

OPERATING SYSTEMS AND SYSTEMS PROGRAMMING

SECOND EDITION

P. Balakrishna Prasad,
Professor in CSE Department,
St.ANN'S College of Engineering,
Chirala



Publishing for future

SCITECH PUBLICATIONS (INDIA) PVT. LTD.

www.scitechpublications.com

CONTENTS

| | |
|--|-----------------|
| 1. Computer System Overview | 1.1-1.25 |
| 1.1 Basic Elements | 1.1 |
| 1.2 Instruction Execution | 1.2 |
| 1.3 Interrupts | 1.4 |
| 1.4 The Memory Hierarchy | 1.9 |
| 1.5 Cache Memory | 1.11 |
| 1.6 I/O Communication techniques | 1.15 |
| 1.6.1 Programmed I/O | 1.15 |
| 1.6.2 Interrupt Initiated I/O | 1.16 |
| 1.6.3 Direct Memory Access (DMA) | 1.17 |
| Short Answer Questions | 1.19 |
| Exercise | 1.20 |
| Short questions | 1.20 |
| 2. Operating Systems Overview | 2.1-2.38 |
| 2.1 The Operating System as a Resource Manager | 2.1 |
| 2.2 Structure of Operating System | 2.2 |
| 2.3 The Evolution of Operating Systems | 2.3 |
| 2.3.1 Serial Processing | 2.3 |
| 2.3.2 Batch Processing | 2.3 |
| 2.3.3 Multiprogramming | 2.4 |
| 2.3.4 Time Sharing Systems | 2.5 |
| 2.3.5 Parallel Systems | 2.6 |
| 2.3.6 Distributed Systems | 2.6 |
| 2.4 Operating System Functions | 2.7 |
| 2.5 Major Achievements | 2.8 |
| 2.5.1 Process | 2.8 |
| 2.5.2 Memory Management | 2.9 |
| 2.5.3 Information Protection & Security | 2.10 |
| 2.5.4 Scheduling & Resource Management | 2.10 |
| 2.5.5 System Structure | 2.11 |
| 2.6 Characteristics of Modern Operating System | 2.11 |
| 2.6.1 Object Oriented Design | 2.11 |
| 2.6.2 Multi Threading | 2.11 |
| 2.6.3 Symmetric Multiprocessing | 2.12 |
| 2.6.4 Distributed Operating System | 2.12 |

| | | |
|-----------|--|-----------------|
| 2.7 | Windows 2000 Overview | 2.12 |
| 2.7.1 | History of Windows 2000 | 2.13 |
| 2.8 | Windows 2000 Architecture | 2.15 |
| 2.9 | Traditional Unix Systems | 2.17 |
| 2.10 | Modern Unix Systems | 2.19 |
| | Short Answer Questions | 2.21 |
| | Exercises | 2.22 |
| | Short Questions | 2.23 |
| 3. | Process Description and Control | 3.1-3.34 |
| 3.1 | Process | 3.1 |
| 3.1.1 | Process Vs programs | 3.1 |
| 3.2 | Process states | 3.2 |
| 3.2.1 | Process Creation | 3.3 |
| 3.2.2 | Process Termination | 3.5 |
| 3.3 | Operations on Process | 3.6 |
| 3.4 | Concurrent Process | 3.6 |
| 3.5 | Threads | 3.7 |
| 3.6 | Unix Svr4 Process Management | 3.9 |
| 3.6.1 | Process States | 3.9 |
| 3.6.2 | Process Description | 3.11 |
| 3.6.3 | Process Control | 3.12 |
| 3.7 | Processes and Threads | 3.13 |
| 3.7.1 | Multithreading | 3.13 |
| 3.8 | Symmetric Multi Processing | 3.17 |
| 3.8.1 | SMPS Organization | 3.18 |
| 3.8.2 | Multiprocessor Operating System-design Considerations | 3.18 |
| 3.9 | Micro Kernels | 3.18 |
| 3.9.1 | Micro Kernel Architecture | 3.18 |
| 3.9.2 | Benefits of Micro Kernel Organization | 3.19 |
| 3.10 | Windows 2000 Thread and SMP Management | 3.19 |
| 3.10.1 | Process and Thread Objects | 3.20 |
| 3.10.2 | SMP Support | 3.22 |
| 3.11 | Solaris Thread and SMP Management | 3.22 |
| 3.11.1 | Multithreaded Architecture | 3.22 |
| 3.11.2 | Process Structure | 3.24 |
| 3.12 | LINUX Process and Thread Management | 3.27 |
| 3.12.1 | Linux Processes | 3.27 |
| 3.12.2 | Linux Threads | 3.29 |

