GANPAT UNIVERSITY

FACULTY OF DIPLOMA ENGINEERING

Programme	Diploma in Computer Engineering/ Information Technology				
Semester	III		Version	1.0.0.0	
Effective from Academic Year		2026-27	Effective for the batch Admitted in		JULY 2025
Course Code	1CEIT3102	Course Name	Operating System		

I. TEACHING-LEARNING AND ASSESSMENT SCHEME

Course Type	Course code	Course Title	Teaching & Learning Scheme									Examination Scheme							
			Credit				Actual Contact Hrs/week			SLH	Total Learning Hrs/Week	TH			PR			SLA	Total
			CL	TL	LL	Total	CL	TL	LL			CE	SEE	Total	CE	SEE	Total		
DSC	1CEIT3102	OPERATING SYSTEM	3	-	1	4	3	-	2	2	7	40	60	100	30	20	50	20	170

Abbreviation:	**CL-** Classroom Learning	**TL -** Tutorial Learning	**LL -** Laboratory Learning
	SLH - Self Learning Hours	**SLA -** Self Learning Assessment	**CE -** Continuous Evaluation
	SEE – Semester End Examination		

II. PRE-REQUISITES

Basic knowledge of computer fundamentals and programming concepts is desirable.

III. INDUSTRY /EMPLOYER EXPECTED OUTCOMES

- Demonstrate the ability to use the Linux Command Line Interface (CLI) for basic file management, directory navigation, and program execution.
- Identify high CPU or RAM usage using standard system tools (such as Task Manager or top) to determine the causes of slow system performance.
- Illustrate how an operating system executes multiple programs simultaneously and describe the fundamental reasons applications may freeze or become unresponsive.
- Execute foundational knowledge of how memory (RAM) and storage limitations affect overall computer performance.
- Implement best practices for system security, including setting strong user passwords and managing basic file read/write permissions to protect personal data.

IV. COURSE LEARNING OUTCOMES

At the end of the course, students will be able to achieve the following course learning outcomes:

CO1. Explain the architectural evolution and structure of Operating Systems, distinguishing between kernel modes, system calls, and modern environments like Mobile and Cloud OS.

CO2. Apply CPU scheduling algorithms such as FCFS, SJF, Priority, and Round Robin to calculate waiting time and turnaround time, comparing their efficiency in different system environments.

CO3. Analyse inter-process communication and concurrency problems to implement deadlock prevention and avoidance strategies, including the Banker's Algorithm.

CO4. Compare memory management techniques, including Paging, Segmentation, and Page Replacement algorithms, to resolve fragmentation issues and optimize effective memory access time.

CO5. Utilize file system structures, disk scheduling algorithms, and basic Linux commands to manage secondary storage efficiently and configure file security permissions.

V. THEORY LEARNING OUTCOMES AND ALIGNED COURSE CONTENT:

Name of Unit	Theory Learning outcomes (TLO's) aligned to CO's	Learning Content mapped with Theory Learning outcomes (TLO's)&CO's	Marks	Hours
Unit-1 **Introduction**	**TLO 1.1** Outline the evolution of Operating Systems, their goals, and the services they provide to users.	**1.1** Operating System services, definitions timeline of OS	**8**	**7**

