No of Pages : 5 Course Code : 12XW44

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

PSG COLLEGE OF TECHNOLOGY, COIMBATORE - 641 004

SEMESTER EXAMINATIONS, APRIL - 2014

MSc - SOFTWARE ENGINEERING Semester : 4

12XW44 OPERATING SYSTEMS

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

INSTRUCTIONS:

1. Group I, Group II and Group III questions should be answered in the Main Answer Book.
2. Ignore the box titled as **"Answers for Group III"** in the Main Answer Book.
3. Answer **ALL** questions from GROUP – I.
4. Answer any **5** questions from GROUP – II.
5. Answer any **ONE** question from GROUP – III.

GROUP - I **Marks : 10 x 3 = 30**

1. What is dual mode operation? Why do we need two modes of execution in an operating system, even though some of the kernel functions run in user mode?
2. What is the use of a process control block? Write the changes in the PCB chain when, a process makes an I/O request and a process completes an I/O operation.
3. Why is it important for the scheduler to distinguish I/O-bound programs from CPU-bound programs? What are the motivations for short term, medium term and long term scheduling levels?
4. What is context switch and mode switch? What is the relationship between them?
5. What is a thread? Assume that process A has one thread and process B has 10 threads, and the scheduler allocates the time slices equally.
   - If threads are in user level, which thread between a thread of A and a thread of B is scheduled longer by how much, or the same?
   - If threads are in kernel level, which process between A and B is scheduled longer by how much, or the same?
   - When a thread in a process invokes a system call, are the other threads in the process also be blocked, or not?
6. What are system calls? Consider the following C program that uses fork() system call to create child process. Suppose the process ID (PID) of parent is 5000. Assume that each new child process will have the next sequential PID that is available (5001, 5002, etc.). Write down the output of these processes.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main()      {
        int pid1, pid2, pid3, pid4;
        pid1= getpid();
        pid2= fork();
```

