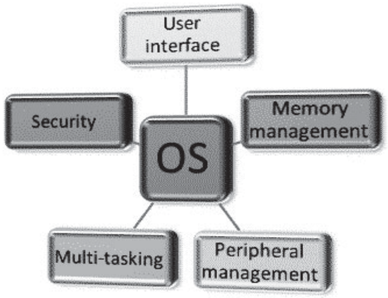

# 16CS304 OPERATING SYSTEMS

Hours Per Week :

| L | T | P | C |
|---|---|---|---|
| 3 | - | 2 | 4 |

Total Hours :

| L | T | P |
|---|---|---|
| 45 | - | 30 |

| WA/RA | SSH/HSH | CS | SA | S | BS |
|---|---|---|---|---|---|
| 8 | 35 | 4 | 7 | 4 | 5 |

## Course Description and Objectives:

This course focusses on how the operating system effectively manages the system resources. The objective of this course is to provide classical internal algorithms and structures of operating systems, including CPU scheduling, memory management and file management concepts.

## Course Outcomes:

The student will be able to:

- understand operating systems and analyze the process scheduling algorithms.
- gain knowledge on resource sharing among the processes.
- understand memory management during the process execution.
- understand the concepts of file management system.

## SKILLS:

- ✓ *Install / remove an operating system in a computer.*
- ✓ *Manage open source operating systems like ubuntu, fedora etc.*
- ✓ *processes scheduling and execution.*
- ✓ *Memory management.*

**UNIT - 1** **L-9**

**INTRODUCTION:** What Operating System do, Operating System structure; Process Concept- Overview, Process scheduling, Operations on process, Inter process communication; Process scheduling-Scheduling criteria, Scheduling algorithms, Multiple-Processor Scheduling; Case study-Process scheduling in Linux.

**UNIT - 2** **L-9**

**PROCESS SYNCHRONIZATION:** The Critical-Section problem, Peterson's solution, Synchronization hardware, Semaphores, Monitors, Classical problems of synchronization.

**UNIT - 3** **L- 9**

**DEADLOCKS:** Deadlock characterization, Methods of handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection and Recovery.

**UNIT - 4** **L-9**

**MEMORY MANAGEMENT:** Continuous memory allocation, Paging, Structure of the page table, Segmentation, Demand paging, Page replacement algorithms.

**SECONDARY-STORAGE STRUCTURE:** Overview of Mass-Storage structure, Disk structure, Disk scheduling.

**UNIT - 5** **L-9**

**FILE SYSTEMS:** File concept, Access methods, Directory structure, File system mounting, File sharing, Protection, File-System structure, File system implementation, Directory implementation, Allocation methods, Free space management.

## LABORATORY EXPERIMENTS

**Course Outcomes:**

The student will be able to:

- Understand the structure and organization of Operating Systems; including functionalities.
- Understand the basic concepts of process scheduling, Deadlock, Page replacement.
- In depth understanding of memory management & File system concepts.

**LIST OF EXPERIMENTS:** Total Hours: 30

1. Write programs using the following system calls of UNIX operating system: fork, exec, getpid, exit, wait, close.
2. Write programs using the I/O system calls of UNIX operating system. (open, read, write, etc)
3. Write C programs to simulate UNIX commands like ls, grep, cp.
4. Obtain the list of processes, their CPU burst times and arrival times through the keyboard. Display/print the Gantt chart for FCFS and SJF. For each of the scheduling policies, compute and print the average waiting time and average turnaround time.

**ACTIVITIES:**

- ○ *Identification and installation of various operating systems*
- ○ *Simulation and comparison of process scheduling.*
- ○ *Simulation of deadlock prevention and avoidance.*
- ○ *Identification of page replacement strategies.*
- ○ *Identification of different disk scheduling methodologies.*

5. Obtain the list of processes, their CPU burst times and arrival times through the keyboard. Display/print the Gantt chart for Priority and Round robin. For each of the scheduling policies, compute and print the average waiting time and average turnaround time.
6. Develop Application using Inter-Process-Communication (Using shared memory, pipes or message queues).
7. Implement the Producer-Consumer problem using semaphores (Using UNIX system calls)
8. Implementation of Banker's Algorithm to Deadlock Avoidance.
9. Implement some Memory management schemes like Paging and Segmentation.
10. Implement some Memory management schemes like First Fit, Best Fit & Worst Fit.
11. Implement any file allocation techniques (Contiguous, Linked or Indexed) calls.

**TEXT BOOK:**

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, "Operating System Concepts", 9th edition, John Wiley & Sons Inc, 2013.

**REFERENCE BOOKS:**

1. Richard. Stevens and Stephen A Rago, "Advanced Programming in the Unix Environment", 3rd edition, Addison-Wesley, 2013.
2. William Stallings, "Operating Systems: Internals and Design Principles", 6th edition, Prentice Hall, 2005.
3. Andrew S Tanenbaum ,"Modern Operating Systems", 3rd edition, Prentice India, 2007.