

OPERATING SYSTEMS

CONCEPTS AND DESIGN

SECOND EDITION

Milan Milenković

IBM Corporation

McGRAW-HILL, INC.

New York St. Louis San Francisco Auckland Bogotá Caracas
Lisbon London Madrid Mexico Milan Montreal New Delhi
Paris San Juan Singapore Sydney Tokyo Toronto

CONTENTS

| | |
|--|-----------|
| PREFACE | xxi |
| PART I: FUNDAMENTAL CONCEPTS | 1 |
| 1 Introduction | 3 |
| 1.1 EVOLUTION OF OPERATING SYSTEMS | 4 |
| 1.1.1 Serial Processing | 5 |
| 1.1.2 Batch Processing | 6 |
| 1.1.3 Multiprogramming | 8 |
| 1.2 TYPES OF OPERATING SYSTEMS | 10 |
| 1.2.1 Batch Operating Systems | 10 |
| 1.2.2 Multiprogramming Operating Systems | 11 |
| <i>Time-Sharing Systems</i> | 12 |
| <i>Real-Time Systems</i> | 13 |
| <i>Combination Operating Systems</i> | 14 |
| 1.2.3 Distributed Operating Systems | 15 |
| 1.3 DIFFERENT VIEWS OF THE OPERATING SYSTEM | 15 |
| 1.3.1 The Command-Language User's View of the Operating System | 15 |
| 1.3.2 The System-Call User's View of the Operating System | 17 |
| 1.4 THE JOURNEY OF A COMMAND EXECUTION | 18 |
| 1.5 DESIGN AND IMPLEMENTATION OF OPERATING SYSTEMS | 20 |
| 1.5.1 Functional Requirements | 20 |
| 1.5.2 Implementation | 22 |
| 1.6 SUMMARY | 25 |
| OTHER READING | 26 |
| 2 Processes | 27 |
| 2.1 THE PROCESS CONCEPT | 28 |
| 2.1.1 Implicit and Explicit Tasking | 30 |
| 2.1.2 Process Relationship | 31 |
| 2.2 SYSTEMS PROGRAMMER'S VIEW OF PROCESSES | 32 |
| 2.2.1 A Multitasking Example | 32 |

| | | |
|----------|---|-----------|
| 2.2.2 | Interprocess Synchronization | 35 |
| 2.2.3 | Behavior of Sample Processes | 37 |
| 2.2.4 | Postlude: The Systems Programmer's View of Processes | 43 |
| 2.3 | THE OPERATING SYSTEM'S VIEW OF PROCESSES | 43 |
| 2.3.1 | Process Control Block (PCB) | 46 |
| 2.3.2 | System State and Process Lists | 47 |
| 2.3.3 | Process State Transitions | 47 |
| 2.3.4 | Process Switch | 50 |
| 2.3.5 | Threads | 52 |
| 2.4 | OPERATING-SYSTEM SERVICES FOR PROCESS MANAGEMENT | 53 |
| | <i>CREATE (processID, attributes);</i> | 54 |
| | <i>DELETE (processID);</i> | 54 |
| | <i>ABORT (processID);</i> | 55 |
| | <i>FORK/JOIN</i> | 55 |
| | <i>SUSPEND (processID);</i> | 56 |
| | <i>RESUME (processID);</i> | 56 |
| | <i>DELAY (processID, time);</i> | 56 |
| | <i>GET_ATTRIBUTES (processID, attribute_set);</i> | 57 |
| | <i>CHANGE_PRIORITY (processID, new_priority);</i> | 57 |
| 2.4.1 | Error Returns | 58 |
| 2.5 | SCHEDULING | 58 |
| 2.5.1 | Types of Schedulers | 59 |
| | <i>The long-term scheduler</i> | 59 |
| | <i>The medium-term scheduler</i> | 60 |
| | <i>The short-term scheduler</i> | 61 |
| 2.5.2 | Scheduling and Performance Criteria | 62 |
| 2.5.3 | Scheduler Design | 64 |
| 2.6 | SCHEDULING ALGORITHMS | 64 |
| 2.6.1 | First-Come-First-Served (FCFS) Scheduling | 65 |
| 2.6.2 | Shortest Remaining Time Next (SRTN) Scheduling | 67 |
| 2.6.3 | Time-Slice Scheduling (Round Robin, RR) | 68 |
| 2.6.4 | Priority-Based Preemptive Scheduling (Event Driven, ED) | 75 |
| 2.6.5 | Multiple-Level Queues (MLQ) Scheduling | 76 |
| 2.6.6 | Multiple-Level Queues With Feedback Scheduling | 78 |
| 2.7 | PERFORMANCE EVALUATION | 78 |
| 2.7.1 | FCFS (Batch) | 81 |
| 2.7.2 | Shortest Job First | 82 |
| 2.7.3 | Round Robin | 82 |
| 2.8 | SUMMARY | 83 |
| | OTHER READING | 84 |
| | EXERCISES | 85 |
| 3 | Interprocess Synchronization | 87 |
| 3.1 | THE NEED FOR INTERPROCESS SYNCHRONIZATION | 88 |
| 3.2 | MUTUAL EXCLUSION | 91 |
| 3.2.1 | The First Algorithm | 92 |
| 3.2.2 | The Second Algorithm | 95 |

