Ex/MCA/224/232/2014

MCA Fourth Semester Examination, 2014
Operating Systems

Time – 3 Hours **Full Marks – 100**

Answer any five questions

Answers to all parts of a question must be written together

1.

a. What are the different states a process can assume? Describe with a suitable diagram how a process transitions from one state to another.

b. Differentiate between a long term scheduler and a short term scheduler. Under which conditions a mid-term scheduler may be invoked?

c. Consider three processes (process id 0, 1, 2, respectively) with compute time bursts 2, 4, and 8 time units. All processes arrive at time zero. Consider the preemptive Longest Remaining Time First (LRTF) scheduling algorithm. In LRTF ties are broken by giving priority to the process with the lowest process id. Find the average waiting time and turn around time. Determine the same for Shortest Remaining Time First (SRTF) scheduling algorithm.

d. What advantage are the advantages of having different time-quantum sizes on different levels of multi-level queue scheduling?

(2+4)+(2+2)+8+2=20

2.

a. What is a Process Control Block? State the requirements that any solution to the critical section problem must satisfy.

b. Suppose we want to synchronize two concurrent processes P and Q using binary semaphores. The code for the processes P and Q is shown below.

```
Process P:
while(1){
    Print '0';
    Print '0';
}
```

```
Process Q:
while(1){
    Print '1';
    Print '1';
}
```

Determine, minimum number of binary semaphores required along with their initial values. Identify places where suitable operations on those semaphores should be inserted in the code of process P and Q such that their execution will always produce an output string of the form "110011001100.............". Also propose a solution without using semaphores (shared variables may be used).

c. For the following code segment determine how many times "Hello" will be printed. Also determine the number of new processes that will be created.

```
int i=0;
do{
        i++;
        if(i%2==0)
                fork();
        else
                printf("Hello");
}while(i<=4);
```

d. What is a "spin lock"? How it can be implemented using TestAndSet instruction? When is a spinlock useful?

(2+3)+7+4+(1+2+1)=20

3.

a. Explain the necessary conditions of deadlock.

b. State how deadlock can be prevented by ensuring that circular wait condition does not hold.

c. Define safe state and safe sequence.

d. 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:

i. What is the content of the matrix *Need?*

ii. Is the system in a safe state?

iii. If a request from process P_1 arrives for (0, 4, 2, 0) can the request be granted immediately?

4+3+3+10=20

4.

a. Differentiate between internal and external fragmentation.

b. A dynamic partitioning scheme is being used, and the following is the memory configuration at a given point in time:

20 M	20 M	40 M	60 M	20 M	10 M	60 M	40 M	20 M	30 M	40 M	40 M

The shaded areas are allocated blocks; the white areas are free blocks. The next three memory requests are for 40M, 20M, and 10M. Indicate the starting address for each of the three blocks using the following placement algorithms:

i. First-fit

ii. Best-fit

iii. Worst-fit

c. A virtual address *a* in a paging system is equivalent to a pair (*p*, *w*), in which *p* is a page number and *w* is the offset within the page. Let *z* be the number of bytes in a page. Find algebraic equations that show *p* and *w* as functions of *z* and *a*.

d. A process uses 2-level page table for virtual to physical address translation. Page tables for both levels are stored in the main memory. Virtual address is 32 bit wide and physical address is 30 bit wide. The memory is byte addressable. For virtual to physical address translation, the 10 most significant bits of the virtual address are used as index into the first level page table while the next 10 bits are used as index into the second level page table. The 12 least significant bits of the virtual address are used as offset within the page. Assume that the page table entries in both levels of page tables are 4 bytes wide. Further, the processor has a translation look-aside buffer (TLB), with a hit rate of 96%. The TLB caches recently used virtual page numbers and the corresponding physical page numbers. The processor also has a physically addressed cache. Main memory access time is 10 ns, cache access time is 1 ns, and TLB access time is also 1 ns. Assume that no page faults occur, determine the minimum cache hit rate so that average time taken to access a virtual address is maximum 3.2 ns. Also determine the maximum number of bits that can be used for storing protection and other information in each page table entry.

3+6+4+6=20

5.

a. Which factors determine minimum number of page frames that must be allocated to a running process in a virtual memory environment?

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

c. Assume that we have a demand-paged memory. The page table is held in memory. It takes 8 milliseconds to service a page fault if an empty frame is available or if the replaced page is not modified and 20 milliseconds if the replaced page is modified. Memory-access time is 80 nanoseconds. Assume that the page to be replaced is modified 75 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 240 nanoseconds?

d. Consider the following page reference string: 1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6. How many page faults would occur for the following replacement algorithms, assuming four frames? Remember all frames are initially empty.
   i. LRU replacement
   ii. FIFO replacement
   iii. Optimal replacement

3+(2+3)+6+6=20

6.

a. Describe physical organization of a moving head disk. Define seek time and latency time.

b. A hard disk has 32 sectors per track, 8 platters each with 2 recording surfaces and 64 cylinders. The address of a sector is given as a triple <c, h, s>, where c is the cylinder number, h is the surface number and s is the sector number. Thus, the 0^{th} sector is addressed as <0, 0, 0>, the 1^{st} sector as <0, 0, 1>, and so on. Determine the sector number that corresponds to the address <40, 12, 24>. Also determine the address of the 1148^{th} sector.

c. A disk has 100 tracks (numbered 0 through 99). At a given time, it was servicing the request of reading data from track 50, and at the previous request, service was for track 40, the pending requests (in order of their arrival) are for the track nos. 30, 70, 85, 60, 20, 80, 65, 25. How many times will the head change its direction for the disk scheduling policies SSTF and FCFS? Also, determine the total number of cylinders that the head has to move for satisfying the pending requests for both the scheduling policies.

d. Differentiate between physical (or low level) formatting and logical formatting.

(4+2)+5+7+2=20

7.

a. Describe how file information is stored in File Allocation Table (FAT). How does FAT-16 differ from FAT-32?

b. Describe the structure of an i-node in Unix file system.

c. In a particular Unix OS, each disk block is of size 2048 bytes, each i-node has 8 direct addresses to data blocks and three additional addresses: one for single indirect block, one for double indirect block and one for triple indirect block. Size of each disk block address is 8 bytes. Determine the approximate maximum size of a file.

d. In a computer system, four files of size 11252 bytes, 4788 bytes, 6184 bytes and 12864 bytes need to be stored. For storing these files on disk, we can use either 128 bytes disk blocks or 256 bytes disk blocks (but can't mix disk blocks). For each block used to store a file, 8 bytes of book keeping information also needs to be stored on the disk. Thus the total space used to store a file is the sum of the space taken to store the file and the space taken to store the book keeping information for the blocks allocated for storing the file. A disk block can store either book keeping information for a file or data from a file, but not both. What is the total space required for storing the files using 128 and 256 bytes disk blocks respectively?

(4+2)+4+4+6=20

--------|-------