Unit - 3

VISION DATE : PAGE NO.

1. Explain three stages of Binding of Instruction and data to memory.

→ Address binding of instruction and data to memory addresses can happen at three different stages.

- Compile time : If memory location known a priori absolute code can be generated; must recompile code if starting location changes.
- load time : Must generate relocatable code if memory location is not known at compile time
- Execution time : Binding delayed until run time if the process can be moved during its execution from one memory segment to another. Need hardware support for address maps

2. Discuss the concept of logical and Physical address space

→ Logical Address space

- The logical address space is the set of all address generated by the CPU. It's the view of memory that a program has.
- The logical address space is generated during Program execution.
- It allows programs to have their own independent address spaces, even if they share the same physical memory.

→ Physical Address space

- The Physical address space is the set of actual memory locations in the computer's RAM.
- These addresses directly correspond to the hardware memory locations.
- The memory Management unit is responsible for translating logical addresses to Physical addresses.

3. Explain Dynamic loading

→ Dynamic loading is a runtime mechanism that allows a Computer Program to load libraries or modules into memory as needed, rather than loading them all at startup.

→ "Routine is not loaded untill it is called"

This is the fundamental principle. Instead of loading all program code at startup, pieces of code are loaded into memory only when they are needed during program execution.

→ "Better memory-space utilization; unused routine is never loaded"

This highlights the primary advantage. By avoiding the loading of unnecessary code, dynamic loading conserves valueable memory resources. This is especially important for large applications with many features and systems with limited memory.

→ "useful when large amounts of code are needed to handle infrequently occuring cases".

This Point to a specific use cases. Consider a program that handles rare error conditions or optional features. Instead of loading the code for these situations into memory all the time, it can be loaded only when the corresponding event occurs. This avoids wasting memory on code that is rarely used.

→ "No special support from the operating System is required implemented through program design"

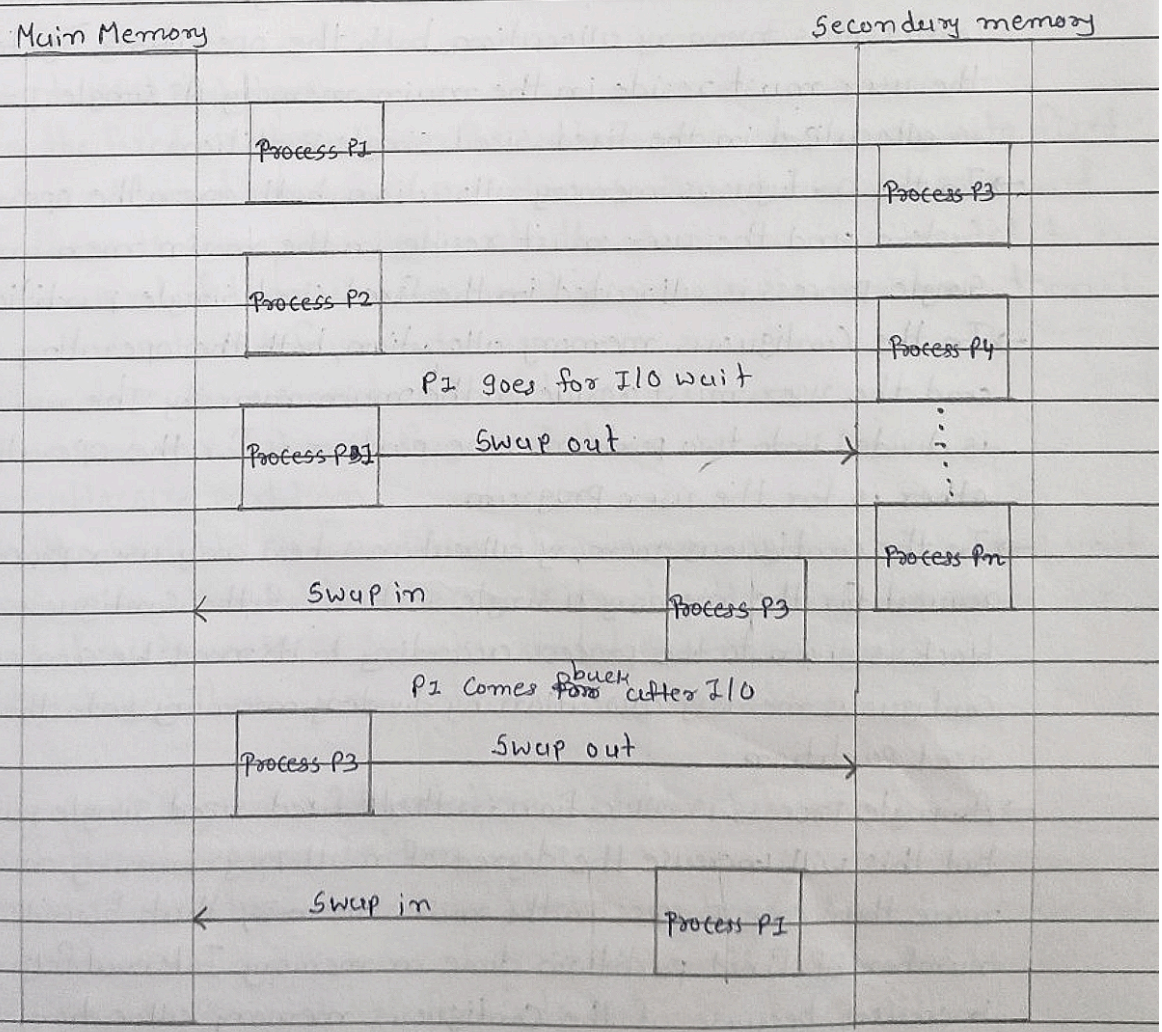
This is crucial point. Dynamic loading can be implemented at the application level, without relying on specific operating system features. This can be done by:

