CONTINUOUS INTERNAL ASSESSMENT I – Question Bank
Regulation – 2022

Programme (s)	Semester	Course Code (s)	Course Title
B.TECH IT & B E CSE	IV	22CS403	Operating Systems

COURSE OUTCOMES	
C403.1	Identify the basic concepts and operations of operating systems.
C403.2	Illustrate the Process management concepts including scheduling, Inter process communication, deadlocks and multithreading in real world problems.
C403.3	Apply the concepts of memory management including Virtual Memory and Page Replacement to the issues that occur in Real time applications.
C403.4	Analyze the concepts related to file system interface and implementation.
C403.5	Describe the disk management, system protection and security mechanisms

PART A

1.	Does Context Switching improve CPU utilization? Justify your answer.
2.	List out the functions of operating system.
3.	Draw the Gantt chart for the following using preemptive priority scheduling.

Process	Arrival Time	Burst Time	Priority
P1	0	4	2
P2	1	2	4
P3	2	3	6
P4	3	5	10

4.	List out the process states available.
5.	Specify any four services provided by an operating system.
6.	Give the benefits of Synchronous and asynchronous communication.
7.	Differentiate pre-emptive and non-pre-emptive scheduling algorithms.
8.	Define semaphore and summarize its importance in operating system.
9.	List out the fields associated with Process Control Blocks.
10.	Justify why deadlocks cannot arise in a bounded buffer producer–consumers system.
11.	Synchronization tool that provides more sophisticated ways (than Mutex locks) for processes to synchronize their activities. Elucidate different kinds of operations that are possible on semaphore

12. Suppose that the following processes arrive for execution at the times indicated. Each process will run for the amount of time listed. In answering the questions, use preemptive scheduling, and base all decisions on the information you have at the time the decision must be made.

Process	Arrival Time	Burst Time
*P*1	0.0	8
*P*2	0.4	4
*P*3	1.0	1

a. What is the average turnaround time for these processes with the SJF scheduling algorithm?

13. Discuss how deadlock is possible with the dining-philosophers problem.

14. List the necessary conditions for the occurrence of a deadlock.

15. Classify the three general methods for passing parameters to the operating system with example.

16. What scheduling policy will you use for each of the following cases?
a. The processes arrive at large time intervals:
b. The system's efficiency is measured by the percentage of jobs completed.
c. All the processes take almost equal amounts of time to complete.

17. Elucidate the purpose of system calls.

18. Illustrate the use of fork and exec system calls.

19. Does Context Switching improve CPU utilization? Justify your answer.

20. Consider the following resource allocation state and compute the need matrix and the total no of resources.

Process	Current Allocation			Max			Available		
	A	B	C	A	B	C	A	B	C
P	0	1	1	6	7	7	3	2	3
Q	2	1	1	4	2	2	2	1	2
R	3	0	0	4	5	3			
S	3	2	2	3	2	2			
T	4	3	2	5	4	5			

21. Mention the main advantage of multiprogramming.

22. List few crucial operating system services.

23. Differentiate between a client-server and distributed computing environment.

24. Infer the concept of system calls and give an example of a system call for any two categories.

25. Compare process virtual machine and system virtual machine

26. Label the data fields associated with Process Control Blocks.

27. Defend timesharing system differs from a multiprogramming system. If so, how?

28. State the advantage of using tightly coupled systems

29.	*#include<stdio.h>* *int main ()* *{* *.* *.* *printf ("Greetings!");* *.* *.* *return 0;* *}* Interpret the above C program using the necessary system call.
30.	Draw the process creation using *fork()* system call.
31.	Sketch the actions taken by a kernel to context-switch between processes
32.	Define race condition with an example
33.	Give an application of thread.
34.	Write down the necessary conditions that must hold simultaneously for a deadlock situation to arise in a system.
35.	Infer how the Convoy effect is implemented with an example
36.	Explain the code snippets for implementing Semaphore with no Busy waiting
37.	Context Switching between processes is expensive than context switching between threads. Justify.
38.	List any two advantages of using multithreading in software development.
39.	Identify synchronization mechanisms used for coordinating processes.
40.	Differentiate between a semaphore and a mutex in terms of their functionality.
41.	State the conditions in which a system encounters a deadlock.

PART B		
1.	i)	Describe about Operating system operations and discuss in detail about User and Abstract view of OS.
2.	i)	Consider the situation while designing the operating system structures, it is sometimes difficult to achieve a layered approach if two components of the operating system are dependent on each other. Which approach will be suitable for designing operating structures in this situation and justify. Also elaborate the other operating system structure design approaches with their benefits. Also list out some of the operating system services.
3.	i)	An operating system executes a variety of programs that run as a process. Process is a program in execution; process execution must progress in sequential fashion. Describe the process states with the help of state diagram. Define process control block and its role in the context switching of the process.