```
wait (NULL);
pid3= fork();
wait (NULL);
pid4= getpid();
printf("[%4d] [%4d] [%4d] [%4d]\n", pid1, pid2, pid3, pid4);  }
```

7. What is the difference between logical address and physical address? Calculate the physical memory address in multi-partition memory allocation if base register is 9055 and limit register is 2000 for the following program address:

   100, 2002, 300, 5000

8. Consider the factors: internal fragmentation, page 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 small page size? Why?

9. A system has five processes P1 through P5 and four resource types R1 through R4. There are 2 units of each resource type. Given that:

   P1 holds 1 unit of R1 and requests 1 unit of R4

   P2 holds 1 unit of R3 and requests 1 unit of R2

   P3 holds 1 unit if Rs and requests 1 unit of R3

   P4 requests 1 unit of R4

   P5 holds 1 unit of R3 and 1 unit of R2 and requests 1 unit of R3

   Show the resource allocation graph for this state of the system. Is the system in deadlock, and if so, which processes are involved?

10. Consider the classic thrashing curve below. Note that on this curve three points are marked: "A", "B", and "C".

Assume that the computer is operating at point "B", so is productively busy. What would be the behavior of the computer, either it will stay near "B" or move much closer to "A" or "C" when,

- the size of the working set parameter, window size Δ is increased
- the size of the working set parameter, Window size Δ is decreased

Justify your answer.

GROUP - II Marks : 5 x 10 = 50

11. Define the essential properties of the following types of operating systems

    (i) Batch (ii) Multiprogramming (iii) Timesharing (iv) Real time (v) Distributed

12. What is process synchronization? Explain the four requirements that a solution to the critical section problem must satisfy. Explain how semaphore can be used as synchronization tool. There exist four processes P1(S1), P2(S2, S3), P3(S4), P4(S5), where Pn(Sx) means Pn=process number, Sx=Statement x belongs to Pn. CPU will execute the process statements based on the following order :

When CPU executes S1 of P1, other process will wait. After executing S1, CPU executes S5 of P4 and so on. To achieve this execution order, write the solution code by using **wait(S)** and **signal(S)** method. Write how many semaphore variables required and write their initial value. Write pseudo code for all four processes.

13. Consider a system with four processes. Total resources in the system are P(13), Q(19), R(15). The current allocation and the maximum need for each process of each resource are as given below:

Processes	*Allocation*			*Maximum*		
	P	Q	R	P	Q	R
P1	2	2	1	11	16	14
P2	3	2	1	4	8	3
P3	2	2	2	5	7	6
P4	4	5	6	9	11	10
P5	0	1	2	3	2	4

By using deadlock avoidance,

i) Find the need matrix?

ii) Is the current state of the system is safe or not? Prove it!

iii) What happen to the system if P3 ask one more resource R? Can it be granted? Why?

14. Consider a system with the following specification:

- Total available physical memory frame: 1200KB
- Frame Size is 4 KB
- Total processes : 6
- The frame needed by each process has the following format (Process ID, Total frame). (0, 40), (1, 60), (2, 100), (3, 20), (4, 80), (5, 100)

Unfortunately the total available frames are limited. The system cannot supply all the requested frames to every process.

i) Determine total frames given to each process if the system uses equal allocation mechanism.

ii) Determine total frames given to each process if the system uses proportional allocation algorithm.

15. Suppose we have files F1, F2, F3 and F4 in sizes 7178, 572, 499 and 1195 bytes. The capacity of the disk is 50 KB with fixed physical block size of 512 bytes for allocation.
    - How many physical blocks would be needed to store these four files? Assume that in case of linked allocation strategy, 5 bytes are needed to store the next block in the link.
    - Using a diagram, show how allocation for these files is done in contiguous, linked and indexed allocation.
    - What is the type of fragmentation associated with each of file allocation policies. For each file, find the internal fragmentation measured as percentage of the file size.
    - Write the bit vector for the disk and calculate the size of the bit-vector.
    - Now, the size of the file F3 is increased to 600 bytes. How many more physical blocks are required? How many disk IO is to be performed to add the new block(s) at the end in each allocation policy? Assume that the new block(s) are in memory and other structures are in disk.

16. 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.

    (i) Give the reference string, assuming a page size of 100 bytes.

    (ii) 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.

    (iii) Calculate the page fault rate if the page replacement algorithm is LRU.

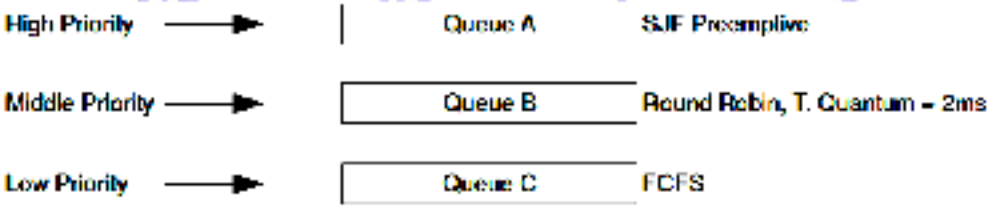
    (iv) If the system has the memory access time of 1000 nanoseconds and a page fault service time of 30 milliseconds (30,000,000 nanoseconds) and the probability of a page fault is 0.3, what is the effective access time?

GROUP - III Marks : 1 x 20 = 20

17. Explain the process state model. An Operating System uses multilevel queue to schedule the processes execution. A multilevel queue scheduling consists of three queues ordered by priority level (high, middle, and low). Each queue has different scheduling algorithm.
    - First Queue (Queue A) uses SJF Preemptive scheduling and eligible for process A1, A2, ..., An
    - Second Queue (Queue B) uses Round Robin scheduling with time quantum = 2ms and eligible for process B1, B2, ..., Bn
    - Third Queue (Queue C) uses First Come First Serve (FCFS) scheduling and eligible for process C1, C2, ..., Cn

    CPU will execute those queues under the following rules:
    - CPU executes the queue based on its priority. If each queue is not empty, the queue that has high priority (Queue A) will be executed first. After that, CPU executes middle priority queue, then low priority queue.
    - If high priority queue is empty, CPU executes process at other less priority queue
    - If there is a process entering an empty high priority queue while CPU is executing process in other less priority queue, CPU must change its execution to high priority

queue to service the process. CPU may move to other less priority queue if no process waiting in high priority queue.



Process	A1	A2	A3	A4	B1	B2	B3	C1	C2	C3
Arrival Time	0	2	8	11	2	4	9	3	5	7
Burst Time (ms)	4	2	1	4	5	4	3	7	8	2

Draw the Gantt chart and determine the **total waiting time and turnaround time** of each process.

18. Consider a disk with the following specifications:

- Using five platters
- One platter consists of two surfaces (top surface #0 and bottom surface #1).
- One surface capacity: 5 Gb
- Total tracks on one surface: 2500 (track#0 - #2499).
- Speed rotation: 6000 rpm.
- Total sectors in one track: 500 (sector #0 - #499).
- Time needed to move from one track to adjacent track: 1 ms
- Assume, only one head active reading/writing. Time needed to move from surface #0 to surface#1 is 0 ms.
- Disk scheduling algorithm: First Come First Served.
- At T=0, the head position is above cylinder #0, sector #0.
- Measurement: 1 kbyte = 1000 byte; 1 Mbyte = 1000 kbyte; 1 Gbyte = 1000 Mbyte.

Answer the following Questions:

i. Determine the capacity of one cylinder and also determine the disk size.

ii. How long it takes to read/write one sector?

iii. Determine the time (mS) to move from surface#0, track#0, sector#0 to surface#0, track#4, sector #399 ([0,0,0] →[0,4,399]).

iv. Determine the time (mS) to move from [0,0,0] → [0,0,499] → [0,3,99] → [0,3,499] → [0,2,249].

v. Some processes request to read/write some cylinders in surface 0. System pools the entire request on the following queue:

Queue(Q)= 200, 1500, 900, 2000, 340, 500, 1100, 1800

Right now the disk is reading cylinder 1000. Determine the cylinder access sequence and its total movement if SSTF and SCAN disk scheduling algorithms are used. Assume the head moves towards the end.

FD/RL /END/