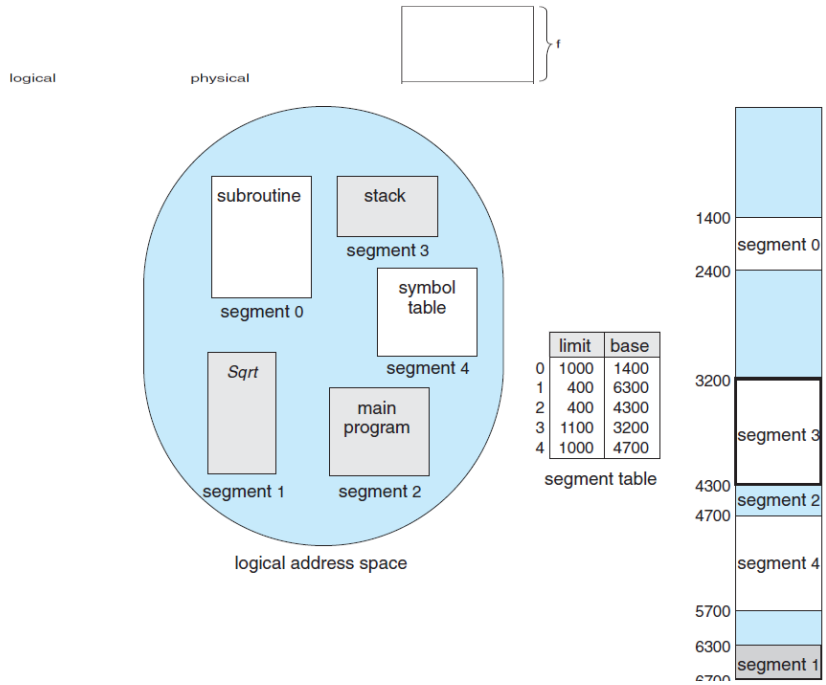
USN | | | | | | | | | | |

Internal Assessment Test 3 – September 2023

Sub:	**Operating Systems**					Sub Code:	21CS44	Branch:	ISE	
Date:	11/09/2023	Duration:	90 min's	Max Marks:	50	Sem/Sec:	IV A, B & C			OBE

Answer any FIVE FULL Questions

		MARKS	CO	RBT
1	Distinguish between the following: i. Internal fragmentation and external fragmentation. **Internal Fragmentation**: • **Definition**: Internal fragmentation occurs when allocated memory space within a memory block is not fully utilized, resulting in wasted memory. • **Cause**: It is typically a result of memory allocation in fixed-size blocks or units where the allocated block is larger than the data it holds. • **Consequence**: Over time, internal fragmentation can lead to inefficient use of memory resources, as unused space accumulates within allocated blocks. **External Fragmentation**: • **Definition**: External fragmentation happens when there is enough free memory space in the system to satisfy a memory allocation request, but the free space is scattered in small, non-contiguous blocks. • **Cause**: It arises in dynamic memory allocation systems when memory allocations and deallocations leave behind small gaps or fragments between allocated regions. • **Consequence**: External fragmentation can lead to inefficient memory utilization, as even though there may be sufficient total free memory, it cannot be efficiently used to fulfill larger memory allocation requests due to the scattered nature of the free space. ii. Paging and segmentation **Paging**: • **Definition**: Paging is a memory management scheme used in computer operating systems to implement virtual memory. It divides physical memory and processes' logical address spaces into fixed-size blocks called "frames" and "pages," respectively. • **Function**: It provides a way to efficiently manage memory by mapping pages in the logical address space to frames in physical memory. This mapping is maintained in a data structure called a "page table." • **Advantages**: Paging allows for more efficient use of physical memory, as it eliminates external fragmentation and enables non-contiguous allocation of memory pages. It also simplifies memory management and provides a straightforward approach to implementing virtual memory and memory protection. • **Disadvantages**: Paging can introduce a small amount of internal fragmentation if the last page in a frame is not fully utilized. Additionally, page tables can become large for processes with large address spaces. **Segmentation**: • **Definition**: Segmentation is another memory management scheme that divides a process's logical address space into different segments, each representing a specific type of data or code (e.g., code, data, stack, heap). • **Function**: It provides a way to organize and protect memory by dividing it into logical segments, each with its own base address and size. Segmentation allows for	10	CO3	L2

	more flexible memory allocation and access control. • **Advantages**: Segmentation is suitable for managing memory in high-level programming languages where data structures have varying sizes and lifetimes. It also offers better protection, as each segment can have its own access control settings. • **Disadvantages**: Segmentation can lead to external fragmentation, as different segments may be of varying sizes, causing gaps between segments. Managing the mapping of segments to physical memory can be more complex compared to paging.			
2	Explain Page Table and Segment Table. **Page Table**: • Maps virtual addresses to physical addresses. • Organized as an array or tree. • Used in virtual memory systems with fixed-size pages.  Diagrams for both are to be drawn.	10	CO3	L2
3	Explain the different techniques to structure the page table for the following: i. Hierarchical ii. Inverted iii. Hashed **1. Hierarchical Page Table:** • **Purpose**: Hierarchical page tables are used to manage large address spaces efficiently by breaking down the page table structure into multiple levels,	10	CO3	L2

reducing the memory overhead required for a flat page table.

