

## **Operating Systems**

**Course Title:** Operating Systems

**Course No:** CSC264

**Nature of the Course:** Theory + Lab

**Semester:** IV

**Full Marks:** 60+ 20+20

**Pass Marks:** 24+8+8

**Credit Hrs:** 3

**Course Description:** This course includes the basic concepts of operating system components. It consists of process management, deadlocks and process synchronization, memory management techniques, File system implementation, and I/O device management principles. It also includes case study on Linux operating system.

### **Course Objectives:**

- Describe need and role of operating system.
- Understand OS components such a scheduler, memory manager, file system handlers and I/O device managers.
- Analyze and criticize techniques used in OS components
- Demonstrate and simulate algorithms used in OS components
- Identify algorithms and techniques used in different components of Linux

### **Course Contents:**

#### **Unit 1: Operating System Overview (4 Hrs.)**

- 1.1. Definition, Two views of operating system, Evolution of operating system, Types of OS.
- 1.2. System Call, Handling System Calls, System Programs, Operating System Structures, The Shell, Open Source Operating Systems

#### **Unit 2: Process Management (10 Hrs.)**

- 2.1. Process vs Program, Multiprogramming, Process Model, Process States, Process Control Block.
- 2.2. Threads, Thread vs Process, User and Kernel Space Threads.
- 2.3. Inter Process Communication, Race Condition, Critical Section
- 2.4. Implementing Mutual Exclusion: Mutual Exclusion with Busy Waiting (Disabling Interrupts, Lock Variables, Strict Alteration, Peterson's Solution, Test and Set Lock), Sleep and Wakeup, Semaphore, Monitors, Message Passing,
- 2.5. Classical IPC problems: Producer Consumer, Sleeping Barber, Dining Philosopher Problem
- 2.6. Process Scheduling: Goals, Batch System Scheduling (First-Come First-Served, Shortest Job First, Shortest Remaining Time Next), Interactive System Scheduling (Round-Robin Scheduling, Priority Scheduling, Multiple Queues), Overview of Real Time System Scheduling

#### **Unit 3: Process Deadlocks (6 Hrs.)**

- 3.1. Introduction, Deadlock Characterization, Preemptable and Non-preemptable Resources, Resource – Allocation Graph, Conditions for Deadlock

- 3.2. Handling Deadlocks: Ostrich Algorithm, Deadlock prevention, Deadlock Avoidance, Deadlock Detection (For Single and Multiple Resource Instances), Recovery From Deadlock (Through Preemption and Rollback)

#### **Unit 4: Memory Management (8 Hrs.)**

- 4.1. Introduction, Monoprogramming vs. Multi-programming, Modelling Multiprogramming, Multiprogramming with fixed and variable partitions, Relocation and Protection.
- 4.2. Memory management (Bitmaps & Linked-list), Memory Allocation Strategies
- 4.3. Virtual memory: Paging, Page Table, Page Table Structure, Handling Page Faults, TLB's
- 4.4. Page Replacement Algorithms: FIFO, Second Chance, LRU, Optimal, LFU, Clock, WS-Clock, Concept of Locality of Reference, Belady's Anomaly
- 4.5. Segmentation: Need of Segmentation, its Drawbacks, Segmentation with Paging(MULTICS)

#### **Unit 5: File Management (6 Hrs.)**

- 5.1. File Overview: File Naming, File Structure, File Types, File Access, File Attributes, File Operations, Single Level, two Level and Hierarchical Directory Systems, File System Layout.
- 5.2. Implementing Files: Contiguous allocation, Linked List Allocation, Linked List Allocation using Table in Memory, Inodes.
- 5.3. Directory Operations, Path Names, Directory Implementation, Shared Files
- 5.4. Free Space Management: Bitmaps, Linked List

#### **Unit 6: Device Management (6 Hrs.)**

- 6.1. Classification of IO devices, Controllers, Memory Mapped IO, DMA Operation, Interrupts
- 6.2. Goals of IO Software, Handling IO(Programmed IO, Interrupt Driven IO, IO using DMA), IO Software Layers (Interrupt Handlers, Device Drivers)
- 6.3. Disk Structure, Disk Scheduling (FCFS, SSTF, SCAN, CSCAN, LOOK, CLOOK), Disk Formatting (Cylinder Skew, Interleaving, Error handling), RAID

#### **Unit 7: Linux Case Study (5 Hrs.)**

- 7.1 History, Kernel Modules, Process Management, Scheduling, Inter-process Communication, Memory Management, File System Management Approaches, Device Management Approaches.

#### **Laboratory Works:**

The laboratory work includes solving problems in operating system. The lab work should include at least;

- Learn basic Linux Commands
- Create process, threads and implement IPC techniques
- Simulate process Scheduling algorithms and deadlock detection algorithms
- Simulate page replacement algorithms
- Simulate free space management techniques and disk scheduling algorithms.

**Text Books:**

1. Modern Operating Systems: Andrew S. Tanenbaum, PHI Publication, Third edition, 2008

**Reference Books:**

1. Abraham Silberschatz, Peter Baer Galvin and Greg Gagne, “Operating System Concepts”, John Wiley & Sons (ASIA) Pvt. Ltd, Seventh edition, 2005.
2. Harvey M. Deitel, Paul J. Deitel, and David R. Choffnes, “Operating Systems, Prentice Hall, Third edition, 2003.