Ex/CSE/MCA/T/211A/2019

MCA Second Year First Semester Examination, 2019
Operating Systems

Time – 3 Hours **Full Marks – 100**

Answer any five questions

1.

a. Define turn-around time and waiting time.

b. Consider the following set of processes with the arrival times and the CPU burst times given in milliseconds

Process	Arrival Time	Burst Time
P_1	0	5
P_2	1	3
P_3	2	3
P_4	4	1

Determine the turnaround time and waiting time for all the processes using Shortest Remaining Time First (SRTF) and Longest Remaining Time First (LRTF) scheduling policy. In both the cases ties are broken by giving priority to the process with lowest id.

c. What will be the effect on the performance of Round Robin scheduling algorithm, when the time quanta is

i. Greater than the largest CPU burst
ii. Lesser than the smallest CPU burst

d. For the following code segment determine how many times "`Hello`" will be printed. Provide justification for your answer.

```
int i=0;
do{
	if(fork()!=0)
		i+=2;
	else
		i+=3;
}while(i<=4);
printf("Hello");
```

2+8+4+6=20

2.

a. What are the different requirements that any solution to the critical section problem must satisfy? Explain each briefly.

b. Consider the methods used by processes P_1 and P_2 for accessing their critical sections as given below. The initial values of shared Boolean variables S_1 and S_2 are randomly assigned. Explain with proper justifications, the requirements (asked in previous question) this solution satisfies.

```
P1

while(1){
        while(S1==S2);
          <Critical Section>
        S1=S2;
}
```

```
P2

while(1){
        while(S1!=S2);
          <Critical Section>
        S1=(not)S2;
}
```

c. What is a “spin lock”? Describe one implementation of semaphore which removes “spin lock”.

d. A certain computation generates two arrays `a` and `b` such that `a[i]=f(i)`, for `0≤i<3,` and `b[i]=g(a[i])` for `0≤i<3`. Suppose this computation is decomposed into two concurrent processes `P` and `Q` such that `P` computes the array `a` and `Q` computes the array `b`. Process `Q` cannot compute `g(a[i])` until process `P` generates `a[i]`, for `0≤i<3`. The processes employ two binary semaphores `X` and `Y`, both initialized to zero. The array `a` is shared by the two processes. The structures of the processes are shown below.

```
Process P:
private i;
for(i=0;i<3;i++){
        a[i] = f(i);
        ExitP(X, Y);
}
```

```
Process Q:
private i;
for(i=0;i<3;i++){
        EntryQ(X,Y);
        b[i]=g(a[i]);
}
```

Derive a correct implementations of `ExitP` and `EntryQ`? Provide necessary justifications for your answer.

3+6+5+6=20

3.

a. What is a `TestAndSet` instruction? Describe how this instruction can be used to solve critical section problem. Explain if your solution satisfies the bounded waiting requirement. Provide justification for your answer.

b. P_1, P_2, and P_3 are three processes executing their respective tasks. They should synchronize among themselves using semaphores such that the string “AABC” is printed infinite times. Determine, minimum number of semaphores required and their initial values. Also identify places where operations on those semaphore should be inserted in the code of P_1, P_2, and P_3. Describe how your solution works.

```
P1
while(true){
        print(“A”);
}
```

```
P2
while(true){
        print(“B”);
}
```

```
P3
while(true){
        print(“C”);
}
```

c. Modify the solution to the Readers-Writers problem to restrict at most 3 simultaneous Readers. Other synchronization requirements remain the same.

6+6+8=20

4.

a. Describe how resource allocation graph can be used to describe deadlock.

b. Suggest a suitable method which prevents deadlock by denying the circular wait condition to hold. Prove that this suggested method actually prevents deadlock.

c. A system has four processes and four allocable resource types each having 3, 12, 10, and 9 instances respectively. The current allocation and maximum requirements for each process are as follows:

Process	Allocation				Maximum			
P_1	0	0	1	1	0	0	1	2
P_2	1	0	0	1	1	6	5	1
P_3	1	2	4	3	2	3	4	5
P_4	0	5	2	2	0	6	5	2

Is the system in a safe state? If a request from process P_2 arrives for (0, 4, 2, 0) can the request be granted immediately.

d. Discuss two mechanisms for recovery from deadlock.

4+6+8+2=20

5.

a. Differentiate between load time and execution time address binding.

b. A computer system implements 8 kilobyte pages and a 32-bit physical address space. Each page table entry contains a valid bit, a dirty bit, three permission bits, and the frame number. If the maximum size of the page table of a process is 24 megabytes, determine the length of the virtual address (in bits) supported by the system.

c. Consider a paging system with 40 bit logical address and 24 bit physical address. Page size is 4 KB and size of each page table entry is 2 bytes. Determine the size of page table if single level page table is used. Now, assume that you want to implement a multi level page table. Determine how many levels of page table is required if you need to store each page of the page table possibly in non contiguous frames in physical memory. Determine the division of bits of the logical address that is required to address each levels of the multi level page table. If physical memory access time is 10 nanoseconds, compare the effective memory access time of the single level and proposed multi level paging scheme.

d. In a system inverted page table is used. Size of the inverted page table is 16 MB. Each entry of the inverted page table stores pid, page number and some protection information. Logical address space is 32GB, physical memory size is 16 GB and page size is 4 KB. Process pid is represented by 8 bit. Determine the maximum number of bits that can be used for storing protection information in each entry of the inverted page table.

2+5+8+5=20

6.

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

b. 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 100 nanoseconds. Assume that the page to be replaced is modified 70 percent of the time. What is the maximum acceptable page-fault rate for an effective access time of no more than 200 nanoseconds?

c. Assume that a main memory with only 4 frames each of 32 bytes is initially empty. The CPU generates the following sequence of virtual addresses and uses the LRU page replacement policy

0, 32, 16, 40, 56, 68, 12, 98, 148, 128, 112, 80

How many page faults does this sequence cause? How many more/less page fault occurs if Optimal page replacement is used.

d. What is thrashing?

2+8+8+2=20

7.

a. How does C-SCAN disk scheduling algorithm differ from SCAN scheduling algorithm.

b. A disk has 200 tracks (numbered 0 through 199). At a given time, it was servicing the request of reading data from track 120, and at the previous request, service was for track 130. The pending request (in order of their arrival) are for track numbers

60 70 115 130 110 80 90 30

How many times will the head change its direction for the disk scheduling policies SSTF (Shortest Seek Time First) and SCAN?

c. What is an i-node? Describe different fields in an i-node.

d. A hard disk has 32 sectors per track, 8 platters each with 2 recording surfaces and 64 cylinders. Determine size of each cylinder and size of the entire disk. Additionally, 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 <1, 4, 8>.

3+7+4+6=20

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