LESSON PLAN 2025-26(SUMMER)

NAME OF THE TEACHER : ANJALI MUNDA, GUEST LECT.(CSE)

Subject: **OPERATING SYSTEMS (Course Code: CSEPC202/TH1)**
Program: Diploma in Computer Science and Engineering
Semester: 4th
Total Contact Hours: 45
Total Marks: 100
Assessment: Internal Assessment – 30, End Term – 70

After completion of the course, the students will be able to:
CO1-Explain the basic concept about the OS,
CO2- Illustrate UNIX/LINUX architecture,
CO3- Explain Process management, memory management and file management,
CO4-Demonstrate I/O system including RAID,
CO5-Explain the concept of deadlocks and measures to prevent them,
CO6-Illustrate OS Security.

Lesson No.	UNIT	Topic/Sub-Topic	Learning Objective	Activity	Homework	COURSE OBJECTIVE
UNIT-1-: Overview of Operating System: (6 Hours)						
1	I	Introduction to Operating Systems	Define Operating System (OS). Understand the role and functions of an OS.	Brainstorm: "What does a computer do?" Group discussion on different OS examples (Windows, macOS, Linux, Android).	List 5 functions of an OS and identify the OS on your phone and PC.	**CO1**
2	I	Overview of Operating System: Basic Concepts (1/2)	Explain key concepts: kernel, user mode vs. kernel mode, batch systems, time-sharing.	Diagram: Draw the basic structure of a computer system showing Hardware, OS, and Applications.	Read about the evolution from batch systems to time-sharing systems	CO1
3	I	Overview of Operating System: Basic Concepts (2/2	Describe multiprogramming, multiprocessing, and distributed systems	Compare & Contrast: Multiprogramming vs. Multiprocessing in a T-chart	Write a short paragraph explaining the difference between a networked OS and a distributed OS.	CO1
4	I	UNIX/LINUX Architecture	Understand the layered architecture of UNIX/LINUX.	Live Demo: Show the basic file system hierarchy in a Linux terminal (ls /).	Draw a simplified diagram of the UNIX architecture (Kernel, Shell, Utilities).	CO2
5	I	Kernel, Services, and System Calls	Explain the role of the Kernel. Define system calls and their purpose.	Trace the steps of a simple read() system call from a user program to the OS.	Find and list 3 common system calls and their functions (e.g., fork(), exec(), open()).	CO2
6	I	System Programs	Differentiate between system calls and system programs (utilities).	Demo: Use common system programs (ls, cp, grep) in a Linux terminal	What is a "shell"? Explain its role as a system program.	CO1
UNIT-2:- Process Management(7 hours)						

7	II	Process Management: Process Concepts (1/2)	Define a "process". Explain the Process Control Block (PCB).	Diagram: Draw the components of a Process Control Block (PCB).	List and describe 5 fields found in a typical PCB.	CO3
8	II	Process Management: Process Concepts (2/2)	Describe the different process states (new, ready, running, waiting, terminated).	Draw the process state transition diagram and discuss the triggers for each transition.	What event could cause a process to move from "running" to "waiting"? From "running" to "ready"?	CO3
9	II	Operations on Processes	Understand process creation (fork()) and termination (exit()).	Live Demo: Write a simple C/Python program demonstrating process creation (fork() or multiprocessing).	Explain the difference between fork() and exec() system calls.	CO3
10	II	Inter-Process Communication (IPC)	Explain the need for IPC. Describe shared memory and message passing models.	Analogy: Compare shared memory (a shared whiteboard) vs. message passing (sending letters).	What is the "producer-consumer" problem? How can IPC help solve it?	CO3
11	II	Process Scheduling (1/2)	Understand the concept of CPU scheduling. Define scheduling criteria (CPU utilization, throughput, etc.).	Discussion: "What makes a scheduler 'good'?" Brainstorm criteria.	Define "preemptive" and "non-preemptive" scheduling. Give an example of each.	CO3
12	II	Process Scheduling (2/2	Explain FCFS, SJF, and Priority scheduling algorithms.	Walkthrough: Manually calculate average wait time for a set of processes using FCFS and SJF.	Solve a given process set for average turnaround time using Priority scheduling.	CO3
13	II	Multi-threaded Programming	Define a "thread". Compare and contrast processes and threads	List pros and cons of multi-threading (e.g., responsiveness, resource sharing).	What are the benefits of using threads over multiple processes for a web server?	CO3
UNIT-3:- Memory management:(6 hours)						
14	III	Memory Management: Allocation (1/2)	Explain the need for memory management. Describe logical vs. physical address space.	Analogy: Logical address (a name) vs. Physical address (a street address).	What is the role of the Memory Management Unit (MMU)?	CO3