- Structureing the program into modules or libraries.
- using conditional loading logic within the program itself.
- This implies that the program itself mamages the loading and unloading of the routies.

4. Explain the swapping of process with memory using diagram.

→ Swapping is a mechanism in which a process can be swapped temporarily out of main memory to secondary storage and make that memory available to other processes. At some later time, the system swaps back the process from the secondary storage to main memory.

→ Swapping is a mechanism in which a process can be swapped temporarily out of main memory to secondary storage and make that memory available to other processes. At some later time, the system swaps back the process from the secondary storage to main memory.



→ Though performance is usually affected by swapping process but it helps in running multiple and big processes in parallel and that's the reason Swapping is also known as a technique for memory Compaction.

→ The total time taken by swapping process includes the time it takes to move the entire process to a secondary disk and then to copy the process back to memory, as well as the time the process takes to regain main memory.

5. Explain the Concept of Contiguous memory allocation with diagram

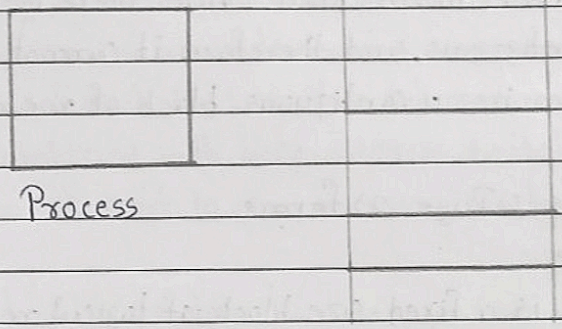
→ It means freely available memory partitions are not scattered here and there across the whole memory space. In the Contiguous memory allocation, both the operating system and the user must reside in the main memory. A single process is allocated in the fixed sized single partition.

→ In the contiguous memory allocation, both the operating System and the user must reside in the main memory. A single process is allocated in the fixed sized single partition.

→ In the Contiguous memory allocation, both the operating system and the user must reside in the main memory. The main memory is divided into two portions one portion is for the operating and other is for the user program.

→ In the Contiguous memory allocation when any user process request for the memory a single section of the contiguous memory block is given to the process according to its need. We can achieve Contiguous memory allocation by dividing memory into the fixed-sized partition.

→ A single process is allocation in the fixed sized single partition But this will increase the degree of multiprogramming means more than one process in the main memory that bounds the number of fixed partition done in memory. Internal fragmentation increases because of the Contiguous memory allocation.



→ Fixed sized partition

In the fixed sized partition the system divides memory into fixed size partition here entire partition is allowed to a process and if there is some wastage inside the partition is allocated to a process and if there is some wastage inside the partition then it is called internal fragmentation.

