Operating System and Unix System Programming Laboratory
(Lab)

Course Code	15CSL57	**Credits**	2
Course type	L1	**CIE Marks**	25 marks
Hours/week: L-T-P	0 – 0 – 3	**SEE Marks**	25 marks
Total Hours:	38	**SEE Duration**	3 Hours for 50 marks

Course learning objectives

1. To present description of the concepts of operating systems to schedule, manage and to improve the utilization of the CPU.
2. To illustrate the process of memory management and file system management.
3. To demonstrate UNIX system programming API's.
4. To get acquainted with knowledge of UNIX operating system environment like race condition, zombie.

Pre-requisites: C Programming, Computer Organization, Basic UNIX Commands, Operating System, Data Structures

List of experiments

PART A

1) Consider the following jobs submitted to a system:

	Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P0	0	0	2	0	0	4	1	0	2
P1	1	0	0	2	0	1			
P2	1	3	5	1	3	7			

Process	**Arrival time (ms)**	**CPU burst time (ms)**	**Priority**
Print	0	7	3
e-mail	2	3	2
File transfer	2	8	1
Web service	3	4	4

a) Implement a scheduling algorithm that schedules all processes in FCFS for a fixed quantum of 4ms and switches till all processes finish their bursts.

b) Schedule the above processes according to the associated priority where a low value indicates higher priority. The algorithm should preempt the current process if a higher priority process arrives.

2) Consider a set of memory partitions and a set of processes. A partition can hold only a single process contiguously. Compare various memory allocation strategies with reference to external and internal fragmentation.

3) Consider a system which has a finite number of instances of resource types A, B and C. A set of processes, P0-P4, compete for these resource instances and are allocated according to the snapshot in Fig.1 at a given point in time.

P3	6	3	2	8	4	2			
P4	1	4	3	1	5	7			

Fig1. Snapshot

i) Determine whether the system is safe.
ii) If a request from process P2 arrives for (0, 0, 2), can the request be granted immediately?

4) A word processor and a spreadsheet process are trying to access a printer that is shared among several processes. Devise a mechanism to ensure that the two processes cooperate in an orderly manner to avoid inconsistent system state.

5) Compare the performance in terms of number of page faults for the algorithm that suffers from Belady's anomaly with the one that replaces a page that has not been used for the longest period of time.

6) Write a C program to simulate the following file organization techniques:
a) Single level directory b) Two level directory c) Hierarchical

PART B

7) Write a C/C++ POSIX compliant program to check the compile time and run time configuration limits:

(i) Number of clock ticks
(ii) Maximum number of child processes
(iii) Maximum path length
(iv) Maximum number of characters in a file name
(v) Maximum number of open files / process

8) Suppose two processes, parent and child, try to access a shared resource such as stdout. The output may not be desirable, called as race condition which occurs due to the order in which the processes are scheduled internally. Develop a C/C++ program to illustrate the race condition.

9) Consider a child process that has been terminated but not yet been reaped leading to a resource leak. Write a C/C++ program to create a resource leak and also identify its presence.

10) Consider system( ) as a higher-level interface and duplicate its functionality using the mechanism of process creation.

11) Consider the last 100 bytes as a region. Write a C/C++ program to check whether the region is locked or not. If the region is locked, print pid of the process which has locked. If the region is not locked, lock the region with an exclusive lock, read the last 50 bytes and unlock the region.

12) Suppose a writer process generates data to be consumed by a reader process on the same machine. Develop a suitable inter process communication mechanism between the two processes that allows not just for one-time but also at subsequent times during execution.

Course Outcome (COs)

At the end of the course, the student will be able to	Bloom's Level
1. **Compare** scheduling algorithms, memory allocation strategies, page replacement algorithms and file organization techniques.	**L4**
2. **Determine** whether a system is safe in context of deadlocks.	**L5**
3. **Design** mechanism for process synchronization.	**L3**
4. **Understand** the working of different API's of Unix system.	**L2**
5. Analyze working of system calls	**L3**

Program Outcome of this course (POs)

		PO No.
1.	Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.	**1**
2.	Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.	**2**
3.	Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.	**4**
4.	Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.	**12**

Assessment methods

1. IA Test
2. Viva-Voce
3. Lab Journal Evaluation.

Scheme of Continuous Internal Evaluation (CIE):

Components	Conduct of the lab	Journal submission	Total Marks
Maximum Marks: 25	10	15	25

- Submission and certification of lab journal is compulsory to qualify for SEE.
- **Minimum marks required to qualify for SEE : 13 marks out of 25**

Scheme of Semester End Examination (SEE):

1. It will be conducted for 50 marks of 3 hours duration**. It will be reduced to 25 marks for the calculation of SGPA and CGPA.**
2. **Minimum marks required in SEE to pass:20**
3.

Initial write up	10 marks	
Conduct of experiments	20 marks	50 marks
Viva- voce	20 marks	

Instructions: In the examination, each student picks one question from the lot of 12 questions.