C0-R4.B2: OPERATING SYSTEM

NOTE :

1.	**Answer question 1 and any FOUR from questions 2 to 7.**
2.	**Parts of the same question should be answered together and in the same sequence.**

Time: 3 Hours **Total Marks: 100**

1. (a) How do we achieve event ordering in distributed systems? How does it helps in process coordination?

(b) What is the main advantage of the layered approach to system design? What are the disadvantages of using the layered approach?

(c) Define race condition. Why is it necessary to avoid race condition?

(d) How does the kernel mode and user mode function for rudimentary form of protection (security) system?

(e) Why do some systems store the operating system in firmware, while others store it on disk?

(f) What is the main difficulty that a programmer must overcome in writing an operating system for a real-time environment?

(g) Give three programming examples in which multithreading provides better performance than a single-threaded solution.

(7×4)

2. (a) Some CPUs support more than two modes of operation. Give two possible uses of these multiple modes.

(b) Consider the following set of processes:

Process	Burst time(ms)	Arrival time(ms)
P1	7	0
P2	5	2
P3	6	4
P4	3	5
P5	4	7

(i) Draw Gantt chart illustrating the execution of the given processes using Round robin policy (time slice = 3ms) and shortest remaining time scheduling algorithms.

(ii) Compute turnaround and total waiting time for each process for both algorithms and find the algorithm with best performance w.r.t. average waiting time.

(6+[6+6])

3. (a) What resources are used when a thread is created? How do they differ from those used when a process is created?

(b) Illustrate how a binary semaphore can be used to implement mutual exclusion among n processes.

(c) To build a robust distributed system, you must know what kinds of failures can occur.

(i) List three possible types of failure in a distributed system.

(ii) Specify which of the entries in your list also are applicable to a centralized system.

(6+6+6)

4. (a) Suppose that a system is in an unsafe state. Show that it is possible for the processes to complete their execution without entering a deadlock state.

(b) Under what circumstances do page faults occur? Describe the actions taken by the operating system when a page fault occurs.

(c) Discuss the hardware support that are required in demand paging.

(6+6+6)

5. (a) Consider a logical address space of 64 pages with 1024 words per page, mapped onto a physical memory of 32 frames.

(i) How many bits are there in the logical address?

(ii) How many bits are there in the physical address?

(b) Give two reasons why caches are useful? What problems do they solve? What problems do they cause? If a cache can be made as large as the device for which it is caching, why not make it that large and eliminate the device?

(c) Capability lists are usually kept within the address space of the user. How does the system ensure that the user cannot modify the contents of the list?

(6+6+6)

6. (a) What is Virtualization? How does virtual machines improve the system utilization as compared to traditional computer systems?

(b) Assume that a distributed system is susceptible to server failure. What mechanisms would be required to guarantee the "exactly once" semantics for execution of RPCs?

(c) Suppose that a disk has 2,000 cylinders numbered 0 to 1999. The disk head is currently at cylinder 1250 and previous request was at 800. The queue of pending requests in FIFO order is:

1470, 750, 600, 1948, 1509, 130, 500

Calculate and show total head movements for the following disk scheduling algorithms:

(i) FCFS

(ii) SCAN

(6+6+6)

7. (a) Distinguish between the client–server and peer-to-peer models of distributed systems.

(b) "In distributed environment, ensuring atomicity of a transaction is difficult". Comment. Which module of the distributed operating system ensures atomicity of a transaction? Write down its basic responsibilities.

(c) Given two sets Process set (P) and Resources set (R)

$P = \{P_1, P_2, P_3, P_4\}$

$R = (R_1, R_2\}$

Two instances of resource type R_1 and R_2 each exist in the system.

$E = \{P_1 \rightarrow R_1, R_1 \rightarrow P_2, R_1 \rightarrow P_3, R_2 \rightarrow P_1, P_3 \rightarrow R_2, R_2 \rightarrow P_4\}$

Using the process requests for resources and allocated resources to the processes given in set E, draw a resource allocation graph and describe whether the system can go into a deadlock state or not.

(6+6+6)