Course Title: Operating Systems Lab	***Course Code: 20CS46L***
Credits: 1.5	***Contact Hours (L: T: P): 0: 0: 39***
Type of Course: Practical	***Category: Professional Core Course***
CIE Marks: 50	***SEE Marks: 50***

Pre-requisite: Computer Organization and Architecture, Data Structures, C Programming.

Course Objectives: The course should enable the students to:

Sl. No.	Course Objectives
1	Familiarize students with LINUX/UNIX OP and provide necessary skills for developing and debugging programs in these environments.
2	Learn shell script, creation and management of processes, IPC using shared memory and multithreads programing.
3	Analyze and develop process scheduling algorithms and process synchronization.

NOTE: For all the scheduling algorithms write the expected output for the given data (table with set of processes, AT, BT) along with the Gantt chart. Execute the scheduling program for the same data. The output should be in the form of table with all the necessary time parameters like AT, BT, CT, TT and WT. Also display the Gantt chart and average WT.

Weeks	List of Programs	No. of Hours
1	a) Write a shell program to check whether a given number is palindrome or not b) Implement SJF Pre-emptive scheduling algorithm by defining process structure.	3
2	a) Write a shell program to generate prime numbers in a given range b) Implement Priority Pre-emptive scheduling algorithm by defining process structure.	3
3	a) Write a shell program to find largest of n numbers, storing numbers in an array. b) Implement Round Robin scheduling algorithm by defining process structure.	3
4	a) Write a shell program to Read two matrices , find addition and display the input matrices and resultant in matrix format b) Implement Priority Non-preemptive scheduling algorithm by defining process structure.	3
5	a) Write a program to perform the following task using I/O system calls for file I/O i. Reading first 10 characters from file ii. Skipping 5 characters from current position in the file iii. Going to 5th last character in the file iv. Going to the 3rd character in the file b) Implement FCFS scheduling algorithm by defining process structure.	3

6	a) Write a program to demonstrate zombie and orphan process. b) Implement SJF Non-preemptive scheduling algorithm by defining process structure.	3
7	a) Write a program to simulate grep command using system calls b) Write a program to generate and print Fibonacci series with the following requirements: - Parent program should create a child and distribute the task of generating Fibonacci no to its child. - The code for generating Fibonacci series should reside in different program. - Child should write the generated Fibonacci series to a shared memory. - Parent process has to print by retrieving the Fibonacci series from the shared memory. i) Implement the above using shmget and shmat Note: Shared object should be removed at the end in the program	3
8	a) Write a program to simulate ls command using system calls. b) Write a program to generate and print Fibonacci series with the following requirements: - Parent program should create a child and distribute the task of generating Fibonacci no to its child. - The code for generating Fibonacci series should reside in different program. - Child should write the generated Fibonacci series to a shared memory. - Parent process has to print by retrieving the Fibonacci series from the shared memory. i) Implement the above using shm_open and mmap Note: Shared object should be removed at the end in the program	3
9	a) Write a shell program to check whether a given no. is a palindrome or not b) Write a program to generate and print N ODD numbers with the following requirements: - Parent program should create a child and distribute the task of generating odd numbers to its child. - The code for generating odd numbers should reside in different program. - Child should write the generated odd numbers to a shared memory. - Parent process has to print the odd numbers by retrieving from the shared memory. i) Implement the above using shmget and shmat Note: Shared object should be removed at the end in the program	3
10	a) Write a program to simulate cat command using system calls. b) Write a program to generate and print Prime nos. between a given	3

	range (between M & N) with the following requirements: - M & N should be passed as command line arguments - Error checking should be done to verify the required no. of arguments at the command line - Parent program should create a child and distribute the task of generating Prime numbers to its child. - The code for generating Prime numbers should reside in different program. - Child should write the generated Prime numbers to a shared memory. - Parent process has to print the Prime numbers by retrieving from the shared memory. i) Implement the above using shm_open and mmap Note: Shared object should be removed at the end in the program.	
11	a) Write a program with two threads and a main thread. Schedule the task of calculating the natural sum upto 'n' terms and factorial of 'n' on these threads. Note: The main thread should read 'n' from command line and pass it as parameter to remaining threads. Terminate the threads using system calls. b) Write a program that implements solution to Producer – Consumer problem using mutex and semaphores.	3
12	a) Write a shell program to find the largest of three numbers b) Write a program that implements solution to Readers-Writers problem using mutex and semaphores.	3
13	**Lab Test/Event**	3

Reference Books:

Sl. No.	Author/s	Title	Publisher Details
1	Abraham Silberschatz, Peter Baer Galvin, Greg Gagne	Operating System Concepts	9th Edition, Wiley India, 2013
2	William Stallings	Operating Systems: Internals and Design Principles	7th Edition, Prentice Hall of India,2017
3	D.M Dhamdhere	Operating systems - A concept-based Approach	4th Edition, Tata McGraw- Hill, 2013
4	P.C.P. Bhatt	Introduction to Operating Systems	Concepts and Practice, 4th Edition, PHI, 2014

Web Resources:

Sl. No.	Web Link
1	https://youtu.be/783KAB-tuE4 - NPTEL IIT, Madras
2	https://nptel.ac.in/courses/106108101/

Course Outcomes:

CO1	Implement shell programs and design process management and file system management with system calls.
CO2	Design and implement Inter Process Communication and multiple threads application.
CO3	Analyze and implement CPU scheduling algorithms and process synchronization.

Mapping Course Outcomes with Program outcomes & Program Specific outcomes:

Course Outcomes	Program Outcomes												PSO's			
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3	PSO4
CO1	3	3	3	-	-	-	-	-	3	3	-	-	3	-	3	-
CO2	3	3	3	3	-	-	-	-	3	3	-	-	3	-	3	-
CO3	3	3	3	3	-	-	-	-	3	3	-	-	3	-	3	-

1-Low association, 2- Moderate association, 3-High association