| | |
|--|-----------------|
| Exercise | 3.29 |
| Short Questions | 3.30 |
| 4. CPU Scheduling | 4.1-4.29 |
| 4.1 Schedulers | 4.1 |
| 4.1.1 Long Term Scheduler | 4.1 |
| 4.1.2 Short Term Scheduler | 4.2 |
| 4.1.3 Medium Term Scheduler | 4.3 |
| 4.2 Scheduling Methodology | 4.3 |
| 4.3 CPU Scheduling Algorithms | 4.4 |
| 4.3.1 First-Come, First-Served Scheduling: (FCFS) | 4.4 |
| 4.3.2 Shortest Job First Scheduling(SJF) | 4.7 |
| 4.3.3 Shortest Remaining Time First(SRTF) | 4.9 |
| 4.3.4 Round Robin Scheduling Algorithm (RR) | 4.11 |
| 4.3.5 Priority Scheduling | 4.12 |
| 4.3.6 Highest Response Ratio Next (HRRN) | 4.14 |
| 4.3.7 Multilevel Queue Scheduling | 4.14 |
| 4.3.8 Multilevel Feedback Queues | 4.15 |
| 4.4 Performance Comparison | 4.16 |
| 4.4.1 Deterministic Modeling | 4.16 |
| 4.4.2 Queuing Analysis | 4.16 |
| 4.4.3 Simulators | 4.17 |
| Exercises (1) | 4.17 |
| Short Answer Questions | 4.18 |
| Exercise (2) | 4.19 |
| Short Questions | 4.21 |
| 5. Concurrency: Deadlock and Starvation | 5.1-5.23 |
| 5.1 Definition | 5.1 |
| 5.2 Resource Allocation Graph | 5.1 |
| 5.3 The Conditions for Deadlock | 5.2 |
| 5.4 Deadlock Prevention | 5.3 |
| 5.5 Deadlock Avoidance | 5.4 |
| 5.6 Deadlock Detection | 5.8 |
| 5.6.1 Single Instance of Resource Type | 5.8 |
| 5.6.2 Deadlock Detection-Several Instance of Resource Type | 5.9 |
| 5.7 Recovery from Deadlock | 5.10 |
| 5.7.1 Process Termination | 5.10 |
| 5.7.2 Resource Preemption | 5.10 |
| 5.8 UNIX Concurrency Mechanism | 5.10 |

| | | |
|-----------|---|-----------------|
| 5.9 | Solaris Thread Synchronization Primitives | 5.13 |
| 5.9.1 | Mutual Exclusion Lock | 5.13 |
| 5.9.2 | Semaphores | 5.13 |
| 5.9.3 | Readers/Writer Lock | 5.14 |
| 5.9.4 | Condition Variables | 5.14 |
| 5.10 | Windows 2000 Concurrency Mechanisms | 5.14 |
| | Short Answer Questions | 5.15 |
| | Exercise | 5.16 |
| 6. | Memory Management | 6.1-6.39 |
| 6.1 | Logical Versus Physical Address Space | 6.2 |
| 6.2 | Swapping | 6.2 |
| 6.3 | Memory Management Requirements | 6.3 |
| 6.3.1 | Relocation | 6.3 |
| 6.3.2 | Protection | 6.4 |
| 6.3.3 | Sharing | 6.5 |
| 6.3.4 | Logical organization | 6.5 |
| 6.3.5 | Physical organization | 6.6 |
| 6.4 | Dynamic Loading and Dynamic Linking | 6.6 |
| 6.5 | Memory Allocation Method | 6.7 |
| 6.5.1 | Single Partition Allocation | 6.8 |
| 6.5.2 | Multiple Partitions Memory Management | 6.10 |
| 6.5.3 | Compaction | 6.16 |
| 6.5.4 | Paging | 6.18 |
| 6.5.5 | Segmentation | 6.23 |
| 6.5.6 | Segmentation with Paging | 6.25 |
| 6.6 | Protection | 6.26 |
| 6.7 | Comparisons between Paging and Segmentation | 6.27 |
| 6.8 | Translation Look Aside Buffer | 6.27 |
| | Short Answer Questions | 6.28 |
| | Exercise | 6.30 |
| 7. | Virtual Memory | 7.1-7.26 |
| 7.1 | Demand Paging | 7.1 |
| 7.2 | Page Fault | 7.2 |
| 7.3 | Page Replacement Algorithms | 7.3 |
| 7.3.1 | FIFO Algorithms (First in first out algorithms) | 7.3 |
| 7.3.2 | Optimal Page Replacement Algorithm | 7.4 |
| 7.3.3 | LRU(Least recently used algorithm) | 7.5 |
| 7.3.4 | Least Frequently used Algorithm (LFU) | 7.5 |

