

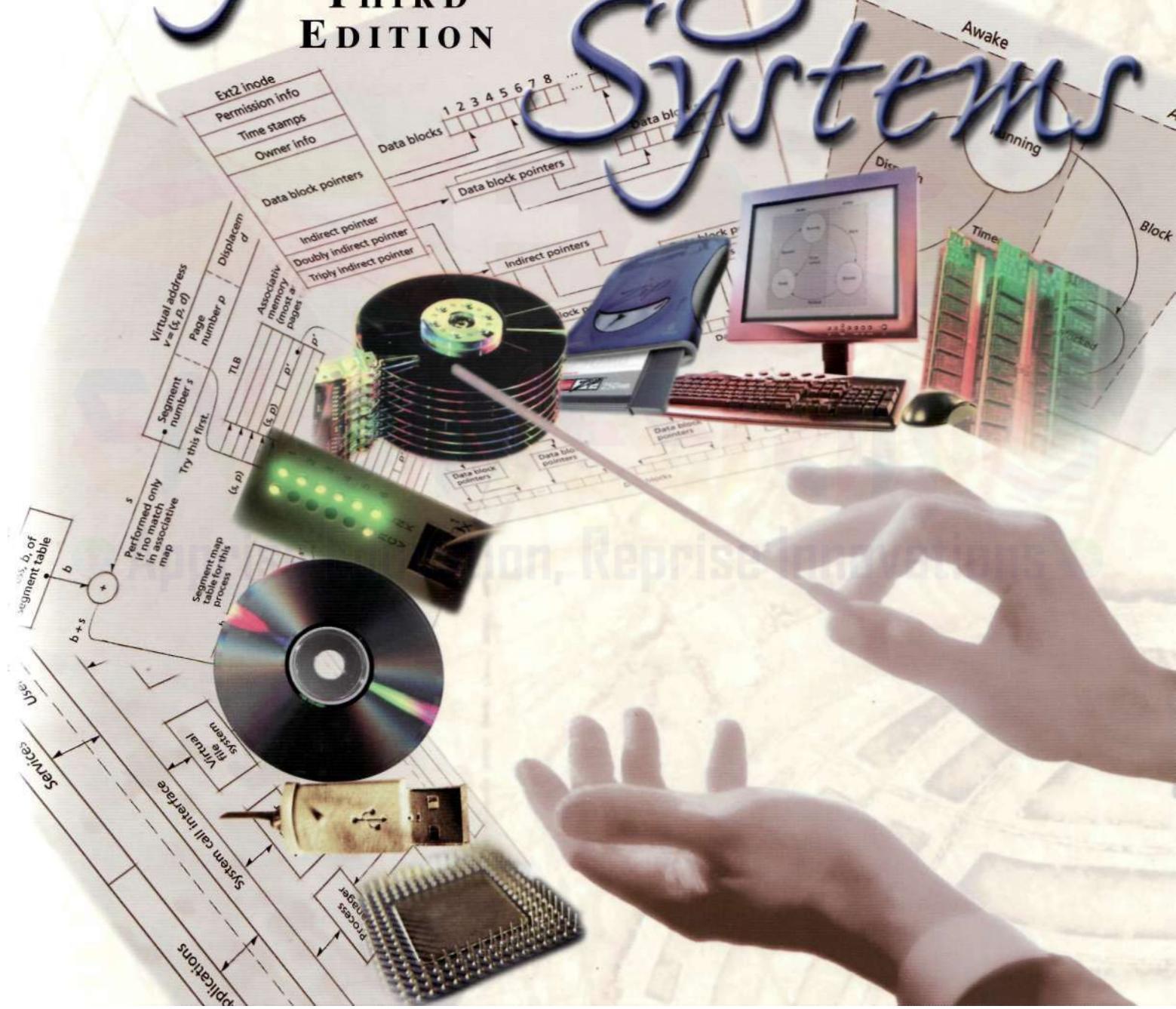
DEITEL

DEITEL

CHOFFNES

Operating Systems

THIRD EDITION



Operating Systems

Third Edition

H. M. Deitel

Deitel & Associates, Inc.

P. J. Deitel

Deitel & Associates, Inc.

D. R. Choffnes

Deitel & Associates, Inc.



Upper Saddle River, NJ 07458

Deitel Books, Cyber Classrooms, Complete Training Courses and Web-Based Training Courses published by Prentice Hall

Computer Science Series

Operating Systems, 3/E

How To Program Series

Advanced Java™ 2 Platform How to Program

C How to Program, 4/E

C++ How to Program, 4/E

C#® How to Program

e-Business and e-Commerce How to Program

Internet and World Wide Web How to Program, 3/E

Java™ How to Program, 5/E

Perl How to Program

Python How to Program

Visual Basic® 6 How to Program

Visual Basic® .NET How to Program, 2/E

Visual C++® .NET How to Program

Wireless Internet & Mobile Business How to Program

XML How to Program

Simply Series

Simply C#®:An Application-Driven Tutorial Approach

Simply Java™ Programming: An Application-Driven Tutorial Approach

Simply Visual Basic® .NET: An Application-Driven Tutorial Approach (Visual Studio .NET2002 Edition)

Simply Visual Basic® .NET: An Application-Driven Tutorial Approach (Visual Studio .NET2003 Edition)

CS1 Programming Series

Java™ Software Design

.NET How to Program Series

C#® How to Program

Visual Basic® .NET How to Program, 2/E

Visual C++® .NET How to Program

Visual Studio® Series

C#® How to Program

Getting Started with Microsoft® Visual

C++® 6 with an Introduction to MFC

Simply C#®:An Application-Driven Tutorial Approach

Simply Visual Basic® .NET: An Application-Driven Tutorial Approach (Visual Studio .NET2002 Edition)

Simply Visual Basic® .NET: An Application-Driven Tutorial Approach (Visual Studio .NET 2003 Edition)

Visual Basic® 6 How to Program

Visual Basic® .NET How to Program, 2/E

Visual C++® .NET How to Program

Deitel® Developer Series

Java™ Web Services for Experienced Programmers

Web Services A Technical Introduction

For Managers Series

e-Business and e-Commerce for Managers

Interactive Web-Based Training Series

Premium CourseCompass Version of Visual Basic® .NET Multimedia Cyber Classroom, 2/E

Premium CourseCompass Version of Java™ 2 Multimedia Cyber Classroom, 5/E

Premium CourseCompass Version of C++ Multimedia Cyber Classroom, 4/E

The Complete Training Course Series

The Complete C++ Training Course, 4/E
The Complete C#® Training Course
The Complete e-Business and e-Commerce Programming Training Course
The Complete Internet and World Wide Web Programming Training Course, 2/E
The Complete Java™ 2 Training Course, 5/E
The Complete Perl Training Course
The Complete Python Training Course
The Complete Visual Basic® 6 Training Course
The Complete Visual Basic® .NET Training Course, 2/E
The Complete Wireless Internet & Mobile Business Programming Training Course
The Complete XML Programming Training Course

