DR. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY, LONERE

Mid Semester Examination – April-2024

Class: M.Tech. **CSE** **Date: 11/11/24** **Time:1 Hr.**

Subject: MTCSEPC103 Advanced Operating Systems **Marks: 20**

Instructions to the Students:

1. All the questions are compulsory.
2. The level of question/expected answer as per OBE or the Course Outcome (CO) on which the question is based is mentioned in ( ) in front of the question.
3. Assume suitable data wherever necessary and mention it clearly.

Q.1	Attempt all of the following	CO	BL	Marks
1)	How does an operating system manage resources to ensure efficient performance and user interaction? 1. To manage hardware resources 2. To write code for applications 3. To provide internet connectivity 4. To design user interfaces	**103_1**	L3	1
2)	Design an operating system architecture where the component that connects user applications to hardware resources is optimized for high-performance computing. 1. Kernel 2. Shell 3. Compiler 4. D) Device driver	**103_1**	L3	1
3)	Consider an operating system running multiple applications. Identify the primary unit of execution within the system A) Kernel B) Thread C) Process D) Task	**103_1**	L3	1
4)	Imagine you are designing an application that requires multiple processes to share data and resources efficiently. What is the term used to describe processes that can share resources and information, and how would you implement this feature in an operating system? A) Isolated processes B) Independent processes C) Cooperating processes D) Autonomous processes	**103_1**	L3	1
5)	Consider a system where multiple processes need to access shared resources simultaneously. Which solution is suitable to address the critical-section problem in concurrent programming? A) Peterson's Solution B) Randomized Solution C) Arbitrary Solution D) Coin toss Solution	**103_1**	L3	1

6)	Imagine you are tasked with designing a multi-threaded application where several threads need to perform tasks simultaneously within the same process. Based on your understanding of threads and their characteristics, analyze the following statements and identify which one accurately describes how threads behave in such an environment. A) Threads have their own memory space. B) Threads share memory and resources with other threads within the same process. C) Threads cannot run concurrently. D) Threads are heavier than processes.	**103_1**	L3	1
Q.2	**Attempt any Two of the following**			
A)	Discuss design goals, policies and implementation of a typical operating system.	**103_1**	L3	3
B)	Define the concept of a critical section in concurrent programming. Describe the key requirements that must be met for a solution to the critical section problem to be effective. OR	**103_1**	L3	3
C)	Consider a scenario where multiple philosophers are seated at a table, each needing two forks to eat. Analyze the four necessary conditions for a deadlock to occur in the Dining Philosophers problem. Then, propose a solution for avoiding deadlock by eliminating one of these conditions. Explain how this modification would prevent deadlock from happening in this scenario.	**103_1**	L3	3
Q.3	**Attempt any One of the following**			
A)	Given a set of processes, each with its corresponding CPU burst time (in ms), analyze the following data and determine the most efficient scheduling algorithm (e.g., First Come First Serve, Shortest Job Next, Round Robin) to minimize overall waiting time and turnaround time. Justify your choice of scheduling algorithm and demonstrate how it applies to the given set of processes. Process Burst Time Priority P1 10 3 P2 1 1 P3 2 3 P4 1 4 P5 5 2 The processes are assumed to have arrived in the order P1,P2,P3,P4,P5, all at time 0. a) Draw Gantt charts that illustrates the execution of these processes using the following scheduling algorithms: FCFS, SJF, non preemptive priority. b) What is the turnaround time of each process for each of the scheduling algorithms? c) What is the waiting time of each process for each of these scheduling algorithms?	**103_1**	L3	8

<table>
<tr><td></td><td>OR</td><td></td><td></td><td></td></tr>
<tr><td>B)</td><td>Given a table of arrival times and burst times for four processes (P1, P2, P3, P4), and a time quantum of 2 ms, apply the Round Robin (RR) scheduling algorithm to determine the average waiting time and response time for these processes. After completing the scheduling, explain how the time quantum affects the waiting time and response time for each process, and evaluate the efficiency of the Round Robin scheduling in comparison to other possible algorithms.