| | | |
|----------|---|------------|
| 3.2.3 | The Third Algorithm | 97 |
| 3.3 | SEMAPHORES | 97 |
| 3.3.1 | Semaphore Definition and Busy-Wait Implementation | 99 |
| 3.3.2 | Some Properties and Characteristics of Semaphores | 102 |
| | <i>Semaphore Service Discipline</i> | 103 |
| | <i>Semaphore Granularity</i> | 103 |
| 3.4 | HARDWARE SUPPORT FOR MUTUAL EXCLUSION | 104 |
| 3.4.1 | Pessimistic and Optimistic Concurrency Control | 104 |
| 3.4.2 | Disable/Enable Interrupts | 106 |
| 3.4.3 | Test-and-Set Instruction | 107 |
| 3.4.4 | Compare-and-Swap Instruction | 110 |
| 3.5 | QUEUING IMPLEMENTATION OF SEMAPHORES | 112 |
| 3.6 | CLASSICAL PROBLEMS IN CONCURRENT PROGRAMMING | 114 |
| 3.6.1 | The Producers/Consumers Problem | 114 |
| | <i>Producers and consumers with an unbounded buffer</i> | 115 |
| | <i>Producers and consumers with a bounded buffer</i> | 117 |
| 3.6.2 | Readers and Writers | 121 |
| 3.7 | SUMMARY | 124 |
| | OTHER READING | 125 |
| | EXERCISES | 126 |
| 4 | Interprocess Communication and Synchronization | 132 |
| 4.1 | CRITICAL REGION AND CONDITIONAL CRITICAL REGION | 133 |
| 4.2 | MONITORS | 135 |
| 4.3 | MESSAGES | 142 |
| 4.3.1 | Issues in Message Implementation | 143 |
| | <i>Naming</i> | 144 |
| | <i>Copying</i> | 145 |
| | <i>Synchronous vs. Asynchronous Message Exchange</i> | 146 |
| | <i>Message Length</i> | 146 |
| 4.3.2 | Interprocess Communication and Synchronization with Messages | 149 |
| 4.3.3 | Interrupt Signaling via Messages | 153 |
| 4.4 | INTERPROCESS SYNCHRONIZATION AND COMMUNICATION IN ADA | 156 |
| 4.4.1 | The Entry-Accept Mechanism | 158 |
| 4.4.2 | The SELECT Statement | 161 |
| 4.5 | DEADLOCKS | 166 |
| 4.5.1 | Reusable and Consumable Resources | 167 |
| 4.5.2 | Deadlock Prevention | 168 |
| 4.5.3 | Deadlock Avoidance | 170 |
| | <i>Resource Request</i> | 173 |
| | <i>Resource Release</i> | 174 |
| 4.5.4 | Deadlock Detection and Recovery | 175 |
| 4.5.5 | Combined Approach | 178 |
| 4.6 | SUMMARY | 179 |
| | OTHER READING | 180 |
| | EXERCISES | 181 |

