Operating Systems

Course Title: Operating Systems **Full Marks:**60+ 20+20

Course No: CSC259 **Pass Marks:** 24+8+8

Nature of the Course: Theory + Lab **Credit Hrs:** 3

Course Description: This course includes the basic concepts of operating system components. It consists of process management, deadlocks and process synchronization, memory management techniques, File system implementation, and I/O device management principles. It also includes case study on Linux operating system.

Course Objectives

- Describe need and role of operating system.
- Understand OS components such a scheduler, memory manager, file system handlers and I/O device managers.
- Analyze and criticize techniques used in OS components
- Demonstrate and simulate algorithms used in OS components
- Identify algorithms and techniques used in different components of Linux

Course Contents:

Unit	Teaching Hour	References
Unit 1: Operating System Overview (4)		
1.1 Introduction: Definition, Two views of operating system, Evolution/*History* of operating system, Types of OS (*Mainframe, Server, Multiprocessor, PC, Real-Time, Embedded, Smart Card Operating Systems*), Operating System Structures **1.2 System Calls:** Definition, Handling System Calls, *System calls for Process, File, and Directory Management*, System Programs, The Shell, Open Source Operating Systems	**2 Hour** **2 Hour**	
Unit 2: Process Management (10)		

2.1 Introduction: Process vs Program, Multiprogramming, Process Model, Process States, Process Control Block/*Process Table*. **2.2 Threads:** Definition, Thread vs Process, *Thread Usage*, User and Kernel Space Threads. **2.3 Inter Process Communication:** Definition Race Condition, Critical Section **2.4 Implementing Mutual Exclusion**: Mutual Exclusion with Busy Waiting (Disabling Interrupts, Lock Variables, Strict Alteration, Peterson's Solution, Test and Set Lock), Sleep and Wakeup, Semaphore, Monitors, Message Passing **2.5 Classical IPC problems**: Producer Consumer, Sleeping Barber, and Dining Philosopher Problem **2.6 Process Scheduling**: Goals, Batch System Scheduling (First-Come First-Served, Shortest Job First, Shortest Remaining Time Next), Interactive System Scheduling (Round-Robin Scheduling, Priority Scheduling, Multiple Queues), Overview of Real Time System Scheduling (*No need to discuss any real time system scheduling algorithm*)	**1 Hour** **1 Hour** **1 Hour** **3 Hour** **1 Hour** **3 Hour**	
Unit 3: Process Deadlocks (6)		
3.1 Introduction: Definition, Deadlock Characterization, Preemptable and Non-Preemptable Resources, Resource–	**1.5 Hour**	

Allocation Graph, *Necessary* Conditions for Deadlock **3.2 Handling Deadlocks**: Ostrich Algorithm, Deadlock prevention, *Safe and Unsafe States,* Deadlock Avoidance (*Bankers algorithm for Single and Multiple Resource Instances*), , Deadlock Detection (*For Single and Multiple Resource Instances*), Recovery From Deadlock (Through Preemption and Rollback)	**4.5 Hour**	
Unit 4: Memory Management (8)		
4.1 Introduction: Monoprogramming vs Multi-programming, Modelling Multi-programming, Multiprogramming with fixed and variable partitions, Relocation and Protection. **4.2 Space Management:** *Fragmentation and Compaction*, Memory management (Bitmaps & Linked-list), Memory Allocation Strategies **4.3 Virtual Memory:** Paging, Page Table, Page Table Structure, *Pages and Frames*, Handling Page Faults, TLB's **4.4 Page Replacement Algorithms**: *Hit Rate and Miss Rate*, Concept of Locality of Reference, FIFO, Belady's Anomaly, Second Chance, LRU, Optimal, LFU, Clock, WS-Clock. **4.5 Segmentation**: Why Segmentation, *Drawbacks of Segmentation*, Segmentation	**1 Hour** **1 Hour** **2 Hour** **3 Hour** **1 Hour**	

