**SRM Institute of Science and Technology**
**College of Engineering and Technology**
**School of Computing**

| Mode of Exam |
|---|
| **OFFLINE** |
| **SET D** |

## DEPARTMENT OF COMPUTING TECHNOLOGIES

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

**Academic Year: 2021-2022 (Even)**

**Test: CLAT-2** | **Date: 26-05-2022**
**Course Code & Title:** 18CSC205J: Operating Systems | **Duration:** 2 Period
**Year & Sem:** 2022 & IV Semester | **Max. Marks:** 50 Marks

**Course Articulation Matrix:** ***(to be placed)***

**Part - A**
**( 10 x 1 = 10 Marks)**

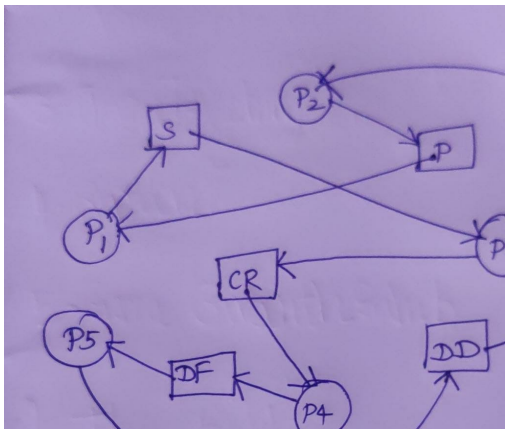
**Instructions: Answer all**

| Q. No | Question | Marks | BL | CO | PO | PI Code |
|---|---|---|---|---|---|---|
| 1 | b. An unsafe state may lead to a deadlock | 1 | 2 | 2 | 3 | 3.6.1 |
| 2 | C. Process j may request for resource i | 1 | 2 | 2 | 3 | 3.6.1 |
| 3 | a. I/O bounded process | 1 | 3 | 2 | 3 | 3.6.1 |
| 4 | b. Short-term scheduler | 1 | 2 | 2 | 3 | 3.5.1 |
| 5 | a.1 printer 1 scanner and 2 disk files | 1 | 1 | 3 | 1 | 1.6.1 |
| 6 | b. Contiguous Allocation | 1 | 1 | 3 | 1 | 1.6.1 |
| 7 | d. Compaction | 1 | 1 | 3 | 1 | 1.6.1 |
| 8 | b. S < STLR | 1 | 1 | 3 | 1 | 1.6.1 |
| 9 | a. Main Memory | 1 | 1 | 3 | 1 | 1.6.1 |
| 10 | c. TLB | 1 | 1 | 3 | 1 | 1.6.1 |

**Part - B**
**( 4 x 5 = 20 Marks)**

**Instructions: Answer Any 5**

| | | | | | | |
|---|---|---|---|---|---|---|
| 11 | **Consider a system consisting of processes P1, P2, ..., Pn, each of which has a unique priority number. Write a monitor that allocates three identical printers to these processes, using the priority numbers for deciding the order of allocation.**<br>type resource = monitor<br>var P: array[0..2] of boolean;<br>X: condition;<br>procedure acquire (id: integer, printer-id: integer);<br>begin<br>if P[0] and P[1] and P[2] then X.wait(id)<br>if not P[0] then printer-id := 0;<br>else if not P[1] then printer-id := 1;<br>else printer-id := 2; | 5 | 2 | 2 | 3 | 3.7.1 |

<table>
<tr><td></td><td>P[printer-id]:=true;<br>end<br>procedure release (printer-id: integer)<br>begin<br>P[printer-id]:=false;<br>X.signal;<br>end<br>begin<br>P[0] := P[1] := P[2] := false;<br>end</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>12</td><td>Five processes , a printer, a scanner , a card reader, a disk file and a disk drive is available in a system. The requests and the allocations are given in the following table
<table>
<tr><td>Requests</td><td>Allocations</td></tr>
<tr><td>Process 1 is requesting for the scanner</td><td>Printer is allocated to process 1</td></tr>
<tr><td>Process 2 is requesting for the printer</td><td>Card reader is allocated to process 4</td></tr>
<tr><td>Process 3 is requesting for the card reader</td><td>Disk drive is allocated to process 2</td></tr>
<tr><td>Process 4 is requesting for the disk file</td><td>Scanner is allocated to process 3</td></tr>
<tr><td>Process 5 is requesting for the disk drive</td><td>Disk file is allocated to process 5</td></tr>
</table>
Draw the resource allocation graph for this system. Draw the wait-for graph and check whether a deadlock is detected.<br><br>Answer:<br> 
<br>Deadlock is detected as the cycle exists in the Graph</td><td>5</td><td>4</td><td>2</td><td>3</td><td>3.7.1</td></tr>
<tr><td>13</td><td>When do page faults occur? Describe the actions taken by the operating system when a page fault occurs.<br><br>A page fault occurs when an access to a page that has not been brought into main memory takes place.</td><td>5</td><td>3</td><td>3</td><td>2</td><td>2.7.1</td></tr>
</table>

