

Course : B.Sc.(H) Computer Science
Semester : III
Paper Title : Operating Systems
Unique Paper Code : 32341302
Admission Year : 2019 onwards
Max. Marks : 75
Time : 3 hours

Instructions for students:

- *Attempt any 4 questions.*
- *All questions carry equal marks.*

Q1.	<p>Suppose that a disk drive has 6,000 cylinders, numbered 0 to 5999. The drive is currently serving a request at cylinder 2350, and the previous request was at cylinder 1800. The queue of pending requests, in FIFO order, is:</p> <p>2069, 1212, 3296, 2800, 244, 1618, 3856, 1523, 965, 3681</p> <p>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 SCAN and C-SCAN disk-scheduling algorithms? Show all the intermediate steps.</p>	8
	<p>Consider a disk drive with 10,000 cylinders. Suppose that 50 percent of the disk access requests are for a small set of cylinders between 100 and 800. Which of the following algorithms will take greater advantage of the given situation? Justify your answer.</p> <p>a) Shortest Seek Time First (SSTF) b) First Come First Serve (FCFS)</p>	6
	<p>Which of the following instructions should be privileged? Give reasons.</p> <p>a) Modify entries in the open-file table. b) Read the clock. c) Access I/O device.</p>	4.75

<p>Q2</p>	<p>A given operating system operates in the user and kernel mode of operation. What is the mode of the system at the following points? Justify your answer.</p> <p>a) Boot time b) When the printer generates “Out of paper” error c) When the operating system gains control</p> <p>Consider a process (id= 2256) creates a child process (id = 2257) and child process further creates a child process (id= 2258). What will be output at statement (1), (2) and (3) along with the process generating this output, for the code given below:</p> <pre>int main() { int id1= fork(); cout<<id1; // (1) if (id1==0) { int id2; id2= fork(); cout<< id2; // (2) } cout<<id1; // (3) }</pre> <p>A computer system with a 32-bit logical address and 4-KB page size supports up to 512 MB of physical memory. Consider a user process that accesses the entire physical memory. How many entries will be there in a conventional single-level page table. How many entries can exist in a two-level hierarchical page table? Justify your answer.</p> <p>A process P is allocated n frames in the memory (initially all empty). The process page-reference string has length q; n distinct page numbers occur in it. What is the lower bound on the number of page faults? What is an upper bound on the number of page faults? Justify your answer.</p>	<p>6</p> <p>5</p> <p>4</p> <p>3.75</p>
<p>Q3</p>	<p>As a process executes, it changes its state. The state of a process is defined in part by the current activity of that process. Draw a process state diagram.</p> <p>A process P changes its state as mentioned in the table below. Perform the following-</p> <p>a) Identify each state of the process P (S. No. 1-5) b) In the process diagram, mark each state of the Process P (S.No. 1-5).</p>	<p>4.75</p>

S. No	Current Activity
1	P arrives in main memory
2	P starts execution
3	Process Q preempts P
4	P resumes execution
5	P needs to accept some input from user

Consider the following code, for creating a child process using fork() system call. Mention the value of 'a' printed by each process at statement (1), (2) and (3). Explain your answer with the help of a parent-child tree diagram.

4

```

void main( )
{
    int a = 7;
    int pid1 = fork();
    if ( pid1 == 0 )
    {
        a++;
        cout<< "value of a is" <<a<<endl; // (1)
    }
    else
    {
        int pid2= fork();
        if (pid2==0)
        {
            a--;
            cout<<"value of a is"<<a<<endl; // (2)
            exit(0);
        }
    }
    cout<< "value of a is" <<a<<endl; // (3)
}

```


	<p>Consider a printing resource having 5 non-shareable instances which may be required by many processes simultaneously. Synchronize the usage of this resource with the help of a semaphore S.</p> <p>a) What type of semaphore will be required for this purpose?</p> <p>b) How will the synchronization be achieved if six processes want to use this resource simultaneously?</p> <p>Consider a system with 4 types of resources R1 (2 units), R2 (2 units), R3 (2 units), R4 (2 units). A non-preemptive resource allocation policy is used. Three processes P1, P2, P3 request and hold the resources as follows:</p> <table border="1" data-bbox="378 701 1243 1146"> <thead> <tr> <th>Process</th> <th>Requesting</th> <th>Holding</th> </tr> </thead> <tbody> <tr> <td>P1</td> <td>1 unit of R1</td> <td>1 unit of R3, 1 unit of R4, 1 unit of R2</td> </tr> <tr> <td>P2</td> <td>1 unit of R2</td> <td>1 Units of R1, 1 unit of R3</td> </tr> <tr> <td>P3</td> <td>1 unit of R3</td> <td>1 unit of R1, 1 unit of R2, 1 unit of R4</td> </tr> </tbody> </table> <p>Draw a resource allocation graph and identify the possibility of any deadlock in the system. Give reasons.</p>	Process	Requesting	Holding	P1	1 unit of R1	1 unit of R3, 1 unit of R4, 1 unit of R2	P2	1 unit of R2	1 Units of R1, 1 unit of R3	P3	1 unit of R3	1 unit of R1, 1 unit of R2, 1 unit of R4	<p>6</p> <p>6</p>
Process	Requesting	Holding												
P1	1 unit of R1	1 unit of R3, 1 unit of R4, 1 unit of R2												
P2	1 unit of R2	1 Units of R1, 1 unit of R3												
P3	1 unit of R3	1 unit of R1, 1 unit of R2, 1 unit of R4												
<p>5</p>	<p>Given memory partitions of 150 KB, 400 KB, 250KB, 300 KB and 650 KB (in order), how would each of the first-fit, best-fit and worst-fit algorithms place processes of 224 KB, 405 KB, 98 KB and 500 KB (in that order)? Which algorithm results in least external fragmentation?</p> <p>Assuming a 2-KB page size, what are the page numbers and offsets for the following address references (provided as decimal numbers)</p> <p>a) 3089 b) 32095 c) 235201</p>	<p>9+1</p> <p>5</p>												