| | | |
|----------|--|------------|
| 5 | Memory Management: Contiguous Allocation | 185 |
| 5.1 | SINGLE-PROCESS MONITOR | 188 |
| 5.2 | PARTITIONED MEMORY ALLOCATION—STATIC | 191 |
| 5.2.1 | Principles of Operation | 191 |
| 5.2.2 | Swapping | 195 |
| 5.2.3 | Relocation | 197 |
| | <i>Static Relocation</i> | 198 |
| | <i>Dynamic Relocation</i> | 198 |
| 5.2.4 | Protection | 200 |
| 5.2.5 | Sharing | 202 |
| 5.2.6 | Concluding Remarks | 203 |
| 5.3 | PARTITIONED MEMORY ALLOCATION—DYNAMIC | 204 |
| 5.3.1 | Principles of Operation | 205 |
| 5.3.2 | Compaction | 210 |
| 5.3.3 | Protection | 213 |
| 5.3.4 | Sharing | 213 |
| 5.3.5 | Concluding Remarks | 216 |
| 5.4 | SEGMENTATION | 217 |
| 5.4.1 | Principles of Operation | 217 |
| | <i>Address Translation</i> | 219 |
| | <i>Segment-Descriptor Caching</i> | 221 |
| 5.4.2 | Protection | 224 |
| 5.4.3 | Sharing | 224 |
| 5.4.4 | Concluding Remarks | 226 |
| 5.5 | SUMMARY | 228 |
| | OTHER READING | 228 |
| | EXERCISES | 229 |
| 6 | Memory Management: Noncontiguous Allocation | 232 |
| 6.1 | PAGING | 233 |
| 6.1.1 | Principles of Operation | 233 |
| 6.1.2 | Page Allocation | 236 |
| 6.1.3 | Hardware Support for Paging | 237 |
| 6.1.4 | Protection and Sharing | 240 |
| 6.1.5 | Concluding Remarks | 241 |
| 6.2 | VIRTUAL MEMORY | 242 |
| 6.2.1 | Principles of Operation | 244 |
| 6.2.2 | Instruction Interruptibility | 245 |
| 6.2.3 | Management of Virtual Memory | 248 |
| 6.2.4 | Program Behavior | 249 |
| 6.2.5 | Replacement Policies | 251 |
| | <i>Memory-Reference Strings</i> | 252 |
| | <i>Replacement Algorithms</i> | 253 |
| | <i>Global and Local Replacement Policies</i> | 258 |
| 6.2.6 | Allocation Policies | 258 |
| | <i>Page-Fault Frequency (PFF)</i> | 261 |

