GANPAT UNIVERSITY

FACULTY OF ENGINEERING & TECHNOLOGY

Programme	Bachelor of Technology	Branch/Spec.	Computer Science & Engineering (CSE/BDA/CS/AI&ML)
Semester	IV	Version	1.0.1.1
Effective from Academic Year	2026-27	Effective for the batch Admitted in	June 2025
Subject code	2CSE402	Subject Name	OPERATING SYSTEM

Teaching scheme						Examination scheme (Marks)			
(Per week)	Lecture (DT)		Practical (Lab.)		Total		CE	SEE	Total
	L	TU	P	TW					
Credit	3	0	1	0	4	Theory	40	60	100
Hours	3	0	2	0	5	Practical	30	20	50

Pre-requisites:

Fundamentals of Computer Organization & Architecture, basic UNIX commands, Data Structures and Algorithms

Learning Outcome:

After successful completion of this course, students will be able to:

- Understand the fundamentals of Process, Memory, IO and File Managements and Operating system architectures
- Learn the various System calls and Design of Unix Operating systems as case study.
- Develop the process, memory , file and IO related programs.
- Implement various algorithms that the OS executes to perform resource management related activities.

Theory syllabus

Unit	Content	Hrs
1	**Linux Kernel - Designer Perspective** Concept of FOSS - Free & Open Source Software, Comparative analytics of various modern OS, Introduction to UNIX system kernel, major building blocks, Dual Mode of nature, Processing environment, OS services, System calls, OS as a Resource Manager, Kernel Data Structures, introduction to various subsystems, Micro Kernel	8
2	**Buffer Caching** Need of buffers, access buffers, Buffer headers, buffer Pools, retrieval of buffers for reading and writing, Reading and Writing to Disk blocks	4
3	**File Systems** Internal Representation of Files, inodes, accessing inodes, structure of files and directories, Concept of Super Blocks, organization and access of files, directories, file sharing, record blocking, secondary storage management, system calls for the file systems	8
4	**Process Management** Programs Vs Process Vs Thread, Process State Transition diagram, CPU switching from process to process, Process Control Blocks and attributes, System calls for process management, Types of Schedules - Job Scheduling Vs CPU scheduling, Short term , medium term and Long term schedulers, Process Scheduling Algorithms and Examples, Process synchronization issue	8

5	**Inter Process Communications** Principles of concurrency,signals, shared memory, pipe, mutual exclusion, semaphore, monitors, message passing, reader's writers problem, deadlock, prevention, avoidance, detection, dining philosophers problem.	6
6	**Memory Management** System Memory Layout, Logical Vs Physical Address Spaces, Memory operations, partitioning, paging, segmentation, virtual memory, swapping, hardware and control structure	8
7	**Input Output System** Device Drivers Interfaces, Drivers entry points, Algorithms for accessing devices, Memory mapped I/O, system calls for I/O, Device Interrupts and Interrupts handlers	3

Self-studies Topics

LINUX kernel module programming basics

Practical content

Practicals will be based on Make and GDB utilities, Implementation of file,memory,IO and process related system calls, Process Synchronization problems and Algorithms to solve various scenarios, Implementation of programs for various concurrency problems using semaphores and without semaphores, Implementation of programs for multithreading, Use of Linux Pipe for redirection and IPC

Text Books

1	Design of the Unix Operating System By Maurice Bach, PHI
2	Operating System Internals and Design Principles by William Stallings - 9th Edition, Pearson publication

Reference Books

1	Operating System Concepts by Silberschatz, Peterson and Galvin
2	Linux Kernel Development by Robert Love, 3rd Edition , Addison-Wesley Publication
3	Linux System Programming by Robert Love, O Reilly, Second Edition

Course Outcomes:

COs	Description
CO1	Understand fundamentals of Linux Kernel Designs.
CO2	Implement various system calls related to processes, file, I/O, Memory operations.
CO3	Modify the linux kernel as per business need & recompile kernel modules using make utility.
CO4	Implement various algorithms that the OS executes to perform resource management related activities.

Mapping of CO and PO:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	0	0	3	0	0	0	2	0	0	2
CO2	3	3	3	0	3	0	0	0	0	0	0	1
CO3	3	2	0	0	2	0	0	0	0	0	2	0
CO4	3	2	2	0	3	0	0	0	0	0	2	0