Advantage: Managment or book keeping is easy.

Disadvantage: Internal fragmentation

→ Variable size partition

In the variable size partition, the memory is treated as one unit and space allocated to a process is exactly the same as required and the leftover space can be reused again.

Advantage: There is no internal fragmentation.

6. Define 1) Internal fragmentation 2) External fragmentation

→ 1) Internal fragmentation

Occurs when allocated memory blocks are larger than the requested memory, resulting in unused space within the allocated block. This wasted space is "internal" to the allocated partition.

→ 2) External Fragmentation

Occurs when there is enough total free memory to satisfy a request, but the available memory is scattered in non-contiguous blocks. This means that while there is sufficient free space, it is not contiguous, and therefore, it cannot be allocated to a process that requires a contiguous block of memory.

7. Define 1) Page 2) frame

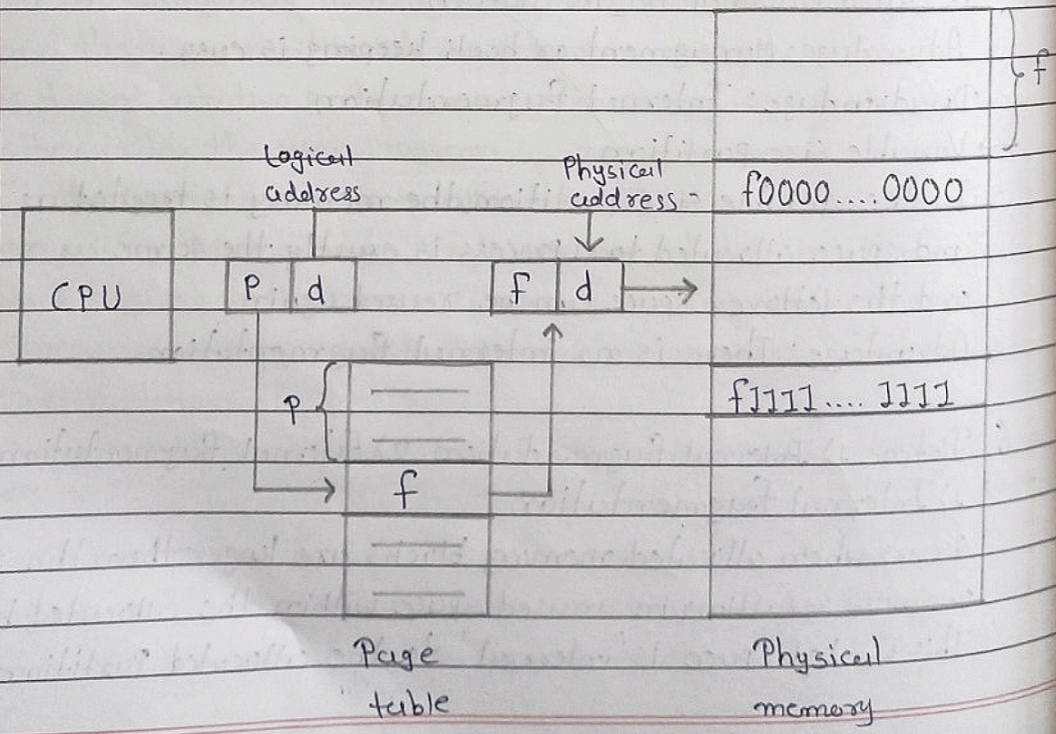
→ 1) Page

A page is a fixed-size block of logical memory. It's the unit of memory that the operating system and memory management unit use to divide a process's logical address space.

→ 2) Frame

A frame is fixed-size block of physical memory. It's the unit of physical memory that corresponds to a page. Essentially, a frame is where a page is stored in physical RAM.

8. Explain address translation architecture with diagram.



→ Address translation architecture is the system that converts logical address, generated by the CPU, into physical addresses, which correspond to actual locations in main memory.

→ Address generated by CPU is divided into:

- Page number (p) - used as an index into a page table with contains base address of each page in physical memory.
- Page offset (d) - combined with base address to define the physical memory address that is sent to the memory unit.