| | |
|---|------------|
| 6.2.7 Working Set: A Theory for Page Replacement and Allocation | 262 |
| 6.2.8 Hardware Support and Considerations | 264 |
| 6.2.9 Protection and Sharing | 266 |
| 6.2.10 Segmentation and Paging | 266 |
| 6.2.11 Hierarchical Address Translation Tables and MMUs | 268 |
| 6.2.12 Unix Considerations | 271 |
| 6.2.13 Concluding Remarks | 272 |
| 6.3 SUMMARY | 272 |
| OTHER READING | 273 |
| EXERCISES | 274 |
| 7 File Management | 277 |
| 7.1 COMMAND-LANGUAGE USER'S VIEW OF THE FILE SYSTEM | 278 |
| 7.1.1 Command-Language File Services | 282 |
| 7.2 SYSTEMS PROGRAMMER'S VIEW OF THE FILE SYSTEM | 285 |
| 7.3 DISK ORGANIZATION | 288 |
| 7.3.1 Disk Access Time | 289 |
| 7.4 DISK CONTROLLER AND DRIVER | 291 |
| 7.5 OPERATING SYSTEM'S VIEW OF FILE MANAGEMENT | 293 |
| 7.5.1 Directories | 296 |
| 7.5.2 Disk Space Management | 301 |
| <i>Contiguous allocation</i> | 303 |
| <i>Noncontiguous allocation</i> | 305 |
| 7.5.3 An Anatomy of Disk Address Translation | 310 |
| 7.5.4 File-Related System Services | 316 |
| 7.5.5 Asynchronous Input/Output | 321 |
| 7.6 DISK CACHES AND UNIX BUFFER CACHE | 322 |
| 7.7 A GENERALIZATION OF FILE SERVICES | 326 |
| 7.8 SUMMARY | 327 |
| OTHER READING | 329 |
| EXERCISES | 329 |
| 8 Security and Protection | 333 |
| 8.1 SECURITY THREATS AND GOALS | 334 |
| 8.2 PENETRATION ATTEMPTS | 335 |
| 8.3 SECURITY POLICIES AND MECHANISMS | 337 |
| 8.3.1 Security Policies | 337 |
| 8.3.2 Security Mechanisms and Design Principles | 338 |
| 8.4 AUTHENTICATION | 339 |
| 8.4.1 Passwords | 340 |
| 8.4.2 Artifact-Based Authentication | 341 |
| 8.4.3 Biometric Techniques | 342 |
| 8.5 PROTECTION AND ACCESS CONTROL | 342 |
| 8.5.1 Protection in Computer Systems | 342 |
| 8.5.2 Access-Matrix Model Of Protection | 343 |
| 8.5.3 Access Hierarchies | 345 |

| | | |
|-------|--|-----|
| 8.5.4 | Access Lists | 347 |
| 8.5.5 | Capabilities | 348 |
| 8.5.6 | Locks and Keys | 352 |
| 8.6 | FORMAL MODELS OF PROTECTION | 352 |
| 8.6.1 | Access-Control Matrix | 353 |
| 8.6.2 | The Take-Grant Model | 355 |
| 8.6.3 | The Bell-LaPadula Model | 356 |
| 8.6.4 | Lattice Model of Information Flow | 358 |
| 8.7 | CRYPTOGRAPHY | 360 |
| 8.7.1 | Conventional Cryptography | 362 |
| 8.7.2 | The Data Encryption Standard (DES) | 364 |
| 8.7.3 | Public-Key Cryptography | 365 |
| | <i>The Rivest, Shamir, Adelman (RSA) Algorithm</i> | 366 |
| | <i>Authentication</i> | 368 |
| | <i>Digital Signatures</i> | 369 |
| 8.8 | WORMS AND VIRUSES | 370 |
| 8.8.1 | Computer Worms | 370 |
| 8.8.2 | Computer Viruses | 371 |
| 8.9 | SUMMARY | 374 |
| | OTHER READING | 375 |
| | EXERCISES | 376 |

PART II: IMPLEMENTATION 379

| | | |
|----------|---|------------|
| 9 | Input/Output: Principles and Programming | 381 |
| 9.1 | THE INPUT/OUTPUT PROBLEM | 382 |
| 9.1.1 | Asynchronous Operation | 382 |
| 9.1.2 | The Speed Gap: Processor versus Peripherals | 383 |
| 9.2 | INPUT/OUTPUT INTERFACES | 384 |
| 9.2.1 | Buffer Registers | 388 |
| 9.2.2 | Command Registers | 389 |
| 9.2.3 | Status Registers | 390 |
| 9.3 | I/O PORT EXAMPLES | 390 |
| 9.3.1 | The Universal Synchronous/Asynchronous Receiver/Transmitter (USART) | 390 |
| 9.3.2 | Programmable Interval Timer (PIT) | 392 |
| 9.4 | PROGRAM-CONTROLLED I/O | 395 |
| 9.4.1 | Controlling a Single Device | 395 |
| 9.4.2 | Controlling Multiple Devices: Polling | 400 |
| 9.5 | INTERRUPT-DRIVEN I/O | 402 |
| 9.5.1 | Controlling a Single Device | 402 |
| | <i>Context Switch</i> | 402 |
| | <i>Interrupt-Service Routine (ISR)</i> | 403 |
| 9.5.2 | Controlling Multiple Devices | 408 |
| | <i>Interrupt Vectoring</i> | 408 |
| | <i>Levels of Interrupt Control</i> | 410 |

