**SRM Institute of Science and Technology**
**College of Engineering and Technology**
**School of Computing**
SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Offline Mode
SET A

**Academic Year: 2021-22 (EVEN)** **Reg No:**

**Test: CLA-T3** **Date: 22.**06.2022
**Course Code & Title: 18CSC205J - Operating Systems** **Duration:** 2 Period
**Year & Sem: II Year / IV Sem** **Max. Marks:** 50 Marks

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

| S.No. | Course Outcome | PO 1 | PO 2 | PO 3 | PO 4 | PO 5 | PO 6 | PO 7 | PO 8 | PO 9 | PO10 | PO11 | PO12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | CO3 | 1 | 3 | | 3 | | | | | | | | |
| 2 | CO4 | 1 | 3 | | 3 | | | | | | | | |
| 3 | CO5 | 1 | 3 | 2 | 3 | | | | | | | | |

| | **Part – A (10 x 1 = 10 Marks)**<br>**Instruction: Answer all** | | | | | |
|---|---|---|---|---|---|---|
| **Q. No** | **Question** | **Mark** | **BL** | **CO** | **PO** | **PI Code** |
| 1) | If hardware does not support _____ then a multi - user and multi - processing operating system cannot be implemented.<br>a) At least two modes of CPU execution<br>b) Demand paging<br>c) DMA for disk transfer<br>d) Address translation<br>Answer:b | 1 | 2 | 3 | 1 | 1.5.1 |
| 2) | After allocating the frames to the process the left-over frames can be used as a free frame buffer pool. This scheme is called as_____<br>a) Equal allocation<br>b) Proportional allocation<br>c) Dynamic allocation<br>d) Static allocation<br>Answer : a | 1 | 2 | 3 | 2 | 2.5.1 |
| 3) | Memory allocation based on Process size is called as _____<br>a) Equal Allocation<br>b) Dynamic allocation<br>c) Proportional allocation<br>d) Static allocation<br>Answer : c | 1 | 1 | 3 | 1 | 1.6.1 |
| 4) | PFF stands for ----------<br>a) Page Find Frequency<br>b) Page Fault Frequency<br>c) Peak Fault Frequency<br>d) Peak Find Frequency<br>Answer : b | 1 | 1 | 3 | 2 | 2.5.1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 5) | Operating system supports different page replacement policy. From the given below option which is not a valid page replacement policy?<br>a) Least Recently Used<br>b) First in first out<br>c) Currently used policy<br>d) Optimal page replacement policy<br>Answer : c | 1 | 2 | 3 | 1 | 1.6.1 |
| 6) | The surface of a platter is logically divided into circular _____, which are Sub divided into sectors.<br>a. Platters<br>b. Disk arm<br>c. Read write head<br>d. Tracks<br>Ans: D Tracks | 1 | 1 | 4 | 1 | 1.5.1 |
| 7) | Random access in magnetic tape is _____ compared to magnetic disks.<br>a) fast<br>b) very fast<br>c) slow<br>d) very slow<br>Ans: d) very slow | 1 | 2 | 4 | 1 | 1.5.1 |
| 8) | Identify the directory structure in which two users keep a subdirectory in their own directories<br>a) tree structure<br>b) cyclic graph directory structure<br>c)two level directory structure<br>d)acyclic graph directory<br>Ans: A – Tree structure | 1 | 2 | 5 | 1 | 1.7.1 |
| 9) | When you rename a file five times then the number of files in the disk is<br>a. 1<br>b. 2<br>c. 3<br>d. 5<br>Ans:a)1 | 1 | 1 | 5 | 4 | 4.6.2 |
| 10) | It is _____ to reread a page from the file system than to write it to swap space and then to reread it from there.<br>A. useless<br>B. less efficient<br>C. more efficient<br><br>D. more protective<br>Ans: C) more efficient | 1 | 2 | 5 | 1 | 1.6.1 |
| **PART B (4 X 5 = 20 Marks)<br>Instruction: Answer any 4** | | | | | | |
| 11) | Describe the steps involved in handling page fault with neat sketch<br>Ans:<br><br>1. First of all, internal table(that is usually the process control block) for this process in order to determine whether the reference was valid or invalid memory access.<br>2. If the reference is invalid, then we will terminate the process. If the reference is valid, but we have not bought in that page so now we just page it in. | 5 | 2 | 3 | 2 | 2.6.1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | 3. Then we **locate the free frame lis**t in order to find the free frame.<br>4. Now a disk operation is scheduled in order to read the **desired page into the newly allocated fram**e.<br>5. When the disk is completely read, then the **internal table is modified** that is kept with the process, and the page table that mainly indicates the page is now in memory.<br>6. Now we will restart the instruction that was interrupted due to the trap. Now the process can access the page as though it had always been in memory. | | | | | |
| 12) | What is the cause of thrashing? How does the system detect thrashing? Once it detects thrashing, what can the system do to eliminate this problem?<br>Ans:<br><br>In case, if the page fault and swapping happens very frequently at a higher rate, then the operating system has to spend more time swapping these pages. This state in the operating system is termed thrashing. Because of thrashing the CPU utilization is going to be reduced.<br><br>**Causes of Thrashing**<br><br>High degree of multiprogramming and lack of frames are two main causes of thrashing in the Operating system<br><br>**Effect of Thrashing**<br><br>**i)Global Page Replacement**<br><br>**ii)Local Page Replacement**<br><br>**Techniques used to handle the thrashing:**<br><br>**i)Working-Set Model**<br><br>**ii)Page fault frequency** | 5 | 1 | 3 | 1 | 1.6.1 |