9. Explain Paging with TLB in detail.

→ In computer operating systems, Paging is a memory managment scheme by which a computer stores and retrieves data form secondary storage for use in main memory. In this scheme, the operating system retrieves data form secondary storage in same-size blocks called pages. Paging is a memory managment scheme that eliminates the need for contiguous allocation of physical memory. This scheme permits the physical address space of a process to be non-contiguous

→ Translation Lookaside Buffer:
The TLB is a hardware cache that stores recent page table entries. It accelerates address translation by providing a fast lookup for frequently used mappings.

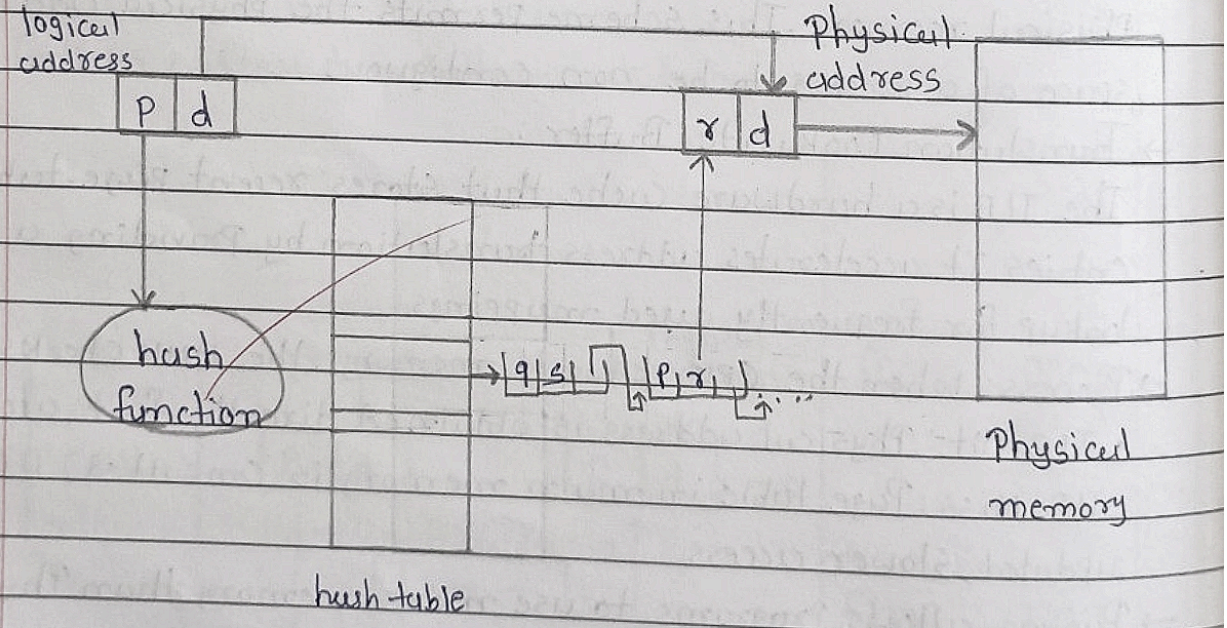
→ Process: When the CPU requests memory, the MMU checks the TLB
- TLB Hit: Physical address is obtained directly, fast access
- TLB Miss: Page table in main memory is consulted, TLB is updated, slower access

→ Paging allows programs to use more memory than physically available. The TLB significantly improves performance by reducing the overhead of page table lookups, making virtual memory efficient.

10) List three techniques of Page table structure, Explain any one with diagram.

→ Three common techniques for Page table structure are:
1. Hierarchical Page Tables
2. Hashed Page Tables
3. Inverted Page Tables

