Operating System (3 — 0 — 2)

Evaluation:

	Theory	Practical	Total
Sessional	30	20	50
Final	50	-	50
Total	80	20	100

Course Objectives:

1. To provide the students with the knowledge of basics design principles of operating systems.
2. To gain knowledge about the Operating Systems concepts such as process, memory, resource, I/O management, CPU and disk scheduling etc.

Course Contents:

1. Types and Structure of Operating Systems (4 hrs)

1.1. Introduction and History of Operating System
1.2. Operating System Concepts and Functionalities
1.2.1 Processes
1.2.2. Files
1.2.3. System Calls
1.2.4. The Shell
1.3. Operating System Structure
1.3.1 Monolithic Systems
1.3.2 Layered
1.3.3 Virtual Machines
1.3.4 Client-Server
1.4 Types and Evolution of Operating Systems

2. Processes and Threads (16 hrs)

1.1. Process Concepts
1.1.1. Introduction
2.1.2. Definition of Process
2.1.3. Process States and Transition 2.1.4. PCB (Process Control Block) 2.1.5. Concurrent Process, Parallel Processing
2.2. IPC (Inter-Process Communication)
2.2.1. Critical Regions and Conditions
2.2.2. Mutual Exclusion
2.2.3. Mutual Exclusion Primitives and Implementation
2.2.3.1. Dekker's Algorithm
2.2.3.2. Peterson's Algorithm
2.2.3.3. TSL (Test and Set Lock)
2.2.3.3. Locks
2.2.4. Producer and Consumer problem

2.2.5. Monitors
2.2.6. Message Passing
2.2.7. Classical IPC Problems
2.2.7.1. The Dining Philosophers Problem
2.2.7.2. The Readers and Writers Problem
2.2.7.3. The Sleeping Barber Problem

2.3. Deadlock and Indefinite Postponement:
2.3.1 Introduction
2.3.1.1 Preempt able and Nonpreemptable Resources
2.3.1.2 Conditions for Deadlock
2.3.1.3 Deadlock Modeling
2.3.2. Deadlock Avoidance
2.3.3. Deadlock Detection and Recovery
2.3.4. Deadlock Prevention
2.3.5. Issues Related to Deadlocks
2.3.5.1. Two Phase Locking
2.3.5.2. Non resource Deadlocks
2.3.5.3. Starvation

2.4. Threads
2.4.1. Introduction
2.4.2. Thread Model
2.4.1 Thread Usage
2.4.4. Advantages of Threads
2.4.5. User Space and Kernel Space Threads
2.4.6. Multithreading Model

2.5. Differences between Threads and Processes

3. Kernel

3.1. Introduction and Architecture of a Kernel
3.2. Types of Kernels
3.3. Context Switching (Kernel mode and User mode)
3.4. First Level Interrupt Handling
3.5. Kernel Implementation of Processes

4. Scheduling

4.1. Introduction:
4.2. Scheduling Levels
4.2.1. Scheduling Objectives and Criteria
4.2.2 Quantum Size
4.3. Preemptive Versus No Preemptive Scheduling
4.4. Scheduling techniques:
4.4.1. Priority Scheduling 4.4.2. Deadline Scheduling 4.4.3. First-In-First-Out Scheduling
4.4.4. Round Robin Scheduling
4.4.5. Shortest-Job-First(SJF) Scheduling

4.4.6. Shortest-Remaining-Time(SRT) Scheduling
4.4.7. Highest-Response-Ration-Next(HRN) Scheduling
4.4.8. Multilevel Feedback Queues

