation 88

2.1.3 Process Termination 90

2.1.4 Process Hierarchies 91

2.1.5 Process States 92

2.1.6 Implementation of Processes 94

2.1.7 Modeling Multiprogramming 95

2.2 THREADS 97

2.2.1 Thread Usage 97

2.2.2 The Classical Thread Model 102

2.2.3 POSIX Threads 106

2.2.4 Implementing Threads in User Space 107

2.2.5 Implementing Threads in the Kernel 111

2.2.6 Hybrid Implementations 112

2.2.7 Making Single-Threaded Code Multithreaded 113

2.3 EVENT-DRIVEN SERVERS 116

2.4 SYNCHRONIZATION AND INTERPROCESS COMMUNICATION 119

2.4.1 Race Conditions 119

2.4.2 Critical Regions 120

2.4.3 Mutual Exclusion with Busy Waiting 121

2.4.4 Sleep and Wakeup 127

2.4.5 Semaphores 129

2.4.6 Mutexes 134

2.4.7 Monitors 138

2.4.8 Message Passing 145

2.4.9 Barriers 148

2.4.10 Priority Inversion 150

2.4.11 Avoiding Locks: Read-Copy-Update 151

2.5	SCHEDULING	152
2.5.1	Introduction to Scheduling	153
2.5.2	Scheduling in Batch Systems	160
2.5.3	Scheduling in Interactive Systems	162
2.5.4	Scheduling in Real-Time Systems	168
2.5.5	Policy Versus Mechanism	169
2.5.6	Thread Scheduling	169
2.6	RESEARCH ON PROCESSES AND THREADS	171
2.7	SUMMARY	172

3 MEMORY MANAGEMENT 179

3.1	NO MEMORY ABSTRACTION	180
3.1.1	Running Multiple Programs Without a Memory Abstraction	181
3.2	A MEMORY ABSTRACTION: ADDRESS SPACES	183
3.2.1	The Notion of an Address Space	184
3.2.2	Swapping	185
3.2.3	Managing Free Memory	188
3.3	VIRTUAL MEMORY	192
3.3.1	Paging	193
3.3.2	Page Tables	196
3.3.3	Speeding Up Paging	200
3.3.4	Page Tables for Large Memories	203
3.4	PAGE REPLACEMENT ALGORITHMS	207
3.4.1	The Optimal Page Replacement Algorithm	208
3.4.2	The Not Recently Used Page Replacement Algorithm	209
3.4.3	The First-In, First-Out (FIFO) Page Replacement Algorithm	210
3.4.4	The Second-Chance Page Replacement Algorithm	210
3.4.5	The Clock Page Replacement Algorithm	211
3.4.6	The Least Recently Used (LRU) Page Replacement Algorithm	212
3.4.7	Simulating LRU in Software	212
3.4.8	The Working Set Page Replacement Algorithm	214
3.4.9	The WSClock Page Replacement Algorithm	218
3.4.10	Summary of Page Replacement Algorithms	220

3.5	DESIGN ISSUES FOR PAGING SYSTEMS	221
3.5.1	Local versus Global Allocation Policies	221
3.5.2	Load Control	224
3.5.3	Cleaning Policy	225
3.5.4	Page Size	226
3.5.5	Separate Instruction and Data Spaces	227
3.5.6	Shared Pages	228
3.5.7	Shared Libraries	230
3.5.8	Mapped Files	232
3.6	IMPLEMENTATION ISSUES	232
3.6.1	Operating System Involvement with Paging	232
3.6.2	Page Fault Handling	233
3.6.3	Instruction Backup	234
3.6.4	Locking Pages in Memory	236
3.6.5	Backing Store	236
3.6.6	Separation of Policy and Mechanism	238
3.7	SEGMENTATION	240
3.7.1	Implementation of Pure Segmentation	242
3.7.2	Segmentation with Paging: MULTICS	243
3.7.3	Segmentation with Paging: The Intel x86	248
3.8	RESEARCH ON MEMORY MANAGEMENT	248
3.9	SUMMARY	250