Interactive Multimedia Cyber Classroom Series

C++ Multimedia Cyber Classroom, 4/E
C#® Multimedia Cyber Classroom
e-Business and e-Commerce Multimedia Cyber Classroom
Internet and World Wide Web Multimedia Cyber Classroom, 2/E
Java™ 2 Multimedia Cyber Classroom, 5/E
Perl Multimedia Cyber Classroom
Python Multimedia Cyber Classroom
Visual Basic® 6 Multimedia Cyber Classroom
Visual Basic® .NET Multimedia Cyber Classroom, 2/E
Wireless Internet & Mobile Business Programming Multimedia Cyber Classroom
XML Multimedia Cyber Classroom

To follow the Deitel publishing program, please subscribe to the free ***Deitel® Buzz Online*** e-mail newsletter at:

www.deitel.com/newsletter/subscribe.html

To communicate with the authors, send e-mail to:

deitel@deitel.com

For information on Deitel instructor-led, corporate training seminars offered worldwide visit:

www.deitel.com

For continuing updates on Prentice Hall and Deitel publications visit:

www.deitel.com,
www.prenhall.com/deitel or
www.InformIT.com/deitel

Vice President and Editorial Director, ECS: ***Marcia J. Horton***

Senior Acquisitions Editor: ***Kate Hargett***

Associate Editor: ***Jennifer Cappello***

Assistant Editor: ***Sarah Parker***

Editorial Assistant: ***Michael Giacobbe***

Vice President and Director of Production and Manufacturing, ESM: ***David W. Riccardi***

Executive Managing Editor: ***Vince O'Brien***

Managing Editor: ***Tom Mansreck***

Production Editors: ***Chirag Thakkar, John F. Loveli***

Production Editor, Media: ***Bob Engelhardt***

Production Assistant: ***Daniela Petrilli***

Director of Creative Services: ***Paul Belfanti***

A/V Production Editor: ***Xiaohong Zhu***

Art Studio: ***Artworks, York, PA***

Creative Director: ***Carole Anson***

Art Director: ***Geoffrey Cassar***

Cover Design: ***Harvey M. Deitel, David R. Choffnes, Suzanne Behnke***

Interior Design: ***Harvey M. Deitel, David R. Choffnes, Geoffrey Cassar, Anne DeMarinis***

Manufacturing Manager: ***Trudy Pisciotti***

Manufacturing Buyer: ***Lisa McDowell***

Marketing Manager: ***Pamela Hersperger***

Marketing Assistant: ***Barrie Reinhold***



© 2004 by Pearson Education, Inc.

Pearson Prentice Hall

Pearson Education, Inc.

Upper Saddle River, New Jersey 07458

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.
Pearson Prentice Hall® is a trademark of Pearson Education, Inc.

The authors 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 authors and publisher make no warranty of any kind, expressed or implied, with regard to these programs or to the documentation contained in this book. The authors 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.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks and registered trademarks. Where those designations appear in this book, and Pearson Education, Inc. and the authors were aware of a trademark claim, the designations have been printed in initial caps or all caps. All product names mentioned remain trademarks or registered trademarks of their respective owners.

DEITEL is a registered trademark of Deitel and Associates, Inc.

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

If you purchased this book within the United States or Canada you should be aware that it has been wrongfully imported without the approval of the Publisher or the Author.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-124696-8

Pearson Education Ltd., *London*

Pearson Education Australia Pty. Ltd., *Sydney*

Pearson Education Singapore, Pte. Ltd.

Pearson Education North Asia Ltd., *Hong Kong*

Pearson Education Canada, Inc., *Toronto*

Pearson Educacion de Mexico, S.A. de CV.

Pearson Education-Japan, *Tokyo*

Pearson Education Malaysia, Pte. Ltd.

Pearson Education, Inc., *Upper Saddle River, New Jersey*

In memory of Edsger W. Dijkstra.

Harvey M. Deitel

Paul J. Deitel

David R. Choffnes

*To my grandparents, Beverly and Jay Pddy:
For your love, generosity and inspiration.*

David

Contents

<i>List of Illustrations</i>	<i>xxvii</i>
<i>Preface.....</i>	<i>xxxvii</i>
<i>Part 1 Introduction to Hardware, Software and Operating System.....</i>	<i>1</i>
<i>1 Introduction to Operating Systems.....</i>	<i>3</i>
<i>1.1 Introduction.....</i>	<i>6</i>
<i>1.2 What Is an Operating System?.....</i>	<i>6</i>
<i>1.3 Early History: The 1940s and 1950s.....</i>	<i>7</i>
<i>Operating Systems Thinking: Innovation.....</i>	<i>8</i>
<i>1.4 The 1960s.....</i>	<i>9</i>
<i>Operating Systems Thinking: Relative Value of Human and Computer Resources.....</i>	<i>10</i>
<i>Biographical Note: Ken Thompson and Dennis Ritchie.....</i>	<i>12</i>
<i>1.5 The 1970s.....</i>	<i>13</i>
<i>Anecdote Abraham Lincoln's Technology Caution.....</i>	<i>14</i>
<i>1.6 The 1980s.....</i>	<i>15</i>
<i>1.7 History of the Internet and World Wide Web.....</i>	<i>16</i>
<i>Biographical Note: Doug Engelbart.....</i>	<i>16</i>
<i>Biographical Note: Tim Berners-lee.....</i>	<i>18</i>
<i>1.8 The 1990s.....</i>	<i>19</i>
<i>Biographical Note: Linus Torvalds.....</i>	<i>21</i>
<i>Biographical Note: Richard Stallman.....</i>	<i>22</i>
<i>1.9 2000 and Beyond.....</i>	<i>23</i>
<i>1.10 Application Bases.....</i>	<i>25</i>
<i>1.11 Operating System Environments.....</i>	<i>26</i>
<i>1.12 Operating System Components and Goals.....</i>	<i>28</i>
<i>12.1.1 Core Operating System Components.....</i>	<i>28</i>

<i>1.12.2 Operating System Goals.....</i>	30
<i>Operating System Thinking: Performance.....</i>	31
1.13 Operating System Architectures.....	33
<i>Operating Systems Thinking: Keep It Simple (KIS)</i>	33
<i>1.13.1 Monolithic Architecture.....</i>	33
 <i>Anecdote: System Architect vs. System Engineer.....</i>	 34
<i>1.13.2 layered Architecture.....</i>	35
<i>Operating Systems Thinking: Architecture.....</i>	35
<i>1.13.3 Microkernel Architecture.....</i>	36
<i>1.13.4 Networked and Distributed Operating Systems.....</i>	38
 2 Hardware and Software Concepts	 53
2.1 Introduction.....	56
2.2 Evolution of Hardware Devices.....	56
<i>Biographical Note: Gordon Moore and Moore's law.....</i>	57
2.3 Hardware Components.....	59
<i>23..1 Mainboard.....</i>	59
<i>2.3.2 Processors</i>	60
<i>2.3.3 Clocks.....</i>	62
<i>2.3.4 Memory Hierarchy</i>	63
<i>Operating Systems Thinking: Caching.....</i>	65
<i>2.3.5 Main Memory.....</i>	65
<i>2.3.6 Secondary Storage.....</i>	66
<i>2.3.7 Buses.....</i>	67
<i>2.3.8 Direct Memory Access (DMA).....</i>	68
<i>Operating Systems Thinking: legacy Hardware and Software.....</i>	70
<i>2.3. 9 Peripheral Devices.....</i>	70
2.4 Hardware Support for Operating Systems.....	72
<i>2.4.1 Processor.....</i>	72
<i>Operating Systems Thinking: Principle of least Privilege.....</i>	73
<i>Operating Systems Thinking: Protection.....</i>	74
<i>Anecdote: Origins of the Term "Glitch"</i>	75
<i>2.4.2 Timers and Clocks</i>	76
<i>2. 4.3 Bootstrapping.....</i>	76
<i>24.4 Plug and Play</i>	77

