
CONTENTS

PREFACE

xix

PART 1: TRADITIONAL OPERATING SYSTEMS

1 INTRODUCTION

1

- 1.1 WHAT IS AN OPERATING SYSTEM? 3
 - 1.1.1 The Operating System as an Extended Machine 3
 - 1.1.2 The Operating System as a Resource Manager 4
- 1.2 HISTORY OF OPERATING SYSTEMS 5
 - 1.2.1 The First Generation 5
 - 1.2.2 The Second Generation 6
 - 1.2.3 The Third Generation 8
 - 1.2.4 The Fourth Generation 11
- 1.3 OPERATING SYSTEM CONCEPTS 12
 - 1.3.1 Processes 12
 - 1.3.2 Files 14
 - 1.3.3 System Calls 16
 - 1.3.4 The Shell 17
- 1.4 OPERATING SYSTEM STRUCTURE 18
 - 1.4.1 Monolithic Systems 19
 - 1.4.2 Layered Systems 20
 - 1.4.3 Virtual Machines 21
 - 1.4.4 Client-Server Model 22
- 1.5 OUTLINE OF THE REST OF THIS BOOK 24
- 1.6 SUMMARY 25

vii

2 PROCESSES

27

- 2.1 INTRODUCTION TO PROCESSES 27
 - 2.1.1 The Process Model 28
 - 2.1.2 Implementation of Processes 31
- 2.2 INTERPROCESS COMMUNICATION 33
 - 2.2.1 Race Conditions 33
 - 2.2.2 Critical Sections 34
 - 2.2.3 Mutual Exclusion with Busy Waiting 35
 - 2.2.4 Sleep and Wakeup 39
 - 2.2.5 Semaphores 41
 - 2.2.6 Event Counters 43
 - 2.2.7 Monitors 45
 - 2.2.8 Message Passing 49
 - 2.2.9 Equivalence of Primitives 51
- 2.3 CLASSICAL IPC PROBLEMS 56
 - 2.3.1 The Dining Philosophers Problem 56
 - 2.3.2 The Readers and Writers Problem 58
 - 2.3.3 The Sleeping Barber Problem 58
- 2.4 PROCESS SCHEDULING 61
 - 2.4.1 Round Robin Scheduling 64
 - 2.4.2 Priority Scheduling 65
 - 2.4.3 Multiple Queues 65
 - 2.4.4 Shortest Job First 67
 - 2.4.5 Guaranteed Scheduling 68
 - 2.4.6 Policy versus Mechanism 68
 - 2.4.7 Two-level Scheduling 69
- 2.5 SUMMARY 70

3 MEMORY MANAGEMENT

74

- 3.1 MEMORY MANAGEMENT WITHOUT SWAPPING OR PAGING 74
 - 3.1.1 Monoprogramming without Swapping or Paging 75
 - 3.1.2 Multiprogramming and Memory Usage 76
 - 3.1.3 Multiprogramming with Fixed Partitions 78
- 3.2 SWAPPING 81
 - 3.2.1 Multiprogramming with Variable Partitions 81
 - 3.2.2 Memory Management with Bit Maps 83
 - 3.2.3 Memory Management with Linked Lists 84
 - 3.2.4 Memory Management with the Buddy System 86
 - 3.2.5 Allocation of Swap Space 87
 - 3.2.6 Analysis of Swapping Systems 88

- 3.3 VIRTUAL MEMORY 89
 - 3.3.1 Paging 89
 - 3.3.2 Page Tables 92
 - 3.3.3 Examples of Paging Hardware 96
 - 3.3.4 Associative Memory 101
 - 3.3.5 Inverted Page Tables 106
- 3.4 PAGE REPLACEMENT ALGORITHMS 107
 - 3.4.1 The Optimal Page Replacement Algorithm 108
 - 3.4.2 The Not-Recently-Used Page Replacement Algorithm 108
 - 3.4.3 The First-In, First-Out 109
 - 3.4.4 The Second Chance Page Replacement Algorithm 110
 - 3.4.5 The Clock Page Replacement Algorithm 111
 - 3.4.6 The Least Recently Used 111
 - 3.4.7 Simulating LRU in Software 112
- 3.5 MODELING PAGING ALGORITHMS 114
 - 3.5.1 Belady's Anomaly 114
 - 3.5.2 Stack Algorithms 114
 - 3.5.3 The Distance String 117
 - 3.5.4 Predicting Page Fault Rates 118
- 3.6 DESIGN ISSUES FOR PAGING SYSTEMS 119
 - 3.6.1 The Working Set Model 119
 - 3.6.2 Local versus Global Allocation Policies 120
 - 3.6.3 Page Size 122
 - 3.6.4 Implementation Issues 124
- 3.7 SEGMENTATION 128
 - 3.7.1 Implementation of Pure Segmentation 131
 - 3.7.2 Segmentation with Paging: MULTICS 132
 - 3.7.3 Segmentation with Paging: The Intel 386 135
- 3.8 SUMMARY 140