15	III	Memory Management: Allocation (2/2)	Explain contiguous memory allocation (fixed and variable partitions). Discuss fragmentation (internal and external).	Diagram: Show how external fragmentation occurs after several process allocations and deallocations.	Explain the difference between internal and external fragmentation. Which one occurs in fixed partitioning	CO3
16	III	Swapping	Define swapping and its purpose.	Diagram: Show the process of "swapping out" and "swapping in" a process between main memory and disk.	What is the major cost associated with swapping? (Hint: disk I/O).	CO3
17	III	Paging	Explain the concept of paging (frames and pages). Describe how a page table is used for address translation.	Step-by-step trace of a logical-to-physical address translation using a simple page table.	Given a logical address, page size, and page table, calculate the corresponding physical address.	CO3
18	III	Segmentation	Explain the concept of segmentation as a user's view of memory.	Compare & Contrast: Paging vs. Segmentation. Why might a programmer prefer segmentation?	How does segmentation aid in protection and sharing of code?	CO3
19	III	Virtual Memory & Various Faults	Define virtual memory. Explain demand paging and page faults.	Trace the steps of handling a page fault.	What is "thrashing"? How can it be detected and prevented?	CO3
UNIT 4:- File management(10 hours)						
20	IV	File Management: Concept of a File	Define a "file" and its attributes (name, type, size, etc.). Describe common file operations.	Activity: Have students list all the attributes they can see in their computer's "File Properties" dialog.	List and describe 6 common file operations (e.g., create, read, write, delete).	CO3
21	IV	Access Methods	Explain sequential access and direct (random) access.	Discussion: For each application, state the best access method 1. Music player 2. Database query 3. Text editor	What is an "index"? How does it help with file access?	CO3
22	IV	Directory Structure (1/2)	Explain the purpose of a directory. Describe single-level and two-level directory structures.	Diagram: Draw the structure of a single-level and two-level directory. Discuss limitations.	What is the main problem with a single-level directory structure?	CO3

23	IV	Directory Structure (2/2)	Describe tree-structured and acyclic-graph directories.	Demo: Navigate a tree-structured directory in the Linux/Windows terminal.	How does an acyclic-graph structure allow for file sharing? What is a "link" or "shortcut"?	CO3
24	IV	File System Mounting	Explain the concept of "mounting" a file system.	Demo: Show the mount command in Linux or the "Disk Management" tool in Windows.	What is a "mount point"? What happens when you mount a USB drive?	CO3
25	IV	Filesharing and Protection	Discuss the need for file sharing and protection. Explain access control lists (ACLs).	Case Study: Analyze file permissions in UNIX (rwx). Set up permissions for a shared project folder.	What do the r, w, and x permissions mean for a file? What do they mean for a directory?	CO3
26	IV	File System Structure & Implementation	Describe the layered structure of a file system. Explain on-disk (e.g., boot block, super block) structures.	Diagram: Draw a simplified layout of a file system on disk.	What is the "superblock"? What information does it contain?	CO3
27	IV	Directory Implementation	Explain linear list and hash table implementations for directories.	Compare & Contrast: Pros and cons of linear list vs. hash table for directory lookups.	Why is deleting a file from a linear list directory slow?	CO3
28	IV	Free Space Management	Describe bitmap (bit vector) and linked list methods for managing free space.	Walkthrough: Show how a bitmap and a free list change as files are allocated and deleted.	What are the pros and cons of the bitmap vs. the free list approach?	CO3
29	IV	Efficiency and Performance	Discuss ways to improve file system performance (e.g., caching, defragmentation).	Discussion: "Why does my computer get slower over time?" Relate to fragmentation.	What is "caching"? How does a disk cache improve performance?	CO3
30	IV	Different Types of File Systems	Briefly introduce common file systems (e.g., FAT32, NTFS, ext4).	Research: Students pick one file system (NTFS or ext4) and list 2-3 of its key features.	What is "journaling" in a file system (like NTFS or ext4)?	CO3
UNIT-5 - I/O System:(6 hours)						
31	V	I/O System: Mass Storage Overview	Describe the hierarchy of storage (registers, cache, main memory, disk).	Draw the storage hierarchy pyramid, labeling speed, cost, and volatility.	Why do we need a storage hierarchy? Why not just have one large, fast memory?	CO4
32	V	Disk Structure & Attachment	Explain the physical structure of an HDD (platter, track, sector). Describe disk attachment (SATA, USB).	Video: Show a short video of a hard disk drive in operation.	Define seek time, rotational latency, and transfer time.	CO4

