

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 13

1.3 COMPUTER HARDWARE REVIEW 17

1.3.1 Processors 17

1.3.2 Memory 21

1.3.3 Disks 24

1.3.4 Tapes 25

1.3.5 I/O Devices 25

1.3.6 Buses 28

1.3.7 Booting the Computer 31

1.4 THE OPERATING SYSTEM ZOO	31
1.4.1 Mainframe Operating Systems	32
1.4.2 Server Operating Systems	32
1.4.3 Multiprocessor Operating Systems	32
1.4.4 Personal Computer Operating Systems	33
1.4.5 Handheld Computer Operating Systems	33
1.4.6 Embedded Operating Systems.	33
1.4.7 Sensor Node Operating Systems	34
1.4.8 Real-Time Operating Systems	34
1.4.9 Smart Card Operating Systems	35
1.5 OPERATING SYSTEM CONCEPTS	35
1.5.1 Processes	36
1.5.2 Address Spaces	38
1.5.3 Files	38
1.5.4 Input/Output	41
1.5.5 Protection	42
1.5.6 The Shell	42
1.5.7 Ontogeny Recapitulates Phylogeny	44
1.6 SYSTEM CALLS	47
1.6.1 System Calls for Process Management	50
1.6.2 System Calls for File Management	54
1.6.3 System Calls for Directory Management	55
1.6.4 Miscellaneous System Calls	56
1.6.5 The Windows Win32 API	57
1.7 OPERATING SYSTEM STRUCTURE	60
1.7.1 Monolithic Systems	60
1.7.2 Layered Systems	61
1.7.3 Microkernels	62
1.7.4 Client-Server Model	65
1.7.5 Virtual Machines	65
1.7.6 Exokernels	69
1.8 THE WORLD ACCORDING TO C	70
1.8.1 The C Language	70
1.8.2 Header Files	71
1.8.3 Large Programming Projects	72
1.8.4 The Model of Run Time	73
1.9 RESEARCH ON OPERATING SYSTEMS	74

- 1.10 OUTLINE OF THE REST OF THIS BOOK 75
- 1.11 METRIC UNITS 76
- 1.12 SUMMARY 77

2 PROCESSES AND THREADS

81

- 2.1 PROCESSES 81
 - 2.1.1 The Process Model 82
 - 2.1.2 Process Creation 84
 - 2.1.3 Process Termination 86
 - 2.1.4 Process Hierarchies 87
 - 2.1.5 Process States 88
 - 2.1.6 Implementation of Processes 89
 - 2.1.7 Modeling Multiprogramming 91
- 2.2 THREADS 93
 - 2.2.1 Thread Usage 93
 - 2.2.2 The Classical Thread Model 98
 - 2.2.3 POSIX Threads 102
 - 2.2.4 Implementing Threads in User Space 104
 - 2.2.5 Implementing Threads in the Kernel 107
 - 2.2.6 Hybrid Implementations 108
 - 2.2.7 Scheduler Activations 109
 - 2.2.8 Pop-Up Threads 110
 - 2.2.9 Making Single-Threaded Code Multithreaded 112
- 2.3 INTERPROCESS COMMUNICATION 115
 - 2.3.1 Race Conditions 115
 - 2.3.2 Critical Regions 117
 - 2.3.3 Mutual Exclusion with Busy Waiting 118
 - 2.3.4 Sleep and Wakeup 123
 - 2.3.5 Semaphores 126
 - 2.3.6 Mutexes 128
 - 2.3.7 Monitors 132
 - 2.3.8 Message Passing 138
 - 2.3.9 Barriers 142

- 2.4 SCHEDULING 143
 - 2.4.1 Introduction to Scheduling 143
 - 2.4.2 Scheduling in Batch Systems 150
 - 2.4.3 Scheduling in Interactive Systems 152
 - 2.4.4 Scheduling in Real-Time Systems 158
 - 2.4.5 Policy versus Mechanism 159
 - 2.4.6 Thread Scheduling 160
- 2.5 CLASSICAL IPC PROBLEMS 161
 - 2.5.1 The Dining Philosophers Problem 162
 - 2.5.2 The Readers and Writers Problem 165
- 2.6 RESEARCH ON PROCESSES AND THREADS 166
- 2.7 SUMMARY 167

