UCS303: OPERATING SYSTEMS

L	T	P	Cr
3	0	2	4.0

Course objective: Role and purpose of the operating system, Functionality of a typical operating system, managing atomic access to OS objects.

Operating System Principles: Structuring methods (monolithic, layered, modular, microkernel models), processes, and resources, Concepts of APIs, Device organization, interrupts: methods and implementations, Concept of user/system state and protection, transition to kernel mode.

Concurrency: Implementing synchronization primitives, Multiprocessor issues (spin locks, reentrancy).

Scheduling and Dispatch: Dispatching and context switching, Preemptive and non-preemptive scheduling, Schedulers and policies, Processes and threads.

Memory Management: Review of physical memory and memory management hardware, Working sets and thrashing, Caching, Paging and virtual memory, Virtual file systems.

File Systems: Files: data, metadata, operations, organization, buffering, sequential, non-sequential, Directories: contents and structure, Naming, searching, access, backups, Journaling and log-structured file systems.

Deadlock: Introduction, Analysis of conditions, Prevention & avoidance, Detection & recovery.

Security and Protection: Overview of system security, Security methods and devices, Protection, access control, and authentication.

Virtual Machines: Types of virtualization (including Hardware/Software, OS, Server, Service, Network).

Device Management: Characteristics of serial and parallel devices, Buffering strategies, Direct memory access, Disk structure, Disk scheduling algorithms.

Laboratory work: To explore different operating systems like Linux, Windows etc. To implement main algorithms related to key concepts in the operating systems.

1. Detailed architecture of linux commands and flow of command execution.
2. Detailed commands related to basics of linux, file handling, process management.
3. Shell program having sequential, decision and loop control constructs.
4. CPU Scheduling Algorithms
5. Threaded programming in Linux (Eg. POSIX threads in LINUX)

Course learning outcomes (CLOs):

On completion of this course, the students will be able to

1. Explain basic operating system concepts such as overall architecture, interrupts, APIs, user mode and kernel mode.
2. Distinguish concepts related to concurrency including, synchronization primitives, race conditions, critical sections and multi-threading.
3. Analyze and apply CPU scheduling algorithms, deadlock detection and prevention algorithms.
4. Examine and categorise various memory management techniques like caching, paging, segmentation, virtual memory, and thrashing.
5. Appraise high-level operating systems concepts such as file systems, security, protection, virtualization and device-management, disk-scheduling algorithms and various file systems.

Text Books:

1. *Silberschatz, A., Galvin, P.B. and Gagne, G., Operating System Concepts, John Wiley (2013).*
2. *Stallings, Willam, Operating Systems Internals and Design Principles, Prentice Hall (2014).*

Reference Books:

1. *Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel, 3rd Ed., O'Reilly Media(2005).*
2. *Michael Kifer, Scott Smolka, Introduction to Operating System Design and Implementation: The OSP 2 Approach, Springer(2007).*

Evaluation Scheme:

S.No.	Evaluation Elements	Weightage (%)
1	MST	20
2	EST	40
3	Sessionals (Assignments/Projects/Tutorials/Quizzes/Lab Evaluations)	40