| | | | | | | |
|---|---|---|---|---|---|---|
| | The operating system verifies the memory access, aborting the program if it is invalid. If it is valid a free frame is located and I/O requested to read the needed page into the free frame. Upon completion of I/O, the process table and page table are updated and the instruction is restarted. | | | | | |
| **14** | **Calculate the number of bits required in the address for memory having size of 16 GB. Assume the memory is 4-byte addressable.**<br><br>Let 'n' number of bits are required. Then, Size of memory = $2^n$ x 4 bytes.<br><br>Since, the given memory has size of 16 GB, so we have-<br><br>$2^n$ x 4 bytes = 16 GB<br><br>$2^n$ x 4 = 16 G<br><br>$2^n$ x $2^2$ = $2^{34}$<br><br>$2^n$ = $2^{32}$<br><br>∴ n = 32 bits | **5** | **3** | **3** | **3** | **3.7.1** |
| **15** | **Draw the Gantt chart for the following scenario (using Round robin algorithm) with time slot "3".**<br><br>Process CPU time Arrival time<br>1 4 0<br>2 3 2<br>3 5 5<br>4 4 6<br>5 8 7<br><br>**Also find the average waiting time.**<br><br>**Answer :**<br><br>P1 P2 P3 P4 P5 P1 P3 P4 P5 P5<br>3 6 9 12 15 16 18 19 23 24<br><br>AWT = 39 | 5 | 4 | 2 | 3 | 3.7.1 |

**Part – C**

**(2 x 10 = 20 Marks)**

**Instructions: Answer All**

| | | | | | | |
|---|---|---|---|---|---|---|
| **16.a** | **Consider the following snapshot of a system:**<br><br>**Process** \| **Allocation** \| **Max** \| **Available**<br>**A B C D** \| **A B C D** \| **A B C D** | 10 | 4 | 2 | 3 | 3.7.1 |

| | | | |
|---|---|---|---|
| **P0** | **2 0 0 1** | **4 2 1 2** | **3 3 2 1** |
| **P1** | **3 1 2 1** | **5 2 5 2** | |
| **P2** | **2 1 0 3** | **2 3 1 6** | |
| **P3** | **1 3 1 2** | **1 4 2 4** | |
| **P4** | **1 4 3 2** | **3 6 6 5** | |

**Answer the following questions using the banker's algorithm:**

**a. Illustrate that the system is in a safe state by demonstrating an order in which the processes may complete.**

**b. If a request from process *P*1 arrives for (1, 1, 0, 0), can the request be granted immediately?**

**c. If a request from process *P*4 arrives for (0, 0, 2, 0), can the request be granted immediately?**

Answer :

a. Yes, the system is safe with sequence <P1,P4,P5,P2,P3>
b. Yes
c. Yes

| | | | | | | |
|---|---|---|---|---|---|---|
| **16.b** | **Consider a scenario where the following processes are arriving in the given time and their CPU burst is given in the milliseconds** | 10 | 4 | 2 | 3 | 3.7.1 |

| Process | Time | Burst time |
|---|---|---|
| 1 | 0 | 5 |
| 2 | 2 | 8 |
| 3 | 4 | 7 |
| 4 | 5 | 10 |
| 5 | 6 | 4 |

**Which of the following algorithm would be more suitable for this given scenario to yield best average waiting and average turnaround time?**

1. **Shortest remaining time first**
2. **Shortest job first**

**Justify your answer whether the preemptive scheduling or non-preemptive scheduling work well for this case. And also specify the difference in the average waiting time by those two algorithms.**

Answer:

SRTF is SJF with Pre-emption

Gantt Chart for both SJF and SRTF

SJF : Avg WT =8

Avg TAT = 14.8

SRTF : Avg WT = 7.6

Avg TAT = 14.4

| | | | | | | |
|---|---|---|---|---|---|---|
| | SRTF is preferred for this processes. | | | | | |
| **17.a** | **Associate the technique used for Structuring the page table,**<br><br>**Case 1: There might be a case where the page table is too big to fit in a contiguous space may have a hierarchy with several levels.**<br><br>**Case 2: Identify the approach is used to handle address spaces that are larger than 32 bits.**<br><br>Some of the common techniques that are used for structuring the Page table are as follows:<br><br>1. Hierarchical Paging<br>2. Hashed Page Tables<br><br>***Hierarchical Paging***<br><br>Another name for Hierarchical Paging is multilevel paging.<br><br>• There might be a case where the page table is too big to fit in a contiguous space, so we may have a hierarchy with several levels.<br>• In this type of Paging the logical address space is broke up into Multiple page tables.<br>• Hierarchical Paging is one of the simplest techniques and for this purpose, a two-level page table and three-level page table can be used.<br><br>Two Level Page Table<br><br>Consider a system having 32-bit logical address space and a page size of 1 KB and it is further divided into:<br><br>• Page Number consisting of 22 bits.<br>• Page Offset consisting of 10 bits.<br><br>As we page the Page table, the page number is further divided into :<br><br>• Page Number consisting of 12 bits.<br>• Page Offset consisting of 10 bits. | **10** | **2** | **3** | **3** | **3.7.1** |

Thus the Logical address is as follows:

| Page Number | | Page |
|---|---|---|
| P1 | P2 | d |

In the above diagram,

P1 is an index into the **Outer Page** table.

