No of Pages : 5 **Course Code : 15XW44 / 15XT44 /15XD44**

Roll No:
(To be filled in by the candidate)

PSG COLLEGE OF TECHNOLOGY, COIMBATORE - 641 004

SEMESTER EXAMINATIONS, APRIL 2018

MSc – SOFTWARE SYSTEMS / TCS / DATA SCIENCE Semester: 4

15XW44 / 15XT44 / 15XD44 OPERATING SYSTEMS

Time : 3 Hours **Maximum Marks :100**

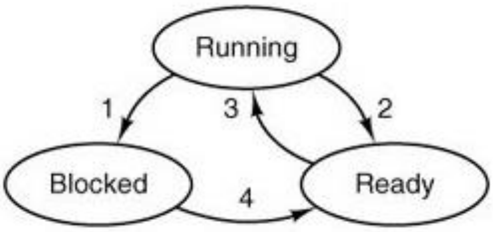
INSTRUCTIONS:
1. Answer **ALL** questions. Each question carries 20 Marks.
2. Subdivision (a) carries 3 marks each, subdivision (b) carries 7 marks each and subdivision (c) carries 10 marks each.
3. Course Outcome : Table

Qn.1	CO 1	Qn.2	CO 2	Qn.3	CO 3	Qn.4	CO 4	Qn.5	CO 5

1. a) What is a Thread? How threads can enhance responsiveness of a single program that contains a GUI and two long processes: one is calculating values and the other is doing disk-related operations?

   b) i) How does the mixture of I/O bound processes with CPU bound processes maximize system utilization? Why is this more important in batch systems than in multiprocessing systems?

   ii) In the following graph you can see the three scheduling states a process can be in. Write an example event that triggers each transition.



   c) Consider two processes, each with two CPU bursts with one I/O burst in between. Process 1 has a CPU burst of 9 units followed by an I/O burst of 7 units and a CPU burst of 6 unit.Process 2 has a CPU burst of 2 units followed by an I/O burst of 1 units and a CPU burst of 5 units.Suppose that Process 1 arrives in the ready queue just before Process 2 and just after Process 2 arrives the process that was in the CPU terminates. No other processes are in the system. For each of the scheduling algorithms below create Ganttcharts as given below. Fill each box with the state of the corresponding process. Use R for running, W for waiting/blocked, and D for ready. Calculate the waiting times and CPU utilization (as a fraction) for each process and fill in the table below.

- First-Come/First-Served
- Shortest Job First (non-preemptive)
- Preemptive Shortest Job First
- Round Robin with a quantum of 3.

Gantt Charts:

a) FCFS	5	10	15	20	25	30
Process 1						
Process 2						

b) SJF	5	10	15	20	25	30
Process 1						
Process 2						

c) PSJF	5	10	15	20	25	30
Process 1						
Process 2						

d) RR 3	5	10	15	20	25	30
Process 1						
Process 2						

Algorithm	CPU Utilization	Waiting Time			Time Finished	
		Process 1	Process 2	Average	Process 1	Process 2
a) FCFS						
b) SJF						
c) PSJF						
d) RR 3						

2. a) How does the use of preemption and priority levels affect the behavior and the performance of CPU scheduling algorithms?

   b) i) With regard to process synchronization what is meant by race condition? In order to avoid race conditions, what four conditions must be satisfied?

   ii) Two processes, P_0 and P_1,are to be run and they update a shared variable. This update is protected by Petersons solution to the mutual exclusion problem.

   - Show Petersons algorithm and show the truth table for the part of the algorithm which dictates if a process is allowed to enter its critical region.
   - P_0 attempts to enter its critical region. Show the state of the variables that are created/updated. Will P_0 be allowed to enter its critical region? Why?
   - P_1 attempts to enter its critical region. Show the state of the variables that are created/updated. Will P_1 be allowed to enter its critical region? Why?
   - P_0 leaves its critical region. What effect does this have on the variables?
   - Assume no processes are running and P0 and P1 try to enter their critical region at exactly the same time. What will happen?

- Critically evaluate the Petersons algorithm with reference to the conditions you listed in part (i)

c) Consider the following snapshot of a system:

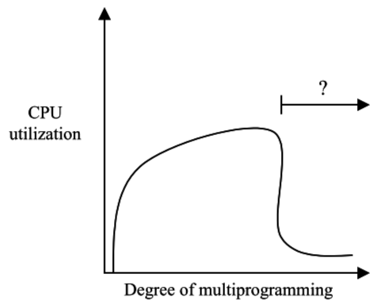
	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P_0	0	0	1	2	0	0	1	2	1	5	2	0
P_1	1	0	0	0	1	7	5	0				
P_2	1	3	5	4	2	3	5	6				
P_3	0	6	3	2	0	6	5	2				
P_4	0	0	1	4	0	6	5	6				

Answer the following questions using the Banker's algorithm:

- Is the system in a safe state? Justify.
- If a request from process P_1 arrives for (0, 4, 2, 0), can the request be granted immediately?