<table>
<tr><td>4.</td><td>i)</td><td>Consider a system consisting of four resources of the same type that are shared by three processes, each of which needs at most two resources. Show that the system is deadlock free.</td></tr>
<tr><td></td><td>ii)</td><td>Describe a solution for the dining philosophers problem using semaphores.</td></tr>
<tr><td>5.</td><td>i)</td><td>Illustrate the classical problems of synchronization in detail with suitable examples.</td></tr>
<tr><td>6.</td><td>i)</td><td>Discuss how wait() and signal() semaphore operations could be implemented in multiprocessor environments using the test and set instruction. The solution should exhibit minimal busy waiting. Develop pseudo code for implementing the operations.</td></tr>
<tr><td>7.</td><td>i)</td><td>Explain the methods for implementing Inter process Communication.</td></tr>
<tr><td>8.</td><td>i)</td><td>Consider the following set of processes, with the length of the CPU – burst time given in Milliseconds:
<table>
<tr><th>Process</th><th>Burst Time</th><th>Priority</th></tr>
<tr><td>P1</td><td>10</td><td>3</td></tr>
<tr><td>P2</td><td>1</td><td>1</td></tr>
<tr><td>P3</td><td>2</td><td>3</td></tr>
<tr><td>P4</td><td>1</td><td>4</td></tr>
<tr><td>P5</td><td>5</td><td>2</td></tr>
</table>
The processes have arrived in the order P1, P2, P3, P4, P5, all at time 0.
a. Visualize Gantt charts illustrating the execution of these processes using FCFS, SJF, Priority and RR (Time Slice = 1) scheduling
b. Determine the turnaround time for each process
c. Calculate the waiting time for each of the process</td></tr>
<tr><td>9.</td><td>i)</td><td>Discuss about Deadlock characteristics and methods for handling deadlock in detail.</td></tr>
<tr><td>10.</td><td>i)</td><td>Consider a system consists of 5 processes P1 through P5; and 3 resource types:
The following Snapshot of the system has been taken:
<table>
<tr><td rowspan="2">Process</td><td colspan="3">Maximum</td><td colspan="3">Allocation</td><td colspan="3">Available (Work)</td></tr>
<tr><td>R1</td><td>R2</td><td>R3</td><td>R1</td><td>R2</td><td>R3</td><td>R1</td><td>R2</td><td>R3</td></tr>
<tr><td>P1</td><td>7</td><td>5</td><td>3</td><td>0</td><td>1</td><td>0</td><td>3</td><td>3</td><td>2</td></tr>
<tr><td>P2</td><td>3</td><td>2</td><td>2</td><td>2</td><td>0</td><td>0</td><td colspan="3" rowspan="4"></td></tr>
<tr><td>P3</td><td>9</td><td>0</td><td>2</td><td>3</td><td>0</td><td>2</td></tr>
<tr><td>P4</td><td>2</td><td>2</td><td>2</td><td>2</td><td>1</td><td>1</td></tr>
<tr><td>P5</td><td>4</td><td>3</td><td>3</td><td>0</td><td>0</td><td>2</td></tr>
</table>
i) Find ‘Available’ resources for each processes.
ii) What is the content of ‘need’ matrix?
iii) Is the system in a safe state?
iv) If a request from P2 arrives for (1,0,2), can the request be granted immediately?</td></tr>
<tr><td>11.</td><td>i.</td><td>Summarize the services and system components of operating systems briefly.</td></tr>
<tr><td>12.</td><td>I</td><td>Explain the various threading models and threading issues involved in a process.</td></tr>
<tr><td></td><td>ii</td><td>Discuss the critical section problem. State and discuss the basic requirements of critical problem solution and methods available to solve it.</td></tr>
<tr><td>13.</td><td>i</td><td>Assume that the following processes arrive in the time given, with the length of the CPU-burst time given in milliseconds.
<table>
<tr><th>Job</th><th>Arrival Time (ms)</th><th>Burst time (ms)</th><th>Priority (ms)</th></tr>
<tr><td>P1</td><td>0</td><td>15</td><td>4</td></tr>
</table></td></tr>
</table>

P2	1	10	3
P3	3	8	5
P4	5	5	2
P5	7	4	1
P6	8	2	6

) Give the Gantt chart illustrating the execution of processes using FIFO, Preemptive SJF scheduling, Preemptive priority scheduling and Round robin algorithm(time quantum= 5 ms)

b) Calculate the average waiting time and average turn-around time for each of the above algorithm

