COURSE CODE	COURSE TITLE	L	T	P	C
1151CS304	OPERATING SYSTEMS LAB	0	0	2	1

Course Category: Program Core

A. Preamble:

Operating systems are the fundamental part of every computing device to run any type of software. The increasing use of computing devices in all areas of life (leisure, work), lead to a variety of operating systems. Yet all operating systems share common principles. These principles are important for computer science students in their understanding of programming languages and software built on top of operating systems. The Operating System Laboratory, OS Lab is a course that will teach students about principles of operating systems using a constructivist approach and problem-oriented learning.

B. Prerequisite Courses:

Sl. No	Course Code	Course Name
1	1151CS102	Data Structures

C. Related Courses:

Sl. No	Course Code	Course Name
1	1156CS601	Minor Project
2	1156CS701	Major Project

D. Course Outcomes:

Upon the successful completion of the course, learners will be able to

CO Nos.	Course Outcomes	Level of learning domain (Based on revised Bloom's)
CO1	Demonstrate the fundamental UNIX commands & system calls	S3
CO2	Apply the scheduling algorithms for the given problem	S3
CO3	Apply the process synchronous concept using message queue, shared memory, semaphore and Dekker's algorithm for the given situation.	S3
CO4	Experiment an algorithm to detect and avoid dead lock	S3
CO5	Apply the various methods in memory allocation and page replacement algorithm.	S3
CO6	Demonstrate the various operations of file system.	S3

K2-Understand, K3-Apply, S3-Processes

D. Correlation of COs with Programme Outcomes:

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	**H**														
CO2	**H**	**M**	**L**	**L**		**L**	**L**					**L**	**L**	**M**	
CO3	**H**	**M**	**L**			**L**	**L**					**L**			**L**
CO4	**H**	**M**	**L**	**L**		**L**	**L**					**L**	**L**	**M**	**L**
CO5	**H**	**M**	**L**	**L**		**L**	**L**					**L**	**L**		**L**
CO6	**M**	**M**													

F. Course Content:

Cycle I

Basics of UNIX Commands

1. Write programs using the following system calls of UNIX operating system: fork, exec, getpid, exit, wait, close.
2. Write programs using the I/O System calls of UNIX operating system (open, read, write, etc).
3. Given the list of processes, their CPU burst times. Display/print the Gantt chart for FCFS scheduling algorithm. Compute and print the average waiting time and average turnaround time.
4. Given the list of processes, their CPU burst times and arrival times. Display the Gantt chart for SJF scheduling algorithm. Compute and print the average waiting time and average turnaround time.

Model Practical Examination I

Cycle II

5. Given the list of processes, their CPU burst times and time quantum. Display the Gantt chart for Round robin scheduling algorithm. Compute and print the average waiting time and average turnaround time.
6. Given the list of processes, their CPU burst times and arrival times. Display the Gantt chart for Priority scheduling algorithm. Compute and print the average waiting time and average turnaround time.
7. Develop application using Inter-Process Communication (using shared memory, pipes or message queues).
8. Implement the Producer-Consumer problem using semaphores (using UNIX system calls)
9. Implement Memory management schemes like paging and segmentation.
10. Implement Memory allocation schemes like First fit, Best fit and Worst fit.

Model Practical Examination II

G. Learning Resources:

i. Reference Books:

1. Universal Command Guide: For Operating Systems – April 15, 2002 ,by Guy Lotgering
2. The Easy Guide to Operating Systems, Larry Miller,2012.