3. a) What is the main purpose of swapping? Can a process be run by an Operating System if some of its pages are swapped out?

b) i) Consider the graph given below that shows a plot of "degree of multiprogramming" on X-axis versus "CPU utilization" on Y-axis. What is the phenomenon observed by a sudden dip in the graph (as shown with the question mark) called? How can it be prevented?



ii) Consider the following factors:

- internal fragmentation
- process table size
- I/O overhead
- locality of reference

Which of these factors could be used to argue for a large page size, and which could be used to argue for a smaller page size? Why?

c) Describe the operation of Translation Look aside Buffers (TLBs) in a paged operating system.
State the sequence of events when

- a TLB is 'hit';
- a TLB is 'missed' but the required page is in main memory;
- a TLB is 'missed' but the required page is not in main memory.

Show clearly how the various components of the system fit together.
In a simple paged system with only one level of page tables the access times are as follows.
TLB HIT:

To read associative memory 1ns
To read main memory 6ns

TLB MISS:

To read associative memory 1ns
To add the page number to the page table origin register 2ns
To read page table 6ns
To read main memory 6ns

- Calculate the effective access time for a hit rate of 93 per cent.
- If the page fault service time is 1.2 ms, what is the maximum page fault rate that the system can tolerate without incurring more than a15 per cent degradation?

4. a) You are responsible for maximizing the utilisation of computing resources for a large scale multiuser system. You are receiving complaints from most users about response and turnaround times when accessing the server. Investigations reveal that the CPU is being utilised only 20% of the time. Which of the following could be reasons for the poor CPU utilisation? Why?
   - insufficient memory
   - frame size too small
   - frame size is too large
   - too much memory
   - one CPU is not enough.

   b) i) Imagine a system with 64KB of physical memory. Suppose the OS and its data structures occupy 16KB and there are two processes, one with 19KB of code and 5KB of data and the other with 15KB of code and 8KB of data. Can these processes fit in physical memory if 2KB pages are used? What if pure segmentation is used?

   ii) How does a buddyallocator allocate memory?Assume a computer with a memory size of 256K, initially empty. Requests are received for blocks of memory of 5K, 25K, 35K and 20K. Show how the buddy system would deal with each request, showing the memory layout at each stage and the status of the lists at the end. After allocating all the processes, what would be the effect of the 25K process terminating and returning its memory?

   c) When do page faults occur? List the actions taken by the operating system when a page fault occurs. Consider the following sequence of memory references from a 460 bytes size program:

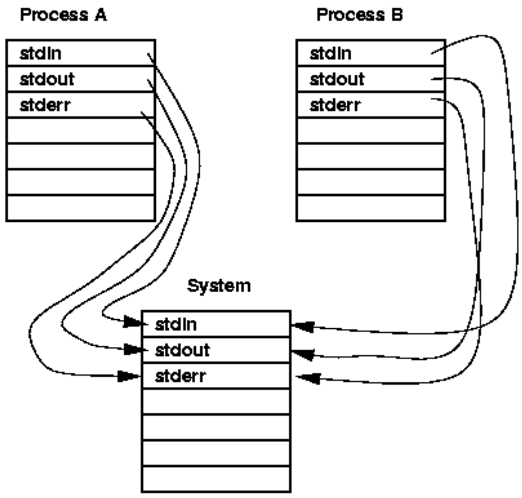
   10, 11, 104, 170, 73, 309, 185, 245, 246, 434, 458, 364.

   - Give the reference string, assuming a page size of 100 bytes.
   - Find the page-fault rate for the reference string in part (i), assuming 200 bytes (2 frames) of main memory is available to the program and a FIFO replacement algorithm is used.
   - Calculate the page fault rate if the page replacement algorithm is LRU.
   - Would increasing the number of frames always decrease the number of page faults for a particular reference string for FIFO? for LRU? Why or why not?

5. a) What is the reason that all I/O operations are done in privileged mode?

   b) i) Why do device drivers need interrupt handlers and what do the interrupt handlers accomplish?

   ii) Consider two independent processes A and B that have just started executing along with per-process and system open file tables with the following initial contents:



- Each of the following pieces of data must be kept in either the per-process open file table or the system open file table. Determine which of the two locations is most appropriate for each piece of information. Justify your answer.
  - Mode (e.g., RDONLY, WRONLY, or RDWR) used to open the file
  - Open file reference count
  - Meta-data (also kept on disk) such as file access and modification time, file size, and pointers to data blocks
  - Current read or write position in file
- Modify the diagram to show the final state of all tables after the following system calls have executed. Show the names of the file assigned to each table entry and the pointers from the per-process table to the system table. Include the **reference count** in the appropriate table(s). Assume that process A and B have the same current working directory and that all system calls complete successfully.
  Process B: fd1 = open("x", mode);
  Process A: fd1 = open("y", mode);
  Process B: fd2 = open("y", mode);
  Process A: close(stdin);
  Process A: fd3 = open("z", mode);

c) Assume a disc has the following characteristics: 1000 sectors (numbered 0 to 999) per surface; 10 sectors per track; and tracks are numbered from 0 (innermost) to 99 (outermost).Consider the following time ordered list of disc I/O requests, where the value given is the sector number requested.
548, 123, 459, 156, 645, 818, 825, 843, 239
Assuming all requests have arrived before accessing commences and that no more requests arrive during processing, compute the average track distance (i.e. number of tracks) the head moves per request, using

- First-in, First-out (FIFO);
- SCAN (in increasing track number);
- Circular SCAN (in increasing track number).

/END/

FD/JU