4 FILE SYSTEMS

145

- 4.1 FILES 146
 - 4.1.1 File Naming 146
 - 4.1.2 File Structure 148
 - 4.1.3 File Types 149
 - 4.1.4 File Access 151
 - 4.1.5 File Attributes 151
 - 4.1.6 File Operations 153
 - 4.1.7 Memory-Mapped Files 156
- 4.2 DIRECTORIES 158
 - 4.2.1 Hierarchical Directory Systems 158
 - 4.2.2 Path Names 159
 - 4.2.3 Directory Operations 161

- 4.3 FILE SYSTEM IMPLEMENTATION 162
 - 4.3.1 Implementing Files 162
 - 4.3.2 Implementing Directories 165
 - 4.3.3 Shared Files 168
 - 4.3.4 Disk Space Management 170
 - 4.3.5 File System Reliability 173
 - 4.3.6 File System Performance 178
- 4.4 SECURITY 180
 - 4.4.1 The Security Environment 181
 - 4.4.2 Famous Security Flaws 182
 - 4.4.3 The Internet Worm 184
 - 4.4.4 Generic Security Attacks 186
 - 4.4.5 Design Principles for Security 188
 - 4.4.6 User Authentication 189
- 4.5 PROTECTION MECHANISMS 192
 - 4.5.1 Protection Domains 193
 - 4.5.2 Access Control Lists 195
 - 4.5.3 Capabilities 196
 - 4.5.4 Protection Models 198
 - 4.5.5 Covert Channels 200
- 4.6 SUMMARY 201

5 INPUT/OUTPUT

205

- 5.1 PRINCIPLES OF I/O HARDWARE 205
 - 5.1.1 I/O Devices 206
 - 5.1.2 Device Controllers 206
 - 5.1.3 Direct Memory Access 208
- 5.2 PRINCIPLES OF I/O SOFTWARE 210
 - 5.2.1 Goals of the I/O Software 211
 - 5.2.2 Interrupt Handlers 212
 - 5.2.3 Device Drivers 212
 - 5.2.4 Device-Independent I/O Software 213
 - 5.2.5 User-Space I/O Software 215
- 5.3 DISKS 216
 - 5.3.1 Disk Hardware 217
 - 5.3.2 Disk Arm Scheduling Algorithms 217
 - 5.3.3 Error Handling 220
 - 5.3.4 Track-at-a-Time Caching 221
 - 5.3.5 RAM Disks 222
- 5.4 CLOCKS 222
 - 5.4.1 Clock Hardware 223
 - 5.4.2 Clock Software 224

- 5.5 TERMINALS 226
 - 5.5.1 Terminal Hardware 226
 - 5.5.2 Memory-Mapped Terminals 228
 - 5.5.3 Input Software 230
 - 5.5.4 Output Software 235
- 5.6 SUMMARY 236

6 DEADLOCKS

240

- 6.1 RESOURCES 241
- 6.2 DEADLOCKS 242
 - 6.2.1 Conditions for Deadlock 242
 - 6.2.2 Deadlock Modeling 243
- 6.3 THE OSTRICH ALGORITHM 245
- 6.4 DEADLOCK DETECTION AND RECOVERY 246
 - 6.4.1 Deadlock Detection with One Resource of Each Type 246
 - 6.4.2 Deadlock Detection with Multiple Resource of Each Type 249
 - 6.4.3 Recovery from Deadlock 251
- 6.5 DEADLOCK AVOIDANCE 252
 - 6.5.1 Resource Trajectories 253
 - 6.5.2 Safe and Unsafe States 254
 - 6.5.3 The Banker's Algorithm for a Single Resource 255
 - 6.5.4 The Banker's Algorithm for Multiple Resources 256
- 6.6 DEADLOCK PREVENTION 258
 - 6.6.1 Attacking the Mutual Exclusion Condition 258
 - 6.6.2 Attacking the Hold and Wait Condition 258
 - 6.6.3 Attacking the No Preemption Condition 259
 - 6.6.4 Attacking the Circular Wait Condition 259
- 6.7 OTHER ISSUES 260
 - 6.7.1 Two-Phase Locking 260
 - 6.7.2 Non-resource Deadlocks 261
 - 6.7.3 Starvation 261
- 6.8 SUMMARY 262