| | |
|--|------------|
| <i>Priority Levels</i> | 411 |
| <i>A Summary of Interrupt Processing</i> | 411 |
| 9.6 CONCURRENT I/O | 412 |
| 9.7 SUMMARY | 419 |
| OTHER READING | 419 |
| EXERCISES | 420 |
| 10 Design of a Kernel of a Multitasking Operating System (KMOS) | 424 |
| 10.1 DEFINING KMOS SERVICES | 426 |
| 10.2 MAJOR DESIGN DECISIONS | 428 |
| 10.3 PROCESS-STATE TRANSITIONS IN KMOS | 430 |
| 10.4 FUNCTIONAL SPECIFICATION OF KMOS | 431 |
| 10.4.1 Process Dispatching | 432 |
| 10.4.2 Interprocess Communication and Synchronization | 433 |
| <i>Mailboxes and messages</i> | 433 |
| <i>SEND and RECEIVE operations</i> | 436 |
| 10.4.3 Interrupt Management | 438 |
| 10.4.4 Process Management | 441 |
| <i>Process creation</i> | 441 |
| <i>Delaying of a process for a specified time</i> | 442 |
| 10.4.5 System Startup | 444 |
| 10.5 IMPLEMENTATION CONSIDERATIONS | 446 |
| 10.5.1 Systems Implementation Languages | 446 |
| <i>Facilities for modular program development</i> | 446 |
| <i>Access to hardware and to physical memory addresses</i> | 449 |
| 10.5.2 Invoking the Operating System | 450 |
| <i>Procedure call</i> | 450 |
| <i>Supervisor call</i> | 451 |
| <i>Software interrupt</i> | 452 |
| 10.6 SUMMARY | 453 |
| OTHER READING | 453 |
| EXERCISES | 454 |
| 11 Implementation of KMOS | 457 |
| 11.1 KMOS SYSTEM LISTS | 458 |
| 11.2 THE READY LIST AND ITS MANIPULATION | 460 |
| 11.2.1 Implementation of the Ready List in KMOS | 460 |
| 11.2.2 Process Control Block | 462 |
| <i>Management of Process Stacks</i> | 462 |
| <i>Structure of the Process Control Block (PCB)</i> | 463 |
| 11.2.3 Insertions Into the Ready List | 464 |
| 11.2.4 The Null Process | 466 |
| 11.3 INTERPROCESS COMMUNICATION AND SYNCHRONIZATION | 467 |
| 11.3.1 Mailboxes and Messages | 468 |
| 11.3.2 The SEND Operation | 470 |

| | | |
|--------|--|-----|
| 11.3.3 | The RECEIVE Operation | 471 |
| 11.4 | PROCESS MANAGEMENT | 472 |
| 11.4.1 | Process Creation | 472 |
| 11.4.2 | Process Deletion | 474 |
| 11.4.3 | Process Dispatching | 474 |
| 11.4.4 | Delaying a Process for a Specified Time | 476 |
| | <i>Timer Management and Delayed List</i> | 476 |
| | <i>The DELAY Operation</i> | 478 |
| | <i>Timer-Interrupt Processing</i> | 481 |
| 11.4.5 | Procedure SSTACK | 483 |
| 11.5 | INTERRUPT MANAGEMENT | 484 |
| 11.5.1 | Interrupt Mailboxes and Priorities | 484 |
| 11.5.2 | Interrupt Servicing in KMOS | 485 |
| 11.5.3 | Enabling Hardware Interrupt-Levels | 488 |
| 11.6 | STARTUP AND INITIAL SYSTEM CONFIGURATION | 488 |
| 11.7 | SUMMARY | 489 |
| | OTHER READING | 489 |
| | EXERCISES | 490 |
| | KMOS SOURCE: PASCAL | 491 |