2.5	<i>Caching and Buffering.....</i>	78
	<i>Operating Systems Thinking: Heuristics.....</i>	79
2.6	<i>Software Overview.....</i>	80
	2.6.1 <i>Machine language and Assembly language.....</i>	80
	2.6.2 <i>Interpreters and Compilers</i>	81
	2.6.3 <i>High-level languages.....</i>	82
	2.6.4 <i>Structured Programming</i>	83
	2.6.5 <i>Object-Oriented Programming.....</i>	83
2.7	<i>Application Programming Interface (APIs).....</i>	85
2.8	<i>Compiling, linking and loading.....</i>	86
	2.8.1 <i>Compiling.....</i>	86
	2.8.2 <i>linking.....</i>	87
	<i>Mini Case Study: Mach.....</i>	90
	2.8.3 <i>loading.....</i>	91
2.9	<i>Firmware.....</i>	93
2.10	<i>Middleware.....</i>	93

Part 2 Processes and Threads..... **106**

3	<i>Process Concepts.....</i>	109
3.1	Introduction.....	III
	<i>Operating Systems Thinking: Customers Ultimately Want Applications.....</i>	111
	3.1.1 <i>Definition of Process</i>	111
	<i>Mini Case Study: CTSS and Multics</i>	112
	<i>Biographical Note: Fernando J. Corbató.....</i>	113
3.2	<i>Process States: Life Cycle of a Process.....</i>	113
3.3	<i>Process Management.....</i>	114
	3.3.1 <i>Process States and State Transitions.....</i>	115
	3.3.2 <i>Process Control Blocks (PCBs)/Process Descriptors.....</i>	116
	<i>Operating Systems Thinking: Data Structures in Operating Systems.....</i>	118
	3.3.3 <i>Process Operations.....</i>	118
	3.3.4 <i>Suspend and Resume.....</i>	120
	3.3.5 <i>Context Switching.....</i>	122
3.4	<i>Interrupts.....</i>	123
	<i>Operating Systems Thinking: Asynchronism vs. Synchronization.....</i>	124

3.4.1	<i>Interrupt Processing</i>	120
4.4.2	<i>Interrupt Classes</i>	127
3.5	<i>Interprocess Communication</i>	129
3.5.1	<i>Signals</i>	130
3.5.2	<i>Message Passing</i>	130
3.6	<i>Case Study: UNIX Processes</i>	132
	<i>Mini Case Study: UNIX Systems</i>	133
4	Thread Concepts	145
4.1	<i>Introduction</i>	147
	<i>Operating Systems Thinking: Concurrency</i>	147
4.2	<i>Definition of Thread</i>	148
4.3	<i>Motivation for Threads</i>	150
	<i>Operating Systems Thinking: Parallelism</i>	151
4.4	<i>Thread States: Life Cycle of a Thread</i>	152
4.5	<i>Thread Operations</i>	154
4.6	<i>Threading Models</i>	155
4.6.1	<i>User-level Threads</i>	155
4.6.2	<i>Kernel-level Threads</i>	157
4.6.3	<i>Combining User- and Kernel-Level Threads</i>	159
	<i>Operating Systems Thinking: Standards Conformance</i>	159
4.7	<i>Thread Implementation Considerations</i>	161
4.7.1	<i>Thread Signal Delivery</i>	161
	<i>Anecdote: Engineering</i>	163
4.7.2	<i>Thread Termination</i>	165
4.8	<i>POSIX and Pthreads</i>	165
	<i>Anecdote: Standards and Conformance: Plug-to-Plug Compatibility</i>	166
4.9	<i>Linux Threads</i>	167
	<i>Operating Systems Thinking: Scalability</i>	168
4.10	<i>Windows XP Threads</i>	169
4.11	<i>Java Multithreading Case Study, Part I: Introduction to Java Threads</i>	172
5	Asynchronous Concurrent Execution	185
5.1	<i>Introduction</i>	187

5.2 Mutual Exclusion.....	187
5.2.1 Java Multithreading Case Study, Part II:	
<i>A Producer/Consumer Relationship in Java.....</i>	188
5.2.2 Critical Section.....	197
5.2.3 Mutual Exclusion Primitives	198
5.3 Implementing Mutual Exclusion Primitives.....	199
5.4 Software Solutions to the Mutual Exclusion Problem.....	200
5.4.1 Dekker's Algorithm.....	200
5.4.2 Peterson's Algorithm.....	211
<i>Anecdote: Why Should you Believe That your Software Is Functioning Correctly?.....</i>	211
5.4.3 N-Thread Mutual Exclusion: Lamport's Bakery Algorithm.....	215
<i>Biographical Note: Leslie Lamport.....</i>	216
5.5 Hardware Solutions to the Mutual Exclusion Problem.....	221
5.5.1 Disabling Interrupts.....	221
5.5.2 Test-and-Set Instruction.....	222
5.5.3 Swap Instruction.....	224
5.6 Semaphores.....	227
<i>Biographical Note: Edsger W. Dijkstra.....</i>	227
5.6.1 Mutual Exclusion with Semaphores	228
5.6.2 Thread Synchronization with Semaphores.....	229
5.6.3 Counting Semaphores.....	231
5.6.4 Implementing Semaphores.....	232
<i>Anecdote: Unclear Requirements.....</i>	233
6 Concurrent Programming.....	245
6.1 Introduction.....	247
<i>Anecdote: Exhaustive Testing Is Impossible.....</i>	248
6.2 Monitors.....	249
<i>Operating Systems Thinking: Information Hiding</i>	250
6.2.1 Condition Variables.....	251
6.2.2 Simple-Resource-Allocation with Monitors	252
<i>Biographical Note: Per Brinch Hansen.....</i>	252
6.2.3 Monitor Example: Circular Buffer.....	254
6.2.4 Monitor Example: Readers and Writers	257
6.3 Java Monitors.....	260