3 MEMORY MANAGEMENT

173

- 3.1 NO MEMORY ABSTRACTION 174
- 3.2 A MEMORY ABSTRACTION: ADDRESS SPACES 177
 - 3.2.1 The Notion of an Address Space 178
 - 3.2.2 Swapping 179
 - 3.2.3 Managing Free Memory 182
- 3.3 VIRTUAL MEMORY 186
 - 3.3.1 Paging 187
 - 3.3.2 Page Tables 191
 - 3.3.3 Speeding Up Paging 192
 - 3.3.4 Page Tables for Large Memories 196
- 3.4 PAGE REPLACEMENT ALGORITHMS 199
 - 3.4.1 The Optimal Page Replacement Algorithm 200
 - 3.4.2 The Not Recently Used Page Replacement Algorithm 201
 - 3.4.3 The First-In, First-Out (FIFO) Page Replacement Algorithm 202
 - 3.4.4 The Second-Chance Page Replacement Algorithm 202
 - 3.4.5 The Clock Page Replacement Algorithm 203
 - 3.4.6 The Least Recently Used (LRU) Page Replacement Algorithm 204
 - 3.4.7 Simulating LRU in Software 205
 - 3.4.8 The Working Set Page Replacement Algorithm 207

3.4.9 The WSClock Page Replacement Algorithm	211
3.4.10 Summary of Page Replacement Algorithms	213
3.5 DESIGN ISSUES FOR PAGING SYSTEMS	214
3.5.1 Local versus Global Allocation Policies	214
3.5.2 Load Control	216
3.5.3 Page Size	217
3.5.4 Separate Instruction and Data Spaces	219
3.5.5 Shared Pages	219
3.5.6 Shared Libraries	221
3.5.7 Mapped Files	223
3.5.8 Cleaning Policy	224
3.5.9 Virtual Memory Interface	224
3.6 IMPLEMENTATION ISSUES	225
3.6.1 Operating System Involvement with Paging	225
3.6.2 Page Fault Handling	226
3.6.3 Instruction Backup	227
3.6.4 Locking Pages in Memory	228
3.6.5 Backing Store	229
3.6.6 Separation of Policy and Mechanism	231
3.7 SEGMENTATION	232
3.7.1 Implementation of Pure Segmentation	235
3.7.2 Segmentation with Paging: MULTICS	236
3.7.3 Segmentation with Paging: The Intel Pentium	240
3.8 RESEARCH ON MEMORY MANAGEMENT	245
3.9 SUMMARY	246

4 FILE SYSTEMS

253

4.1 FILES	255
4.1.1 File Naming	255
4.1.2 File Structure	257
4.1.3 File Types	258
4.1.4 File Access	260
4.1.5 File Attributes	261

- 4.1.6 File Operations 262
- 4.1.7 An Example Program Using File System Calls 263
- 4.2 DIRECTORIES 266
 - 4.2.1 Single-Level Directory Systems 266
 - 4.2.2 Hierarchical Directory Systems 266
 - 4.2.3 Path Names 267
 - 4.2.4 Directory Operations 270
- 4.3 FILE SYSTEM IMPLEMENTATION 271
 - 4.3.1 File System Layout 271
 - 4.3.2 Implementing Files 272
 - 4.3.3 Implementing Directories 278
 - 4.3.4 Shared Files 281
 - 4.3.5 Log-Structured File Systems 283
 - 4.3.6 Journaling File Systems 285
 - 4.3.7 Virtual File Systems 286
- 4.4 FILE SYSTEM MANAGEMENT AND OPTIMIZATION 290
 - 4.4.1 Disk Space Management 290
 - 4.4.2 File System Backups 296
 - 4.4.3 File System Consistency 302
 - 4.4.4 File System Performance 305
 - 4.4.5 Defragmenting Disks 309
- 4.5 EXAMPLE FILE SYSTEMS 310
 - 4.5.1 CD-ROM File Systems 310
 - 4.5.2 The MS-DOS File System 316
 - 4.5.3 The UNIX V7 File System 319
- 4.6 RESEARCH ON FILE SYSTEMS 322
- 4.7 SUMMARY 322

5 INPUT/OUTPUT

327

- 5.1 PRINCIPLES OF I/O HARDWARE 327
 - 5.1.1 I/O Devices 328
 - 5.1.2 Device Controllers 329