PART III: ADVANCED TOPICS

517

| | | |
|--------|--|-----|
| 12 | Multiprocessor Systems | 519 |
| 12.1 | MOTIVATION AND CLASSIFICATION | 520 |
| 12.1.1 | Advantages of Multiprocessors | 520 |
| 12.1.2 | Multiprocessor Classification | 521 |
| 12.2 | MULTIPROCESSOR INTERCONNECTIONS | 522 |
| 12.2.1 | Bus-Oriented Systems | 523 |
| 12.2.2 | Crossbar-Connected Systems | 524 |
| 12.2.3 | Hypercubes | 525 |
| 12.2.4 | Multistage Switch-Based Systems | 527 |
| 12.3 | TYPES OF MULTIPROCESSOR OPERATING SYSTEMS | 529 |
| 12.3.1 | Separate Supervisors | 529 |
| 12.3.2 | Master/Slave | 530 |
| 12.3.3 | Symmetric | 530 |
| 12.4 | MULTIPROCESSOR OS FUNCTIONS AND REQUIREMENTS | 531 |
| 12.5 | OS DESIGN AND IMPLEMENTATION ISSUES | 532 |
| 12.5.1 | Processor Management and Scheduling | 533 |
| | <i>Support for Multiprocessing</i> | 533 |
| | <i>Allocation of Processing Resources</i> | 534 |
| | <i>Scheduling</i> | 535 |
| 12.5.2 | Memory Management | 536 |
| 12.6 | INTRODUCTION TO PARALLEL PROGRAMMING | 537 |
| 12.6.1 | Speedup | 537 |
| 12.6.2 | An Example of Parallel Programming: | |
| | Matrix Multiplication | 538 |

| | |
|---|------------|
| 12.6.3 FORK and JOIN in Multiprocessors | 541 |
| 12.7 MULTIPROCESSOR SYNCHRONIZATION | 542 |
| 12.7.1 Test-and-Set | 542 |
| 12.7.2 Compare-and-Swap | 543 |
| 12.7.3 Fetch-and-Add | 546 |
| 12.8 SUMMARY | 547 |
| OTHER READING | 548 |
| EXERCISES | 549 |
| | |
| 13 Distributed Operating Systems: Algorithms | 551 |
| 13.1 RATIONALE FOR DISTRIBUTED SYSTEMS | 552 |
| 13.1.1 Why Distributed | 552 |
| 13.1.2 What Is Distributed | 555 |
| 13.2 COMPUTER NETWORKS | 556 |
| 13.2.1 Wide-Area Networks | 556 |
| 13.2.2 Local-Area Networks | 560 |
| 13.2.3 Communication Protocols and OSI Model | 562 |
| <i>Physical Layer</i> | 563 |
| <i>Data Link Layer</i> | 563 |
| <i>Network Layer</i> | 564 |
| <i>Transport Layer</i> | 564 |
| <i>Session Layer</i> | 565 |
| <i>Presentation Layer</i> | 565 |
| <i>Application Layer</i> | 565 |
| 13.3 ALGORITHMS FOR DISTRIBUTED PROCESSING | 565 |
| 13.3.1 The Environment and Common Assumptions | 566 |
| 13.3.2 Time and Ordering of Events | 568 |
| 13.3.3 Mutual Exclusion in Distributed Systems | 569 |
| <i>Lamport's Algorithm</i> | 571 |
| <i>Ricart and Agrawala's Algorithm</i> | 571 |
| 13.3.4 Transactions | 572 |
| 13.3.5 Distributed Concurrency Control and Deadlocks | 573 |
| 13.4 COPING WITH FAILURES | 576 |
| 13.4.1 Failures in Distributed Systems | 577 |
| 13.4.2 Election of a Successor | 578 |
| 13.4.3 Regeneration of a Lost Token | 580 |
| 13.4.4 Reaching Agreement | 581 |
| 13.5 SUMMARY | 586 |
| OTHER READING | 586 |
| EXERCISES | 587 |
| | |
| 14 Distributed Operating Systems: Implementation | 589 |
| 14.1 MODELS OF DISTRIBUTED SYSTEMS | 590 |
| 14.1.1 The Host-based Model | 591 |
| 14.1.2 The Processor Pool Model | 591 |
| 14.1.3 The Workstation/Server Model | 591 |