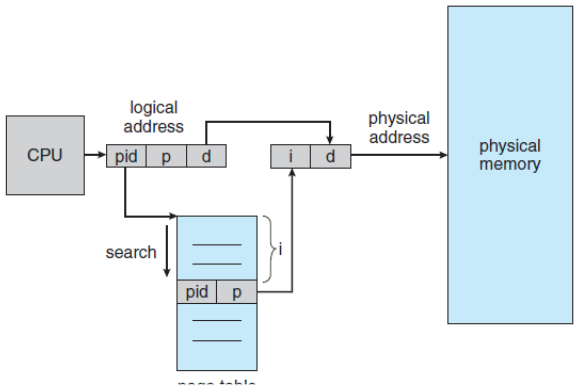
- **Structure**:
  - The page table is organized as a multi-level tree-like structure.
  - At the top level, there is a master page table or page directory.
  - Each entry in the top-level page table points to a second-level page table (page directory entry).
  - Second-level page tables may point to third-level tables and so on.
  - The lowest level tables contain the actual page table entries mapping virtual to physical addresses.
- **Advantages**:
  - Efficient memory utilization for sparse address spaces.
  - Reduced memory overhead compared to a flat page table.
- **Disadvantages**:
  - Increased complexity in address translation due to multiple levels.
  - Slightly slower access times due to multiple table lookups.

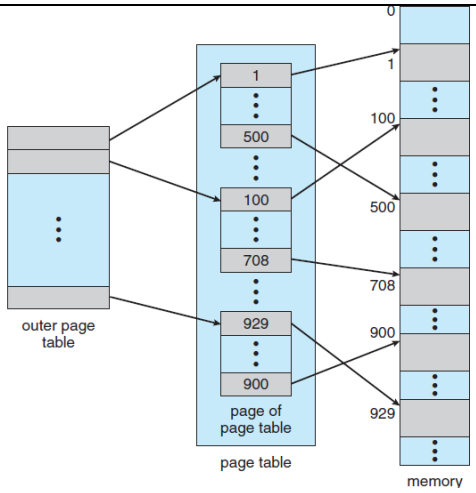
2. Inverted Page Table:

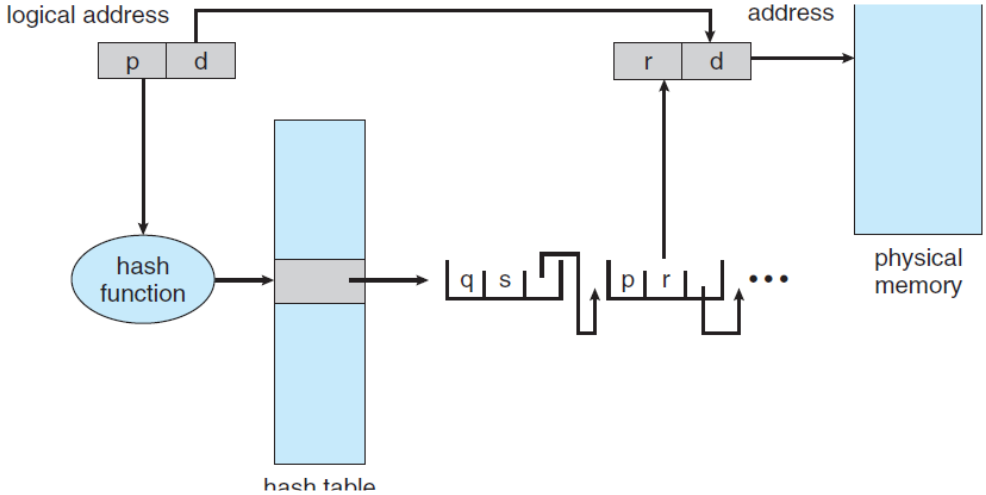
- **Purpose**: Inverted page tables are used to manage memory efficiently in systems with a large amount of physical memory but relatively small address spaces.
- **Structure**:
  - Instead of one entry per virtual page, there is one entry per physical page frame.
  - Each entry contains the virtual address and process identifier (PID) of the owning process.
- **Advantages**:
  - Efficient for systems with a large amount of physical memory.
  - Reduces memory overhead since there are fewer entries.
- **Disadvantages**:
  - Slower address translation because it requires searching the entire table to find a mapping.
  - May not be suitable for systems with large address spaces or a high degree of address space sharing.