14. i Explain process state diagram, PCB and the operations performed on processes

15. i Assume that there are 5 processes, P0 through P4, and 4 types of resources. At T0 we have the following system state:

Process	**Allocation**	**Max**	**Available**
	A, B, C, D	A, B, C, D	A, B, C, D
P0	0 1 1 0	0 2 1 0	1 5 2 0
P1	1 2 3 1	1 6 5 2	
P2	1 3 6 5	2 3 6 6	
P3	0 6 3 2	0 6 5 2	
P4	0 0 1 4	0 6 5 6	

Using Banker's algorithm, answer the following questions:-

i) Calculate the number of resources of A, B, C, D

ii) Find the contents of need matrix.

iii) Is the system in a safe state? If so, show the sequence of processes executing in order. If not, justify which process cannot complete and the reason.

16. i Assume the situation where we have a file share between many people. If one of the people tries editing the file, no other person should be reading or writing at the same time. Otherwise, changes will not be visible to him/her. However, if some person is reading the file, then others may read it at the same time.

i) Identify the application used.

ii) Consider the following problem parameters to solve this situation.

- One set of data is shared among a number of processes
- Once a writer is ready, it performs its write. Only one writer may write at a time,
- If a process is writing, no other process can read it.
- If atleast one reader is reading, no other process can write.
- Readers may not write and only read.

ii Construct a Resource Allocation Graph for the following scenario. At time't', process P1 requests for a resource R1, P2 requests R2. Both the resources are available and they are allocated to the requesting process. At time t1, where t1>t,

		both the processes are still holding the resources, however P1 requests for R2 which is held by P2, P2 requests R1 held by P1. Will there be a deadlock? Justify your answer with the conditions.
17.	i	Elucidate the various methods and mechanisms involved in implementing inter process communication.
18.	i	Write short notes on i) Loosely coupled systems ii) Scheduling queues iii) Real time systems iv) Variation in FCFS Scheduling
19.	i.	Considering the situation while designing the operating system structures, it is sometimes difficult to achieve a layered approach if two components of the operating system are dependent on each other. Which approach will be suitable for designing operating structures in this situation and justify. Also elaborate the other operating system structure design approaches with their benefits.
	ii.	Using simple system calls as examples (e.g. getpid, or fork), describe what is generally involved in providing the result, from the point of invoking the system call to what it returns.
20.	i	Discuss the classic problems of synchronization with examples.
21.	i)	An operating system executes a variety of programs that run as a process. The process is a program in execution; process execution must progress sequentially. Describe the process states with the help of a state diagram. Define the process control block and its role in the context switching of the process.
22.	i)	Discuss in detail about multithreading models. Cite the necessity for process synchronization. Illustrate a software-based solution to Producer-Consumer Problem using Shared Memory and Message Passing
23.	i)	Consider the following set of processes, with the length of the CPU burst given in milliseconds: **Process** **Burst Time** **Priority** P1 2 2 P2 1 1 P3 8 5 P4 4 4 P5 5 3 The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time 0. a. Draw four Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: FCFS, SJF, Priority (a larger priority number implies a higher priority), and RR (quantum = 4). b. What is the turnaround time of each process for each of the scheduling algorithms in part a? c. What is the waiting time of each process for each of these scheduling algorithms? d. Which of the algorithms results in the minimum average waiting time (over all processes)?
24.	i)	Assume that the following processes arrive in the time given, with the length of the CPU-burst time given in milliseconds. Job Arrival Time (ms) Burst time (ms) P 0 15