6.4	<i>Java Multithreading Case Study, Part III. Producer/Consumer Relationship in Java ...</i>	261
6.5	<i>Java Multithreading Case Study, Part IV Circular Buffer in Java.....</i>	270
7	<i>Deadlock and Indefinite Postponement.....</i>	289
7.1	<i>Introduction.....</i>	291
7.2	<i>Examples of Deadlock</i>	291
7.2.1	<i>Traffic Deadlock</i>	292
7.2.2	<i>Simple Resource Deadlock</i>	293
<i>Anecdote: One-lane Bridge.....</i>	293	
7.2.3	<i>Deadlock in Spooling Systems.....</i>	294
<i>Operating Systems Thinking: Waiting, Deadlock and Indefinite Postponement.....</i>	295	
7.2.4	<i>Example: Dining Philosophers.....</i>	296
7.3	<i>Related Problem: Indefinite Postponement</i>	298
7.4	<i>Resource Concepts.....</i>	298
7.5	<i>Four Necessary Conditions for Deadlock</i>	299
7.6	<i>Deadlock Solutions.....</i>	300
7.7	<i>Deadlock Prevention.....</i>	301
7.7.1	<i>Denying the "Wait-For" Condition.....</i>	302
<i>Anecdote: No Nuts, Bolts or Screws Allowed.....</i>	302	
7.7.2	<i>Denying the "No-Preemption" Condition.....</i>	303
7.7.3	<i>Denying the "Circular-Wait" Condition.....</i>	304
7.8	<i>Deadlock Avoidance with Dijkstra's Banker's Algorithm.....</i>	306
<i>Anecdote: Acronyms</i>	307	
7.8.1	<i>Example of a Safe States.....</i>	309
7.8.2	<i>Example of an Unsafe State.....</i>	309
7.8.3	<i>Example of Safe-State-to-Unsafe-State Transition.....</i>	310
7.8.4	<i>Banker's Algorithm Resource Allocation.....</i>	311
7.8.5	<i>Weaknesses in the Banker's Algorithm.....</i>	312
7.9	<i>Deadlock Detection.....</i>	312
7.9.1	<i>Resource-Allocation Graphs.....</i>	313
7.9.2	<i>Reduction of Resource-Allocation Graphs.....</i>	314
7.10	<i>Deadlock Recovery</i>	316
7.11	<i>Deadlock Strategies in Current and Future Systems.....</i>	318

8 Processor Scheduling.....	333
8.1 <i>Introduction.....</i>	335
8.2 <i>Scheduling Levels.....</i>	335
8.3 <i>Preemptive vs. Nonpreemptive Scheduling</i>	337
<i>Operating Systems Thinking: Overhead.....</i>	338
8.4 <i>Priorities.....</i>	339
8.5 <i>Scheduling Objectives</i>	340
<i>Operating Systems Thinking: Predictability.....</i>	341
<i>Operating Systems Thinking: Fairness.....</i>	342
8.6 <i>Scheduling Criteria.....</i>	342
8.7 <i>Scheduling Algorithms.....</i>	344
8.7.1 <i>First-In-First-Out (FIFO) Scheduling.....</i>	344
8.7.2 <i>Round-Robin (RR) Scheduling.....</i>	345
8.7.3 <i>Shortest-Process-First (SPF) Scheduling.....</i>	348
8.7.4 <i>Highest-Response-Ratio-Next (HRRN) Scheduling.....</i>	349
8.7.5 <i>Shortest-Remaining-Time- (SRT) Scheduling.....</i>	349
8.7.6 <i>Multilevel Feedback Queues</i>	350
8.7.7 <i>Fair Share Scheduling.....</i>	354
8.8 <i>Deadline Scheduling.....</i>	356
<i>Operating Systems Thinking: Intensity of Resource Management vs. Relative Resource Value ...</i>	357
8.9 <i>Real-Time-Scheduling.....</i>	357
<i>Mini Case Study: Real-Time Operating Systems.....</i>	358
8.10 <i>Java Thread Scheduling.....</i>	360
 Part 3 Physical and Virtual Memory.....	 374
 9 Real Memory Organization and Management.....	 377
9.1 <i>Introduction.....</i>	379
9.2 <i>Memory Organization.....</i>	379
<i>Operating Systems Thinking: There Are No Upper Limits to Processing Power, Memory, Storage and Bandwidth.....</i>	380
9.3 <i>Memery Management</i>	380
9.4 <i>Memory Hierarchy.....</i>	382
9.5 <i>Memory Management Strategies.....</i>	383

9.6	<i>Contiguous vs. Noncontiguous Memory Allocation</i>	384
9.7	<i>Single-User Contiguous Memory Allocation</i>	385
9.7.1	<i>Overlays</i>	386
9.7.2	<i>Protection in a Single-User System</i>	387
9.7.3	<i>Single-Stream Batch Processing</i>	389
	<i>Operating Systems Thinking: Change Is the Rule Rather Than the Exception</i>	389
9.8	<i>Fixed-Partition Multiprogramming</i>	390
	<i>Anecdote: Compartmentalization</i>	393
	<i>Operating Systems Thinking: Spatial Resources and Fragmentation</i>	394
9.9	<i>Variable-Partition Multiprogramming</i>	395
9.9.1	<i>Variable-Partition Characteristics</i>	395
9.9.2	<i>Memory Placement Strategies</i>	399
9.10	<i>Multiprogramming with Memory Swapping</i>	401
10	<i>Virtual Memory Organization</i>	413
10.1	<i>Introduction</i>	415
	<i>Mini case Study: Atlas</i>	415
	<i>Operating Systems Thinking: Virtualization</i>	416
	<i>Biographical Note: Peter Denning</i>	417
10.2	<i>Virtual Memory: Basic Concepts</i>	418
	<i>Anecdote: Virtual Memory Unnecessary</i>	418
10.3	<i>Block Mapping</i>	420
10.4	<i>Paging</i>	424
10.4.1	<i>Paging Address Translation by Direct Mapping</i>	428
10.4.2	<i>Paging Address Translation by Associative Mapping</i>	429
10.4.3	<i>Paging Address Translation with Direct/Associative Mapping</i>	430
	<i>Operating Systems Thinking: Empirical Results: locality-Based Heuristics</i>	432
10.4.4	<i>Multilevel Page Tables</i>	433
10.4.5	<i>Inverted Page Tables</i>	436
10.4.6	<i>Sharing in a Paging System</i>	439
	<i>Operating Systems Thinking: lazy Allocation</i>	441
10.5	<i>Segmentation</i>	442
10.5.1	<i>Segmentation Address Translation by Direct Mapping</i>	444
10.5.2	<i>Sharing in a Segmentation System</i>	446

10.5.3	<i>Protection and Access Control in Segmentation Systems</i>	447
10.6	<i>Segmentation Paging Systems</i>	451
	10.6.1 <i>Dynamic Address Translation in a Segmentation/Paging System</i>	451
	10.6.2 <i>Sharing and Protection in a Segmentation/Paging System</i>	455
10.7	<i>Case Study: IA-32 Intel Architecture Virtual Memory</i>	455
	<i>Mini Case Study: IBM Mainframe Operating Systems</i>	459
	<i>Mini Case Study: Early History of the VM Operating System</i>	461
 11 Virtual Memory Management		 477
11.1	<i>Introduction</i>	479
11.2	<i>Locality</i>	479
11.3	<i>Demand Paging</i>	480
	<i>Operating Systems Thinking, Computer Theory in Operating Systems</i>	481
	<i>Operating Systems Thinking, Space-Time Trade-offs</i>	482
11.4	<i>Anticipatory Paging</i>	483
11.5	<i>Page Replacement</i>	485
11.6	<i>Page-Replacement Strategies</i>	486
	11.6.1 <i>Random Page Replacement</i>	486
	11.6.2 <i>First-In-First-Out (FIFO) Page Replacement</i>	487
	11.6.3 <i>FIFO Anomaly</i>	488
	11.6.4 <i>Least-Recently-Used (LRU) Page Replacement</i>	489
	11.6.5 <i>Least-Frequently-Used (LFU) Page-Replacement</i>	491
	11.6.6 <i>Not-Used-Recently (NUR) Page Replacement</i>	491
	11.6.7 <i>Modifications to FIFO: Second-Chance and Clock Page Replacement</i>	493
	11.6.8 <i>Far Page Replacement</i>	494
11.7	<i>Working Set Model</i>	496
11.8	<i>Page-Fault-Frequency (PFF) Page Replacement</i>	500
11.9	<i>Page Release</i>	502
11.10	<i>Page Size</i>	502
11.11	<i>Program Behavior under Paging</i>	505
11.12	<i>Global vs. local Page Replacement</i>	507
11.13	<i>Case Study, Linux Page Replacement</i>	508