4 FILE SYSTEMS

259

4.1	FILES	261
4.1.1	File Naming	261
4.1.2	File Structure	263
4.1.3	File Types	264
4.1.4	File Access	266
4.1.5	File Attributes	267
4.1.6	File Operations	269
4.1.7	An Example Program Using File-System Calls	270

4.2	DIRECTORIES	272
4.2.1	Single-Level Directory Systems	272
4.2.2	Hierarchical Directory Systems	273
4.2.3	Path Names	274
4.2.4	Directory Operations	277
4.3	FILE-SYSTEM IMPLEMENTATION	278
4.3.1	File-System Layout	278
4.3.2	Implementing Files	280
4.3.3	Implementing Directories	285
4.3.4	Shared Files	288
4.3.5	Log-Structured File Systems	290
4.3.6	Journaling File Systems	292
4.3.7	Flash-based File Systems	293
4.3.8	Virtual File Systems	298
4.4	FILE-SYSTEM MANAGEMENT AND OPTIMIZATION	301
4.4.1	Disk-Space Management	301
4.4.2	File-System Backups	307
4.4.3	File-System Consistency	312
4.4.4	File-System Performance	315
4.4.5	Defragmenting Disks	320
4.4.6	Compression and Deduplication	321
4.4.7	Secure File Deletion and Disk Encryption	322
4.5	EXAMPLE FILE SYSTEMS	324
4.5.1	The MS-DOS File System	324
4.5.2	The UNIX V7 File System	327
4.6	RESEARCH ON FILE SYSTEMS	330
4.7	SUMMARY	331

5 INPUT/OUTPUT

337

5.1	PRINCIPLES OF I/O HARDWARE	337
5.1.1	I/O Devices	338
5.1.2	Device Controllers	338
5.1.3	Memory-Mapped I/O	340

5.1.4	Direct Memory Access	344
5.1.5	Interrupts Revisited	347
5.2	PRINCIPLES OF I/O SOFTWARE	352
5.2.1	Goals of the I/O Software	352
5.2.2	Programmed I/O	354
5.2.3	Interrupt-Driven I/O	355
5.2.4	I/O Using DMA	356
5.3	I/O SOFTWARE LAYERS	357
5.3.1	Interrupt Handlers	357
5.3.2	Device Drivers	359
5.3.3	Device-Independent I/O Software	362
5.3.4	User-Space I/O Software	368
5.4	MASS STORAGE: DISK AND SSD	370
5.4.1	Magnetic Disks	370
5.4.2	Solid State Drives (SSDs)	381
5.4.3	RAID	385
5.5	CLOCKS	390
5.5.1	Clock Hardware	390
5.5.2	Clock Software	391
5.5.3	Soft Timers	394
5.6	USER INTERFACES: KEYBOARD, MOUSE, & MONITOR	395
5.6.1	Input Software	396
5.6.2	Output Software	402
5.7	THIN CLIENTS	419
5.8	POWER MANAGEMENT	420
5.8.1	Hardware Issues	421
5.8.2	Operating System Issues	422
5.8.3	Application Program Issues	428
5.9	RESEARCH ON INPUT/OUTPUT	428
5.10	SUMMARY	430

6 DEADLOCKS 437

- 6.1 RESOURCES 438
 - 6.1.1 Preemptable and Nonpreemptable Resources 438
 - 6.1.2 Resource Acquisition 439
 - 6.1.3 The Dining Philosophers Problem 440
- 6.2 INTRODUCTION TO DEADLOCKS 444
 - 6.2.1 Conditions for Resource Deadlocks 445
 - 6.2.2 Deadlock Modeling 445
- 6.3 THE OSTRICH ALGORITHM 447
- 6.4 DEADLOCK DETECTION AND RECOVERY 449
 - 6.4.1 Deadlock Detection with One Resource of Each Type 449
 - 6.4.2 Deadlock Detection with Multiple Resources of Each Type 451
 - 6.4.3 Recovery from Deadlock 454
- 6.5 DEADLOCK AVOIDANCE 455
 - 6.5.1 Resource Trajectories 456
 - 6.5.2 Safe and Unsafe States 457
 - 6.5.3 The Banker's Algorithm for a Single Resource 458
 - 6.5.4 The Banker's Algorithm for Multiple Resources 459
- 6.6 DEADLOCK PREVENTION 461
 - 6.6.1 Attacking the Mutual-Exclusion Condition 461
 - 6.6.2 Attacking the Hold-and-Wait Condition 462
 - 6.6.3 Attacking the No-Preemption Condition 462
 - 6.6.4 Attacking the Circular Wait Condition 463
- 6.7 OTHER ISSUES 464
 - 6.7.1 Two-Phase Locking 464
 - 6.7.2 Communication Deadlocks 465
 - 6.7.3 Livelock 467
 - 6.7.4 Starvation 468
- 6.8 RESEARCH ON DEADLOCKS 469
- 6.9 SUMMARY 470