7 CASE STUDY 1: UNIX

- 7.1 HISTORY OF UNIX 265
 - 7.1.1 UNICS 266
 - 7.1.2 PDP-11 UNIX 267
 - 7.1.3 Portable UNIX 268
 - 7.1.4 Berkeley UNIX 269
 - 7.1.5 Standard UNIX 269

- 7.2 OVERVIEW OF UNIX 271
 - 7.2.1 UNIX Goals 271
 - 7.2.2 Interfaces to UNIX 272
 - 7.2.3 Logging into UNIX 273
 - 7.2.4 The UNIX Shell 274
 - 7.2.5 Files and Directories in UNIX 276
 - 7.2.6 UNIX Utility Programs 277
- 7.3 FUNDAMENTAL CONCEPTS IN UNIX 279
 - 7.3.1 Processes in UNIX 279
 - 7.3.2 The UNIX Memory Model 284
 - 7.3.3 The UNIX File System 287
 - 7.3.4 Input/Output in UNIX 290
- 7.4 UNIX SYSTEM CALLS 293
 - 7.4.1 Process Management System Calls in UNIX 293
 - 7.4.2 Memory Management System Calls in UNIX 297
 - 7.4.3 File and Directory System Calls in UNIX 297
 - 7.4.4 Input/Output System Calls in UNIX 299
- 7.5 IMPLEMENTATION OF UNIX 299
 - 7.5.1 Implementation of Processes in UNIX 300
 - 7.5.2 Implementation of Memory Management in UNIX 303
 - 7.5.3 Implementation of the UNIX File System 307
 - 7.5.4 Implementation of Input/Output in UNIX 310
- 7.6 SUMMARY 312

8 CASE STUDY 2: MS-DOS

315

- 8.1 HISTORY OF MS-DOS 316
 - 8.1.1 The IBM PC 316
 - 8.1.2 MS-DOS Version 1.0 317
 - 8.1.3 MS-DOS Version 2.0 318
 - 8.1.4 MS-DOS Version 3.0 319
 - 8.1.5 MS-DOS Version 4.0 320
 - 8.1.6 MS-DOS Version 5.0 320
- 8.2 OVERVIEW OF MS-DOS 321
 - 8.2.1 Using MS-DOS 321
 - 8.2.2 MS-DOS Shell 324
 - 8.2.3 Configuring MS-DOS 326
- 8.3 FUNDAMENTAL CONCEPTS IN MS-DOS 327
 - 8.3.1 Processes in MS-DOS 328
 - 8.3.2 The MS-DOS Memory Model 332
 - 8.3.3 The MS-DOS File System 340
 - 8.3.4 Input/Output in MS-DOS 342

- 8.4 MS-DOS SYSTEM CALLS 343
 - 8.4.1 Process Management System Calls in MS-DOS 344
 - 8.4.2 Memory Management System Calls in MS-DOS 344
 - 8.4.3 File and Directory System Calls in MS-DOS 346
 - 8.4.4 Input/Output System Calls in MS-DOS 346
- 8.5 IMPLEMENTATION OF MS-DOS 347
 - 8.5.1 Implementation of Processes in MS-DOS 348
 - 8.5.2 Implementation of Memory Management in MS-DOS 350
 - 8.5.3 Implementation of the MS-DOS File System 352
 - 8.5.4 Implementation of Input/Output in MS-DOS 356
- 8.6 SUMMARY 358

PART 2: DISTRIBUTED OPERATING SYSTEMS

9 INTRODUCTION TO DISTRIBUTED SYSTEMS

362

- 9.1 GOALS 363
 - 9.1.1 Advantages of Distributed Systems over Centralized Ones 363
 - 9.1.2 Advantages of Distributed Systems over Independent PCs 365
 - 9.1.3 Disadvantages of Distributed Systems 365
- 9.2 HARDWARE CONCEPTS 366
 - 9.2.1 Bus-Based Multiprocessors 369
 - 9.2.2 Switched Multiprocessors 370
 - 9.2.3 Bus-Based Multicomputers 371
 - 9.2.4 Switched Multicomputers 372
- 9.3 SOFTWARE CONCEPTS 373
 - 9.3.1 Network Operating Systems and NFS 374
 - 9.3.2 True Distributed Systems 382
 - 9.3.3 Multiprocessor Timesharing Systems 383
- 9.4 DESIGN ISSUES 385
 - 9.4.1 Transparency 385
 - 9.4.2 Flexibility 387
 - 9.4.3 Reliability 389
 - 9.4.4 Performance 390
 - 9.4.5 Scalability 391
- 9.5 SUMMARY 393

