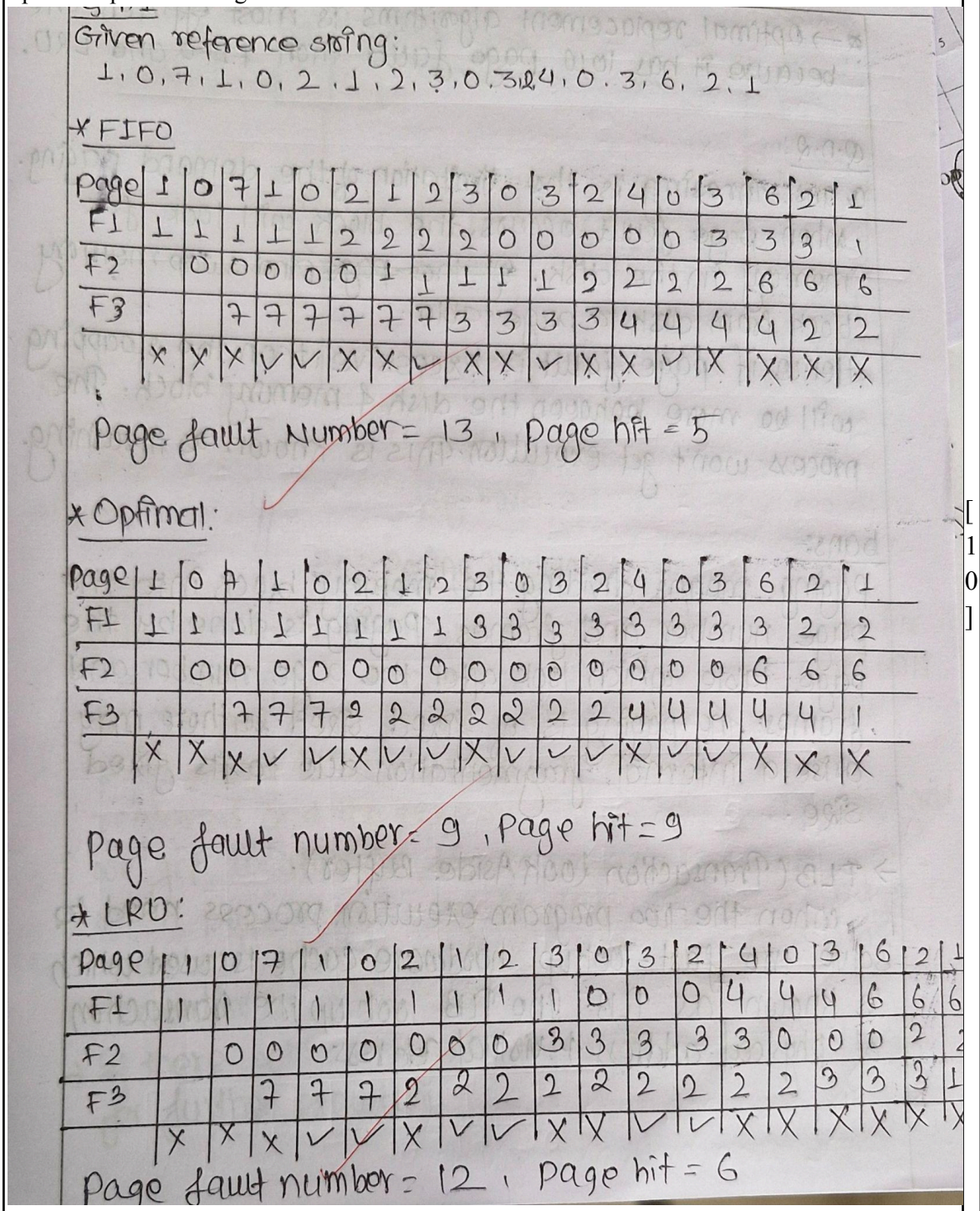
USN | | | | | | | | | | |

Internal Assessment Test 2 – December 2024

Sub:	**OPERATING SYSTEMS ANSWER SCHEME**					Sub Code:	**BCS303**	Branch:	**AINDS / CS (DS)**	
Date:	**17/12/2024**	Duration:	**90 minutes**	Max Marks:	**50**	Sem	**III**		**OBE**	
Answer any FIVE Questions								**MARKS**	**CO**	**RBT**

1	a	Consider the page reference string: 1,0,7,1,0,2,1,2,3,0,3,2,4,0,3,6,2,1 for a memory with three frames. Determine the number of page faults using the FIFO,Optimal,and LRU replacement algorithms. Which algorithm is most efficient? Optimal replacement algorithms is the most efficient one.  Given reference string: 1, 0, 7, 1, 0, 2, 1, 2, 3, 0, 3, 2, 4, 0, 3, 6, 2, 1 * FIFO Page fault Number = 13, page hit = 5 * Optimal: Page fault number = 9, page hit = 9 * LRU: Page fault number = 12, page hit = 6	[10]	4	L3

