Course: BTech **Semester:** 4

Prerequisite: Fundamentals of Computer Systems

Rationale: This course is an introduction to the theory and practice behind modern computer operating systems. Topics will include what an operating system does (and doesn't) do, system calls and interfaces, processes, concurrent programming, resource scheduling and management, virtual memory, deadlocks, algorithms, programming, and security. The approach of the subject is from both a theoretical perspective as well as a practical one.

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	T	CE	P	T	P	
3	0	0	0	3	20	20	-	60	-	100

SEE - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Content

W - Weightage (%) , **T** - Teaching hours

Sr.	Topics	W	T
1	**INTRODUCTION:** Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS-Layered,Monolithic, Microkernel Operating Systems, Concept of Virtual Machine.	5	3
2	**PROCESSES, THREAD & PROCESS SCHEDULING:** Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching. Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads. Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization,Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre- emptive, FCFS, SJF, RR.	20	9
3	**INTER-PROCESS COMMUNICATION:** Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson' s Solution, The Producer\ ConsumerProblem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader' s & Writer Problem, Dinning Philosopher Problem etc	15	6
4	**DEADLOCKS:** Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker' s algorithm, Deadlock detection and Recovery.	10	5
5	**MEMORY MANAGEMENT & VIRTUAL MEMORY:** Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation ‘⍰Fixed and variable partition‘Internal and External fragmentation and Compaction; Paging: Principle of operation ‘⍰Page allocation ‘⍰Hardware support for paging, Protection and sharing, Disadvantages of paging. Virtual Memory: Basics of Virtual Memory ‘ ⍰Hardware and control structures‘ ⍰Locality of reference, Page fault , Working Set , Dirty page/Dirty bit ‘ ⍰Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).	30	13
6	**I/O SYSTEMS, FILE & DISK MANAGEMENT:** I/O Hardware: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software. File Management: Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance. Disk Management: Disk structure, Disk scheduling algorithms - FCFS, SSTF, SCAN, C-SCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks	20	9

Reference Books

1.	**Operating System Concepts Essentials (TextBook)** By byAviSilberschatz, Peter Galvin,Greg Gagne | 9th Edition Wiley Asia Student Edition.
2.	**Operating Systems Internals and Design Principles** By William Stallings | PHI | 5th Edition
3.	**Operating System: A Design-oriented Approach** By Charles Crowley, | 1st Edition - Irwin Publishing
4.	**Operating Systems: A Modern Perspective** By by Gary J. Nutt | Addison-Wesley; 2nd Edition | 2nd Edition
5.	**Design of the Unix Operating Systems** By Maurice Bach, | Prentice-Hall of India | 8th Edition
6.	**Understanding the Linux Kernel** By Daniel P. Bovet, Marco Cesati, | O'Reilly and Associates | 3rd Edition

Course Outcome

After Learning the Course the students shall be able to:

After Learning the Course the students shall be able to:
1. Distinguish different styles of operating system design.
2. Understand device and I/O management functions in operating systems as part of a uniform device abstraction.
3. Understand disk organization and file system structure
4. Give the rationale for virtual memory abstractions in operating systems.
5. Understand the main principles and techniques used to implement processes and threads as well as the different algorithms for process scheduling.
6. Understand the main mechanisms used for inter-process communication.

Miscellaneous

Exam Requirement

It consists of Assignments/Seminars/Presentations/Quizzes/Surprise Tests (Summative/MCQ) etc

Course: BTech **Semester:** 4

Prerequisite: Data Structures and Algorithms, Good working knowledge of C, and Fundamentals of Computer Systems.

Rationale: This course is an introduction to the theory and practice behind modern computer operating systems. Topics will include what an operating system does (and doesn't) do, system calls and interfaces, processes, concurrent programming, resource scheduling and management, virtual memory, deadlocks, and algorithms, programming, and security. We will approach the subject from both a theoretical perspective as well as a practical one

Teaching and Examination Scheme

Teaching Scheme					Examination Scheme					Total
					Internal Marks			External Marks		
Lecture Hrs/Week	Tutorial Hrs/Week	Lab Hrs/Week	Hrs/Week	Credit	T	CE	P	T	P	
0	0	2	0	1	-	-	20	-	30	50

SEE - Semester End Examination, **CIA** - Continuous Internal Assessment (It consists of Assignments/Seminars/Presentations/MCQ Tests, etc.)

Course Outcome

After Learning the Course the students shall be able to:

1. Experiment with Linux commands and shell programming.
2. Able to build shell program for process and file system management with system calls.
3. Able to implement and analyse the performance of CPU scheduling algorithm.
4. Able to implement and analyse the performance of page replacement algorithms.
5. Able to implement and analyse the performance of deadlock avoidance and detection algorithm.

List of Practical

1.	**Study of Basic commands of Linux.**
2.	**Study the basics of shell programming.**
3.	**Write a Shell script to print given numbers sum of all digits.**
4.	**Write a shell script to validate the entered date. (eg. Date format is: dd-mm-yyyy).**
5.	**Write a shell script to check entered string is palindrome or not.**
6.	**Write a Shell script to say Good morning/Afternoon/Evening as you log in to system.**
7.	**Write a C program to create a child process.**
8.	**Finding out biggest number from given three numbers supplied as command line arguments.**
9.	**Printing the patterns using for loop.**
10.	**Shell script to determine whether given file exist or not.**
11.	**Write a program for process creation using C. (Use of gcc compiler.**
12.	**Implementation of FCFS &Round Robin Algorithm.**
13.	**Implementation of Banker's Algorithm.**