

Course No.	Course Name	L-T-P Credits	Year of Introduction
CS331	SYSTEM SOFTWARE LAB	0-0-3-1	2015
<b>Course Objectives</b> <i>To build an understanding on design and implementation of different types of system software.</i>			
<b>List of Exercises/Experiments: (Exercises/experiments marked with * are mandatory from each part. Total 12 Exercises/experiments are mandatory)</b>			
<u>Part A</u>			
1. Simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time. a) FCFS      b) SJF      c) Round Robin (pre-emptive)      d) Priority			
2. Simulate the following file allocation strategies. a) Sequential    b) Indexed    c) Linked			
3. Implement the different paging techniques of memory management.			
4. Simulate the following file organization techniques * a) Single level directory      b) Two level directory      c) Hierarchical			
5. Implement the banker's algorithm for deadlock avoidance.*			
6. Simulate the following disk scheduling algorithms. * a) FCFS      b)SCAN      c) C-SCAN			
7. Simulate the following page replacement algorithms a) FIFO b)LRU      c) LFU			
8. Implement the producer-consumer problem using semaphores. *			
9. Write a program to simulate the working of the dining philosopher's problem.*			

Part B

10. Implement the symbol table functions: create, insert, modify, search, and display.
11. Implement pass one of a two pass assembler. \*
12. Implement pass two of a two pass assembler. \*
13. Implement a single pass assembler. \*
14. Implement a two pass macro processor \*
15. Implement a single pass macro processor.
16. Implement an absolute loader.
17. Implement a relocating loader.
18. Implement pass one of a direct-linking loader.
19. Implement pass two of a direct-linking loader.
20. Implement a simple text editor with features like insertion / deletion of a character, word, and sentence.
21. Implement a symbol table with suitable hashing.\*

**Expected Outcome**

Student is able to

1. *Compare and analyze CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority.*
2. *Implement basic memory management schemes like paging.*
3. *Implement synchronization techniques using semaphores etc.*
4. *Implement banker's algorithm for deadlock avoidance.*
5. *Implement memory management schemes and page replacement schemes and file allocation and organization techniques.*
6. *Implement system software such as loaders, assemblers and macro processor.*