Master of Technology in Computer Technology

First Year Second Semester Examination, 2024

Subject: Advanced Operating Systems

Time: 3 Hours | Full Marks: 100

Answer Any Five Questions

1. (a) What is a Process Control Block? What are the different information stored in it?

   (b) What is the difference between a long term and a short term scheduler? What is the role of a mid term scheduler?

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

   (d) Consider the following set of processes, with the length of the CPU burst given in milliseconds:
   The processes are assumed to have arrived in the order P_1, P_2, P_3, P_4, P_5, all at time 0.

Process	Burst Time	Priority
P_1	2	2
P_2	1	1
P_3	8	4
P_4	4	2
P_5	5	3

   Draw Gantt charts that illustrate the execution of these processes using the following scheduling algorithms: SJF, nonpreemptive priority (a larger priority number implies a higher priority), and RR (quantum = 2). Determine the turnaround time and waiting time of each process for each of the scheduling algorithms.

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

```
int x = 3;
do{
    if(fork()!=0){
        x=x-1;
    }
    else{
        x=x-2;
        printf("Hello\n");
    }
}while(x>0);
```

3+3+3+6+5=20

2. (a) Processes P_1 and P_2 are executing their respective tasks. They should synchronize among themselves using semaphores such that the string "ABABBABA" 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 and P_2. Provide necessary justifications.

```
P1
while(true){
    print("A");
}
```

```
P2
while(true){
    print("B");
}
```

(b) Explain with a suitable example the priority inversion problem. Also explain how does priority ceiling approach solve this problem.

(c) State the requirements that any solution to the critical section problem must satisfy.

(d) Propose a solution to a variant of the Readers-Writers problem where maximum three readers allowed. Other synchronization requirements remain the same.

8+5+2+5=20

3. (a) Show with an example how resource allocation graph can be used for deadlock detection.

(b) Define safe state and safe sequence.

(c) 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				

Is the system in a safe state?

(d) If there are 6 units of resource R in the system and each process in the system requires 3 units of resource R, then how many processes can be present at maximum so that no deadlock will occur?

(e) Describe two different ways that an operating system can recover from deadlock.

4+3+6+4+3=20

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

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

(c) Consider three concurrent processes P_1, P_2 and P_3 as shown below, which access a shared variable X that has been initialized to 5.

P_1	P_2	P_3
...	...	...
$X = X + 2$	$X = X - 3$	$X = X + 1$
...	...	...

The processes are executed on a uniprocessor system running a time-shared operating system. Determine the minimum and maximum possible values of X after the three processes have completed execution.

(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	20	10
B	30	5

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

2+4+6+8=20

5. (a) What is external fragmentation? How can external fragmentation be reduced?
(b) Consider a four-level page table to translate a 42-bit virtual address to a physical address as follows:

<------42-bit virtual address------>

Level 1 offset	Level 2 offset	Level 3 offset	Level 4 offset	Page offset
2 bits	9 bits	9 bits	9 bits	13 bits

The page size is 8 KB and page table entry size at every level is 16 bytes. Determine the amount of memory required for storing the page tables of of all levels.
Determine the effective memory access time if main memory access time is 100 ns.

(c) In a system inverted page table is used. Size of the inverted page table is 32 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 32 bit. Determine the maximum number of bits that can be used for storing protection information in each entry of the inverted page table.

3+10+7=20

6. (a) What is the use of dirty bit in page replacement?
(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 supported by the system.
(c) Consider a system with 30 bit logical address and 28 bit physical address. The system implements an inverted page table. It uses a page size of 8 KB and uses 15 bits as process id. What is the size of each entry in the inverted page table? Also, determine the size of the inverted page table.
(d) Assume that a main memory with only 3 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, 160, 40, 56, 68, 12, 198, 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.

2+6+6+6=20

7. (a) Suppose that a disk drive has 2,000 cylinders, numbered 0 to 1999. The drive is currently serving a request at cylinder 143, and the previous request was at cylinder 125. The queue of pending requests, in FIFO order, is:
86, 913, 1774, 948, 1509, 1022, 1750, 130
Starting from the current head position, what is the total distance (in cylinders) that the disk arm moves to satisfy all the pending requests for each of the following disk-scheduling algorithms?

(i) FCFS
(ii) SSTF
(iii) SCAN

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

(b) In a particular Unix OS, each data block is of size 1024 bytes, each node 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 in the file system?

(c) What are the advantages and disadvantages of large chunk size in Google File System?

(d) What are the different metadata that are stored at the master node in Google File System? What happens to these metadata when the master node restarts after a failure?

7+5+4+4=20