*Ex/PG/MTCT/T/121A/2023*

# Master of Technology in Computer Technology

## First Year Second Semester Examination, 2023

## Subject: Advanced Operating Systems

Time: 3 Hours | Full Marks: 100

**Answer Any Five Questions**

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) State if starvation is possible in priority based process scheduling. If so, propose a solution to overcome this problem.

   (d) What is multi-level feedback scheduling?

   (e) Consider the following four processes with arrival times (in milliseconds) and their length of CPU bursts (in milliseconds) as shown below:

| Processes | $P_1$ | $P_2$ | $P_3$ | $P_4$ |
|---|---|---|---|---|
| Arrival Time | 0 | 1 | 3 | 4 |
| Burst Time | 3 | 1 | 3 | 3 |

   These processes are run on a single processor using preemptive Shortest Remaining Time First scheduling algorithm. Determine the average waiting time and average turn around time of all the processes.

   3+3+3+3+8=20

2. (a) What is a Process Control Block?

   (b) For the following code segment determine the output and state the number of new processes that will be created. Provide necessary justifications.

```
for (i=0; i<=4; i++){
    if(i%2==0)
        fork();
    else
        printf("%d\n", i);
}
```

   (c) What is race condition? Explain with a suitable example.

   (d) $P_1$, $P_2$ and $P_3$ are three processes executing their respective tasks. They should synchronize among themselves using semaphores such that the string "ABABCA" gets 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 $P_3$. Provide necessary justifications.

$P_1$
```
while(true){
    print("A");
}
```

$P_2$
```
while(true){
    print("B");
}
```

```
P3
while(true){
    print("C");
}
```

2+6+4+8=20

3. (a) What is busy waiting? Propose one implementation of semaphore which overcomes the busy wait problem.

   (b) Explain how the critical section problem can be solved in a system which supports `TestAndSet` machine instruction.

   (c) What are the synchronization requirements of the producer and consumer processes in both bounded-buffer and unbounded-buffer producer consumer problem?

   (d) Consider a set of two periodic tasks with the execution profile as given in the following table.

| **Task** | **Period (in ms)** | **Execution Time (in ms)** |
|---|---|---|
| A | 30 | 10 |
| B | 40 | 15 |

   Draw the scheduling diagrams (between 0-120 ms) considering the following scheduling policies.

   i. Fixed priority scheduling (A has priority over B)
   ii. Fixed priority scheduling (B has priority over A)
   iii. Earliest Deadline First using completion deadline

4+4+2+10=20

4. (a) What are the necessary conditions of deadlock? Explain each briefly.

   (b) Propose a method that will deny circular wait condition in a system. Prove the correctness of this method.

   (c) What is a safe sequence? What is its significance in the context of deadlock handling?

   (d) Consider a system with 3 processes that share 4 instances of the same resource type. Each process can request a maximum of $K$ instances. Resource instances can be requested and released only one at a time. Find the largest value of $K$ that will always avoid deadlock.

6+5+3+6=20

5. (a) 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 P1 arrives for (0, 4, 2, 0) can the request be granted immediately?

(b) What is the difference between deadlock prevention and deadlock avoidance mechanisms?

(c) Describe how resource allocation graphs are used for deadlock detection.

(d) Differentiate between load time address binding and execution time address binding.

10+3+4+3=20

6. (a) Differentiate between internal and external fragmentation.

(b) Consider a paging system with 40 bit logical address and 36 bit physical address. Page size is 16 KB and size of each page table entry is 4 bytes. Determine the size of page table if single level page table is used. How many bits in each page table entry can be used for storing protection and other information?

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

(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. LRU replacement, FIFO replacement, Optimal replacement.

2+5+4+9=20

7. (a) What is thrashing? Explain the causes of it.

(b) Differentiate between seek time and latency time.

(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) 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.

3+3+8+6=20