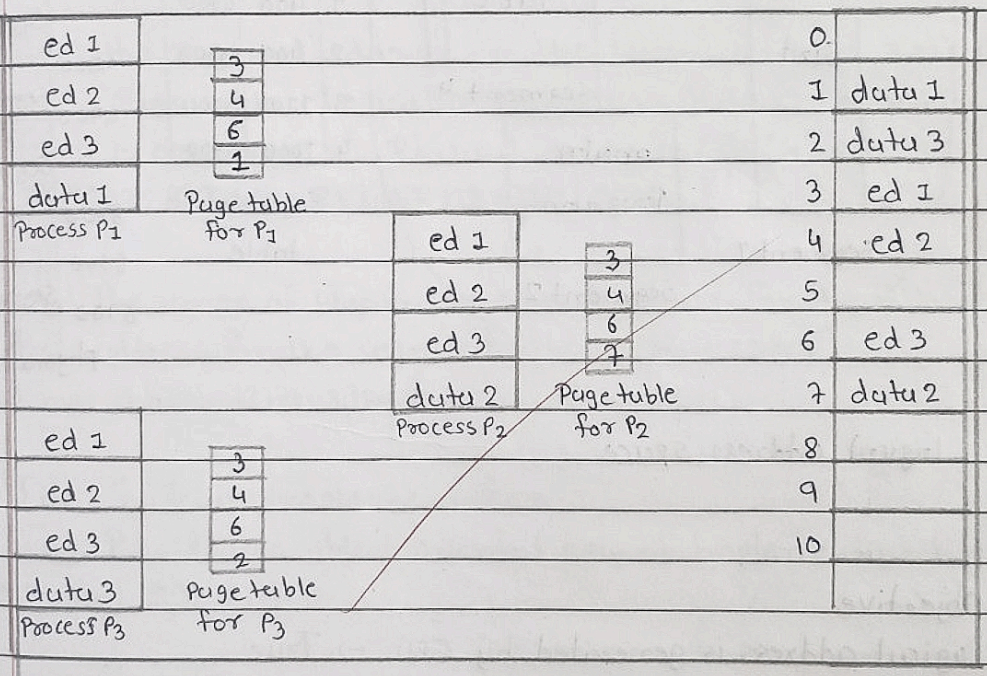
→ 2. Hashed Page Tables
- Common in address spaces > 32 bits
- The virtual page number is hashed into a page table. This Page Table contains a chain of elements hashing to the same location
- Virtual page numbers are compared in this chain searching for a match. If a match is found, the corresponding physical frame is extracted.



11. Explain shared Page example with diagram

→ Shared Code
- One copy of read-only code shared among Processes
- Shared code must appear in same location in the logical address space of all processes.

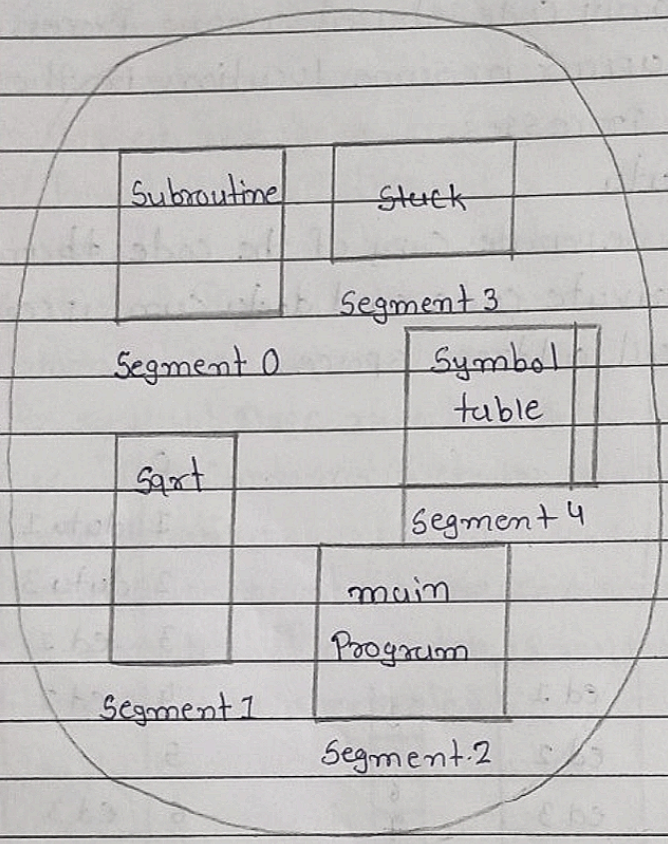
→ Private code and data
- Each process keeps a separate copy of the code ~~and~~ and data.
- The Pages for the private code and data can appear anywhere in the logical address space.