10 COMMUNICATION IN DISTRIBUTED SYSTEMS

- 10.1 LAYERED PROTOCOLS 396
 - 10.1.1 The Physical Layer 399
 - 10.1.2 The Data Link Layer 399
 - 10.1.3 The Network Layer 400
 - 10.1.4 The Transport Layer 401
 - 10.1.5 The Session Layer 401
 - 10.1.6 Presentation Layer 402
 - 10.1.7 Application Layer 402
- 10.2 THE CLIENT-SERVER MODEL 402
 - 10.2.1 Clients and Servers 403
 - 10.2.2 An Example Client and Server 404
 - 10.2.3 Addressing 406
 - 10.2.4 Blocking versus Nonblocking Primitives 409
 - 10.2.5 Buffered versus Unbuffered Primitives 412
 - 10.2.6 Reliable versus Unreliable Primitives 414
 - 10.2.7 Implementing the Client-Server Model 415
- 10.3 REMOTE PROCEDURE CALL 417
 - 10.3.1 Basic RPC Operation 418
 - 10.3.2 Parameter Passing 422
 - 10.3.3 Dynamic Binding 426
 - 10.3.4 RPC Semantics in the Presence of Failures 428
 - 10.3.5 Implementation Issues 433
 - 10.3.6 Problem Areas 442
- 10.4 GROUP COMMUNICATION 445
 - 10.4.1 Introduction to Group Communication 446
 - 10.4.2 Design Issues 447
 - 10.4.3 Group Communication in ISIS 456
- 10.5 SUMMARY 459

11 SYNCHRONIZATION IN DISTRIBUTED SYSTEMS

- 11.1 CLOCK SYNCHRONIZATION 464
 - 11.1.1 Logical Clocks 465
 - 11.1.2 Physical Clocks 469
 - 11.1.3 Clock Synchronization Algorithms 471
- 11.2 MUTUAL EXCLUSION 476
 - 11.2.1 A Centralized Algorithm 477
 - 11.2.2 A Distributed Algorithm 478
 - 11.2.3 A Token Ring Algorithm 480
 - 11.2.4 A Comparison of the Three Algorithms 482

- 11.3 ELECTION ALGORITHMS 483
 - 11.3.1 The Bully Algorithm 483
 - 11.3.2 A Ring Algorithm 484
- 11.4 ATOMIC TRANSACTIONS 485
 - 11.4.1 Introduction to Atomic Transactions 486
 - 11.4.2 The Transaction Model 487
 - 11.4.3 Implementation 491
 - 11.4.4 Concurrency Control 494
- 11.5 DEADLOCKS IN DISTRIBUTED SYSTEMS 498
 - 11.5.1 Distributed Deadlock Detection 499
 - 11.5.2 Distributed Deadlock Prevention 502
- 11.6 SUMMARY 504

2 PROCESSES AND PROCESSORS IN DISTRIBUTED SYSTEMS

507

- 12.1 THREADS 507
 - 12.1.1 Introduction to Threads 508
 - 12.1.2 Thread Usage 509
 - 12.1.3 Design Issues for Threads Packages 512
 - 12.1.4 Implementing a Threads Package 515
 - 12.1.5 Threads and RPC 518
 - 12.1.6 An Example Threads Package 519
- 12.2 SYSTEM MODELS 523
 - 12.2.1 The Workstation Model 524
 - 12.2.2 Using Idle Workstations 527
 - 12.2.3 The Processor Pool Model 530
 - 12.2.4 A Hybrid Model 534
- 12.3 PROCESSOR ALLOCATION 534
 - 12.3.1 Allocation Models 534
 - 12.3.2 Design Issues for Processor Allocation Algorithms 536
 - 12.3.3 Implementation Issues for Processor Allocation Algorithms 537
 - 12.3.4 Example Processor Allocation Algorithms 540
- 12.4 SCHEDULING IN DISTRIBUTED SYSTEMS 545
- 12.5 SUMMARY 546

DISTRIBUTED FILE SYSTEMS

549

- 13.1 DISTRIBUTED FILE SYSTEM DESIGN 550
 - 13.1.1 The File Service Interface 550
 - 13.1.2 The Directory Server Interface 552
 - 13.1.3 Semantics of File Sharing 556