| 13) | Explain how to manage Consistency semantics of shared files in distributed environment<br>Answer:<br>• **Consistency semantics** represent an important criterion for evaluating any file system that supports file sharing.<br>• These semantics specify how multiple users of a system are to access a shared file simultaneously.<br>• In particular, they specify when modifications of data by one user will be observable by other users. | 5 | 2 | 5 | 2 | 2.8.3 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | • These semantics are typically implemented as code with the file system.<br>• Consistency semantics are directly related to the process synchronization algorithms | | | | | |
| 14) | Discuss the importance of Swap space Management.<br>Swap-space management: Swap-space management is low- level task of the operating system. The main goal for the design and implementation of swap space is to provide the best throughput for the virtual memory system. Swap-space use: The operating system needs to release sufficient main memory to bring in a process that is ready to execute. Operating system uses this swap space in various way. Paging systems may simply store pages that have been pushed out of main memory. Unix operating system allows the use of multiple swap spaces. These swap space are usually put on separate disks, so the load placed on the I/O system by paging and swapping can be spread over the systems I/O devices. Swap-space location:<br>Swap space can reside in two places: 1. Separate disk partition 2. Normal file system | 5 | 2 | 5 | 2 | 2.6.2 |
| 15) | How file protection is provided in Unix system. Explain with access control matrix.<br>Answer:<br>• The most general scheme to implement identity dependent access is to associate with each file and directory an **access-control list (ACL)** specifying user names and the types of access allowed for each user.<br>• When a user requests access to a particular file, the operating system checks the access list associated with that file.<br>• If that user is listed for the requested access, the access is allowed. Otherwise, a protection violation occurs, and the user job is denied access to the file.<br><br>Mode of access: read(4 bits), write(2 bits), execute(1 bit)<br>Three classes of users on Unix / Linux(Default Permission)<br>RWX<br>a) **owner access** 7 ⇒ 1 1 1<br>RWX<br>b) **group access** 6 ⇒ 1 1 0<br>RWX<br>c) **public access** 1 ⇒ 0 0 1<br>To change Permissions the following commands are used :<br>Chmod: Change Mode(For files)<br>Chown: Change Owner(For Files)<br>Chgroup: Change group permission(For files) | 5 | 1 | 5 | 1 | 1.6.1 |
| **PART C (2 X 10 = 20 Marks)**<br>**Instruction: Answer All** | | | | | | |
| 16 ) A) | Given page reference string: 1,2,3,4,2,1,5,6,2,1,2,3,7,6,3,2,1,2,3,6 . Compare the number of page faults for LRU, FIFO and Optimal page replacement algorithm with 3 frames.<br>ANS:<br>LRU:<br>3 frames:<br>1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6<br>Frame 1: 1 4 5 1 7 2 -<br>Frame 2: 2 - 6 3 - - | 10 | 3 | 3 | 2 | 2.5.3 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | Frame 3: 3 1 2 - 6 1 6<br>15 faults<br>FIFO:<br>3 frames:<br>1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6<br>Frame 1: 1 4 6 3 - 2 - 6<br>Frame 2: 2 - 1 2 - 7 1<br>Frame 3: 3 5 1 6 3<br>16 faults<br>Optimal:<br>3 frames:<br>1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6<br>Frame 1: 1 - - 3 - -<br>Frame 2: 2 - - - 7 2 -<br>Frame 3: 3 4 5 6 - 1 6<br>11 faults<br><br>(OR) | | | | | |
| 16 ) B) | Describe the working of Frame Allocation mechanism.<br>ANS:<br><br>**Frame allocation algorithms –**<br>The two algorithms commonly used to allocate frames to a process are:<br>1. **Equal allocation:** In a system with x frames and y processes, each process gets equal number of frames, i.e. x/y. For instance, if the system has 48 frames and 9 processes, each process will get 5 frames. The three frames which are not allocated to any process can be used as a free-frame buffer pool.<br>2. **Proportional allocation:** Frames are allocated to each process according to the process size.<br>For a process $p_i$ of size $s_i$, the number of allocated frames is $\mathbf{a_i = (s_i/S)*m}$, where S is the sum of the sizes of all the processes and m is the number of frames in the system. For instance, in a system with 62 frames, if there is a process of 10KB and another process of 127KB, then the first process will be allocated (10/137)*62 = 4 frames and the other process will get (127/137)*62 = 57 frames.<br><br>**Global vs Local Allocation –**<br><br>1. **Local replacement:** When a process needs a page which is not in the memory, it can bring in the new page and allocate it a frame from its own set of allocated frames only.<br>2. **Global replacement:** When a process needs a page which is not in the memory, it can bring in the new page and allocate it a frame from the set of all frames, even if that frame is currently allocated to some other process; that is, one process can take a frame from another. | 10 | 1 | 3 | 2 | 2.5.1 |
| 17) A) | Read Request sequence = {176, 79, 34, 60, 92, 11, 41, 114} . Initial head position = 50. Implement Apply any 4-disk scheduling algorithm and find the total head movement for fetching the content from the given track numbers.<br>Answer: | 10 | 3 | 4 | 4 | 4.6.1 |