| | | |
|----------|--------------------------------------|-----------------|
| 7.3.5 | Most Frequently used Algorithm (MFU) | 7.6 |
| 7.3.6 | Belady's Anomaly | 7.6 |
| 7.3.7 | Clock page Replacement Algorithm | 7.7 |
| 7.4 | Thrashing | 7.7 |
| 7.5 | Operating System Software | 7.8 |
| 7.6 | UNIX / Solaris Memory Management | 7.9 |
| 7.6.1 | Paging Memory Management | 7.9 |
| 7.6.2 | Page Replacement Algorithm | 7.11 |
| 7.6.3 | Kernel Memory Allocator | 7.12 |
| 7.6.4 | Parameter in Lazy Buddy System | 7.12 |
| 7.6.5 | Lazy Buddy System Algorithm | 7.13 |
| 7.7 | Linux Memory Management | 7.13 |
| 7.7.1 | Linux Virtual Memory | 7.13 |
| 7.7.2 | Page Allocation | 7.14 |
| 7.7.3 | Page Replacement Algorithm | 7.14 |
| 7.7.4 | Kernal Memory Allocation | 7.14 |
| 7.8 | Windows 2000 Memory Management | 7.14 |
| 7.8.1 | Virtual Address MAP | 7.14 |
| 7.8.2 | Windows 2000 Paging | 7.15 |
| 7.8.3 | Traditional UNIX Scheduling | 7.15 |
| 7.8.4 | Example of Process Scheduling | 7.15 |
| 7.8.5 | Algorithm for Process Scheduling | 7.17 |
| | Short Answer Questions | 7.17 |
| | Exercise | 7.18 |
| 8 | I/O Management | 8.1-8.32 |
| 8.1 | Introduction | 8.1 |
| 8.2 | I/O Hardware | 8.2 |
| 8.3 | I/O Buffering | 8.3 |
| 8.3.1 | Operating System Design Issues | 8.6 |
| 8.4 | Disk I/O | 8.6 |
| 8.4.1 | Disk Performance Parameter | 8.7 |
| 8.4.2 | Disk Scheduling Algorithms | 8.8 |
| 8.5 | Raid | 8.13 |
| 8.6 | Disk Cache | 8.17 |
| 8.6.1 | Design Considerations | 8.17 |
| 8.7 | UNIX SVR4 I/O | 8.20 |
| 8.7.1 | Buffer Cache | 8.20 |

| | | |
|------------|---|-------------------|
| 8.8 | Windows 2000 I/O | 8.22 |
| | Short Answer Questions | 8.23 |
| | Exercise | 8.24 |
| 9. | Concurrency : Mutual Exclusion and Synchronization | 9.2-9.25 |
| 9.1 | Critical – Section | 9.2 |
| 9.2 | Mutual Exclusion | 9.3 |
| | 9.2.1 Requirements for Mutual Exclusion | 9.3 |
| 9.3 | Semaphores | 9.7 |
| 9.4 | Inter Process Communication | 9.9 |
| | 9.4.1 The Producer–Consumer Problem | 9.9 |
| | 9.4.2 The Dining philosophers Problem | 9.11 |
| | 9.4.3 The Readers and Writers Problem | 9.14 |
| | 9.4.4 The Sleeping Barber Problem | 9.15 |
| 9.5 | Monitors | 9.17 |
| 9.6 | Message Passing | 9.19 |
| | Exercise | 9.21 |
| | Short Questions | 9.21 |
| 10. | File Management | 10.1-10.23 |
| 10.1 | Overview | 10.1 |
| 10.2 | File Management System | 10.4 |
| 10.3 | File Accessing Methods | 10.5 |
| | 10.3.1 Sequential file access | 10.5 |
| | 10.3.2 Direct Access | 10.6 |
| | 10.3.3 Indexed Sequential File | 10.6 |
| 10.4 | File Directories | 10.7 |
| 10.5 | File Allocation Methods | 10.10 |
| | 10.5.1 Contiguous Allocation | 10.10 |
| | 10.5.2 Linked Allocation | 10.11 |
| | 10.5.3 Grouped Allocation (or) Indexed Allocation | 10.13 |
| 10.6 | Free Space Management (or) Disk Space Management | 10.14 |
| | 10.6.1 Bit Vector or Bit Table | 10.14 |
| | 10.6.2 Chain Free Points (or) Linked Free Space List | 10.14 |
| | 10.6.3 Index Block List | 10.14 |
| 10.7 | Record Blocking | 10.16 |
| 10.8 | Unix File Management | 10.17 |
| 10.9 | Windows 2000 File System | 10.18 |
| | 10.9.1 Key Feature of NTFS | 10.19 |
| | 10.9.2 Desk Storage Concepts | 10.19 |