| | | |
|--------|--|-----|
| 14.1.4 | The Integrated Model | 594 |
| 14.2 | NAMING | 594 |
| 14.2.1 | Static Maps | 596 |
| 14.2.2 | Broadcasting | 597 |
| 14.2.3 | Name Servers | 597 |
| 14.2.4 | Prefix Tables | 598 |
| 14.3 | PROCESS MIGRATION | 599 |
| 14.4 | REMOTE PROCEDURE CALLS | 602 |
| 14.4.1 | Transfer of Control | 603 |
| 14.4.2 | Binding | 605 |
| 14.4.3 | Flow of Data | 606 |
| 14.4.4 | RPC Server Design Issues | 607 |
| 14.4.5 | RPC versus Message Passing | 608 |
| 14.5 | DISTRIBUTED SHARED MEMORY | 609 |
| 14.6 | DISTRIBUTED FILE SYSTEMS | 611 |
| 14.6.1 | Client/Server Division of Labor | 612 |
| 14.6.2 | File Caching and Consistency Semantics | 615 |
| 14.6.3 | Statefulness and Performance | 618 |
| 14.6.4 | Fault Tolerance | 619 |
| 14.7 | SUMMARY | 620 |
| | OTHER READING | 622 |
| | EXERCISES | 622 |

PART IV: CASE STUDIES

625

| | | |
|-----------|--|------------|
| 15 | Case Studies | 627 |
| 15.1 | PC-DOS (MS-DOS) OPERATING SYSTEM | 628 |
| 15.1.1 | Command-Language User's View of PC-DOS | 628 |
| 15.1.2 | System-Call User's View of PC-DOS | 630 |
| 15.1.3 | PC-DOS Implementation | 632 |
| 15.1.4 | PC-DOS Summary | 634 |
| 15.2 | THE UNIX OPERATING SYSTEM | 635 |
| 15.2.1 | Command-Language User's View of Unix | 636 |
| 15.2.2 | System-Call User's View of Unix | 642 |
| 15.2.3 | Implementation of Unix | 645 |
| 15.2.4 | Unix Summary | 650 |
| 15.3 | iRMX 86 OPERATING SYSTEM | 650 |
| 15.3.1 | Command-Language User's View of iRMX 86 | 652 |
| 15.3.2 | System-Call User's View of iRMX 86 | 653 |
| 15.3.3 | Implementation of iRMX 86 | 659 |
| 15.3.4 | iRMX 86 Summary | 662 |
| 15.4 | DESIGN OF A REMOTE-TELEMETRY UNIT (RTU) | 662 |
| 15.4.1 | Computer Data-Acquisition and Control Systems | 663 |
| 15.4.2 | The Role of a Remote-Telemetry Unit (RTU) | 666 |
| 15.4.3 | Functional Organization and Activities of an RTU | 668 |

| | |
|---|------------|
| 15.4.4 RTU Software Organization (Processes) | 671 |
| 15.4.5 RTU Data Structures and Interprocess Communication | 674 |
| 15.4.6 Operation of RTU Processes | 676 |
| 15.4.7 RTU Processes and KMOS | 680 |
| 15.4.8 Concluding Remarks | 681 |
| OTHER READING | 682 |
| Appendix A: KMOS Sampler: Pascal | 684 |
| Appendix B: KMOS Source: C | 693 |
| Appendix C: KMOS Sampler: C | 717 |
| Bibliography | 724 |
| Index | 741 |