

Contents

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

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

2.5	Caching and Buffering	78
	<i>Operating Systems Thinking: Heuristics</i>	79
2.6	Software Overview	80
2.6.1	Machine language and Assembly language	80
2.6.2	Interpreters and Compilers	81
2.6.3	High-level languages	82
2.6.4	Structured Programming	83
2.6.5	Object-Oriented Programming	83
2.7	Application Programming Interfaces (APIs)	85
2.8	Compiling, Linking and Loading	86
2.8.1	Compiling	86
2.8.2	Linking	87
	<i>Mini Case Study: Mach</i>	90
2.8.3	Loading	91
2.9	Firmware	93
2.10	Middleware	93
Part 2 Processes and Threads		106
3	Process Concepts	109
3.1	Introduction	111
	<i>Operating Systems Thinking: Customers Ultimately Want Applications</i>	111
3.1.1	Definition of Process	111
	<i>Mini Case Study: CTSS and Multics</i>	112
	<i>Biographical Note: Fernando J. Corbato</i>	113
3.2	Process States: Life Cycle of a Process	113
3.3	Process Management	114
3.3.1	Process States and State Transitions	115
3.3.2	Process Control Blocks (PCBs)/Process Descriptors	116
	<i>Operating Systems Thinking: Data Structures in Operating Systems</i>	118
3.3.3	Process Operations	118
3.3.4	Suspend and Resume	120
3.3.5	Context Switching	122
3.4	Interrupts	123
	<i>Operating Systems Thinking: Asynchronism vs. Synchronism</i>	124

3.4.1	Interrupt Processing	125
3.4.2	Interrupt Classes	127
3.5	Interprocess Communication	129
3.5.1	Signals	130
3.5.2	Message Passing	130
3.6	Case Study: UNIX Processes	132
	Mini Case Study: UNIX Systems	133
4	Thread Concepts	145
4.1	Introduction	147
	Operating Systems Thinking: Concurrency	147
4.2	Definition of Thread	148
4.3	Motivation for Threads	150
	Operating Systems Thinking: Parallelism	151
4.4	Thread States: Life Cycle of a Thread	152
4.5	Thread Operations	154
4.6	Threading Models	155
4.6.1	User-Level Threads	155
4.6.2	Kernel-Level Threads	157
4.6.3	Combining User- and Kernel-Level Threads	159
	Operating Systems Thinking: Standards Conformance	159
4.7	Thread Implementation Considerations	161
4.7.1	Thread Signal Delivery	161
	Anecdote: Engineering	163
4.7.2	Thread Termination	165
4.8	POSIX and Pthreads	165
	Anecdote: Standards and Conformance: Plug-to-Plug Compatibility	166
4.9	Linux Threads	167
	Operating Systems Thinking: Scalability	168
4.10	Windows XP Threads	169
4.11	Java Multithreading Case Study, Part I: Introduction to Java Threads	172
5	Asynchronous Concurrent Execution	185
5.1	Introduction	187

5.2	Mutual Exclusion	187
5.2.1	Java Multithreading Case Study, Part II: A Producer/Consumer Relationship in Java	188
5.2.2	Critical Sections	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	Java Multithreading Case Study, Part III: Producer/Consumer Relationship in Java . .	261
6.5	Java Multithreading Case Study, Part IV: Circular Buffer in Java	270
7	Deadlock and Indefinite Postponement	289
7.1	Introduction	291
7.2	Examples of Deadlock	291
7.2.1	Traffic Deadlock	292
7.2.2	Simple Resource Deadlock	293
	Anecdote: One-lane Bridge	293
7.2.3	Deadlock in Spooling Systems	294
	Operating Systems Thinking: Waiting, Deadlock and Indefinite Postponement	295
7.2.4	Example: Dining Philosophers	296
7.3	Related Problem: Indefinite Postponement	298
7.4	Resource Concepts	298
7.5	Four Necessary Conditions for Deadlock	299
7.6	Deadlock Solutions	300
7.7	Deadlock Prevention	301
7.7.1	Denying the "Wait-For" Condition	302
	Anecdote: No Nuts, Bolts or Screws Allowed	302
7.7.2	Denying the "No-Preemption" Condition	303
7.7.3	Denying the "Circular-Wait" Condition	304
7.8	Deadlock Avoidance with Dijkstra's Banker's Algorithm	306
	Anecdote: Acronyms	307
7.8.1	Example of a Safe State	309
7.8.2	Example of an Unsafe State	309
7.8.3	Example of Safe-State-to-Unsafe-State Transition	310
7.8.4	Banker's Algorithm Resource Allocation	311
7.8.5	Weaknesses in the Banker's Algorithm	312
7.9	Deadlock Detection	312
7.9.1	Resource-Allocation Graphs	313
7.9.2	Reduction of Resource-Allocation Graphs	314
7.10	Deadlock Recovery	316
7.11	Deadlock Strategies in Current and Future Systems	318