2	a	What is thrashing ? **Thrashing** is a condition in an operating system where excessive paging occurs, leading to a significant drop in performance. It happens when the working set of a process exceeds the available physical memory, causing the system to spend more time swapping pages in and out of memory than executing processes.	[2]	4	L1

	b	What is Paging ? Explain with neat Diagram paging Hardware with TLB. Paging is a memory management scheme that eliminates the need for contiguous allocation of physical memory. In a system using paging, the physical memory is divided into small fixed-size blocks called *frames*, while the logical memory (or virtual memory) is divided into blocks of the same size called *pages*. [2] **Paging Hardware with Translation Lookaside Buffer (TLB)** Paging works by breaking both the physical and logical memory into equal-sized units: ● **Pages:** These are blocks of logical memory (virtual memory), and their size is typically 4 KB or 8 KB. ● **Frames:** These are blocks of physical memory, also of the same size as the pages. **Page Table:** The page table is used to store the mapping between the logical page number and the physical frame number. Each entry in the page table holds the physical address of the corresponding frame. --- **Paging with TLB (Translation Lookaside Buffer)** The Translation Lookaside Buffer (TLB) is a small, high-speed cache used to improve the performance of address translation. The TLB stores a subset of the most recently used page table entries. When a process accesses a memory address, the system first checks the TLB. If the entry is found (a "hit"), the corresponding physical address is immediately available. If the entry is not found (a "miss"), the system accesses the page table in main memory to perform the translation and then updates the TLB. **Paging System with TLB:** 1. **Logical Address (Virtual Address):** The CPU generates a logical address, which is divided into two parts: ○ **Page Number (P):** This part is used as an index into the page table. ○ **Page Offset (D):** This part is the offset within the page, which is used to access the data within the page. 2. **TLB Lookup:** ○ The system first looks up the page number in the TLB. ○ If a TLB hit occurs, the corresponding physical address is directly retrieved. ○ If there is a TLB miss, the page number is used to index the page table to find the corresponding frame in physical memory. The TLB is then updated with this new mapping. 3. **Frame Number and Physical Address:** ○ The frame number from the page table (or the TLB) is combined with the page offset to form the physical address. ○ The data is then accessed from the physical memory using the physical address.[4] DIAGRAM [2]	[8]	4	L2

3		What is demand paging? Explain the steps in handling page faults using the appropriate diagram **Demand Paging** **Demand Paging** is a memory management scheme in which pages are loaded into physical memory only when they are needed, i.e., when a page fault occurs. It is an efficient technique to handle virtual memory, where the system does not load all pages of a process into memory at once. Instead, pages are brought into memory on demand when the program tries to access them, reducing the amount of memory required for a process. In demand paging, when a process tries to access a page that is not in memory, a **page fault** occurs. The operating system (OS) then handles this fault by loading the required page from secondary storage (usually a disk) into physical memory. This technique allows systems to run larger programs than the available physical memory by relying on swapping pages in and out of memory. **Steps in Handling Page Faults:** 1. **Page Access Request:** ○ The CPU generates a virtual address for a memory access. The virtual address is divided into a page number and a page offset. 2. **Check in Memory (TLB or Page Table Lookup):** ○ The system first checks if the page number is available in memory. ○ The **Translation Lookaside Buffer (TLB)** is checked first. If the TLB does not have the mapping, the **page table** is checked for the mapping between the virtual page and the physical frame. 3. **Page Fault Occurs (Page Not in Memory):** ○ If the page is not found in memory (i.e., it's not in the TLB and the page table entry indicates it is not in physical memory), a **page fault** occurs. 4. **OS Handles the Page Fault:** ○ The **Operating System** now handles the page fault by performing several steps: ■ **Save the State of the Process:** If necessary, the state of the process is saved, and the OS ensures no further memory access takes place while it handles the fault. ■ **Locate the Page on Disk (Swap Space):** The OS locates the page on the disk or swap space. This page is identified by its virtual page number. ■ **Check for Free Frame:** The OS checks if there is a free frame in physical memory. If no free frame exists, the OS may choose a page to evict (swap out). ■ **Evict a Page (if necessary):** If memory is full, a page is chosen for eviction. The **page replacement algorithm** (e.g., LRU, FIFO) is used to select which page to swap out. If the page being swapped out has been modified, it is written back to disk. ■ **Load the Required Page into Memory:** The required page is loaded from secondary storage (disk) into a free frame in memory. ■ **Update Page Table and TLB:** The page table is updated with the new mapping, and the TLB is updated as well. 5. **Return to Process Execution:** ○ Once the page is loaded into memory and the page table is updated, the process can resume execution from the point where the page fault occurred. 6. **Repeat the Process:** ○ If additional pages are needed, further page faults will occur, and the system will repeat the steps of handling the fault.	[10]	4	L2