P2 indicates the displacement within the page of the **Inner page** Table.

As address translation works from outer page table inward so is known as **forward-mapped Page Table**.

Below given figure below shows the Address Translation scheme for a two-level page

level scheme then the addresses will look like this:

| outer page | inner page | offset |
|---|---|---|
| p1 | p2 | d |
| 42 | 10 | 12 |

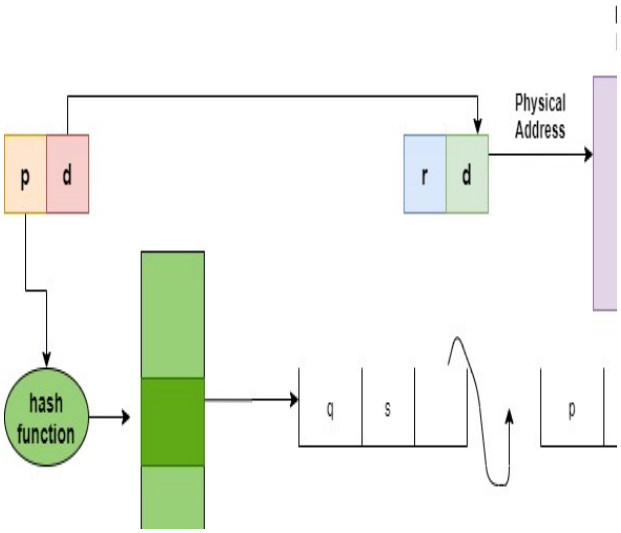
Thus in order to avoid such a large table, there is a solution and that is to divide the outer page table, and then it will result in a **Three-level page table:**

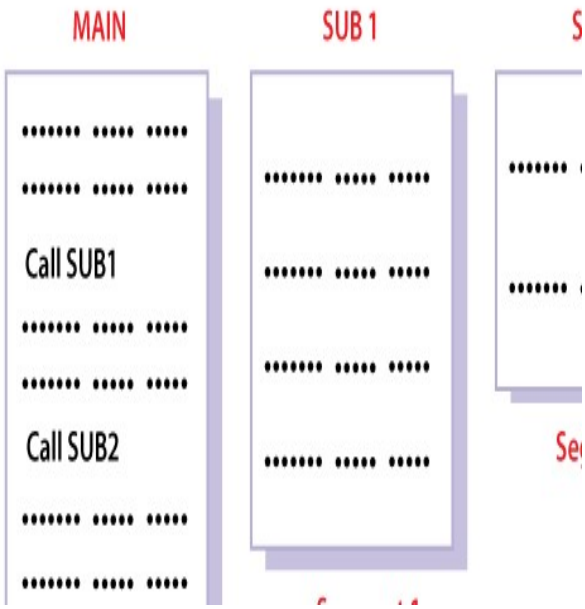
| 2nd outer page | outer page | inner page | offset |
|---|---|---|---|
| p1 | p2 | p2 | d |
| 32 | 10 | 10 | 12 |

***Hashed Page Tables***

This approach is used to handle address spaces that are larger than 32 bits.

- In this virtual page, the number is hashed into a page table.
- This Page table mainly contains a chain of elements hashing to the same elements.

| | | | | | | |
|---|---|---|---|---|---|---|
| | Each element mainly consists of :<br><br>1. The virtual page number<br>2. The value of the mapped page frame.<br>3. A pointer to the next element in the linked list.<br><br>Given below figure shows the address translation scheme of the Hashed Page Table:<br><br>  <br> | | | | | |
| **17.b** | **In an Operating system, the OS doesn't care about the User's view of the process. It may divide the same function into different pages and those pages may or may not be loaded at the same time into the memory. It decreases the efficiency of the system. So identify which technique is suitable to overcome the drawback and also helps out in better efficiency and performances.**<br><br>**<u>Segmentation</u>**<br><br>In Operating Systems, Segmentation is a memory management technique in which the memory is divided into the variable size parts. Each part is known as a segment which can be allocated to a process.<br><br>The details about each segment are stored in a table called a segment table. Segment table is stored in one (or many) of the segments.<br><br>Segment table contains mainly two information | **10** | **4** | **3** | **3** | **3.7.1** |

| | | | | | | |
|---|---|---|---|---|---|---|
| | about segment:<br><br>1. Base: It is the base address of the segment<br>2. Limit: It is the length of the segment.<br><br>It is better to have segmentation which divides the process into the segments. Each segment contains the same type of functions such as the main function can be included in one segment and the library functions can be included in the other segment.<br><br>  | | | | | |

**Question Paper Setter**

**Approved by Audit Professor/ Course Coordinator**