| | | | | | | |
|---|---|---|---|---|---|---|
| | First Come First Serve (FCFS)-510<br>Shortest Seek Time First (SSTF)-204<br>SCAN (Elevator) algorithm-226<br>C-SCAN-190<br>Look-291<br>c-look-156<br>(OR). | | | | | |
| 17) B) | Consider a disk with a rotational rate of 10,000 RPM, an average seek time of 8 ms, and an average of 500 sectors per track. Estimate the average time to read a random sector from disk. Do this by summing the estimates of the seek time, rotational latency, and transfer time.<br>**Answer:**<br>• The average access time is the sum of the seek time, rotational latency, and transfer time.<br>• The seek time is given as 8ms. Once the head is in the right place, on average we will need to wait for half a rotation of the disk for the correct sector to come under the head. Thus, on average, the rotational latency is half the time it takes the disk to make a complete revolution.<br>• The disk spins at 10000 RPM, so it takes 1/10000 of a minute to make one revolution. Equivalently, (1000 ms/sec × 60 sec/minute) / 10000 RPM = 6 ms to make one revolution. So rotational latency is 3ms.<br>• The transfer time is the time it takes for the head to read all of the sector. The head is now at the start of the sector, so how long does it take for the entire sector to go past the head? Since there are on average 500 sectors per track, we need 1/500th of a revolution of the disk to read the entire sector. We can work this out as (time for one revolution of disk) / 500 = 6ms / 500 = 0.012ms.u<br>• So the total time is 8ms + 3ms + 0.012ms ≈ 11ms. We can clearly see that getting to read the first byte of the sector takes a long time, but reading the rest of the bytes in the sector is essentially | 10 | 5 | 5 | 2 | 2.8.1 |