with Paging(MULTICS)		
Unit 5: File Management (6)		
5.1 File Overview: File Naming, File Structure, File Types, File Access, File Attributes, File Operations, Single Level, Two Level and Hierarchical Directory Systems, File System Layout. **5.2 Implementing Files:** Contiguous allocation, Linked List Allocation, *Linked List Allocation using Table in Memory/ File Allocation Table*, Inodes. **5.3 Directory:** Directory Operations, Path Names, Directory Implementation, Shared Files **5.4 Free Space Management:** Bitmaps, Linked List	**1 Hour** **3 Hour** **1 Hour** **1 hour**	
Unit 6: Device Management (6)		
6.1 Introduction: Classification of IO devices, Controllers, Memory Mapped IO, DMA Operation, Interrupts **6.2 IO Handling**: Goals of IO Software, Handling IO(*Programmed IO, Interrupt Driven IO, IO using DMA*), IO Software Layers (Interrupt Handlers, Device Drivers) **6.3 Disk Management:** Disk Structure, Disk Scheduling (*FCFS, SSTF, SCAN, CSCAN, LOOK, CLOOK*), Disk Formatting (*Cylinder Skew, Interleaving, Error handling*), RAID	**1 Hour** **2 Hour** **3 Hour**	
Unit 7: Linux Case Study (5)		
7.1 History, Kernel Modules, Process	**5 Hour**	

Management, Scheduling, Inter-process Communication, Memory Management, File System Management Approaches, Device Management Approaches.		

Text Book

- Modern Operating Systems: Andrew S. Tanenbaum, PH1 Publication, Third edition, 2008

Reference

- Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, "Operating System Concepts", John Wiley & Sons (ASIA) Pvt. Ltd, Seventh edition, 2005.
- Harvey M. Deitel, Paul J. Deitel, and David R. Choffnes, "Operating Systems, Prentice Hall, Third edition, 2003.

Laboratory Work

The laboratory work includes solving problems in operating system. The lab work should include;

1 Demonstration of basic Linux Commands
2 Process creation and termination, thread creation and termination
3 Simulation of IPC techniques
4 Simulation process Scheduling algorithms
5 Simulation of deadlock avoidance and deadlock detection algorithms
6 Simulation of page replacement algorithms
7 Simulation of File allocation techniques
8 Simulate free space management techniques
9 Simulation of disk scheduling algorithms

Model Question

Long Questions

Attempt any two questions. (2 × 10 = 20)

1 What is sleep and wakeup? Demonstrate problem with suitable code snippet and illustration.

2 When page fault occurs and how it is handled? Demonstrate Second Chance, and LRU page replacement algorithm for memory with three frames and following reference string: *1,3,7,4,5,2,3,6,4,5,7,8, 5,1,4*

3 What is Inode? Why it is superior to other file allocation approaches? Consider 20-GB disk with 8-KB block size. How much memory space will be occupied if contiguous, and File allocation table is used for file allocation. Assume that each FAT entry takes 4 byte.

Short Questions

Attempt any eight questions. (8 × 5 = 40)

4 Define the terms shell and system call? How it is handled? Illustrate with suitable example.

5 What are main goals of interactive system scheduling? Discuss priority scheduling along with its pros and cons.

6 How starvation differs from deadlock? Consider the following situation of processes and resources:

Process	Has	Max
P1	2	6
P2	1	5
P3	2	5
P4	2	6

Free=3

- What will happen if process P3 requests 1 resource?
- What will happen if process P4 requests 1 resource?

7 Consider a virtual memory and physical memory of size 128-MB and 32-MB respectively. Assume that page size is 4-KB. What will be the number of bits required for page number, frame number, and offset? Find physical address for the virtual address 20500. (Assume that value at index 5 of page table is 2)

8 Define the term race condition? Justify that race condition leads data loss or incorrect data.

9 Explain directory implementation techniques employed in operating systems briefly.

10 What is the main purpose of disk scheduling algorithms? Which disk scheduling technique is best but impractical? Explain the algorithm with example.

11 How threads differ from processes? Explain thread usages.

12 Write short notes on:

a) Linux Scheduling

b) Fragmentation