5.1.3 Memory-Mapped I/O	330
5.1.4 Direct Memory Access (DMA)	334
5.1.5 Interrupts Revisited	337
5.2 PRINCIPLES OF I/O SOFTWARE	341
5.2.1 Goals of the I/O Software	341
5.2.2 Programmed I/O	342
5.2.3 Interrupt-Driven I/O	344
5.2.4 I/O Using DMA	345
5.3 I/O SOFTWARE LAYERS	346
5.3.1 Interrupt Handlers	346
5.3.2 Device Drivers	347
5.3.3 Device-Independent I/O Software	351
5.3.4 User-Space I/O Software	357
5.4 DISKS	358
5.4.1 Disk Hardware	359
5.4.2 Disk Formatting	374
5.4.3 Disk Arm Scheduling Algorithms	377
5.4.4 Error Handling	380
5.4.5 Stable Storage	383
5.5 CLOCKS	386
5.5.1 Clock Hardware	386
5.5.2 Clock Software	388
5.5.3 Soft Timers	391
5.6 USER INTERFACES: KEYBOARD, MOUSE, MONITOR	392
5.6.1 Input Software	392
5.6.2 Output Software	397
5.7 THIN CLIENTS	413
5.8 POWER MANAGEMENT	415
5.8.1 Hardware Issues	416
5.8.2 Operating System Issues	417
5.8.3 Application Program Issues	422
5.9 RESEARCH ON INPUT/OUTPUT	423
5.10 SUMMARY	424

6 DEADLOCKS

431

6.1 RESOURCES 432

6.1.1 Preemptable and Nonpreemptable Resources 432

6.1.2 Resource Acquisition 433

6.2 INTRODUCTION TO DEADLOCKS 435

6.2.1 Conditions for Resource Deadlocks 436

6.2.2 Deadlock Modeling 436

6.3 THE OSTRICH ALGORITHM 439

6.4 DEADLOCK DETECTION AND RECOVERY 440

6.4.1 Deadlock Detection with One Resource of Each Type 440

6.4.2 Deadlock Detection with Multiple Resources of Each Type 442

6.4.3 Recovery from Deadlock 445

6.5 DEADLOCK AVOIDANCE 446

6.5.1 Resource Trajectories 447

6.5.2 Safe and Unsafe States 448

6.5.3 The Banker's Algorithm for a Single Resource 449

6.5.4 The Banker's Algorithm for Multiple Resources 450

6.6 DEADLOCK PREVENTION 452

6.6.1 Attacking the Mutual Exclusion Condition 452

6.6.2 Attacking the Hold and Wait Condition 453

6.6.3 Attacking the No Preemption Condition 453

6.6.4 Attacking the Circular Wait Condition 454

6.7 OTHER ISSUES 455

6.7.1 Two-Phase Locking 455

6.7.2 Communication Deadlocks 456

6.7.3 Livelock 457

6.7.4 Starvation 459

6.8 RESEARCH ON DEADLOCKS 459

6.9 SUMMARY 460

7 MULTIMEDIA OPERATING SYSTEMS

465

- 7.1 INTRODUCTION TO MULTIMEDIA 466
- 7.2 MULTIMEDIA FILES 470
 - 7.2.1 Video Encoding 471
 - 7.2.2 Audio Encoding 474
- 7.3 VIDEO COMPRESSION 476
 - 7.3.1 The JPEG Standard 476
 - 7.3.2 The MPEG Standard 479
- 7.4 AUDIO COMPRESSION 482
- 7.5 MULTIMEDIA PROCESS SCHEDULING 485
 - 7.5.1 Scheduling Homogeneous Processes 486
 - 7.5.2 General Real-Time Scheduling 486
 - 7.5.3 Rate Monotonic Scheduling 488
 - 7.5.4 Earliest Deadline First Scheduling 489
- 7.6 MULTIMEDIA FILE SYSTEM PARADIGMS 491
 - 7.6.1 VCR Control Functions 492
 - 7.6.2 Near Video on Demand 494
 - 7.6.3 Near Video on Demand with VCR Functions 496
- 7.7 FILE PLACEMENT 497
 - 7.7.1 Placing a File on a Single Disk 498
 - 7.7.2 Two Alternative File Organization Strategies 499
 - 7.7.3 Placing Files for Near Video on Demand 502
 - 7.7.4 Placing Multiple Files on a Single Disk 504
 - 7.7.5 Placing Files on Multiple Disks 506
- 7.8 CACHING 508
 - 7.8.1 Block Caching 509
 - 7.8.2 File Caching 510
- 7.9 DISK SCHEDULING FOR MULTIMEDIA 511
 - 7.9.1 Static Disk Scheduling 511
 - 7.9.2 Dynamic Disk Scheduling 513
- 7.10 RESEARCH ON MULTIMEDIA 514
- 7.11 SUMMARY 515