to Operating Systems	**TLO 1.2** Illustrate the system boot process and distinguish between Monolithic and Layered structural approaches. **TLO 1.3** Demonstrate the mechanism of System Calls and the transition between User Mode and Kernel Mode. **TLO 1.4** Describe the architecture of the Linux Operating System and the significance of the Open-Source model. **TLO 1.5** Compare the features of Mobile and Cloud Operating Systems with traditional Desktop environments. **TLO 1.6** Role of operating systems in cloud and edge computing (overview)	evolution (Batch to Time-Sharing Systems). **1.2** Flowchart of the Booting Process (BIOS → Kernel). **1.3** Diagram of User/Kernel transition via System Calls (open, read, write). **1.4** Basic Linux Architecture diagram (Kernel vs. Shell). **1.5** Comparison table: Mobile (Android/iOS) versus Desktop OS. **1.6** Cloud OS: conceptual overview. Virtualization and containerization: concept and use cases		
Unit-2 **Process Management & Scheduling**	**TLO 2.1** Illustrate the Process Concept, including the Process Control Block (PCB) and the transitions between Process States. **TLO 2.2** Differentiate between Process and Thread, and describe Single vs. Multi-threaded environments. **TLO 2.3** Define Scheduling Basics, including Pre-emptive vs. Non-pre-emptive modes, Scheduling Queues, and Criteria (Burst/Arrival Time). **TLO 2.4** Implement the First-Come, First-Served (FCFS) algorithm to calculate turnaround and waiting times. **TLO 2.5** Implement the Shortest Job First (SJF) algorithm for non-pre-emptive scheduling tasks. **TLO 2.6** Demonstrate the Round Robin scheduling algorithm using Time Quantum and simulate context switching. **TLO 2.7** Implement the Shortest Remaining Time Next (SRTN) algorithm for pre-emptive scheduling tasks. **TLO 2.8** Analyse the concept of Context Switching, including hardware support and performance overhead.	**2.1** Process Life Cycle, Process States (New, Ready, Run, Wait, Terminate), and Process Control Block (PCB). **2.2** Overview of Threads, distinction between Process & Thread, and Multithreading models. **2.3** Scheduling Criteria, Scheduling Queues, and basic definitions (Burst Time, Arrival Time). **2.4** Scheduling Algorithm: First-Come, First-Served (FCFS) logic and Gantt Charts. **2.5** Scheduling Algorithm: Shortest Job First (SJF) - Non-pre-emptive mode. **2.6** Scheduling Algorithm: Round Robin (RR) with Time Quantum. **2.7** Scheduling Algorithm: Shortest Remaining Time Next (SRTN) - Pre-emptive mode. **2.8** Context Switching: Definition, Hardware support, and Overhead analysis. **2.9** Comparison of Scheduling	**14**	**10**

	TLO 2.9 Compare various process scheduling algorithms based on fairness, efficiency, and response time.	algorithms (FCFS, SJF, RR, SRTN).		
Unit-3 **Synchronization & Deadlocks**	**TLO 3.1** Compare Inter-Process Communication (IPC) mechanisms, specifically Shared Memory versus Message Passing. **TLO 3.2** Analyse concurrency issues, including Race Conditions and the Critical Section concept **TLO 3.3** Solve the classic Producer-Consumer problem using synchronization primitives. **TLO 3.4** Define Deadlocks and identify the four necessary conditions required for a deadlock to occur. **TLO 3.5** Solve Deadlock Prevention strategies by attacking the necessary conditions (e.g., Mutual Exclusion). **TLO 3.6** Analyse Deadlock Avoidance techniques using state modelling to distinguish Safe vs. Unsafe states. **TLO 3.7** Comparison of mutex vs semaphore	**3.1** Shared Memory architecture. **3.2** Critical Section. **3.3** Concept of "Buffer" management concepts. **3.4** Conditions for Deadlock. **3.5** Mutual Exclusion **3.6** Semaphores. **3.7** Overview of mutex vs semaphore	**12**	**8**
Unit-4 **Memory Management**	**TLO 4.1** Outline the Memory Hierarchy and analyse the trade-off between speed, size, and cost. **TLO 4.2** Apply Contiguous Memory Allocation techniques, distinguishing between partitioning methods and solving fragmentation **TLO 4.3** Differentiate between Logical and Physical addresses and outline the concept of address mapping. **TLO 4.4** Analyse the Paging hardware mechanism, including page tables and address translation. **TLO 4.5** Illustrate the concept of Segmentation and its logical view of memory (Text, Data, Stack). **TLO 4.6** Implement Page Replacement Algorithms (FIFO, LRU, Optimal) to calculate page faults. **TLO 4.7** Analyse Thrashing, its causes, and its impact on CPU	**4.1** The Memory Pyramid concept; Swapping. **4.2** Contiguous Allocation: Fixed vs. Variable partitions; Internal vs. External Fragmentation and Compaction **4.3** Logical and physical address map; Memory relocation and protection mechanisms. **4.4** Paging: Address translation using the basic method of paging. **4.5** Non-Contiguous Allocation: Overview of Segmentation. **4.6** Page replacement algorithms: FIFO, LRU, Optimal.	**14**	**10**

