UCS303: OPERATING SYSTEMS

L	T	P	Cr
3	0	2	4.0

Course Objectives: To understand the role, responsibilities, and the algorithms involved for achieving various functionalities of an Operating System.

Introduction and System Structures: Computer-System Organization, Computer-System Architecture, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security, Computing Environments, Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, System Programs, Operating-System Design and Implementation, Operating-System Structure.

Process Management: Process Concept, Process Scheduling, Operations on Processes, Inter-process Communication, Multi-threaded programming: Multi-core Programming, Multithreading Models, Process Scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms, Multiple-Processor Scheduling, Algorithm Evaluation.

Deadlock: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

Memory Management: Basic Hardware, Address Binding, Logical and Physical Address, Dynamic linking and loading, Shared Libraries, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table, Virtual Memory Management: Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing, Allocating Kernel Memory.

File Systems: File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing, Protection, File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management.

Disk Management: Mass Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, Disk Management, Swap-Space Management, RAID Structure.

Protection and Security: Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of the Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, The Security Problem, Program Threats, System and Network Threats, User Authentication, Implementing Security Defenses, Firewalling to Protect Systems and Networks.

Concurrency: The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors.

Laboratory work:

To explore detailed architecture and shell commands in Linux / Unix environment, and to simulate CPU scheduling, Paging, Disk-scheduling and process synchronization algorithms.

Course learning outcome (CLO) / Course Objectives (COs):

After the completion of the course, the student will be able to:

1. Explain the basic of an operating system viz. system programs, system calls, user mode and kernel mode.
2. Select a particular CPU scheduling algorithms for specific situation, and analyze the environment leading to deadlock and its rectification.
3. Explicate memory management techniques viz. caching, paging, segmentation, virtual memory, and thrashing.
4. Understand the concepts related to file systems, disk-scheduling, and security, protection.
5. Comprehend the concepts related to concurrency.

Text Books:

1. Silberschatz A., Galvin B. P. and Gagne G., Operating System Concepts, John Wiley & Sons Inc (2013) 9th ed.
2. Stallings W., Operating Systems Internals and Design Principles, Prentice Hall (2018) 9th ed.

Reference Books:

1. Bovet P. D., Cesati M., Understanding the Linux Kernel, O'Reilly Media (2006), 3rd ed.
2. Kifer M., Smolka A. S., Introduction to Operating System Design and Implementation: The OSP 2 Approach, Springer (2007).

UCS521: ARTIFICIAL INTELLIGENCE

L	T	P	Cr
3	**0**	**2**	**4.0**

Course Objectives: To be familiar with the applicability, strengths, and weaknesses of the basic knowledge representation, problem solving, machine learning, knowledge acquisition and learning methods in solving particular engineering problems.

Overview: foundations, scope, problems, and approaches of AI.

Intelligent agents: reactive, deliberative, goal-driven, utility-driven, and learning agents

Problem-solving through Search: forward and backward, state-space, blind, heuristic, problem-reduction, A, A*, AO*, minimax, constraint propagation, neural, stochastic, and evolutionary search algorithms, sample applications.

Knowledge Representation and Reasoning: ontologies, foundations of knowledge representation and reasoning, representing and reasoning about objects, relations, events, actions, time, and space; predicate logic, situation calculus, description logics, reasoning with defaults, reasoning about knowledge, sample applications.

Planning: Planning as search, partial order planning, construction and use of planning graphs, existing expert systems like MYCIN, RI, Expert system shells.

Representing and Reasoning with Uncertain Knowledge: probability, connection to logic, independence, Bayes rule, Bayesian networks, probabilistic inference, sample applications.
Decision-Making: basics of utility theory, decision theory, sequential decision problems, elementary game theory, sample applications.

Machine Learning and Knowledge Acquisition: learning from memorization, examples, explanation, and exploration. Learning nearest neighbor, naive Bayes, and decision tree classifiers, Q-learning for learning action policies, applications.

Languages for AI problem solving: Introduction to PROLOG syntax and data structures, representing objects and relationships, built-in predicates. Introduction to LISP- Basic and intermediate LISP programming.

Expert Systems: Architecture of an expert system.

Laboratory work: