

Course code	Course Name	L-T-P Credits	Year of Introduction
CS331	SYSTEM SOFTWARE LAB	0-0-3-1	2016
Prerequisite: Nil			
Course Objectives			
<ul style="list-style-type: none"> To build an understanding on design and implementation of different types of system software. 			
List of Exercises/Experiments: (Exercises/experiments marked with * are mandatory from each part. Total 12 Exercises/experiments are mandatory)			
<i>Part A</i>			
<ol style="list-style-type: none"> Simulate the following non-preemptive CPU scheduling algorithms to find turnaround time and waiting time. <ol style="list-style-type: none"> FCFS SJF Round Robin (pre-emptive) Priority Simulate the following file allocation strategies. <ol style="list-style-type: none"> Sequential Indexed Linked Implement the different paging techniques of memory management. Simulate the following file organization techniques * <ol style="list-style-type: none"> Single level directory Two level directory Hierarchical Implement the banker's algorithm for deadlock avoidance.* Simulate the following disk scheduling algorithms. * <ol style="list-style-type: none"> FCFS SCAN C-SCAN Simulate the following page replacement algorithms <ol style="list-style-type: none"> FIFO LRU LFU Implement the producer-consumer problem using semaphores. * Write a program to simulate the working of the dining philosopher's problem.* 			
<i>Part B</i>			
<ol style="list-style-type: none"> Implement the symbol table functions: create, insert, modify, search, and display. Implement pass one of a two pass assembler. * Implement pass two of a two pass assembler. * Implement a single pass assembler. * Implement a two pass macro processor * Implement a single pass macro processor. Implement an absolute loader. Implement a relocating loader. Implement pass one of a direct-linking loader. Implement pass two of a direct-linking loader. Implement a simple text editor with features like insertion / deletion of a character, word, and sentence. Implement a symbol table with suitable hashing.* 			

Expected Outcome

The students will be able to

- i. Compare and analyze CPU Scheduling Algorithms like FCFS, Round Robin, SJF, and Priority.
- ii. Implement basic memory management schemes like paging.
- iii. Implement synchronization techniques using semaphores etc.
- iv. Implement banker's algorithm for deadlock avoidance.
- v. Implement memory management schemes and page replacement schemes and file allocation and organization techniques.
- vi. Implement system software such as loaders, assemblers and macro processor.

TECHNOLOGICAL
UNIVERSITY