8 MULTIPLE PROCESSOR SYSTEMS**521**

- 8.1 MULTIPROCESSORS 524
 - 8.1.1 Multiprocessor Hardware 524
 - 8.1.2 Multiprocessor Operating System Types 532
 - 8.1.3 Multiprocessor Synchronization 536
 - 8.1.4 Multiprocessor Scheduling 540
- 8.2 MULTICOMPUTERS 546
 - 8.2.1 Multicomputer Hardware 547
 - 8.2.2 Low-Level Communication Software 551
 - 8.2.3 User-Level Communication Software 553
 - 8.2.4 Remote Procedure Call 556
 - 8.2.5 Distributed Shared Memory 558
 - 8.2.6 Multicomputer Scheduling 563
 - 8.2.7 Load Balancing 563
- 8.3 VIRTUALIZATION 566
 - 8.3.1 Requirements for Virtualization 568
 - 8.3.2 Type 1 Hypervisors 569
 - 8.3.3 Type 2 Hypervisors 570
 - 8.3.4 Paravirtualization 572
 - 8.3.5 Memory Virtualization 574
 - 8.3.6 I/O Virtualization 576
 - 8.3.7 Virtual Appliances 577
 - 8.3.8 Virtual Machines on Multicore CPUs 577
 - 8.3.9 Licensing Issues 578
- 8.4 DISTRIBUTED SYSTEMS 578
 - 8.4.1 Network Hardware 581
 - 8.4.2 Network Services and Protocols 584
 - 8.4.3 Document-Based Middleware 588
 - 8.4.4 File-System-Based Middleware 589
 - 8.4.5 Object-Based Middleware 594
 - 8.4.6 Coordination-Based Middleware 596
 - 8.4.7 Grids 601
- 8.5 RESEARCH ON MULTIPLE PROCESSOR SYSTEMS 602
- 8.6 SUMMARY 603

9 SECURITY

609

- 9.1 THE SECURITY ENVIRONMENT 611
 - 9.1.1 Threats 611
 - 9.1.2 Intruders 613
 - 9.1.3 Accidental Data Loss 614

- 9.2 BASICS OF CRYPTOGRAPHY 614
 - 9.2.1 Secret-Key Cryptography 615
 - 9.2.2 Public-Key Cryptography 616
 - 9.2.3 One-Way Functions 617
 - 9.2.4 Digital Signatures 617
 - 9.2.5 Trusted Platform Module 619

- 9.3 PROTECTION MECHANISMS 620
 - 9.3.1 Protection Domains 620
 - 9.3.2 Access Control Lists 622
 - 9.3.3 Capabilities 625
 - 9.3.4 Trusted Systems 628
 - 9.3.5 Trusted Computing Base 629
 - 9.3.6 Formal Models of Secure Systems 630
 - 9.3.7 Multilevel Security 632
 - 9.3.8 Covert Channels 635

- 9.4 AUTHENTICATION 639
 - 9.4.1 Authentication Using Passwords 640
 - 9.4.2 Authentication Using a Physical Object 649
 - 9.4.3 Authentication Using Biometrics 651

- 9.5 INSIDER ATTACKS 654
 - 9.5.1 Logic Bombs 654
 - 9.5.2 Trap Doors 655
 - 9.5.3 Login Spoofing 656

- 9.6 EXPLOITING CODE BUGS 657
 - 9.6.1 Buffer Overflow Attacks 658
 - 9.6.2 Format String Attacks 660
 - 9.6.3 Return to libc Attacks 662
 - 9.6.4 Integer Overflow Attacks 663
 - 9.6.5 Code Injection Attacks 664
 - 9.6.6 Privilege Escalation Attacks 665

