

Define operating Systems. Explain the dual-mode operating system with a neat diagram.

An operating system is system software that acts as an intermediary between a user of a computer and the computer hardware. It is software that manages the computer hardware and allows the user to execute programs in a convenient and efficient manner. (2 Marks)

The system can be assumed to work in two separate modes of operation:

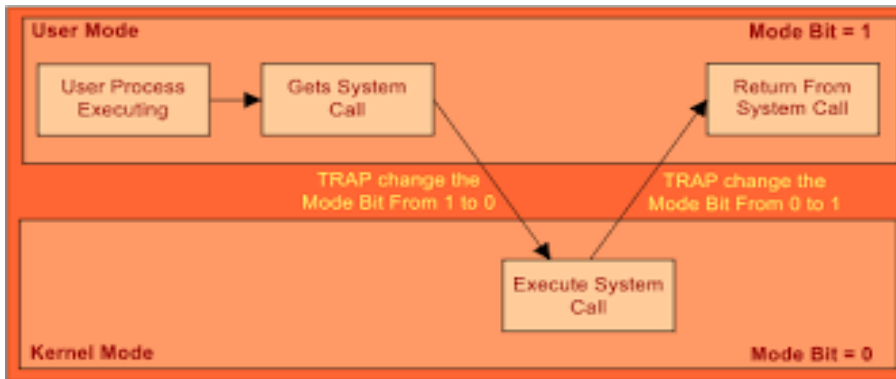
1. User mode
2. Kernel mode

User Mode: This is a restricted mode where applications and user-level programs run. In user mode, programs have limited access to system resources, ensuring they cannot directly interact with critical hardware or OS functions. Any request for resources or services outside this restricted set must be made through the OS via system calls. This limitation helps protect the system from accidental or malicious damage by user applications.

Kernel Mode (or Supervisor Mode): This mode has full access to all system resources, including hardware (e.g., memory, CPU) and privileged instructions. The operating system runs in kernel mode, where it can execute any instruction and access any memory location. Kernel mode is essential for performing low-level tasks like process management, memory allocation, and handling I/O operations.

(3 Marks)

Diagram (2 Marks)



[7]

1

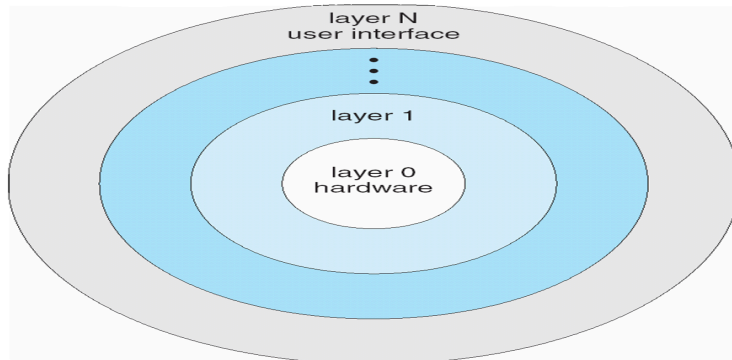
L2

	<p>What are system calls? List out its types.</p> <p>System Calls are interfaces through which a program requests a service from the operating system's kernel. They allow user-level processes to interact with the system hardware and resources indirectly by invoking functions within the OS. (1 mark)</p> <p>List out its types.</p> <p>b Process Control</p> <p>File Management</p> <p>Device Management</p> <p>Information Maintenance</p> <p>Communication (2 marks)</p>	[3]	1	L2
2 a	<p>Explain the services of the operating system that are helpful for the user and the system.</p> <p>Operating systems provide essential services that are beneficial for both users and the system.</p> <ol style="list-style-type: none"> 1. User Interface: Operating systems provide a user interface, which could be command-line (CLI) or graphical (GUI), allowing users to interact with the system easily. This improves usability and efficiency for end-users. 2. Program Execution: The OS enables the loading, execution, and termination of applications. It manages resources to run multiple programs simultaneously, providing a stable environment for users to run applications effectively. 3. File System Management: The OS offers services for file creation, deletion, reading, and writing, as well as directory structures to organize and manage files. This service allows users to store and retrieve data in a structured manner. 4. I/O Operations: Operating systems manage input and output devices, abstracting hardware complexity from the user. It controls access to printers, disks, and other I/O devices, ensuring smooth communication between hardware and applications. 5. Error Detection and Handling: The OS monitors for hardware and software errors and takes necessary actions, such as logging or alerting users. This service improves system stability and protects user data from corruption or loss due to errors. <p>These services improve the efficiency and usability of the system for users and ensure that resources are managed effectively to maintain overall system stability. (5*1 = 5 Marks)</p>	[5]	1	L2