		Q 3 6 R 2 10 S 4 7 T 1 13 a) Give the Gantt chart illustrating the execution of processes using FIFO, Preemptive and non-preemptive SJF scheduling, and Round robin algorithm (time quantum= 4 ms). Note highest priority is 5 and the lowest priority is 1. b) Calculate the average waiting time and average turn-around time for each of the above algorithms.
25.	i)	Briefly explain the hardware solutions used to solve critical – sections.
	ii)	Discuss the methods and mechanisms used to implement interprocess-communications
26.	i)	Assume a multithreaded application uses only reader-writer locks for synchronization. Applying the four necessary conditions for deadlock, is deadlock still possible if multiple reader-writer locks are used? Can a system detect that some of its processes are starving? If you answer “yes,” explain how it can. If you answer “no,” explain how the system can deal with the starvation problem
27.	i)	Explain readers-writers problem. Give its solution with semaphore.
	ii)	Summarize the responsibility of operating systems concerning system components. Write short notes on Tightly coupled systems & Monitors
28.	i)	Various types of errors are generated when programmers use Semaphores incorrectly to solve critical-section problems. With a schematic view of a monitor and its condition variables, discuss the usage of Monitors in dealing with such errors. Also, illustrate monitor concepts by presenting deadlock-free solutions to Dining – Philosophers’ problem.
29.	i)	Consider the following snapshot of a system: **Allocation** **Max** **Available** **A B C D** **A B C D** **A B C D** P_0 0 0 1 2 0 0 1 2 1 5 2 0 P_1 1 0 0 0 1 7 5 0 P_2 1 3 5 4 2 3 5 6 P_3 0 6 3 2 0 6 5 2 P_4 0 0 1 4 0 6 5 6 Answer the following questions using the banker’s algorithm: a. What is the content of the matrix Need? b. Is the system in a safe state? c. If a request from process P_1 arrives for (0,4,2,0), can the request be granted immediately?
30.	i)	Consider the deadlock situation that occurs in a dining philosopher’s problem, when the philosophers obtain the chopsticks one at a time. Analyze the four necessary conditions for deadlock for this setting. • If all five philosophers are hungry simultaneously, and each of them pickup one chopstick, then a deadlock situation occurs because they will be waiting for another chopstick forever. • Mutual exclusion • Hold and wait

<table>
<tr><td></td><td></td><td>• No preemption
• Circular wait</td></tr>
<tr><td>31.</td><td>i)</td><td>Explain about the emerging trends and technologies shaping the future of computing environments, and how might they impact businesses and individuals.</td></tr>
<tr><td>32.</td><td>i)</td><td>Distinguish between blocking and non-blocking system calls. Enumerate the advantages and disadvantages of each approach</td></tr>
<tr><td></td><td>ii)</td><td>Explain the purpose and importance of System Calls in detail with examples.</td></tr>
<tr><td>33.</td><td>i)</td><td>Summarize the activities of an operating system with regard to process, file and memory management.</td></tr>
<tr><td>34.</td><td>i)</td><td>Discuss the concept of virtualization and its role in modern operating system structures.</td></tr>
<tr><td>35.</td><td>i)</td><td>Elucidate on multitasking, and how does the operating system achieve it? Provide examples of how multitasking benefits users.</td></tr>
<tr><td>36.</td><td>i)</td><td>Illustrate the concept of boot process of a computer, including the various stages involved.</td></tr>
<tr><td></td><td>ii)</td><td>Discuss the concept of process states (e.g., running, ready, blocked) in a process control block (PCB). How does the OS maintain and make transition between these states?</td></tr>
<tr><td>37.</td><td>i)</td><td>Consider the set of processes given that arrive at time zero. The length of the CPU burst time given in millisecond. Calculate the average waiting time, average turnaround time and throughput using FCFS.

<table><tr><th>Process</th><th>Burst Time(ms)</th></tr><tr><td>P1</td><td>5</td></tr><tr><td>P2</td><td>24</td></tr><tr><td>P3</td><td>16</td></tr><tr><td>P4</td><td>10</td></tr><tr><td>P5</td><td>3</td></tr></table></td></tr>
<tr><td>38.</td><td>i)</td><td>Consider the set of processes given. Calculate the average waiting time, average turnaround time and throughput using Priority Scheduling

<table><tr><th>Process</th><th>Burst Time(ms)</th><th>Priority</th></tr><tr><td>P1</td><td>5</td><td>2</td></tr><tr><td>P2</td><td>24</td><td>4</td></tr><tr><td>P3</td><td>16</td><td>5</td></tr><tr><td>P4</td><td>10</td><td>1</td></tr><tr><td>P5</td><td>3</td><td>3</td></tr></table>.</td></tr>
<tr><td>39.</td><td>i)</td><td>System consists of 4 processes P1 through P4 ; and 2 resource types: R1 (4 instances) and R2 (3 instances) The following Snapshot of the system has been taken:</td></tr>
</table>

		<table><tr><th></th><th colspan="2">Maximum</th><th colspan="2">Allocation</th></tr><tr><th></th><th>R1</th><th>R2</th><th>R1</th><th>R2</th></tr><tr><td>P1</td><td>2</td><td>2</td><td>0</td><td>1</td></tr><tr><td>P2</td><td>2</td><td>1</td><td>1</td><td>0</td></tr><tr><td>P3</td><td>1</td><td>2</td><td>0</td><td>1</td></tr><tr><td>P4</td><td>3</td><td>3</td><td>2</td><td>0</td></tr></table> i) Is the current allocation in a safe state? ii) If a request from P1 arrives for (0,1). Can the request be granted immediately?
40.	i)	Explain the concept of deadlock prevention and how it aims to eliminate one or more of the necessary conditions for deadlock.