9.7 MALWARE	665
9.7.1 Trojan Horses	668
9.7.2 Viruses	670
9.7.3 Worms	680
9.7.4 Spyware	682
9.7.5 Rootkits	686
9.8 DEFENSES	690
9.8.1 Firewalls	691
9.8.2 Antivirus and Anti-Antivirus Techniques	693
9.8.3 Code Signing	699
9.8.4 Jailing	700
9.8.5 Model-Based Intrusion Detection	701
9.8.6 Encapsulating Mobile Code	703
9.8.7 Java Security	707
9.9 RESEARCH ON SECURITY	709
9.10 SUMMARY	710

10 CASE STUDY 1: LINUX

715

10.1 HISTORY OF UNIX AND LINUX	716
10.1.1 UNICS	716
10.1.2 PDP-11 UNIX	717
10.1.3 Portable UNIX	718
10.1.4 Berkeley UNIX	719
10.1.5 Standard UNIX	720
10.1.6 MINIX	721
10.1.7 Linux	722
10.2 OVERVIEW OF LINUX	724
10.2.1 Linux Goals	725
10.2.2 Interfaces to Linux	726
10.2.3 The Shell	727
10.2.4 Linux Utility Programs	730
10.2.5 Kernel Structure	732

10.3 PROCESSES IN LINUX	735
10.3.1 Fundamental Concepts	735
10.3.2 Process Management System Calls in Linux	737
10.3.3 Implementation of Processes and Threads in Linux	741
10.3.4 Scheduling in Linux	748
10.3.5 Booting Linux	751
10.4 MEMORY MANAGEMENT IN LINUX	754
10.4.1 Fundamental Concepts	754
10.4.2 Memory Management System Calls in Linux	757
10.4.3 Implementation of Memory Management in Linux	758
10.4.4 Paging in Linux	764
10.5 INPUT/OUTPUT IN LINUX	767
10.5.1 Fundamental Concepts	768
10.5.2 Networking	769
10.5.3 Input/Output System Calls in Linux	771
10.5.4 Implementation of Input/Output in Linux	771
10.5.5 Modules in Linux	775
10.6 THE LINUX FILE SYSTEM	775
10.6.1 Fundamental Concepts	776
10.6.2 File System Calls in Linux	781
10.6.3 Implementation of the Linux File System	784
10.6.4 NFS: The Network File System	792
10.7 SECURITY IN LINUX	799
10.7.1 Fundamental Concepts	799
10.7.2 Security System Calls in Linux	801
10.7.3 Implementation of Security in Linux	802
10.8 SUMMARY	802

11 CASE STUDY 2: WINDOWS VISTA

809

11.1 HISTORY OF WINDOWS VISTA	809
11.1.1 1980s: MS-DOS	810
11.1.2 1990s: MS-DOS-based Windows	811
11.1.3 2000s: NT-based Windows	811
11.1.4 Windows Vista	814

11.2 PROGRAMMING WINDOWS VISTA	815
11.2.1 The Native NT Application Programming Interface	818
11.2.2 The Win32 Application Programming Interface	821
11.2.3 The Windows Registry	825
11.3 SYSTEM STRUCTURE	827
11.3.1 Operating System Structure	828
11.3.2 Booting Windows Vista	843
11.3.3 Implementation of the Object Manager	844
11.3.4 Subsystems, DLLs, and User-Mode Services	854
11.4 PROCESSES AND THREADS IN WINDOWS VISTA	857
11.4.1 Fundamental Concepts	857
11.4.2 Job, Process, Thread, and Fiber Management API Calls	862
11.4.3 Implementation of Processes and Threads	867
11.5 MEMORY MANAGEMENT	875
11.5.1 Fundamental Concepts	875
11.5.2 Memory Management System Calls	880
11.5.3 Implementation of Memory Management	881
11.6 CACHING IN WINDOWS VISTA	890
11.7 INPUT/OUTPUT IN WINDOWS VISTA	892
11.7.1 Fundamental Concepts	893
11.7.2 Input/Output API Calls	894
11.7.3 Implementation of I/O	897
11.8 THE WINDOWS NT FILE SYSTEM	902
11.8.1 Fundamental Concepts	903
11.8.2 Implementation of the NT File System	904
11.9 SECURITY IN WINDOWS VISTA	914
11.9.1 Fundamental Concepts	915
11.9.2 Security API Calls	917
11.9.3 Implementation of Security	918
11.10 SUMMARY	920

12 CASE STUDY 3: SYMBIAN OS

925

- 12.1 THE HISTORY OF SYMBIAN OS 926
 - 12.1.1 Symbian OS Roots: Psion and EPOC 926
 - 12.1.2 Symbian OS Version 6 927
 - 12.1.3 Symbian OS Version 7 928
 - 12.1.4 Symbian OS Today 928

