Ex/CSE/MCA/T/121A/2023

# Master of Computer Application
## First Year Second Semester Examination, 2023
### Subject: Operating Systems

Time: 3 Hours

Full Marks: 100

**Answer Any Five Questions**

1. (a) What is a process? Describe the different states a process may assume with the help of process state transition diagram.

   (b) An operating system uses a shortest remaining compute time first scheduling algorithm and schedules a new process either when the running process gets blocked on I/O or when the running process finishes its compute burst. Assume that all I/O operations can be overlapped as much as possible. Consider the set of 4 processes whose arrival times and burst times are given below:

| Process | Arrival Time | CPU Burst | I/O Burst | CPU Burst |
|---|---|---|---|---|
| $P_1$ | 0 | 3 | 2 | 2 |
| $P_2$ | 0 | 2 | 4 | 1 |
| $P_3$ | 2 | 1 | 3 | 2 |
| $P_4$ | 5 | 2 | 2 | 1 |

   Determine average turn around time and average waiting time.

   (c) For the following code segment determine how many times "Hello" will be printed. Provide necessary Justifications.

```
for(i=0;i<5;i++){
   if(fork()==0){
      printf("Hello\n");
      i+=2;
   }
}
```

4+10+6=20

2. (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) Two processes $P_1$ and $P_2$ use a shared variable $B$ which is initialized to 4. Determine the possible values of $B$ after both the processes finish their execution. $X$ is a local variable.

```
P1(){
    X=B-1;
    B=2*X;
}
```

```
P2(){
    X=2*B;
    B=X-1;
}
```

(c) A certain computation generates three arrays $a$, $b$ and $c$ such that $a[i] = f(i)$ for $0 \leq i < n$, $b[i] = g(a[i])$ for $0 \leq i < n$ and $c[i] = h(b[i])$ for $0 \leq i < n$. Suppose this computation is decomposed among three concurrent processes $X$, $Y$ and $Z$ such that $X$ computes the array $a$, $Y$ computes the array $b$ and $Z$ computes the array $c$. The processes employ three binary semaphores $R$, $S$ and $T$, all initialized to zero. The array $a$ is shared by $X$ and $Y$. The array $b$ is shared by $Y$ and $Z$. The structures of the processes are shown below.

```
Process X:
private i;
for(i=0; i<n; i++){
   a[i] = f(i);
}
```

```
Process Y:
private i;
for(i=0; i<n; i++){
    b[i]=g(a[i]);
}
```

```
Process Z:
private i;
for(i=0; i<n; i++){
    c[i]=h(b[i]);
}
```

State the synchronization requirement so that the correct computation takes place. Insert appropriate operations on the these semaphores - such that the desired synchronization is achieved. Provide necessary justifications.

6+6+8=20

3. (a) Describe how resource allocation graph can be used for deadlock detection.

(b) What are the different alternatives for recovering from deadlock?

(c) A system is having 3 user processes $P_1$, $P_2$ and $P_3$ where $P_1$ requires 2 instances of resource type $R$, $P_2$ requires 3 instances of $R$, $P_3$ requires 4 instances of resource $R$. Determine the minimum number of instances of $R$ that will ensure no deadlock.

(d) A system has four processes and five allocable resource types. The current allocation and maximum needs are as follows:

| Process | Allocated | | | | | Maximum | | | | | Available | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $P_1$ | 1 | 0 | 2 | 1 | 1 | 1 | 1 | 2 | 1 | 3 | 0 | 0 | 2 | 1 | 1 |
| $P_2$ | 2 | 0 | 1 | 1 | 0 | 2 | 2 | 2 | 1 | 0 | | | | | |
| $P_3$ | 1 | 1 | 0 | 1 | 1 | 2 | 1 | 3 | 1 | 1 | | | | | |
| $P_4$ | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 2 | 2 | 1 | | | | | |

Determine if the system is in a safe state? Now process $P_3$ makes a request [1,0,1,0,0]. Can the request be granted?

4+3+5+8=20

4. (a) Differentiate between load time address binding and execution time address binding.

(b) What is external fragmentation? How does paging overcome this problem? Does paging remove internal fragmentation?

(c) Consider a computer system with 39-bit virtual addressing using multi-level page tables with $L$ levels for virtual to physical address translation. The page size is 4 KB and a page table entry at any of the levels occupies 8 bytes. Determine the value of $L$ and also the sum total of sizes of page tables of all the levels.

(d) In a system inverted page table is used. Each entry of the inverted page table stores process id and page number. Logical address is 36 bit, physical memory size is 32 GB and page size is 4 KB. Process id is represented by 8 bit. Determine size of the inverted page table.

3+3+8+6=20

5. (a) What are the different steps that an operating system adopts while servicing a page fault?

(b) What is the usage of a dirty bit?

(c) In a demand paging system memory access time is 10 ns and page fault service time is 25 ms. What should be maximum page fault rate so that performance does not degrade more than 20%?

(d) Assume that there are 4 page frames which are initially empty. If the page reference string is 7, 2, 7, 3, 2, 5, 3, 4, 6, 7, 7, 1, 5, 6, 1 - determine the number of page faults using the FIFO, LRU and optimal page replacement policies.

4+2+4+10=20

6. (a) What is thrashing? When does thrashing occur?

(b) How does C-SCAN differ from SCAN scheduling?

(c) A disk pack has 16 surfaces, 256 tracks per surface and 64 sectors per track. Size of each sector is 512 bytes. Determine the size of the disk pack. Also determine the number of bits required to address each cylinder and each sector.

(d) 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 45. The pending requests (in order of their arrival) are for the track nos. 40, 80, 65, 90, 60, 85, 75, 35. How many times will the head change its direction for the disk scheduling policies FCFS, SSTF and SCAN? Also, determine the total number of cylinders that the head has to move for satisfying the pending requests for all the scheduling policies.

2+3+5+10=20

7. (a) A hard disk has 64 sectors per track, 16 platters each with 2 recording surfaces and 128 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 $< 2, 16, 32 >$. Also determine the address of the $4000^{th}$ sector.

(b) Compare linked and indexed allocation strategies for allocation of disk space to files.

(c) What is an index node or i-node?

(d) In a particular Unix OS, each data block is of size 2048 bytes, each i-node has 8 direct data block addresses and three additional addresses: one for single indirect block, one for double indirect block and one for triple indirect block. Also each block can contain addresses for 128 blocks. What is the approximate maximum size of a file?

8+3+3+6=20