| | |
|--|-------------------|
| Short Answer Questions | 10.19 |
| Exercise | 10.20 |
| 11. Unix | 11.1-11.8 |
| 11.1 Introduction | 11.1 |
| 11.2 Security in Unix | 11.2 |
| 11.3 The Unix System Software | 11.3 |
| 11.4 Process Control in Unix | 11.4 |
| 11.5 Pipe Lines in Unix | 11.5 |
| 11.6 File Protection in Unix | 11.7 |
| 12. Protection / Security | 12.1-12.14 |
| 12.1 Mechanisms and Policies | 12.1 |
| 12.2 Threats | 12.2 |
| 12.3 Intruders | 12.4 |
| 12.4 Accidental Data Loss | 12.4 |
| 12.5 Protection Mechanisms | 12.5 |
| 12.5.1 Cryptography | 12.5 |
| 12.5.2 Digital Signatures | 12.6 |
| 12.5.3 Trusted System | 12.7 |
| 12.6 User Authentication | 12.8 |
| 12.6.1 Passwords | 12.8 |
| 12.6.2 Physical Identification | 12.10 |
| 12.6.3 Authentication Using Biometrics | 12.10 |
| 12.7 Attacks From Inside the System | 12.10 |
| 12.7.1 Trojan Horse | 12.11 |
| 12.7.2 Login Spoofing | 12.11 |
| 12.7.3 Trap Doors | 12.12 |
| 12.8 Virus | 12.12 |
| 12.8.1 How Virus Work | 12.12 |
| 12.8.2 Boot Sector Virus | 12.12 |
| 12.8.3 Device Driver Viruses | 12.13 |
| 12.8.4 Macroviruses | 12.13 |
| 12.8.5 Polymorphic Virus | 12.13 |
| 12.8.6 Memory Resident Virus | 12.13 |
| 12.9 Antivirus Approaches | 12.14 |
| Review Questions | 12.14 |
| 13. Case Study - Windows NT | 13.1-13.6 |

| | |
|--|-------------------|
| 14. Case Study - Windows 2000 | 14.1-14.9 |
| 14.1 History of Windows 2000 | 14.1 |
| 14.1.1 MS – DOS | 14.1 |
| 14.1.2 Consumer Windows (windows 95/98/me) | 14.1 |
| 14.1.3 Windows NT | 14.2 |
| 14.1.4 Windows 2000 | 14.2 |
| 14.2 The Application-Programming Interface | 14.3 |
| 14.3 Windows 2000 Architecture | 14.4 |
| 14.4 Security Subsystem Architecture | 14.6 |
| 14.5 Directory Service Architecture | 14.7 |
| 14.6 Data Storage | 14.8 |
| 14.7 Active Directory Architecture | 14.8 |
| 14.8 Authentication in Windows 2000 | 14.9 |
| 15. Assemblers | 15.1-15.18 |
| 15.1 Elements of Assembly Language Programming | 15.1 |
| 15.2 Assembly Language Program | 15.2 |
| 15.3 Assembler Directives | 15.3 |
| 15.4 A Single Pass Assembler for IBM PC | 15.9 |
| 15.5 Intel 8088 Instructions | 15.11 |
| 15.6 Operators Supported by Intel 8088 | 15.12 |
| 15.7 Design of Single Pass Assembler | 15.13 |
| 16. Macro Processor | 16.1-16.14 |
| 16.1 Features of a Macro Facility | 16.3 |
| 16.2 Design of a Macro Processor | 16.8 |
| 16.3 Algorithm for Pass 1 | 16.10 |
| 16.4 Macro Assembler | 16.14 |
| Question and Answers | 16.14 |
| 17. Linkers | 17.1-17.10 |
| 17.1 Relocation | 17.2 |
| 17.2 Editors | 17.9 |
| Short Questions | 17.10 |
| Sample Programs in OS (52 Programs) | 1-113 |
| Sample Programs in SP (45 Programs) | 1-41 |
| Glossary | 1-14 |
| Acronyms | |
| Bit Questions | 1-13 |
| University Question Papers | 1-4 |
| Operating Systems Lab Cycle | |