	utilization using the degree of multiprogramming.	**4.7** Graph: CPU Utilization vs. Degree of Multiprogramming.		
Unit-5 **File Systems & I/O**	**TLO 5.1** Outline File System concepts, including file attributes, operations, and various file types. **TLO 5.2** Analyse Directory Structures, comparing Single-level and Tree-structured organizations. **TLO 5.3** Apply Contiguous and Linked Allocation methods to manage file storage. **TLO 5.4** Implement Indexed Allocation method to solve external fragmentation and file size limitations. **TLO 5.5** Implement Disk Scheduling Algorithms (FCFS, SSTF, SCAN, C-SCAN) to optimize seek time. **TLO 5.6** Illustrate I/O Hardware principles, distinguishing between Polling, Interrupts, and DMA. **TLO 5.7** Apply basic file system security mechanisms. **TLO 5.8** Create and Execute basic Linux commands for file and directory management.	**5.1** File Concepts: Attributes, Operations, and File Types. **5.2** Directory Structures: Single-level vs. Tree-structured. **5.3** Allocation Methods I: Contiguous and Linked Allocation. **5.4** Allocation Methods II: Indexed Allocation. **5.5** Disk Management: Disk Structure and Scheduling Algorithms (FCFS, SSTF, SCAN, C-SCAN). **5.6** I/O Hardware: Polling, Interrupts, and Direct Memory Access (DMA). **5.7** File System Security: User Authentication, Permissions & Access Control. **5.8** Introduction to the Linux, Shell and basic commands.	**12**	**10**

VI. LABORATORY LEARNING OUTCOME AND ALIGNED PRACTICAL			
SR. NO	**PRACTICAL/LABORATORY LEARNING OUTCOME(LLO)**	**PRACTICAL TITLES**	**RELEVANT COs**
1	**LLO 1.1** Identify different types of OS	Compare Windows, Linux, and Mobile OS.	CO1
2	**LLO 2.1** Implement Non-Pre-emptive CPU Scheduling Algorithms with example	Solve examples with SJF, FCFS and Draw Gantt chart.	CO2
3	**LLO 2.2** Identify Pre-emptive Scheduling & Context Switching Algorithms with example	Solve examples with round robin and SRTN algorithm and Draw Gantt chart.	CO2
4	**LLO 3.1** Implement various memory allocation algorithms such as First Fit, Best Fit, and Worst Fit by solving examples	Solve examples using following algorithms: First fit, best fit, Worst fit	CO3
5	**LLO 3.2** Apply and analyse Page Replacement Algorithms	Solve examples using FIFO Page replacement algorithms	CO3
6	**LLO 3.3** Apply and analyse Page Replacement Algorithms	Solve examples using LRU Page replacement algorithms	CO3
7	**LLO 4.1** Interpret Disk scheduling algorithm with the help of examples.	Solve examples using FCFS and SSTF Disk Scheduling Algorithms	CO4

8	**LLO 4.2** Interpret Disk scheduling algorithm with the help of examples.	Solve examples using SCAN and CSCAN Disk Scheduling Algorithms	CO4
9	**LLO 5.1** To test and execute basic UNIX command	Test and run basic UNIX commands.	CO5
10	**LLO 5.2** To test and execute advanced UNIX command	Test and run Advanced UNIX commands.	CO5
11	**LLO 5.3** To test and use file editing commands using editors	Test commands related with File editing with Vi, Vim, gedit.	CO5
12	**LLO 5.4** Create Basic Shell Script	Create a shell script to read from command line and print "Hello".	CO5
13	**LLO 5.6** Create File handling Shell Scripts	Create a Shell script to read and display content of a file. And append content of one file to another	CO5
14	**LLO 5.7** Implement Shell Scripts using string	Create a Shell script to accept a string in lower case letters from a user, & convert to upper case letters.	CO5
15	**LLO 5.8** Implement Shell Script using Operators, if…. else, switch	Create a shell script to implement a calculator using operators, if–else, and switch.	CO5