<i>Part 4 Secondary Storage, Files and Databases.....</i>	522
<i> 12 Disk Performance Optimization.....</i>	525
12.1 <i>Introduction.....</i>	528
12.2 <i>Evolution of Secondary Storage.....</i>	528
12.3 <i>Characteristics of Moving Head Disk Storage.....</i>	529
12.4 <i>Why Disk Scheduling Is Necessary</i>	531
12.5 <i>Disk Scheduling Strategies</i>	533
12.5.1 <i>First-Come-First-Served (FCFS) Disk Scheduling</i>	535
12.5.2 <i>Shortest-Seek-Time-First (SSTF) Disk Scheduling.....</i>	536
<i>Anecdote: Every Problem Has a Solution and Every Solution Has a Problem.....</i>	536
12.5.3 <i>SCAN Disk Scheduling</i>	537
12.5.4 <i>C-SCAN Disk Scheduling.....</i>	538
12.5.5 <i>FSCAN and N-Step SCAN Disk Scheduling.....</i>	539
12.5.6 <i>LOOK and C-LOOK Disk Scheduling.....</i>	541
12.6 <i>Rotational Optimization.....</i>	542
12.6.1 <i>SLTF Scheduling.....</i>	542
12.6.2 <i>SPTF and SATF Scheduling.....</i>	543
12.7 <i>System Considerations.....</i>	546
<i>Operating Systems Thinking: Saturation and Bottlenecks.....</i>	546
12.8 <i>Caching and Buffering.....</i>	548
12.9 <i>Other Disk Performances Techniques.....</i>	549
<i>Operating Systems Thinking: Compression and Decompression.....</i>	550
<i>Operating Systems Thinking: Redundancy.....</i>	551
12.10 <i>Redundant Arrays of Independent Disks (RAID)</i>	552
12.10.1 <i>RAID Overview.....</i>	552
<i>Operating Systems Thinking: Fault Tolerance.....</i>	554
<i>Operating Systems Thinking: Mission-n-Critical Systems.....</i>	554
12.10.2 <i>Level 0 (Striping).....</i>	555
12.10.3 <i>Level 1 (Mirroring).....</i>	556
12.10.4 <i>Level 2 (Bit-level Hamming ECC Parity).....</i>	558
12.10.5 <i>Level 3 (Bit-level XOR ECC Parity)</i>	560
12.10.6 <i>Level 4 (Block-level XOR ECC Parity).....</i>	562
12.10.7 <i>level 5 (Block-level Distributed XOR ECC Parity).....</i>	563

13 File and Database Systems.....	583
13.1 Introduction.....	586
13.2 Data Hierarchy.....	586
13.3 Files.....	587
13.4 File Systems.....	589
<i>Operating Systems Thinking: Encryption and Decryption.....</i>	590
13.4.1 Directories	590
13.4.2 Metadata.....	594
13.4.3 Mounting.....	596
13.5 File Organization.....	598
13.6 File Allocation.....	599
13.6.1 Contiguous File Allocation.....	599
13.6.2 Linked-list Noncontiguous File Allocation.....	600
13.6.3 Tabular Noncontiguous File Allocation.....	602
<i>Mini Case Study: MS-DOS.....</i>	604
13.6.4 Indexed Noncontiguous File Allocation.....	605
13.7 Free Space Management.....	608
13.8 File Access Control.....	610
13.8.1 Access Control fMatrix.....	610
<i>Operating Systems Thinking: Security.....</i>	610
13.8.2 Access Control by User Classes.....	611
13.9 Data Access Techniques.....	612
13.10 Data Integrity Protection.....	613
13.10.1 Backup and Recovery.....	613
<i>Operating Systems Thinking: Backup and Recovery.....</i>	614
13.10.2 Data Integrity and log Structured File Systems.....	610
<i>Operating Systems Thinking: Murphy's law and Robust Systems</i>	616
13.11 File Servers and Distributed Systems.....	619
13.12 Database Systems.....	619
13.12.1 Advantages of Database Systems.....	620
13.12.2 Data Access.....	620
13.12.3 Relational Database Model.....	621
13.12.4 Operating Systems and Database Systems.....	623

<i>Part 5 Performance, Processors and Multiprocessor Management.....</i>	638
<i>14 Performance and Processor Design.....</i>	641
<i>14.1 Introduction.....</i>	643
<i>14.2 Important Trends Affecting Performance Issues.....</i>	643
<i>14.3 Why Performance Monitoring and Evaluation Are Needed.....</i>	644
<i>14.4 Performance Measures</i>	645
<i>14.5 Performance Evaluation Techniques.....</i>	647
<i>14.5.1 Tracing and Profiling.....</i>	648
<i>14.5.2 Timings and Microbenchmarks</i>	649
<i>14.5.3 Application Specific Evaluation.....</i>	650
<i>14.5.4 Analytic Models.....</i>	652
<i>14.5.5 Benchmarks.....</i>	652
<i>14.5.6 Synthetic Programs.....</i>	654
<i>14.5.7 Simulation.....</i>	655
<i>14.5.8 Performance Monitoring.....</i>	656
<i>14.6 Bottlenecks and Saturation.....</i>	658
<i>14.7 Feedback Loops.....</i>	659
<i>14.7.1 Negative Feedback.....</i>	659
<i>14.7.2 Positive Feedback.....</i>	659
<i>14.8 Performance Techniques in Processor Design.....</i>	660
<i>14.8.1 Complex Instruction Set Computing (CISC)</i>	661
<i>14.8.2 Reduced Instruction Set Computing (RISC).....</i>	662
<i>14.8.3 Post-RISC Processors.....</i>	665
<i>14.8.4 Explicitly Parallel Instruction Computing (EPIC).....</i>	668
<i>15 Multiprocessor Management.....</i>	683
<i>15.1 Introduction.....</i>	686
<i>Mini Case Study: Supercomputers.....</i>	686
<i>Biographical Note: Seymour Cray</i>	687
<i>15.2 Multiprocessor Architecture.....</i>	688
<i>15.2.1 Classifying Sequential and Parallel Architectures.....</i>	688
<i>15.2.2 Processor Interconnection Schemes.....</i>	690
<i>15.2.3 Loosely Coupled vs. Tightly Coupled Systems.....</i>	696