- 12.2 AN OVERVIEW OF SYMBIAN OS 928
 - 12.2.1 Object Orientation 929
 - 12.2.2 Microkernel Design 930
 - 12.2.3 The Symbian OS Nanokernel 931
 - 12.2.4 Client/Server Resource Access 931
 - 12.2.5 Features of a Larger Operating System 932
 - 12.2.6 Communication and Multimedia 933

- 12.3 PROCESSES AND THREADS IN SYMBIAN OS 933
 - 12.3.1 Threads and Nanothreads 934
 - 12.3.2 Processes 935
 - 12.3.3 Active Objects 935
 - 12.3.4 Interprocess Communication 936

- 12.4 MEMORY MANAGEMENT 937
 - 12.4.1 Systems with No Virtual Memory 937
 - 12.4.2 How Symbian OS Addresses Memory 939

- 12.5 INPUT AND OUTPUT 941
 - 12.5.1 Device Drivers 941
 - 12.5.2 Kernel Extensions 942
 - 12.5.3 Direct Memory Access 942
 - 12.5.4 Special Case: Storage Media 943
 - 12.5.5 Blocking I/O 943
 - 12.5.6 Removable Media 944

- 12.6 STORAGE SYSTEMS 944
 - 12.6.1 File Systems for Mobile Devices 944
 - 12.6.2 Symbian OS File Systems 945
 - 12.6.3 File System Security and Protection 945

- 12.7 SECURITY IN SYMBIAN OS 946

12.8 COMMUNICATION IN SYMBIAN OS 949

12.8.1 Basic Infrastructure 949

12.8.2 A Closer Look at the Infrastructure 950

12.9 SUMMARY 953

13 OPERATING SYSTEM DESIGN**955****13.1 THE NATURE OF THE DESIGN PROBLEM 956**

13.1.1 Goals 956

13.1.2 Why Is It Hard to Design an Operating System? 957

13.2 INTERFACE DESIGN 959

13.2.1 Guiding Principles 959

13.2.2 Paradigms 961

13.2.3 The System Call Interface 964

13.3 IMPLEMENTATION 967

13.3.1 System Structure 967

13.3.2 Mechanism versus Policy 971

13.3.3 Orthogonality 972

13.3.4 Naming 973

13.3.5 Binding Time 974

13.3.6 Static versus Dynamic Structures 975

13.3.7 Top-Down versus Bottom-Up Implementation 976

13.3.8 Useful Techniques 977

13.4 PERFORMANCE 983

13.4.1 Why Are Operating Systems Slow? 983

13.4.2 What Should Be Optimized? 984

13.4.3 Space-Time Trade-offs 984

13.4.4 Caching 987

13.4.5 Hints 988

13.4.6 Exploiting Locality 989

13.4.7 Optimize the Common Case 989

13.5 PROJECT MANAGEMENT 990

13.5.1 The Mythical Man Month 990

13.5.2 Team Structure 991

13.5.3 The Role of Experience	993
13.5.4 No Silver Bullet	994
13.6 TRENDS IN OPERATING SYSTEM DESIGN	994
13.6.1 Virtualization	995
13.6.2 Multicore Chips	995
13.6.3 Large Address Space Operating Systems	996
13.6.4 Networking	996
13.6.5 Parallel and Distributed Systems	997
13.6.6 Multimedia	997
13.6.7 Battery-Powered Computers	998
13.6.8 Embedded Systems	998
13.6.9 Sensor Nodes	999
13.7 SUMMARY	999

14 READING LIST AND BIBLIOGRAPHY **1003**

14.1 SUGGESTIONS FOR FURTHER READING	1003
14.1.1 Introduction and General Works	1004
14.1.2 Processes and Threads	1004
14.1.3 Memory Management	1005
14.1.4 Input/Output	1005
14.1.5 File Systems	1006
14.1.6 Deadlocks	1006
14.1.7 Multimedia Operating Systems	1006
14.1.8 Multiple Processor Systems	1007
14.1.9 Security	1008
14.1.10 Linux	1010
14.1.11 Windows Vista	1010
14.1.12 The Symbian OS	1011
14.1.13 Design Principles	1011
14.2 ALPHABETICAL BIBLIOGRAPHY	1012