3. Hashed Page Table:

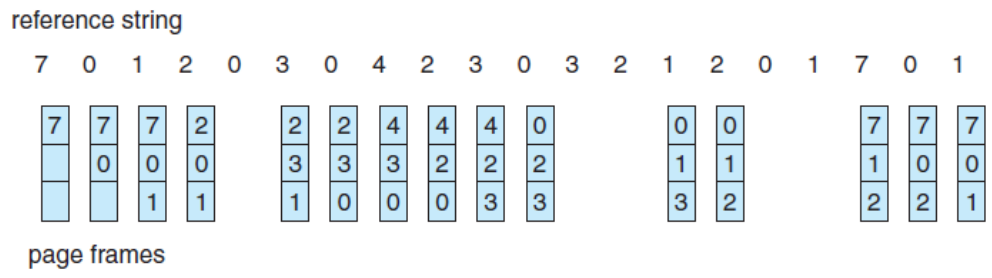
- **Purpose**: Hashed page tables are used to reduce the time complexity of address translation by using a hash function to index the page table entries.
- **Structure**:
  - A hash table is used, where the hash function maps virtual page numbers to hash indices.
  - Each hash table entry contains a linked list of page table entries corresponding to the same hash index.
- **Advantages**:
  - Provides faster address translation compared to a linear search (as in inverted page tables)



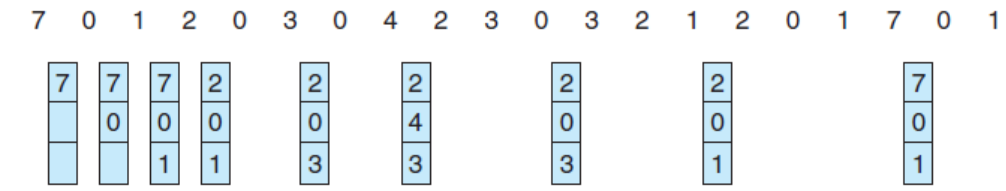




4	Discuss the various directory structures with suitable diagrams. Directory structures are used in file systems to organize and manage files and directories. Different operating systems use various directory structures, each with its own way of arranging and accessing files. Here, I'll describe four common directory structures and provide a brief overview of each, without specific diagrams: 1. **Single-Level Directory Structure**: • **Description**: In a single-level directory structure, all files and directories are organized in a single directory or folder. Each file or directory must have a unique name within that directory. • **Advantages**: Simple and straightforward, suitable for small-scale systems. • **Disadvantages**: Limited scalability and organization, can become cluttered with a large number of files. 2. **Two-Level Directory Structure**: • **Description**: In a two-level directory structure, there is a separate user directory for each user. Each user can have their own set of files and directories within their user directory. • **Advantages**: Better organization and separation of user files, suitable for multi-user systems. • **Disadvantages**: Limited hierarchy, as there are only two levels (user and 3. **Tree-Structured Directory Structure**:	10	CO2	L2

	• **Description**: A tree-structured directory structure arranges files and directories in a hierarchical tree-like fashion. Each directory can contain subdirectories and files, creating a more extensive hierarchy. • **Advantages**: Provides a flexible and scalable organization with multiple levels. 4. **Acyclic-Graph Directory Structure**: • **Description**: In this structure, directories and files are organized as a directed acyclic graph (DAG). It allows for multiple parents and offers a high degree of flexibility and sharing. • **Advantages**: Allows for complex relationships between directories and files, supports shared resources. • **Disadvantages**: Complex to implement and navigate Each directory structure has its strengths and weaknesses, and the choice depends on the specific requirements of the file system and the intended usage patterns. The tree-structured directory structure is the most common and widely used in modern operating systems like Windows, macOS, and Linux.			
5a	Explain any two page replacement algorithms. **1. FIFO (First-In-First-Out)**: • Replaces the oldest page in memory. • Simple to implement but can suffer from Belady's Anomaly. • Doesn't consider the frequency of page access. **2. LRU (Least Recently Used)**: • Replaces the least recently used page. • More sophisticated than FIFO, as it considers recent page accesses. • Requires tracking and updating timestamps for each page.	6	CO3	L2
5b	Explain Thrashing in detail. Thrashing is a severe performance issue in computer systems, occurring when the system spends excessive time swapping data between RAM and disk due to insufficient physical memory or high memory demand. It leads to slow performance, frequent disk I/O operations, and unresponsiveness.	4	CO4	L2
6	For the following page reference string 7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1. Calculate the page fault ratio and hit ratio using FIFO, Optimal and LRU using 3 frames. FIFO  Optimal	10	CO3	L3

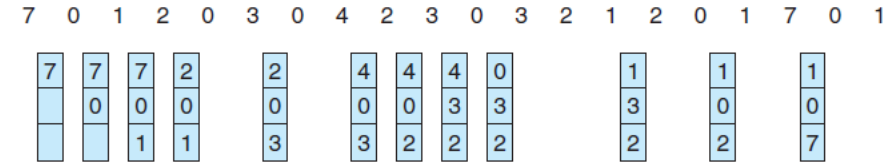
reference string



page frames

LRU

reference string



page frames