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

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

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

Part 4 Secondary Storage, Files and Databases	522
12 Disk Performance Optimization	525
12.1 Introduction	528
12.2 Evolution of Secondary Storage	528
12.3 Characteristics of Moving-Head Disk Storage	529
12.4 Why Disk Scheduling Is Necessary	531
12.5 Disk Scheduling Strategies	533
12.5.1 First-Come-First-Served (FCFS) Disk Scheduling	535
12.5.2 Shortest-Seek-Time-First (SSTF) Disk Scheduling	536
<i>Anecdote: Every Problem Has a Solution and Every Solution Has a Problem</i>	536
12.5.3 SCAN Disk Scheduling	537
12.5.4 C-SCAN Disk Scheduling	538
12.5.5 FSCAN and N-Step SCAN Disk Scheduling	539
12.5.6 LOOK and C-LOOK Disk Scheduling	541
12.6 Rotational Optimization	542
12.6.1 SRTF Scheduling	542
12.6.2 SPTF and SATF Scheduling	543
12.7 System Considerations	546
<i>Operating Systems Thinking: Saturation and Bottlenecks</i>	546
12.8 Caching and Buffering	548
12.9 Other Disk Performance Techniques	549
<i>Operating Systems Thinking: Compression and Decompression</i>	550
<i>Operating Systems Thinking: Redundancy</i>	551
12.10 Redundant Arrays of Independent Disks (RAID)	552
12.10.1 RAID Overview	552
<i>Operating Systems Thinking: Fault Tolerance</i>	554
<i>Operating Systems Thinking: Mission-Critical Systems</i>	554
12.10.2 Level 0 (Striping)	555
12.10.3 Level 1 (Mirroring)	556
12.10.4 Level 2 (Bit-Level Hamming ECC Parity)	558
12.10.5 Level 3 (Bit-Level XOR ECC Parity)	560
12.10.6 Level 4 (Block-Level XOR ECC Parity)	562
12.10.7 Level 5 (Block-Level Distributed XOR ECC Parity)	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
	Operating Systems Thinking: Encryption and Decryption	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
	Mini Case Study: MS-DOS	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 Matrix	610
	Operating Systems Thinking: Security	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
	Operating Systems Thinking: Backup and Recovery	614
13.10.2	Data Integrity and Log-Structured File Systems	615
	Operating Systems Thinking: Murphy's Law and Robust Systems	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

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

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

Part 6 Networking and Distributed Computing

746

16	Introduction to Networking	749
16.1	Introduction	751

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

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

18.7	Java Distributed Computing	843
18.7.1	Java Servlets and JavaServer Pages (JSP)	843
18.7.2	Jini	845
18.7.3	JavaSpaces	847
18.7.4	Java Management Extensions (JMX)	849
18.8	Web Services	850
18.8.1	Microsoft's .NET Platform	852
18.8.2	Sun Microsystems and the Sun ONE Platform	853

Part 7 Security

19	Security	869
19.1	Introduction	872
	<i>Operating Systems Thinking: Ethical Systems Design</i>	872
19.2	Cryptography	873
19.2.1	Secret-Key Cryptography	875
19.2.2	Public-Key Cryptography	877
	<i>Biographical Note: Rivest, Shamir and Adleman</i>	880
19.3	Authentication	880
19.3.1	Basic Authentication	881
19.3.2	Biometrics and Smart Cards	882
19.3.3	Kerberos	884
19.3.4	Single Sign-On	885
19.4	Access Control	886
19.4.1	Access Rights and Protection Domains	886
19.4.2	Access Control Models and Policies	887
19.4.3	Access Control Mechanisms	888
19.5	Security Attacks	892
19.5.1	Cryptanalysis	892
19.5.2	Viruses and Worms	892
19.5.3	Denial-of-Service (DoS) Attacks	894
19.5.4	Software Exploitation	896
19.5.5	System Penetration	896
19.6	Attack Prevention and Security Solutions	897

19.6.1	Firewalls	897
19.6.2	Intrusion Detection Systems (IDSs)	899
19.6.3	Antivirus Software	900
19.6.4	Security Patches	902
19.6.5	Secure File Systems	903
	Mini Case Study: OpenBSD	904
	Mini Case Study: Macintosh	905
19.6.6	Orange Book Security	906
19.7	Secure Communication	907
19.8	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

Part 8 Operating System Case Studies 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	Standards	947
20.4	Kernel Architecture	948
20.4.1	Hardware Platforms	950
	Mini Case Study: User-Mode Linux (UML)	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 (ext2fs)	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
	Biographical Note: Bill Gates	1037
	Biographical Note: David Cutler	1039
	Mini Case Study: OS/2	1040
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
21.13.4	Other Features	1112
Glossary	1138
Index	1188