Course Name: Operating System

Course Outcome

CO1 - Understand the classification of operating system environment
CO2 - Understand the basic of process management
CO3 - Apply the concept of CPU process scheduling for the given scenarios.
CO4 - Illustrate the process synchronization and concurrency process in operating system.
CO5 - Analyze the occurrence of deadlock in operating system
CO6 - Describe and analyze the memory management and its allocation policies.
CO7 - Understand the concepts of disk scheduling

Printed Pages: 03 **University Roll No.** .....

Mid Term Examination, Odd Semester 2023-24

B. Tech- CSE/ EC with Minor CS/AIML/CCV/DA/CSF/IIOT, II Year III Semester

Operating System (BCSC 0004)

Time: 2 Hours **Maximum Marks: 30**

Section – A

Attempt All Questions 3 X 5 = 15 Marks

No.	Detail of Question	Marks	CO	BL	KL
1	i) Give two reasons why caches are useful. What problems do they solve? ii) What is the main difficulty that a programmer must overcome in writing an operating system for a real-time environment?	1+2	1	An	C
2	Discuss the merits and demerits of following operating systems: a) Multiprogramming b) Multitasking	1.5+1.5	1	U	F
3	Consider three CPU-bound processes arriving to a First Come First Served (FCFS) scheduler as follows: **Process** **Arrival time** **Burst time** P1 0 2 P2 1 12 P3 5 1 i) Calculate the average waiting time across all three processes. ii) A weakness of FCFS is that its performance is sensitive to the arrival process. Assume process arrival times remain t = 0, 1, 5 but the arrival order changes. Give an alternative arrival order that improves the average waiting time. Define	1+2	3	A	P

	the convoy effect and give a second alternative arrival order displaying it.				
4	Describe the five-state process model, describe what transitions are valid between the five states, and describe an event that might cause such a transition.	3	2	U	P
5	You are a developer working on a multi-threaded application. The application frequently encounters race conditions, leading to unpredictable behavior and crashes. a) Define what race conditions are and why they occur in multi-threaded applications? b) Discuss synchronization mechanisms and best practices you would use to eliminate race conditions and ensure the application behaves predictably.	1+2	4	An	M

Section – B

*Attempt **All** Questions* 5 X 3 = 15 Marks

No.	Detail of Question	Marks	CO	BL	KL
6	i) You are tasked with simulating the Dining Philosophers problem, where five philosophers sit around a circular table, and they alternate between thinking and eating. To eat, a philosopher must pick up two forks placed between them and their neighbors. Describe the challenges and potential solutions to this problem using Semaphore. ii) If we add a 6th chopstick to the center of the table, have we cured the deadlock problem? If yes, what condition have we removed? If no, explain why not.	4+1	4	A	P
7	A shared variable x, initialized to zero, is operated on by four concurrent processes W, X, Y, Z as follows. Each of the processes W and X reads x from memory, increments by one, stores it to memory, and then terminates. Each of the processes Y and Z	5	4	An	P

	reads x from memory, decrements by two, stores it to memory, and then terminates. Each process before reading x invokes the P operation (i.e., wait) on a counting semaphore S and invokes the V operation (i.e., signal) on the semaphore S after storing x to memory. Semaphore S is initialized to two. What is the maximum possible value of x after all processes complete execution?				
8	Consider a uniprocessor system executing three tasks T1, T2 and T3, each of which is composed of an infinite sequence of jobs (or instances) which arrive periodically at intervals of 3, 7 and 20 milliseconds, respectively. The priority of each task is the inverse of its period and the available tasks are scheduled in order of priority, with the highest priority task scheduled first. Each instance of T1, T2 and T3 requires an execution time of 1, 2 and 4 milliseconds, respectively. Given that all tasks initially arrive at the beginning of the 1st milliseconds and task preemptions are allowed. In how many milliseconds the first instance of T3 completes its execution?	5	3	E	M