<table>
<tr><td>Process</td><td>Burst Time</td><td>Arrival Time</td></tr>
<tr><td>P1</td><td>5 ms</td><td>0 ms</td></tr>
<tr><td>P2</td><td>4 ms</td><td>1 ms</td></tr>
<tr><td>P3</td><td>2 ms</td><td>2 ms</td></tr>
<tr><td>P4</td><td>1 ms</td><td>4 ms</td></tr>
</table>
</td><td>103_1</td><td>L3</td><td>8</td></tr>
</table>

End

Model Answer

DR. BABASAHEB AMBEDKAR TECHNOLOGICAL UNIVERSITY, LONERE

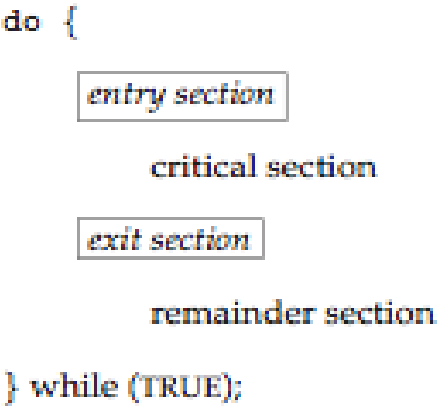
Mid Semester Examination – April-2024

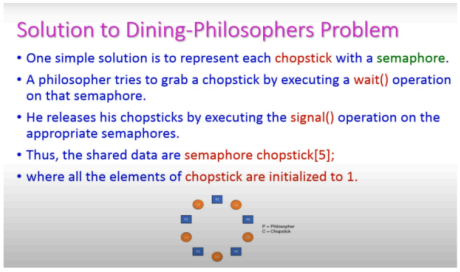
Class: S.Y. B. Tech. CSE **Date: 22/4/2024** **Time:1 Hr.**

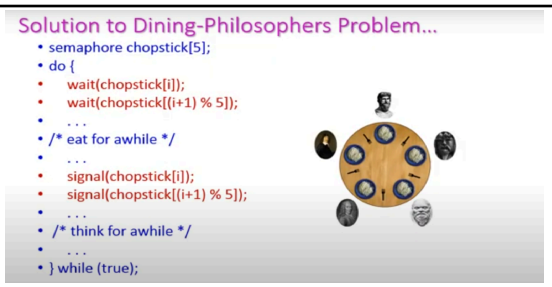
Subject: Operating Systems (BTCOC402)) **Marks: 20**

Q.1	Attempt all of the following	Marks
1)	How does an operating system manage resources to ensure efficient performance and user interaction? A) To manage hardware resources B) To write code for applications C) To provide internet connectivity D) To design user interfaces **Answer: A) To manage hardware resources**	1
2)	Design an operating system architecture where the component that connects user applications to hardware resources is optimized for high-performance computing. A) Kernel B) Shell C) Compiler D) Device driver **Answer: A) Kernel**	1
3)	Consider an operating system running multiple applications. Identify the primary unit of execution within the system. A) Kernel B) Thread C) Process D) Task **Answer : C) Process**	1
4)	Imagine you are designing an application that requires multiple processes to share data and resources efficiently. What is the term used to describe processes that can share resources and information, and how would you implement this feature in an operating system? A) Isolated processes B) Independent processes C) Cooperating processes D) Autonomous processes **Answer c) Cooperating processes**	1

5)	Which solution is used to address the critical-section problem in concurrent programming? A) Peterson's Solution B) Randomized Solution C) Arbitrary Solution D) Coin toss Solution **Answer : A)** Peterson's Solution	1
6)	Which of the following is true about threads? i. Threads have their own memory space. ii. Threads share memory and resources with other threads within the same process. iii. Threads cannot run concurrently. iv. Threads are heavier than processes. **Answer: B) Threads share memory and resources with other threads within the same process.**	1
Q.2	**Attempt any Two of the following**	
A)	Ans- The design of an operating system is a broad and complex topic that touches on many aspects of computer science. This article will cover the design of operating systems in general and then focus on the implementation aspect. **Design Goals: (1 Mark)** Design goals are the objectives of the operating system. They must be met to fulfill design requirements and they can be used to evaluate the design. These goals may not always be technical, but they often have a direct impact on how users perceive their experience with an operating system. While designers need to identify all design goals and prioritize them, they also need to ensure that these goals are compatible with each other as well as compatible with user expectations or expert advice The process of identifying design goals, conflicts, and priorities is often referred to as "goal-driven design." The goal of this approach is to ensure that each design decision is made with the best interest of users and other stakeholders in mind. **Mechanisms and Policies: (1 Mark)** An operating system is a set of software components that manage a computer's resources and provide overall system management. Mechanisms and policies are the two main components of an operating system. Mechanisms handle low-level functions such as scheduling, memory management, and interrupt handling; policies handle higher-level functions such as resource management, security, and reliability. A well-designed OS should provide both mechanisms and policies for each component in order for it to be successful at its task: Mechanisms should ensure that applications have access to appropriate hardware resources (seats). They should also make sure that applications don't interfere with each other's use of these resources (for example through mutual exclusion). Policies determine how processes will interact with one another when they're running simultaneously on multiple CPUs within a single machine instance – what processor affinity should occur during multitasking operations? Should all processes be allowed access simultaneously or just those belonging specifically within group 'A'?' These are just some of the many questions that policies must answer. The OS is responsible for enforcing these mechanisms and policies, as well as handling exceptions when they occur. The operating system also provides a number of services to applications, such as file access and networking capabilities. The operating system is also responsible for making sure that all of these tasks are done efficiently and in a timely manner. The OS provides applications with access to the underlying hardware resources and ensures that they're properly utilized by the	3 (1 marks each)