33	V	Disk Scheduling Algorithms (1/2)	Explain the need for disk scheduling. Describe FCFS and SSTF algorithms.	Walkthrough: Calculate total head movement for a list of track requests using FCFS and SSTF.	What is the main problem with the SSTF (Shortest Seek Time First) algorithm? (Hint: starvation).	CO4
34	V	Disk Scheduling Algorithms (2/2)	Describe SCAN, C-SCAN, and LOOK algorithms.	Walkthrough: Calculate total head movement for the same list of requests using SCAN and C-SCAN.	Solve a new disk scheduling problem using FCFS, SSTF, SCAN, and C-SCAN.	CO4
35	V	Swap Space Management	Explain how swap space is used by the OS. Discuss swap space location and management.	Discussion: "Where should swap space be? On its own partition or in a file?" Pros and cons.	How does swap space management relate to virtual memory?	CO4
36	V	RAID Types	Explain the concept of RAID. Describe RAID 0 (striping), RAID 1 (mirroring), and RAID 5 (parity).	Diagram: Draw the disk layouts for RAID 0, 1, and 5. Discuss pros/cons (speed, redundancy).	Which RAID level provides the best redundancy? Which provides the best speed? Which provides a balance?	CO4
UNIT-6 Dead Locks(5 Hours)						
37	VI	Dead Locks: Concept & Resources	Define "deadlock". dentify the four necessary conditions for deadlock.	Analogy: Use the "Dining Philosophers" problem or a traffic gridlock analogy to explain deadlock.	List and explain the four necessary conditions for deadlock (Mutual Exclusion, Hold & Wait, No Preemption, Circular Wait).	CO5
38	VI	Deadlock Prevention (1/2)	Explain methods for preventing deadlock by breaking one of the four conditions.	Brainstorm: How could we break the "Circular Wait" condition? (e.g., resource ordering).	For each of the four conditions, propose one strategy to prevent it.	CO5
39	VI	Deadlock Prevention (2/2): Banker's Algorithm	Explain the concept of deadlock avoidance. Introduce the Banker's Algorithm (safety algorithm).	High-Level Walkthrough: Step through a simple state (Available, Max, Allocation) to see if it is "safe".	What is the difference between "deadlock prevention" and "deadlock avoidance"?	CO5
40	VI	Banker's Algorithm & Safety Algorithm	Detailed numerical example of the Banker's Algorithm.	Problem Solving: Given a system state, students apply the Safety Algorithm to determine if it's safe.	Solve a new Banker's Algorithm problem.	CO5
41	VI	The Ostrich Algorithm	Explain the "Ostrich Algorithm" (ignoring the problem).	Discussion: "When is it appropriate to use the Ostrich Algorithm?" (e.g., if deadlocks are very rare).	Why do many common desktop OSes (like Windows and Linux) primarily use the Ostrich Algorithm?	CO5

42	VI	Deadlock Detection and Recovery	Explain how to detect a deadlock (e.g., wait-for graph). Describe methods for recovery.	Diagram: Draw a wait-for graph from a given resource allocation state to detect a cycle.	List two ways to recover from a deadlock once it is detected. What are the pros and cons of each?	CO5
UNIT-7- OS Security(3 hours)						
43	VII	OS Security: Authentication	Define authentication. Describe methods: passwords, biometrics, two-factor authentication (2FA).	Discussion: "What makes a 'strong' password?" "Why is 2FA important?"	What is "multi-factor authentication"? Give an example.	CO6
44	VII	Access Control & Access Rights	Define access control. Explain the concept of an Access Control List (ACL) vs. capability lists.	Case Study: Set up permissions for three users (Alice, Bob, Charlie) on two files (ProjectA, ProjectB).	What is the "Principle of Least Privilege"? Why is it important for security?	CO6
45	VII	System Logs & Review	Explain the purpose of system logs for security and auditing. Course Review.	Demo: Show a snippet of a system log (/var/log/syslog in Linux) and point out key information.	Write a short paragraph summarizing the 7 main topics covered in this course.	CO6

22.12.25
Signature of Teacher

22.12.2025
Signature of HOD