<i>15.3 Multiprocessor Operating System Organizations.....</i>	698
15.3.1 <i>Master/Slave.....</i>	698
15.3.2 <i>Separate Kernels.....</i>	699
15.3.3 <i>Symmetrical Organization.....</i>	700
<i>Operating Systems Thinking: Graceful Degradation.....</i>	701
<i>15.4 Memory Access Architectures.....</i>	701
15.4.1 <i>Uniform Memory Access.....</i>	702
15.4.2 <i>Nonuniform Memory Access.....</i>	703
15.4.3 <i>Cache-Only Memory Architecture.....</i>	704
15.4.4 <i>No Remote Memory Access.....</i>	705
<i>15.5 Multiprocessor Memory Sharing.....</i>	707
<i>Operating Systems Thinking: Data Replication and Coherency.....</i>	707
15.5.1 <i>Cache Coherence.....</i>	708
15.5.2 <i>Page Replication and Migration.....</i>	709
15.5.3 <i>Shared Virtual Memory</i>	710
<i>15.6 Multiprocessor Scheduling.....</i>	713
15.6.1 <i>Job-Blind Multiprocessor Scheduling.....</i>	714
15.6.2 <i>Job-Aware Multiprocessor Scheduling.....</i>	715
<i>15.7 Process Migration.....</i>	719
15.7.1 <i>Flow of Process Migration.....</i>	719
15.7.2 <i>Process Migration Concepts.....</i>	721
15.7.3 <i>Process Migration Strategies.....</i>	722
<i>15.8 Load Balancing.....</i>	723
15.8.1 <i>Static load Balancing.....</i>	724
15.8.2 <i>Dynamic load Balancing.....</i>	725
<i>15.9 Multiprocessor Mutual Exclusion.....</i>	729
15.9.1 <i>Spin locks.....</i>	729
15.9.2 <i>Sleep/Wakeup locks</i>	730
15.9.3 <i>Read/Write locks.....</i>	731
<i>Part 6 Networking and Disiributed Computing.....</i>	746
<i>16 Introduction to Networking.....</i>	749
<i>16.1 Introduction.....</i>	751

16.2	<i>Network Topology</i>	751
	<i>Mini Case Study: Symbian OS</i>	754
16.3	<i>Network Types</i>	754
16.4	<i>TCP/IP Protocol Stack</i>	755
16.5	<i>Application Layer</i>	756
	16.5.1 <i>Hypertext Transfer Protocol (HTTP)</i>	757
	16.5.2 <i>File Transfer Protocol (FTP)</i>	758
16.6	<i>Transport Layer</i>	758
	<i>Anecdote: Unreliable Communication lines Can lead to Embarrassment</i>	759
	16.6.1 <i>Transmission Control Protocol (TCP)</i>	760
	16.6.2 <i>User Datagram Protocol (UDP)</i>	761
16.7	<i>Network Layer</i>	762
	16.7.1 <i>Internet Protocol (IP)</i>	763
	16.7.2 <i>Internet Protocol version 6 (IPv6)</i>	763
16.8	<i>Link Layer</i>	764
	16.8.1 <i>Ethernet</i>	765
	16.8.2 <i>Token Ring</i>	766
	16.8.3 <i>Fiber Distributed Data Interface (FDDI)</i>	766
	16.8.4 <i>IEEE 802.11 (Wireless)</i>	768
16.9	<i>Client/Server Model</i>	768
17	<i>Introduction to Distributed Systems</i>	781
17.1	<i>Introduction</i>	783
17.2	<i>Attributes of Distributed Systems</i>	784
	17.2.1 <i>Performance and Scalability</i>	784
	17.2.2 <i>Connectivity and Security</i>	784
	17.2.3 <i>Reliability and Fault Tolerance</i>	785
	17.2.4 <i>Transparency</i>	786
	17.2.5 <i>Network Operating Systems</i>	787
	17.2.6 <i>Distributed Operating Systems</i>	787
17.3	<i>Communication in Distributed Systems</i>	788
	17.3.1 <i>Middleware</i>	788
	<i>Anecdote: Consequences of Errors</i>	789
	17.3.2 <i>Remote Procedure Call (RPC)</i>	790

17.3.3 <i>Remote Method Invocation (RMI)</i>	791
17.3.4 <i>CORBA (Common Object Request Broker Architectures)</i>	792
17.3.5 <i>DCOM (Distributed Component Object Model)</i>	793
17.3.6 <i>Process Migration in Distributed Systems</i>	793
17.4 <i>Synchronization in Distributed Systems</i>	794
17.5 <i>Mutual Exclusion Distributed Systems</i>	795
17.5.1 <i>Mutual Exclusion without Shared Memory</i>	795
17.5.2 <i>Agrawala and Ricart's Distributed Mutual Exclusion Algorithm</i>	796
17.6 <i>Deadlock in Distributed Systems</i>	797
17.6.1 <i>Distributed Deadlock</i>	797
17.6.2 <i>Deadlock Prevention</i>	798
17.6.3 <i>Deadlock Detection</i>	800
17.6.4 <i>A Distributed Resource Deadlock Algorithm</i>	801
17.7 <i>Case Study: The Sprite Distributed Operating System</i>	803
17.8 <i>Case Study: The Amoeba Distributed Operating System</i>	804
 18 <i>Distributed System and Web Services</i>	815
18.1 <i>Introduction</i>	817
18.2 <i>Distributed File Systems</i>	817
18.2.1 <i>Distributed file System Concepts</i>	817
18.2.2 <i>Network File System (NFS)</i>	820
18.2.3 <i>Andrew File System (AFS)</i>	822
18.2.4 <i>Coda File System</i>	824
18.2.5 <i>Sprite File System</i>	827
18.3 <i>Multicomputer Systems</i>	830
18.4 <i>Clustering</i>	831
18.4.1 <i>Clustering Types</i>	832
18.4.2 <i>Clustering Benefits</i>	833
18.4.3 <i>Clustering Examples</i>	833
18.5 <i>Peer-to-Peer Distributed Computing</i>	836
18.5.1 <i>Client/Server and Peer-to-Peer Applications</i>	836
18.5.2 <i>Centralized vs. Decentralized P2P Applications</i>	836
18.5.3 <i>Peer Discovery and Searching</i>	839
18.5.4 <i>JXTA</i>	840
18.6 <i>Grid Computing</i>	842

18.7	<i>Java Distributed Computing</i>	843
18.7.1	<i>Java Servlets and JavaServer Pages (JSP)</i>	843
18.7.2	<i>Jini</i>	845
18.7.3	<i>JavaSpaces</i>	847
18.7.4	<i>Java Management Extensions (JMX)</i>	849
18.8	<i>Web Services</i>	850
18.8.1	<i>Microsoft's .NET Platform</i>	852
18.8.2	<i>Sun Microsystems and the Sun ONE Platform</i>	853
 <i>Part 7 Security</i>		 866
19	<i>Security</i>	869
19.1	<i>Introduction</i>	872
	<i>Operating Systems Thinking: Ethical Systems Design</i>	872
19.2	<i>Cryptography</i>	873
19.2.1	<i>Secret-Key Cryptography</i>	875
19.2.2	<i>Public-Key Cryptography</i>	877
	<i>Biographical Note: Rivest, Shamir and Adleman</i>	880
19.3	<i>Authentication</i>	880
19.3.1	<i>Basic Authentication</i>	881
19.3.2	<i>Biometrics and Smart Cards</i>	882
19.3.3	<i>Kerberos</i>	884
19.3.4	<i>Single Sign-On</i>	885
19.4	<i>Access Control</i>	886
19.4.1	<i>Access Rights and Protection Domains</i>	886
19.4.2	<i>Access Control Models and Policies</i>	887
19.4.3	<i>Access Control Mechanisms</i>	888
19.5	<i>Security Attacks</i>	892
19.5.1	<i>Cryptanalysis</i>	892
19.5.2	<i>Viruses and Worms</i>	892
19.5.3	<i>Denial-of-Service (DoS) Attacks</i>	894
19.5.4	<i>Software Exploitation</i>	896
19.5.5	<i>System Penetration</i>	896
19.6	<i>Attack Prevention and Security Solutions</i>	897

