23IT22P1 - OPERATING SYSTEMS & SOFTWARE ENGINEERING LAB

(IT)

Course Category:	Professional Core	**Credits:**	1.5
Course Type:	Practical	**Lecture-Tutorial-Practical:**	0-0-3
Prerequisite:	Understanding of fundamental operating system concepts such as processes, threads, scheduling algorithms, synchronization, deadlock handling, memory management (paging, segmentation), file systems, and I/O management. Understanding of fundamental Software engineering concepts such as SRD, Design, Testing.	**Sessional Evaluation:** **Univ. Exam Evaluation:** **Total Marks:**	30 70 100
Objectives:	**Students undergoing this course are expected:**		
	• Provide insights into system calls, file systems, semaphores, • Develop and debug CPU Scheduling algorithms, page replacement algorithms, thread implementation • Implement Bankers Algorithms to Avoid the Dead Lock • Acquire the generic software development skill through various stages of software life cycle • Generate test cases for software testing		

Course Outcomes	**Upon successful completion of the course, the students will be able to:**	
	CO1	Trace different CPU Scheduling algorithms (L2).
	CO2	Implement Bankers Algorithms to Avoid the Dead Lock (L3).
	CO3	Evaluate Page replacement algorithms (L5).
	CO4	Illustrate the file organization techniques (L4).
	CO5	Illustrate Inter process Communication and concurrent execution of threads (L4)
	CO6	Generate test cases for software testing
	CO7	Implement Design and Requirements documentation
Course Content	**Experiments in Operating Systems** 1. Practicing of Basic UNIX Commands. 2. Write programs using the following UNIX operating system calls fork, exec, getpid, exit, wait, close, stat, opendir and readdir 3. Simulate UNIX commands like cp, ls, grep, etc., 4. Simulate the following CPU scheduling algorithms a) FCFS b) SJF c) Priority d) Round Robin 5. Control the number of ports opened by the operating system with a) Semaphore b) Monitors. 6. Write a program to illustrate concurrent execution of threads using pthreads library.	

	7. Write a program to solve producer-consumer problem using Semaphores. 8. Implement the following memory allocation methods for fixed partition a) First fit b) Worst fit c) Best fit 9. Simulate the following page replacement algorithms a) FIFO b) LRU c) LFU 10. Simulate Paging Technique of memory management. 11. Implement Bankers Algorithm for Dead Lock avoidance and prevention 12. Simulate the following file allocation strategies a) Sequential b) Indexed c) Linked **Experiments in Software Engineering** 1. Perform the following, for the following experiments: i. Do the Requirement Analysis and Prepare SRS ii. Draw E-R diagrams, DFD, CFD and structured charts for the project. 2. Course Registration System 3. Students Marks Analyzing System 4. Online Ticket Reservation System 5. Stock Maintenance 6. Consider any application, using COCOMO model, estimate the effort. 7. Consider any application, Calculate effort using FP oriented estimation model. 8. Draw the UML Diagrams for the problem a, b, c, d. 9. Design the test cases for e-Commerce application (Flipcart, Amazon) 10. Design the test cases for a Mobile Application (Consider any example from Appstore) 11. Design and Implement ATM system through UML Diagrams.
Text Books & References Books	**REFERENCE BOOKS:** 1. Operating System Concepts, Silberschatz A, Galvin P B, Gagne G, 10th Edition, Wiley, 2018. 2. Modern Operating Systems, Tanenbaum A S, 4th Edition, Pearson, 2016 3. Operating Systems -Internals and Design Principles, Stallings W, 9th edition, Pearson, 2018 4. Operating Systems: A Concept Based Approach, D.M Dhamdhere, 3rd Edition, McGraw- Hill, 2013 5. Software Engineering, Ian Sommerville,10th Edition, Pearson. 6. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
E-Resources	1. https://www.cse.iitb.ac.in/~mythili/os/ 2. http://peterindia.net/OperatingSystems.html 3. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012605895063871488 27_shared/overview 4. https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_013382690411003