VII. SUGGESTED MICRO PROJECT/ASSIGNMENTS/ACTIVITIES FOR SELF LEARNING/SKILL DEVELOPMENT (SELF LEARNING)

Micro Projects (Mini Applications / Use Cases)

- Programming & Simulation Projects (These projects turn the numerical problems from class into working software.)
  Skills: C, C++, or Python
- Create a program that simulates CPU scheduling algorithms.
  Skills: C++, or Python
- Calculate the physical movement of a hard disk head.
  Skills: Understanding FCFS, SSTF, and SCAN Algorithms
- Compare the efficiency of FIFO vs. LRU.
  Skills: Understanding of Algorithms.
- A tool to teach and verify file security.
  Skills: Basic I/O operations and Shell Scripting.

Self-Learning / Skill Building Activities

1. Create a Bootable USB Drive using a lightweight Linux distribution (e.g., Ubuntu or Puppy Linux) and a tool like Rufus or Etcher.
2. Visualize the abstract concept of a PCB (Process Control Block) through task manager.
3. Students manually draw Gantt Charts for FCFS, SJF, and Round Robin on graph paper.
4. Students navigate to their OS settings (e.g., Windows "Advanced System Settings" → Performance → Virtual Memory).
5. Download a monitoring tool (e.g., Process Monitor for Windows or use strace on Linux).
   Goal: Observe the invisible communication between software and the Kernel.
6. Demonstrate Inter-Process Communication (IPC) using the command line.
7. Implement logical organization to file systems example.

VIII. LIST OF INSTRUMENTS / EQUIPMENT / TRAINER BOARD

1	Computer with latest configuration running Windows or Linux operating system.
2	Compiler: gcc (GNU Compiler Collection).

3	strace (Linux) for kernel monitoring.
4	Text Editors: Vi, Vim, gedit.

IX. LIST OF REFERENCE BOOKS

Sr.No	Title	Author	Publication
1	Operating System Concepts, 9th Edition	Abraham Silberschatz, Peter B Galvin, Gerg Gagne	Wiley
2	Unix Concepts and Application, 4th Edition	Sumitabha Das	McGraw-Hill Education (MGH)
3	Modern Operating Systems, 3rd Edition	Andrew Tanenbaum, Herbert Bos	Pearson
4	Operating Systems: 2nd Edition	Milan Milenkovic	McGraw-Hill Education (MGH),
5	Linux –Application and Administration	Ashok Kumar Harnal	Tata McGraw-Hill (TMH)

X. LINK OF LEARNING WEB RESOURCE

1	https://boonsuen.com/process-scheduling-solver
2	https://codepen.io/faso/pen/zqWGQW
3	http://cpuburst.com/ganttcharts.html
4	https://nptel.ac.in/courses/106106144
5	https://www.tutorialspoint.com
6	https://nptel.ac.in/courses/106102132

XI. SUGGESTED WEIGHTAGE TO LEARNING EFFORTS & ASSESSMENT PURPOSE

Unit	Unit Title	Aligned COs	Learning Hours	R-Level	U-Level	A-Level	Total Marks
1	Introduction of Operating System	CO1	7	4	3	1	8
2	Process Management & Scheduling	CO2	10	4	5	5	14
3	Synchronization & Deadlocks	CO3	8	4	4	4	12
4	Memory Management	CO4	10	5	5	4	14
5	File Systems & I/O	CO5	10	4	4	4	12
	Grand Total		**45**	**21**	**21**	**18**	**60**

XII. COs AND POs AND PSOs MAPPING

Course Outcome (Cos)	Programme Outcomes (POs)							Programme Specific Outcomes (PSOs)		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PSO1	PSO2	PSO3
CO1	3	1	0	0	0	2	0	1	3	0
CO2	3	3	1	0	0	1	0	2	2	0

CO3	3	3	2	1	0	1	0	3	1	0
CO4	3	3	2	1	0	1	0	2	2	0
CO5	2	2	2	3	1	2	1	2	3	1

Legends: - *3-High; 2-Moderate/Medium; 1-Slight/Low; 0-None*