19.6.1 Firewalls.....	897
19.6.2 Intrusion Detection Systems (<i>IDSs</i>).....	899
19.6.3 Antivirus Software.....	900
19.6.4 Security Patches	902
19.6.5 Secure File Systems.....	903
<i>Mini Case Study: OpenBSD.....</i>	904
<i>Mini Case Study: Macintosh.....</i>	905
19.6.6 Orange Book Security.....	906
'9.7 Secure Communication.....	907
'9.8 Key Agreement Protocols.....	908
19.8.1 Key Management.....	908
19.8.2 Digital Signatures	910
19.9 Public-Key Infrastructure, Certificates and Certificate Authorities.....	911
19.10 Secure Communication Protocols.....	913
19.10.1 Secure Sockets layer (SSL).....	913
19.10.2 Virtual Private Networks (VPNs) and IP Security (IPSec)	914
19.10.3 Wireless Security.....	915
19.11 Steganography.....	917
19.12 Proprietary and Open-Source Security.....	918
19.13 Case Study: UNIX Systems Security.....	919
<i>Part 8 Operating System Case Studies.....</i>	936
20 Case Study: Linux.....	939
20.1 Introduction.....	942
20.2 History.....	942
20.3 Linux Overview.....	945
20.3.1 Development and Community.....	945
20.3.2 Distributions.....	946
20.3.3 User Interface.....	947
20.3.4 Stafmards.....	947
20.4 Kernel Architecture.....	948
20.4.1 Hardware Platforms.....	950
<i>Mini Case Study: User-Mode Linux (UML).....</i>	951
20.4.2 Loadable Kernel Modules.....	952

20.5 Process Management.....	952
20.5.1 Process and Thread Organization.....	953
20.5.2 Process Scheduling.....	955
20.6 Memory Management.....	961
20.6.1 Memory Organization.....	961
20.6.2 Physical Memory Allocation and Deallocation.....	966
20.6.3 Page Replacement.....	968
20.6.4 Swapping.....	970
20.7 File Systems.....	971
20.7.1 Virtual File System.....	972
20.7.2 Virtual File System Caches.....	975
20.7.3 Second Extended File System (<i>ext2fs</i>).....	977
20.7.4 Proc File System.....	982
20.8 Input/Output Management.....	983
20.8.1 Device Drivers.....	983
20.8.2 Character Device I/O.....	986
20.8.3 Block Device I/O.....	987
20.8.4 Network Device I/O.....	991
20.8.5 Unified Device Model.....	992
20.8.6 Interrupts.....	995
20.9 Kernel Synchronization.....	997
20.9.1 Spin locks.....	997
20.9.2 Reader/Writer locks.....	998
20.9.3 Seqlocks.....	998
20.9.4 Kernel Semaphores.....	999
20.10 Interprocess Communication.....	1000
20.10.1 Signals.....	1001
20.10.2 Pipes.....	1002
20.10.3 Sockets.....	1003
20.10.4 Message Queues.....	1005
20.10.5 Shared Memory.....	1005
20.10.6 System V Semaphores.....	1007
20.11 Networking.....	1008
20.11.1 Packet Processing.....	1008
20.11.2 Netfilter Framework and Hooks.....	1010

20.12 Scalability.....	1010
20.12.1 Symmetric Multiprocessing (SMP).....	1011
20.12.2 Nonuniform Memory Access (NUMA).....	1012
20.12.3 Other Scalability Features.....	1013
20.12.4 Embedded Linux.....	1014
20.13 Security	1014
20.13.1 Authentication.....	1015
20.13.2 Access Control Methods.....	1015
20.13.3 Cryptography.....	1017
21 Case Study: Windows XP.....	1033
21.1 Introduction.....	1036
21.2 History.....	1036
<i>Biographical Note: Bill Gates.....</i>	<i>1037</i>
<i>Biographical Note: David Cutler.....</i>	<i>1039</i>
<i>Mini Case Study: OS/2.....</i>	<i>1040</i>
21.3 Design Goals.....	1041
21.4 System Architecture.....	1042
21.5 System Management Mechanisms.....	1044
21.5.1 Registry.....	1044
21.5.2 Object Manager.....	1045
21.5.3 Interrupt Request Levels (IRQLs)	1046
21.5.4 Asynchronous Procedure Calls (APCs).....	1048
21.5.5 Deferred Procedure Calls (DPCs)	1049
21.5.6 System Threads.....	1051
21.6 Process and Thread Management.....	1051
21.6.1 Process and Thread Organization.....	1052
21.6.2 Thread Scheduling.....	1055
21.6.3 Thread Synchronization.....	1059
21.7 Memory Management	1064
21.7.1 Memory Organization.....	1065
21.7.2 Memory Allocation.....	1067
21.7.3 Page Replacement.....	1071
21.8 File Systems Management.....	1073
21.8.1 File System Drivers.....	1074

21.8.2 NTFS.....	1074
21.9 Input/Output Management.....	1080
21.9.1 Device Drivers.....	1081
21.9.2 Input/Output Processing.....	1084
21.9.3 Interrupt Handling.....	1089
21.9.4 File Cache Management.....	1090
21.10 Interprocess Communication.....	1091
21.10.1 Pipes.....	1091
21.10.2 Mailslots.....	1093
21.10.3 Shared Memory.....	1094
21.10.4 Local and Remote Procedure Calls.....	1094
21.10.5 Component Object Model (COM).....	1097
21.10.6 Drag-and-Drop and Compound Documents.....	1098
21.11 Networking.....	1099
21.11.1 Network Input/Output.....	1099
21.11.2 Network Driver Architecture.....	1101
21.11.3 Network Protocols.....	1103
21.11.4 Network Services.....	1105
21.11.5 .NET.....	1105
21.12 Scalability.....	1106
21.12.1 Symmetric Multiprocessing (SMP).....	1106
21.12.2 Windows XP Embedded.....	1108
21.13 Security.....	1108
21.13.1 Authentication.....	1109
21.13.2 Authorization.....	1110
21.13.3 Internet connection Firewall.....	1111
21.13.3 Other Features.....	1112
Glossary.....	1138
Index.....	1188

Illustration

Part 1 Introduction to Hardware, Software and Operating System..... 1

1 Introduction to Operating Systems

<i>1.1 Interaction between applications and the operating system.</i>	25
<i>1.2 Schematic of a virtual machine.....</i>	28
<i>1.3 Monolithic operating system kernel architecture.....</i>	34
<i>1.4 Layers of THE operating system.....</i>	36
<i>1.5 Microkernel operating system architecture.....</i>	37
<i>1.6 Client/server networked operating system model.....</i>	38