		DIAGRAM[3]			
4	a	**What is a file? Explain different file allocation methods? (1 mark)** A **file** is a collection of data or information that is stored on a storage medium, such as a hard disk, SSD, or network storage. It can contain data in various forms, such as text, images, videos, or program instructions. Files are a basic unit of storage used by operating systems to organize and manage data. Each file is typically associated with attributes like name, type, size, creation date, and permissions. In a computer system, files are stored in **file systems**, which provide mechanisms to create, read, write, and delete files, as well as manage their storage.(2 marks) **File Allocation Methods(1 mark)** When it comes to storing files on a disk, the operating system needs a way to manage the allocation of space on the disk for these files. There are several methods for file allocation, each with its pros and cons. The common file allocation methods are: 1. **Contiguous Allocation** 2. **Linked Allocation** 3. **Indexed Allocation** **Explanation(3 marks)**	[5]	5	L2

	b	**List the Goals of Protection and Write a short note on access matrix.(1 mark)** The goals of protection in computer systems are to ensure the integrity, security, and efficient management of resources, preventing unauthorized access or modification. These goals are essential to protect data and resources from threats like malicious software, accidental damage, and unauthorized access. The key goals of protection are,(2marks) 1. Confidentiality 2. Integrity 3. Availability 4. Authentication 5. Authorization 6. Accountability 7. Least Privilege **Access Matrix(2marks)** The Access Matrix is a conceptual model used to represent the permissions or access rights that subjects (users, processes, etc.) have on objects (files, resources, etc.) in a system. It is a table where each row corresponds to a subject, each column corresponds to an object, and the matrix entries specify the access rights or permissions granted to the subject for each object. Structure of an Access Matrix: ● Subjects (Rows): These are entities (users, processes, or groups) that interact with the system's resources. ● Objects (Columns): These are the resources (files, devices, or databases) that the subjects can access or modify. ● Access Rights (Matrix Entries): These are the permissions or operations a subject can perform on an object, such as read, write, execute, delete, or modify.	[5]	6	L2
5	a	**Suppose that a disk drive has 5000 cylinders, numbered 0 to 4999.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 86,1470,913,1774,948,1509,1022,1750, 130.Starting from current head position, what is a total distance (in cylinders) that the Disk arm move to satisfy all a pending requests, for each of the following disk Scheduling algorithms?** **1. FCFS 2. SSTF 3. SCAN 4. LOOK** **what is a total distance (in cylinders) that the Disk arm move to satisfy all a pending request:** FCFS:7081 SSTF:1745 SCAN:9769 LOOK:3319	[10]	5	L3

6		**Write note on** **Discuss various directory structures with neat diagrams** **List and explain free space management Methods.** **Directory Structures in Operating Systems(2 marks)** A **directory structure** is a way of organizing and storing files in a file system. It provides a hierarchy or organization of files, making it easier to manage and access them. A directory can contain files and other directories (subdirectories), forming a tree-like structure. The design of the directory structure plays a crucial role in how efficiently a file system can locate, store, and manage files. Various types of directory structures are used in different operating systems. Types of Directory Structures**(1 mark)** 1. Single-Level Directory 2. Two-Level Directory 3. Tree-Structured Directory 4. Acyclic-Graph Directory 5. General Graph Directory **explanation and diagram:(2 marks)** **Free Space Management Methods** In file systems, free space refers to the areas on the storage medium (like a hard drive or SSD) that are not occupied by files. These unused spaces are available to store new files or expand existing ones. Efficient management of free space is crucial for optimizing the performance of the file system and ensuring that files are allocated in an organized manner.(2 marks) The main methods of free space management aim to track, allocate, and reclaim free space efficiently. The common free space management methods are:(1 mark) 1. Bit Vector (Bitmap) 2. Linked List 3. Free Block List 4. Grouping 5. Counting **Comparison of Free Space Management Methods or explanation (2marks)**	[5+5]	5	L2

Method	Advantages	Disadvantages
Bit Vector	Efficient in space, fast allocation	Large bitmap for large disks, scanning overhead
Linked List	Dynamic, easy to manage	Pointer overhead, slower allocation
Free Block List	Direct access, relatively quick	Memory or disk space overhead, list maintenance
Grouping	Space-efficient, reduces fragmentation	Complex management, requires updates
Counting	Efficient for large contiguous blocks	Limited flexibility, complex maintenance in fragmented space

CI **CCI** **HOD**