Explain the layered approach of operating system structure with a supporting diagram.

The operating system is divided into a number of layers (levels), each built on top of lower layers. The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface.

With modularity, layers are selected such that each uses functions (operations) and services of only lower-level layers **(3 Marks)**



Advantages: (2 Marks)

- Easier Debugging and Testing: Each layer can be tested independently.
- Flexibility: Different layers can be modified or replaced without affecting others.
- Improved Reliability: The isolation of functions in layers reduces the chances of system-wide failures.

b

[5]

1

L2

Discuss message passing and the shared

Interprocess Communication- Processes executing may be either co-operative or independent processes Cooperating processes require some type of inter-process communication.

This is allowed by two models:

1. SharedMemorysystems
2. Message passing systems.

3

a

Shared Memory : Shared Memory is faster once it is set up, because no system calls are required and access occurs at normal memory speeds. Shared memory is generally preferable when large amountsof information must be shared quickly on the same computer. Explain with Diagram **(3 Marks)**

Message Passing requires system calls for every message transfer, and is therefore slower, but it is simpler to set up and works well across multiple coy8huomputers.

Messaging is generally preferable when the amount and/or frequency of data transfers is small. Explanation with Diagram .**(3 Marks)**

[6]

1

L2

	<p>Explain the states of a process with a transition diagram.</p> <p>Process : A process is a program under execution. (1 Mark)</p> <p>Five states: new, ready, running, blocked, and exit. (1 Mark)</p> <p>Diagram (2 Marks)</p>	[4]	2	L2
a	<p>Explain different scheduling criteria used in the computing scheduling mechanism.</p> <p>Burst Time , Waiting Time , TurnAround Time , Response Time , Throughput (2 Marks)</p>	[2]	2	L2

Processes	Arrival Time	Burst Time	Priority
P1	0	8	3
P2	1	4	2
P3	2	9	1
P4	3	5	4

Calculate the average waiting time and the average turnaround time by drawing the Gantt chart using **SJF and the Priority scheduling algorithm** (Preemptive Scheduling)

SJF

P	C	T	W
rocesses	omputing time	urnaround time	aiting time
P1	17	17	9
P2	54	40	
P3	264	245	15
P4	107	72	

b

[8] 2 L3

GANTT CHART:



Average Turn Around Time=13
 Average Waiting Time=6.5 (4 Marks)

Priority scheduling algorithm

P	A	B	C	T	W
rocesses	rrival time	urst time	omputing time	urnaround time	aiting time
P1	0	8	8	8	8
P2	1	4	4	4	4
P3	2	9	9	9	9
P4	3	5	5	5	5

			m	m	
	e	e			
P	0	8	2	2	1
1			1	1	3
P	1	4	1	1	9
2			4	3	
P	2	9	1	9	0
3			1		
P	3	5	2	2	1
4			6	3	8

P	P	P	P	P	P
1	2	3	2	1	4

1 2 11 14 21 26

Average Turn Around Time=16.5
Average Waiting Time=10 (4 Marks)

Define deadlock. Explain different methods to recover from deadlocks.