2 Hardware and Software Concepts

<i>2.1 Transistor or count plotted against time for Intel processors.</i>	58
<i>2.2 Process or components.</i>	61
<i>2.3 Memory hierarchy.....</i>	64
<i>2.4 Direct memory access (DMA).</i>	69
<i>2.5 Peripheral devices.</i>	71
<i>2.6 Bootstrapping.....</i>	77
<i>2.7 Application programming interface (API).</i>	85
<i>2.8 Compiler phases.</i>	86
<i>2.9 Object module.....</i>	87
<i>2.10 Linking process.</i>	88
<i>2.11 Symbol resolution.....</i>	89
<i>2.12 Loading.....</i>	92
<i>2.13 Compiling, linking and loading.....</i>	92

Part 2 Processes and Threads.....106

3 Process Concepts

<i>3.1 Process state transitions.....</i>	115
<i>3.2 Process table and process control blocks.....</i>	117
<i>3.3 Process creation hierarchy.....</i>	119

3.4 Process hierarchy in Linux.....	120
3.5 Process state transitions with suspend and resume.....	121
3.6 Context switch.....	122
3.7 Handling interrupts.....	126
3.8 Common interrupt types recognized in the Intel IA-32 architecture.....	128
3.9 Intel IA-32 exception classes.	128
3.10 UNIX system calls. ..	135
3.11 Code for Exercise 3.4.	139

4 Thread Concepts

4.1 Thread relationship to processes.....	149
4.2 Thread life cycle.....	153
4.3 User-level threads.....	156
4.4 Kernel-level threads.....	157
4.5 Hybrid threading model	160
4.6 Signal masking.....	164
4.7 Linux task state-transition diagram.	169
4.8 Windows XP thread state-transition diagram.	171
4.9 Java-threads being created, starting, sleeping and printing.....	174

5 Asynchronous Concurrent Execution

5.1 Buffer interface used in producer/consumer examples.	190
5.2 Producer class represents the producer thread in a producer/consumer relationship.....	191
5.3 Consumer class represents the consumer thread in a producer/consumer relationship.	192
5.4 UnsynchronizedBuffer class maintains the shared integer that is accessed by a producer thread and a consumer thread via methods set and get.....	194
5.5 SharedBuffer class enables threads to modify a shared object without synchronization.	195
5.6 Mutual exclusion implementation—version 1.....	201
5.7 Mutual exclusion implementation—version 2.	203
5.8 Mutual exclusion implementation—version 3.	205
5.9 Mutual exclusion implementation—version 4.	206
5.10 Dekker's Algorithm for mutual exclusion.	208
5.11 Peterson's Algorithm for mutual exclusion.....	212
5.12 Lamport's Bakery Algorithm.....	217
5.13 testAndSet instruction for mutual exclusion.....	223
5.14 swap instruction for mutual exclusion.....	225
5.15 Mutual exclusion with semaphores.....	228
5.16 Producer/consumer relationship implemented with semaphores.	230
5.17 Code for exercise 5.5.....	237
5.18 New mutual exclusion primitives for Exercise 5.16(a).....	238
5.19 Code for Exercise 5.16(b).....	238
5.20 Algorithm for Exercise 5.28.....	239

6 Concurrent Programming

6.1 Simple resource allocation with a monitor in pseudocode	253
6.2 Monitor pseudocode implementation of a circular buffer.....	254
6.3 Monitor pseudocode for solving the readers and writers problem.	258
6.4 SynchronizedBuffer synchronizes access to a shared integer.....	262
6.5 Threads modifying a shared object with synchronization.....	267
6.6 SynchronizedBuffer controls access to a shared array of integers.	271
6.7 CircularBufferTest instantiates producer and consumer threads.....	276

7 Deadlock and Indefinite Postponement

7.1 Traffic deadlock example.....	292
7.2 Resource deadlock example: This system is deadlocked because each process holds a resource being requested by the other process and neither process is willing to release the resource it holds.	294
7.3 Dining philosopher behavior.....	296
7.4 Implementation of method eat	297
7.5 Havender's linear ordering of resources for preventing deadlock.....	306
7.6 Safe state.	309
7.7 Unsafe state.....	310
7.8 Safe-state-to-unsafe-state transition.....	310
7.9 State description of three processes.	311
7.10 Resource allocation and request graphs.....	314
7.11 Graph reductions determining that no deadlock exists.....	315
7.12 Typical philosopher behavior for Exercise 7.9.....	323
7.13 Philosopher behavior for Exercise 7.9(a).....	323
7.14 Philosopher behavior for Exercise 7.9(b).....	324
7.15 Philosopher behavior for Exercise 7.9(c).....	324
7.16 Philosopher behavior for Exercise 7.9(d).....	325
7.17 Resource description for State A	326
7.18 Resource description for State B.	327
7.19 System state describing current loan, maximum need and current claim.....	327
7.20 System state describing total number of resources and available resources.....	327
7.21 Example of a system in an unsafe state.	328

8 Processor Scheduling

8.1 Scheduling levels.....	336
8.2 First-in-first-out scheduling.	344
8.3 Round-robin scheduling.....	345
8.4 Multilevel feedback queues.	351
8.5 Standard UNIX process scheduler. The scheduler grants the processor to users, each of whom may have many processes. (Property of AT&T Archives. Reprinted with permission of AT&T).	355

8.6 Fair share scheduler. The fair share scheduler divides system resource capacity into portions, which are then allocated by process schedulers assigned to various fair share groups. (Property of AT&T Archives. Reprinted with permission of AT&T)	356
8.7 Java thread priority scheduling.....	362

Part 3 Physical and Virtual Memory 374

9 Real Memory Organization and Management

9.1 Microsoft Windows operating system memory requirements.	381
9.2 Hierarchical memory organization.....	383
9.3 Single-user contiguous memory allocation.....	385
9.4 Overlay structure.	386
9.5 Memory protection with single-user contiguous memory allocation.	388
9.6 Processor utilization on a single-user system. [Note: In many single user jobs, I/O waits are much longer relative to processor utilization periods indicated in this diagram]......	390
9.7 Fixed-partition multiprogramming with absolute translation and loading.....	391
9.8 Memory waste under fixed-partition multiprogramming with absolute translation and loading.	392
9.9 Fixed-partition multiprogramming with relocatable translation and loading.....	392
9.10 Memory protection in contiguous-allocation multiprogramming systems.....	393
9.11 Internal fragmentation in a fixed-partition multiprogramming system	395
9.12 Initial partition assignments in variable-partition programming.	396
9.13 Memory "holes" in variable-partition multiprogramming.	397
9.14 Coalescing memory "holes" in variable-partition multiprogramming.	398
9.15 Memory compaction in variable-partition multiprogramming.....	398
9.16 First-fit, best-fit and worst-fit memory placement strategies.....	400
9.17 Multiprogramming in a swapping system in which only a single process at a time is in main memory.	402

10 Virtual Memory Organization

10.1 Evolution of memory organizations.....	416
10.2 Two-level storage	419
10.3 Pieces of address spaces exist in memory and in secondary storage.....	420
10.4 Mapping virtual addresses to real addresses.	421
10.5 Artificial contiguity.....	422
10.6 Virtual address format in a block mapping system.....	422
10.7 Virtual address translation with block mapping.	423
10.8 Virtual address format in a pure paging system.....	424
10.9 Main memory divided into page frames.....	425
10.10 Correspondence between virtual memory addresses and physical memory addresses in a pure paging system.....	426