5. Memory Management

5.1. Introduction,
5.1.1. Storage Organization, Hierarchy and Management
5.1.2. Storage Allocation
5.1.3. Contiguous versus Noncontiguous Storage Allocation
5.1.4. Logical and Physical Memory
5.2. Fragmentation
5.3. Fixed Partition Multiprogramming
5.4. Variable Partition Multiprogramming
5.5. Relocation and Protection
5.6. Coalescing and Compaction
5.7. Virtual Memory:
5.7.1. Introduction
5.7.2. Paging
5.7.3. Page Tables
5.7.4. Block Mapping
5.7.5. Direct Mapping
5.7.6. TLB (Translation Look Aside Buffers)
5.7.7. Page Fault
5.7.8. Thrashing
5.8. Page Replacement Algorithms
5.8.1. Optimal Page Replacement Algorithm
5.8.2. Not Recently Used Page Replacement Algorithm
5.8.3. First-In-First-Out Algorithm
5.8.4. Second Chance Page Replacement Algorithm
5.8.5. Least Recently Used Replacement Algorithm
5.8.6. Clock Page Replacement Algorithm
5.8.7. Working Set Page Replacement Algorithm
5.8.8. WS Clock Page Replacement Algorithm
5.9. Segmentation
5.9.1. Implementation of Pure Segmentation
5.9.2. Segmentation with Paging

6. Input/output

6.1. Introduction
6.2. Principles of I/O Hardware
6.2.1. I/O Devices
6.2.2. Device Controllers
6.2.3. Memory-mapped 1/0
6.2.4. DMA (Direct Memory Access)
6.3. Principles of I/O Software
6.3.1. Goals of 1/0 Software

6.3.2. Polled I/O versus Interrupt Driven I/O
6.3.3. Character User Interface and Graphical User Interface
6.3.4. Goals of I/O Software
6.3.5. Device Drivers
6.3.6. Device independent I/O software
6.3.7. User Space I/O Software
6.4. Disk
6.4.1. Disk hardware
6.4.2. Disk Scheduling
6.4.2.1. Seek Time
6.4.2.2. Transfer Time
6.4.2.3. Disk Arm Scheduling Algorithms
6.4.3. RAID (Redundant Array of Inexpensive Disks)
6.4.4. RAM Disks

7. File Systems (3 hrs)

7.1. File Organization
7.1.1. Blocking and Buffering
7.1.2. File Descriptor
7.1.3. File Naming
7.1.4. File Structure
7.1.5. File Types
7.1.6. File Attributes
7.1.7. File Operations
7.1.8. File Access Methods
7.2. Sharing
7.3. ACL (Access Control List)
7.4. Directories
7.4.1. Directory Hierarchy
7.4.2. Directory Operations
7.5. File System Implementation
7.5.1. Contiguous Allocation
7.5.2. Linked list Allocation
7.5.3. I-nodes
7.6. Security and Multi-media files

8. Distributed Operating Systems (5 hrs)

8.1. Introduction
8.1.1. Advantages and Disadvantages of Distributed Operating System
8.1.2. Network Architecture
8.1.3. Hardware and Software Concepts
8.2. Communication in Distributed Systems
8.2.1. ATM (Asynchronous Transfer Mode)
8.3. Client-Server Model
8.4. RPC (Remote Procedure Call)
8.5. Group Communication

8.6. Processes and Processors in Distributed System
8.7. Clock Synchronization
8.8. Scheduling in Distributed System.

9 Case Studies (6 hrs)

DOS Operating System: System Configurations, Filing and Disk Management, Graphical Capabilities, Memory management.

Unix/Linux Operating System: File Systems and Disk Management, Filters, Pipelining, Sockets, Shell, Memory Management, Networking feature, Multiprocessing Feature.

Window 2000: File System and Disk management, Networking, Security.

AMOEBA: The System Architecture of Amoeba, Memory, Process Management and Communication in AMOEBA

Laboratory:

1. Housekeeping in DOS and Windows.
2. Housekeeping in UNIX/LINUX
3. Shell management in UNIX/LINUX and Shell Programming
4. Implementation of Resource and Memory Management Schemes in UNIX/LINUX
5. Process and Thread Creation, Scheduling and Management
6. Implementation of Inter-Process Communication Using Buffers
7. Implementation of Deadlock Prevention Algorithms
8. Device Programming

Text Books:

1. A.S. Tanenbaum, Operating systems, Design and Implementation, Prentice hall India.
2. UM. Dietel, An Introduction to Operating System, Addison Wesley

References:

1. A.S. Tanenbaum, Modern Operating System, Second Edition. Prentice hall India.
2. W. Stallings. Operating Systems, Prentice hall India.