7 VIRTUALIZATION AND THE CLOUD 477

- 7.1 HISTORY 480
- 7.2 REQUIREMENTS FOR VIRTUALIZATION 482
- 7.3 TYPE 1 AND TYPE 2 HYPERVISORS 484
- 7.4 TECHNIQUES FOR EFFICIENT VIRTUALIZATION 486
 - 7.4.1 Virtualizing the Unvirtualizable 487
 - 7.4.2 The Cost of Virtualization 489
- 7.5 ARE HYPERVISORS MICROKERNELS DONE RIGHT? 490
- 7.6 MEMORY VIRTUALIZATION 493
- 7.7 I/O VIRTUALIZATION 497
- 7.8 VIRTUAL MACHINES ON MULTICORE CPUS 501
- 7.9 CLOUDS 501
 - 7.9.1 Clouds as a Service 502
 - 7.9.2 Virtual Machine Migration 503
 - 7.9.3 Checkpointing 504
- 7.10 OS-LEVEL VIRTUALIZATION 504
- 7.11 CASE STUDY: VMWARE 507
 - 7.11.1 The Early History of VMware 507
 - 7.11.2 VMware Workstation 509
 - 7.11.3 Challenges in Bringing Virtualization to the x86 509
 - 7.11.4 VMware Workstation: Solution Overview 511
 - 7.11.5 The Evolution of VMware Workstation 520
 - 7.11.6 ESX Server: VMware's type 1 Hypervisor 521
- 7.12 RESEARCH ON VIRTUALIZATION AND THE CLOUD 523
- 7.13 SUMMARY 524

8 MULTIPLE PROCESSOR SYSTEMS 527

- 8.1 MULTIPROCESSORS 530
 - 8.1.1 Multiprocessor Hardware 530
 - 8.1.2 Multiprocessor Operating System Types 541
 - 8.1.3 Multiprocessor Synchronization 545
 - 8.1.4 Multiprocessor Scheduling 550
- 8.2 MULTICOMPUTERS 557
 - 8.2.1 Multicomputer Hardware 558
 - 8.2.2 Low-Level Communication Software 562
 - 8.2.3 User-Level Communication Software 565
 - 8.2.4 Remote Procedure Call 569
 - 8.2.5 Distributed Shared Memory 571
 - 8.2.6 Multicomputer Scheduling 575
 - 8.2.7 Load Balancing 576
- 8.3 DISTRIBUTED SYSTEMS 579
 - 8.3.1 Network Hardware 581
 - 8.3.2 Network Services and Protocols 585
 - 8.3.3 Document-Based Middleware 588
 - 8.3.4 File-System-Based Middleware 590
 - 8.3.5 Object-Based Middleware 594
 - 8.3.6 Coordination-Based Middleware 595
- 8.4 RESEARCH ON MULTIPLE PROCESSOR SYSTEMS 598
- 8.5 SUMMARY 600

9 SECURITY 605

- 9.1 FUNDAMENTALS OF OPERATING SYSTEM SECURITY 607
 - 9.1.1 The CIA Security Triad 608
 - 9.1.2 Security Principles 609
 - 9.1.3 Security of the Operating System Structure 611
 - 9.1.4 Trusted Computing Base 612
 - 9.1.5 Attackers 614
 - 9.1.6 Can We Build Secure Systems? 617

