



MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

COURSE CONTENT

OPERATING SYSTEMS LAB								
IV Semester: CSD / CSE / CSM								
Course Code	Category	Hours/ Week			Credits	Maximum Marks		
		L	T	P		CIA	SEE	Total
2540582	CORE	0	0	2	1	40	60	100
Contact Classes: Nil		Tutorial Classes: Nil		Practical Classes: 30		Total Classes: 30		
Prerequisites: Programming for Problem Solving, Computer Organization and Architecture								

Course Overview:

This course offers hands-on experience with core Operating System concepts through C programming on UNIX/Linux platforms. It focuses on process and memory management, CPU scheduling, deadlock handling, synchronization, file systems, and inter-process communication using practical simulations and system calls. The course bridges theory with real-world system-level programming, equipping students with strong problem-solving skills and a solid foundation for advanced studies and careers in operating systems and systems programming.

Course Objectives:

1. To provide an understanding of the design aspects of operating system concepts through simulation.
2. Introduce basic Unix commands, system call interface for process management, interprocess communication and I/O in Unix.

Course Outcomes: After Completion of the Course, Students should be able to

1. Apply various CPU scheduling algorithms and analyze their performance using C programming.
2. Develop C programs utilizing UNIX/Linux system calls to perform file operations, process control, and I/O management.
3. Simulate deadlock avoidance and resource allocation strategies such as the Bankers Algorithm effectively.
4. Implement inter-process communication and synchronization mechanisms including semaphores, pipes, FIFOs, message queues, and shared memory.
5. Demonstrate memory management and page replacement techniques through C programming, analyzing efficiency and system behavior.

List of Experiments:

1. Write C programs to simulate the following CPU Scheduling algorithms a) FCFS b) SJF c) Round Robin d) priority
2. Write programs using the I/O system calls of UNIX/LINUX operating system (open, read, write, close, lseek, stat, fork, exit)
3. Write a C program to simulate Bankers Algorithm for Deadlock Avoidance.
4. Write a C program to implement the Producer – Consumer problem using semaphores using UNIX/LINUX system calls.
5. Write C programs to illustrate the following IPC mechanisms a) Pipes b) FIFOs c) Message Queues d) Shared Memory
6. Write C programs to simulate the following memory management techniques a) Paging b)

Segmentation

7. Write C programs to simulate Page replacement policies a) FCFS b) LRU c) Optimal

TEXT BOOKS:

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley.
2. Advanced programming in the Unix environment, W. R. Stevens, Pearson education.

REFERENCE BOOKS:

1. Operating Systems – Internals and Design Principles, William Stallings, Fifth Edition–2005, Pearson Education/PHI.
2. Operating System - A Design Approach-Crowley, TMH.
3. Modern Operating Systems, Andrew S Tanenbaum, 2nd edition, Pearson/PHI.
4. UNIX Programming Environment, Kernighan and Pike, PHI/Pearson Education.
5. UNIX Internals: The New Frontiers, U. Vahalia, Pearson Education.

ELECTRONIC RESOURCES:

1. <https://khushalip.github.io/OS-lab/index.html>
2. <https://www.scribd.com/document/791182025/5-6075789087724276027>
3. <https://studyglance.in/labprograms/r22osdisplay.php?url1=r22os/exp1.html>

MATERIALS ONLINE:

1. Lab Manual
2. Open-ended experiments