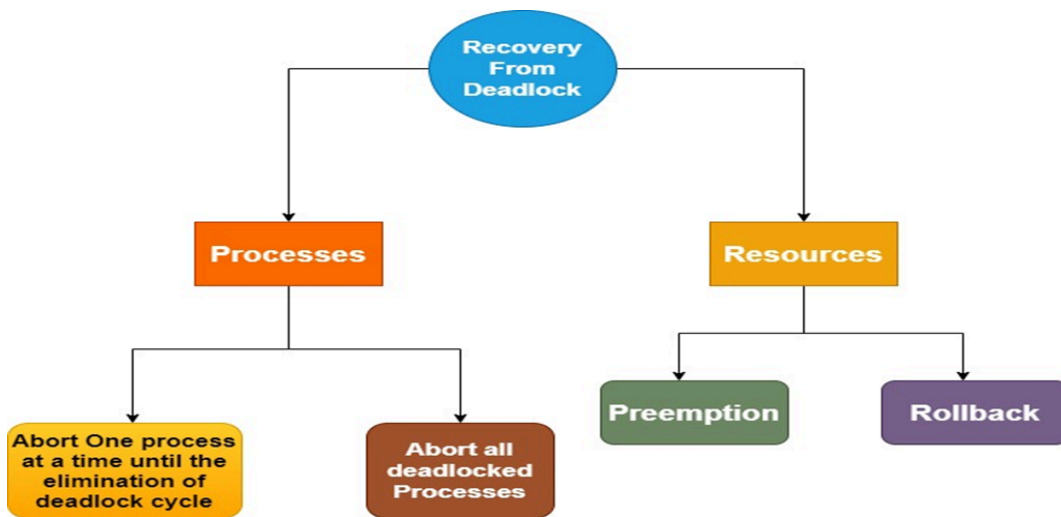
A process requests resources and if the resources are not available at that time the process enters into waiting state. The waiting process is never again able to change state because the resource it has requested are held by other waiting processes. This situation is called a deadlock (1 Mark)

5 a

[2]

3

L2



(1 Mark)

Given Matrices

	<u>Allocation Matrix</u> (NO of the allocated resources By a process)				<u>Max Matrix</u> Max resources that may be used by a process				<u>Available Matrix</u> Not Allocated Resources			
	A	B	C	D	A	B	C	D	A	B	C	D
P ₀	0	1	1	0	0	2	1	0	1	5	2	0
P ₁	1	2	3	1	1	6	5	2				
P ₂	1	3	6	5	2	3	6	6				
P ₃	0	6	3	2	0	6	5	2				
P ₄	0	0	1	4	0	6	5	6				

b

[8]

3

L3

Assume that there are 5 processes, P₀ through P₄, and 4 types of resources .

What will be the content of the Need matrix?

Is the system in a safe state? If yes, then what is the safe sequence?

Need Matrix (2 Marks)

The system is in a safe state and the processes will be executed in the following order:

P₀,P₃,P₄,P₁,P₂ (6 Marks)

What is a semaphore? State a Dining Philosopher problem gives a solution using semaphore.

The various hardware based solutions to the critical section problem are complicated for application programmers to use. To overcome this difficulty a synchronization tool called semaphore is used Semaphore S is an integer variable, apart from initialization is accessed only through two standard atomic operations **(2 Marks)**

Dining Philosopher Problem : Consider five philosophers sitting around a table, in which there are five chopsticks evenly distributed and an endless bowl of rice in the center, as shown in the diagram below. (There is exactly one chopstick between each pair of dining philosophers.)

These philosophers spend their lives alternating between two activities: eating and thinking. When it is time for a philosopher to eat, it must first acquire two chopsticks - one from their left and one from their right. When a philosopher thinks, it puts down both chopsticks in their original locations. **(3 Marks)**

6

[10]

3

L2

Dining-Philosophers Problem (Cont.)

- The structure of Philosopher is:

```
do {  
    wait ( chopstick[i] ); // left chopstick  
    wait ( chopstick[ (i + 1) % 5] ); //right  
    chopstick  
  
    Critical Section // eat  
    signal ( chopstick[i] ); //release left chopstick  
    signal ( chopstick[ (i + 1) % 5] ); // release right  
    chopstick  
  
    Remainder Section // think  
  
} while (TRUE);
```

(5 Marks)