	application. It also handles any exceptions that occur during execution so that they don't cause the entire system to crash. Implementation**(1 Mark)** Implementation is the process of writing source code in a high-level programming language, compiling it into object code, and then interpreting (executing) this object code by means of an interpreter. The purpose of an operating system is to provide services to users while they run applications on their computers. **The design and implementation of an operating system is a complex process that involves many different disciplines. The goal is to provide users with a reliable, efficient, and convenient computing environment, so as to make their work more efficient.**	
B)	Ans-Define **(1 Mark)** **Critical Section** refers to the segment of code or the program which tries to access or modify the value of the variables in a shared resource.  **Figure 6.1** General structure of a typical process P_i. Solutions to critical section problem **(2 Marks)** **Mutual Exclusion** - If process P_i is executing in its critical section, then no other processes can be executing in their critical sections 2. **Progress** - If no process is executing in its critical section and there exist some processes that wish to enter their critical section, then the selection of the processes that will enter the critical section next cannot be postponed indefinitely 3. **Bounded Waiting** - A bound must exist on the number of times that other processes are allowed to enter their critical sections after a process has made a request to enter its critical section and before that request is granted ● Assume that each process executes at a nonzero speed ● No assumption concerning **relative speed** of the n processes	3 (2 marks explanation) (1 marks fig.)
C)	Ans- Four conditions : **(1 Mark)** Mutual exclusion, hold and wait, circular wait and no prepmption Solution to dining philosopher problem: **(2 Marks)**	3





Q.3 **Attempt any One of the following**

A) Ans- 8 (2 Marks Each difference)

FCFS

Gantt Chart

P1	P2	P3	P4	P5
10	11	13	14	19

Criteria: Non preemptive

Arrival for all is 0

Process	BT	priority	CT	TAT	WT	RT
P1	10	3	10	10	0	0
P2	1	1	11	11	10	10
P3	2	3	13	13	11	11
P4	1	4	14	14	13	13
P5	5	2	19	19	14	14

Average WT= 9.6 ms

SJF

Gantt Chart

P2	P4	P3	P5	P1
1	2	4	9	29

Case : if arrival time is same, then FCFS is used to break the tie.

Criteria: Non preemptive

Consider burst time

Arrival for all is 0

Process	BT	priority	CT	TAT	WT	RT
P1	10	3	29	29	19	
P2	1	1	1	1	0	
P3	2	3	4	4	2	
P4	1	4	2	2	1	
P5	5	2	9	9	4	

Average WT=5.2ms

Non-Preemptive priority

Gantt Chart

P2	P5	P1	P3	P4

1 6 16 18 19

Case : if arrival time is same, then FCFS is used to break the tie.

Criteria: Non preemptive, 0 IS HIGH PRIORITY

Consider burst

Arrival for all is 0 time

Process	BT	priority	CT	TAT	WT	RT
P1	10	3	16	16	6	
P2	1	1	1	1	0	
P3	2	3	18	18	16	
P4	1	4	19	19	18	
P5	5	2	6	6	1	

Average
WT=8.2ms

OR

B) Ans- 8

RR

Gantt Chart(ready queue)

P1	P2	P3	P1	P4	P2	P1

2 4

Gantt Chart(Running Queue)

P1	P2	P3	P1	P4	P2	P1

2 4 6 8 9 11 12

Criteria: preemptive Time Quantum: 2ms

Rule: Put arraivall as per clock first, them premmpted is attached at end.

Process	BT	AT	CT	TAT	WT	RT
P1	3	0	12	12	7	0
P2	4	1	11	10	6	1
P3	2	2	6	4	2	2
P4	1	4	9	5	4	4

Average WT= 4.75 **Average TAT=7.75ms**