- 13.2 DISTRIBUTED FILE SYSTEM IMPLEMENTATION 559
 - 13.2.1 File Usage 559
 - 13.2.2 System Structure 561
 - 13.2.3 Caching 564
 - 13.2.4 Replication 570
 - 13.2.5 An Example: The Andrew File System 573
 - 13.2.6 Lessons Learned 579
- 13.3 TRENDS IN DISTRIBUTED FILE SYSTEMS 580
 - 13.3.1 New Hardware 580
 - 13.3.2 Scalability 582
 - 13.3.3 Wide Area Networking 583
 - 13.3.4 Mobile Users 584
 - 13.3.5 Fault Tolerance 585
- 13.4 SUMMARY 585

14 CASE STUDY 3: AMOEB

- 14.1 INTRODUCTION TO AMOEB
- 14.1.1 History of Amoeba 588
- 14.1.2 Research Goals 589
- 14.1.3 The Amoeba System Architecture 590
- 14.1.4 The Amoeba Microkernel 592
- 14.1.5 The Amoeba Servers 594
- 14.2 OBJECTS AND CAPABILITIES IN AMOEB
- 14.2.1 Capabilities 596
- 14.2.2 Object Protection 596
- 14.2.3 Standard Operations 598
- 14.3 PROCESS MANAGEMENT IN AMOEB
- 14.3.1 Processes 599
- 14.3.2 Threads 601
- 14.4 MEMORY MANAGEMENT IN AMOEB
- 14.4.1 Segments 603
- 14.4.2 Mapped Segments 603
- 14.5 COMMUNICATION IN AMOEB
- 14.5.1 Remote Procedure Call 604
- 14.5.2 Group Communication in Amoeba 608
- 14.5.3 The Fast Local Internet Protocol 616
- 14.6 THE AMOEB SERVERS
- 14.6.1 The Bullet Server 623
- 14.6.2 The Directory Server 626
- 14.6.3 The Replication Server 631
- 14.6.4 The Run Server 632
- 14.6.5 The Boot Server 633

14.6.6 The TCP/IP Server 633

14.6.7 Other Servers 634

14.7 SUMMARY 634

5 CASE STUDY 4: MACH

637

15.1 INTRODUCTION TO MACH 637

15.1.1 History of Mach 637

15.1.2 Goals of Mach 639

15.1.3 The Mach Microkernel 639

15.1.4 The Mach BSD UNIX Server 641

15.2 PROCESS MANAGEMENT IN MACH 641

15.2.1 Processes 641

15.2.2 Threads 644

15.2.3 Scheduling 647

15.3 MEMORY MANAGEMENT IN MACH 649

15.3.1 Virtual Memory 650

15.3.2 Memory Sharing 653

15.3.3 External Memory Managers 656

15.3.4 Distributed Shared Memory in Mach 659

15.4 COMMUNICATION IN MACH 660

15.4.1 Ports 660

15.4.2 Sending and Receiving Messages 666

15.4.3 The Network Message Server 670

15.5 BSD UNIX EMULATION IN MACH 672

15.6 COMPARISON OF AMOEBA AND MACH 674

15.6.1 Philosophy 674

15.6.2 Objects 675

15.6.3 Processes 675

15.6.4 Memory Model 676

15.6.5 Communication 677

15.6.6 Servers 678

15.7 SUMMARY 678

READING LIST AND BIBLIOGRAPHY

682

A.1 SUGGESTIONS FOR FURTHER READING 682

A.1.1 Introduction and General Works 682

A.1.2 Processes 683

A.1.3 Memory Management 684

A.1.4 File Systems 684

A.1.5 Input/Output 685

A.1.6	Deadlocks	685
A.1.7	UNIX	686
A.1.8	MS-DOS	686
A.1.9	Introduction to Distributed Systems	687
A.1.10	Communication in Distributed Systems	688
A.1.11	Synchronization in Distributed Systems	689
A.1.12	Processes and Processors in Distributed Systems	689
A.1.13	Distributed File Systems	690
A.1.14	Amoeba	690
A.1.15	Mach	691
A.2	ALPHABETICAL BIBLIOGRAPHY	692

B INTRODUCTION TO C

704

B.1	FUNDAMENTALS OF C	704
B.2	BASIC DATA TYPES	705
B.3	CONSTRUCTED TYPES	706
B.4	STATEMENTS	708
B.5	EXPRESSIONS	710
B.6	PROGRAM STRUCTURE	712
B.7	THE C PREPROCESSOR	713
B.8	IDIOMS	713

INDEX

715