Seat No.: _____ Enrolment No._____

GUJARAT TECHNOLOGICAL UNIVERSITY

MCA - SEMESTER– III EXAMINATION – SUMMER - 2017

Subject Code:2630004 **Date: 05-06-2017**

Subject Name: Operating Systems

Time: 02:30 pm - 05:00 pm **Total Marks: 70**

Instructions:

1. **Attempt all questions.**
2. **Make suitable assumptions wherever necessary.**
3. **Figures to the right indicate full marks.**

Q.1 **(a)** Consider the following set of processes, with the length of the CPU-burst time given in milliseconds: **07**

Process	Burst Time	Priority
A	10	3
B	1	1
C	2	3
D	1	4
E	5	5

The processes are assumed to have arrived in the sequential order all at time 0.

a) Draw four Gantt charts describing execution of these processes using FCFS, SPN, a non preemptive priority (a smaller priority number implies a higher priority), and RR (quantum=1) scheduling.
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 the scheduling algorithms?

(b) A process contains five virtual pages on disk and is assigned a fixed allocation of four page frames in main memory. The following page trace occurs: **07**

1, 2, 3, 5, 4, 5, 2, 3, 4, 5, 1, 3, 2, 4, 5.

a) Show the successive pages residing in the four frames using the LRU replacement policy. Compute the hit ratio in main memory. Assume that the frames are initially empty.
b) Repeat part a) for the FIFO replacement policy.
c) Compare the two hit ratios and comment on the effectiveness of using FIFO to approximate LRU with respect to this particular trace.

Q.2 **(a)** Discuss the deadlock avoidance using Banker's algorithm. Also discuss data structure for implementing this algorithm. **07**

(b) Discuss ULT and KLT in detail. **07**

OR

(b) Explain Six and Seven-state Process Models mentioning all the transitions. **07**

Q.3 **(a)** What is paging? Explain the following terms : **07**

(i) page
(ii) frames
(iii) page table

Also explain with examples how Logical –to Physical Address Translation takes place.

(b) What is Monitor? Explain the solution to the Bounded-Buffer Producer/Consumer Problem using a Monitor. **07**

OR

Q.3 **(a)** Explain Internal Fragmentation and External Fragmentation. **07**

(b) State dining philosopher problem. **07**

Q.4 **(a)** What is Real time Scheduling? Explain the Characteristics of Real –Time Operating System. **07**

(b) Explain the different methods for Secondary Storage Management **07**

OR

Q.4 **(a)** Suppose that a disk has 200 tracks, numbered 0 to 199. The disk head is currently located at track 100. The random queue of pending requests, in FIFO order, is **07**

55, 58,39, 18, 90, 160, 150, 38, 184

Starting from the current head position, what is the total distance (in tracks) that the disk head moves to satisfy all the pending requests, for each of the following disk-scheduling algorithms? Also show the next track accessed with each new request serviced.

a. FIFO b. SSTF c. SCAN

(b) Explain four general approaches of thread scheduling in detail. **07**

Q.5 **(a)** Define: **07**

1. Jacketing
2. Response time
3. Thrashing
4. Dispatcher
5. Multiprocessing
6. Kernel
7. Spawning

(b) What is RAID? Briefly discuss its levels. **07**

OR

Q.5 **(a)** Do as directed: **07**

1. Monolithic kernel architecture assigns only a few essential functions to the kernel, including address spaces, inter-process communication and basic scheduling. True/False?
2. A consumable resource is one that can be safely used by only one process at a time and is not depleted by that use. True/False?
3. A semaphore that does not specify the order in which processes are removed from the queue is a _____. (Fill in the Blank)
4. The _____ is a directed graph that depicts a state of the system of resources and processes, with each process and each resource represented by a node. (Fill in the Blank)
5. Give full forms of: DMA and ORB
6. What is instruction trace?
7. Differentiate persistent and non-persistent binding.

(b) Differentiate between: **07**

1. Fat client vs Thin Client
2. Synchronous RPC vs Asynchronous RPC

****