Syllabus

| Course code | Course Name | Hours/week | | | Credit | Max. Marks |
|---|---|---|---|---|---|---|
| | **Operating System** | **L** | **T** | **P** | **C** | **100** |
| | | 3 | 0 | 1 | 4 | |
| **Pre-requisite** | **NA** | | | | | |
| **Evaluation Scheme** | **Theory** | | | | **Hours** | **Marks** |
| | **End Semester Examination** | | | | 2-2.5 | 50 |
| | **Internal Assessment** | | | | - | 50 |
| | 1) Mid Semester Examination (Theory)/Continuous Assessment (Practical) | | | | 1.5 | 20 |
| | 2) Attendance | | | | - | 05 |
| | 3) Assignment/Seminar/Activity/ School Approved Certificate Course/Events/Professional Societies Membership and Participants | | | | - | 15 |
| **UNIT-I** | **INTRODUCTION** | | | | | **9** |
| Introduction of Operating System, Simple Batch Systems, Multi-programmed Batches systems, Time Sharing Systems, Personal-computer systems, Parallel systems, Distributed Systems, Real-Time Systems, OS – A Resource Manager. Virtual Memory: Demand Paging, Page Replacement, Page-replacement Algorithms, Performance of Demand Paging, Thrashing, Demand Segmentation, and Overlay Concepts. | | | | | | |
| **UNIT-II** | **PROCESSES AND PROCESS SYNCHRONIZATION** | | | | | **9** |
| Introduction, Process states, process management, Interrupts, Inter process Communication Threads: Introduction, Thread states, Thread Operation, Threading Models. Scheduling levels, pre-emptive vs no pre-emptive scheduling, priorities, scheduling objective, scheduling criteria, scheduling algorithms, demand scheduling, real-time scheduling. Process Synchronization: Mutual exclusion, software solution to Mutual exclusion problem, hardware solution to Mutual exclusion problem, semaphores, Critical section problems. Case study on Dining philosopher problem, Barbershop problem etc. | | | | | | |
| **UNIT-III** | **MEMORY MANAGEMENT, DEADLOCKS AND DEVICE MANAGEMENT** | | | | | **9** |
| Memory Organization & Management: Memory Organization, Memory Hierarchy, Memory Management Strategies, Contiguous versus non- Contiguous memory allocation, Partition Management Techniques, Logical versus Physical Address space, swapping, Paging, Segmentation, Segmentation with Paging. Deadlocks: examples of deadlock, resource concepts, necessary conditions for deadlock, deadlock solution, deadlock prevention, deadlock avoidance with Banker's algorithms, deadlock detection, deadlock recovery. | | | | | | |

| | | |
|---|---|---|
| Disk Scheduling Strategies, Rotational Optimization, System Consideration, Caching and Buffering. | | |
| **UNIT-IV** | **FILE SYSTEM** | **9** |
| Introduction, File Organization, Logical File System, Physical File System, File Allocation strategy, Free Space Management, File Access Control, Data Access Techniques, Data Integrity Protection, Case study on file system viz FAT32, NTFS, Ext2/Ext3 etc. | | |
| **UNIT-V** | **I/O SYSTEMS** | **9** |
| Kernel I/O Subsystem, Disk Structure, Disk Scheduling, Disk Management, Swap Space Management, RAID Structure, Case study on Linux System, Case study on Windows XP. | | |
| | Total hours | **45** periods |

**30** Period

**List of Experiments**

1. Perform a case study by installing and exploring various types of operating systems on a physical or logical (virtual) machine.
2. Write C programs to simulate Page Replacement Algorithms: FIFO, LRU.
3. Write C Program to implement optimal page replacement technique.
4. Write C programs to demonstrate various process related concepts.
5. Write C programs to simulate CPU scheduling algorithm : FCFS, SJF
6. Write C program to simulate the CPU scheduling algorithm round-robin.
7. Write a C program to simulate the following contiguous memory allocation techniques
   a) Worst-fit b) Best-fit c) First-fit
8. Write a C program to implement deadlock prevention technique
9. Write a C program to simulate disk scheduling algorithms.
10. Write a C programs to implement UNIX system calls and file management.
11. Write a C program to simulate the following file organization techniques:
    (a) Single level directory (b) Two level directory (c) Hierarchical
12. Write a C program to Implement Different File operations.
13. Write C programs to simulate implementation of Disk Scheduling Algorithms: FCFS, SSTF.
14. Case Study on Different Operating Systems: Linux (UNIX) OS, Windows OS. Etc.
15. Write C program to implement System Call in Operating System

Course Outcomes: At the end of the course, the students will be able to:

| COs | Statements | Bloom's Level |
|---|---|---|
| **CO1** | Understand the need for a multi-programmed operating system and explain the general structure of such systems. | **L2** |

| | | |
|---|---|---|
| **CO2** | Apply different process scheduling algorithms and memory management techniques to optimize system performance. | **L3** |
| **CO3** | Compare the process and thread abstractions, memory management and assess possible issues related to concurrent execution and Deadlock mechanism. | **L4** |
| **CO4** | Explain the File abstraction and system level I/O. | **L2** |
| **CO5** | Evaluate various disk scheduling algorithms for I/O systems. | **L5** |

**TEXTBOOK:**


1. Silberschatz and Galvin, "Operating System Concepts", Pearson, 5th Ed., 2001.
2. Deitel & Dietel, "Operating System", Pearson, 3rd Ed, 2011.
3. Madnick & Donovan, "Operating System", TMH,1st Ed, 2001.


**REFERENCES:**


1. Chauhan, "Principles of Operating Systems", Oxford Uni. Press, 2014.
2. Dhamdhere, "Operating Systems", Tata McGraw Hill, 3rd edition, 2012.
3. Modern Operating Systems (Fourth Edition) by Andrew S Tanenbaum, Prentice Hall India (2008)


## Course Outcomes with Program Outcomes

| | PO1 | PO2 | PO3 | PO4 | PO5 | PO6 | PO7 | PO8 | PO9 | PO10 | PO11 | PO12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **CO1** | 3 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| **CO2** | 3 | 3 | 3 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 2 |
| **CO3** | 3 | 2 | 3 | 2 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 2 |
| **CO4** | 2 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 2 |
| **CO5** | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 2 | 2 | 1 | 2 |