12. Draw example of Segmentation with suitable values of base and limit.

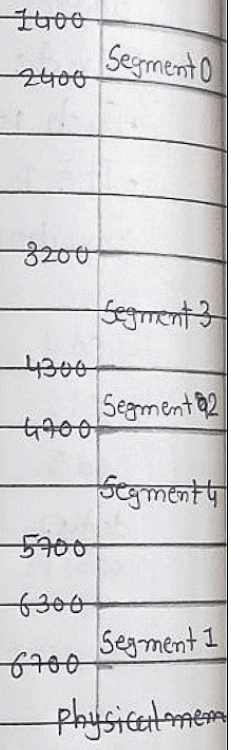
→ In operating System, Segmentation is a memory managment technique in which, the memory is divided into the variable size parts. Each part is known as Segment ~~which~~ which can be allocated to a process. The details about each segment are stored in table called as segment table.

→ Segment table contains mainly two information about segment
1. Base : It is the base address of the segment
2. Limit : It is the length of the segment



	limit	base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

Segment table



Logical address space

⟹ Objective

1. Logical address is generated by CPU → True

2. ~~limit~~ Full forms : MMU → Memory Managment Unit

3. limit register contains range of Physical address. → false

4. What do you mean by hole in memory?
→ A hole in memory refers to a contiguous block of free spa
is physical memory that is not currently allocated to any Pro

5. How first fit and best are better than worst fit?
→ Worst-fit takes memory from the largest free region, giving the process the most space to grow. As a general rule, first-fit is fastest, but increases fragmentation. Best-fit can result in small unusable holes, but in general studies have shown that it can produce better utilization than worst-fit.

6.

0	a
1	b
2	c
3	d

Page table

0	5
1	6
2	1
3	2

Page size = 4kb

Find the physical address for the logical address 3 as per the given page table. Logical address is on page 0 - offset in 3
→ Page 0 is on frame 5. Physical address = (5*4) + 3 = 23

7. Find the range of physical memory address for the following segment:
Base address for the segment: 1400, Limit: 1000
→ Range = base address + limit = 1400 + 100 = 1500

⟹ File System implementation

1. ____ Provide an object-oriented way of implementing file systems.
→ Virtual file system

2. Linear list with hash data structure is called ____ table.
→ Directory

3. Situations where two file names hashed to the same location is called
→ Collision

4. A file consists of one or more extents
→ True

5. When address pointer contains not data but the address of blocks which contain data, it is called ——

→ Double indirect

⟹ Descriptive

1. Write about file System Structure?

→ A file system structure is a method that an operating system uses to manage and organize data stored on storage devices like hard drives, SSDs and USB drives.

→ File structure

- Logical Storage unit: Files are treated as logical storage units that allow data organization
- Collection of Related Information: File often contain related data grouped together for easy access.

→ File System Storage

- The file system reside on secondary storage since this type of storage is non-volatile and ideal for long-term data storage

→ Layered Organization

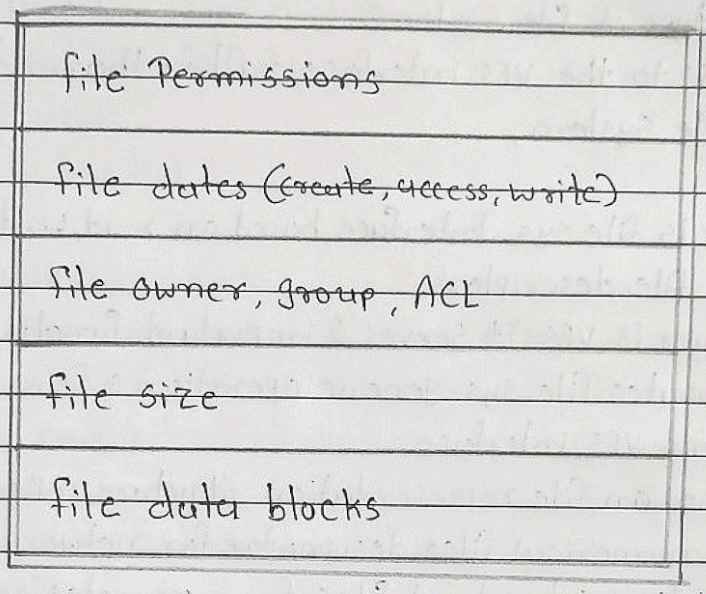
- The file system is organized into layers, which helps manages complex data structures efficiently
  - Application layer - Interfaces for user interaction
  - Logical file system - Manages metadata and file control blocks
  - Physical file system - Handles data blocks on storage media

→ File Control Block

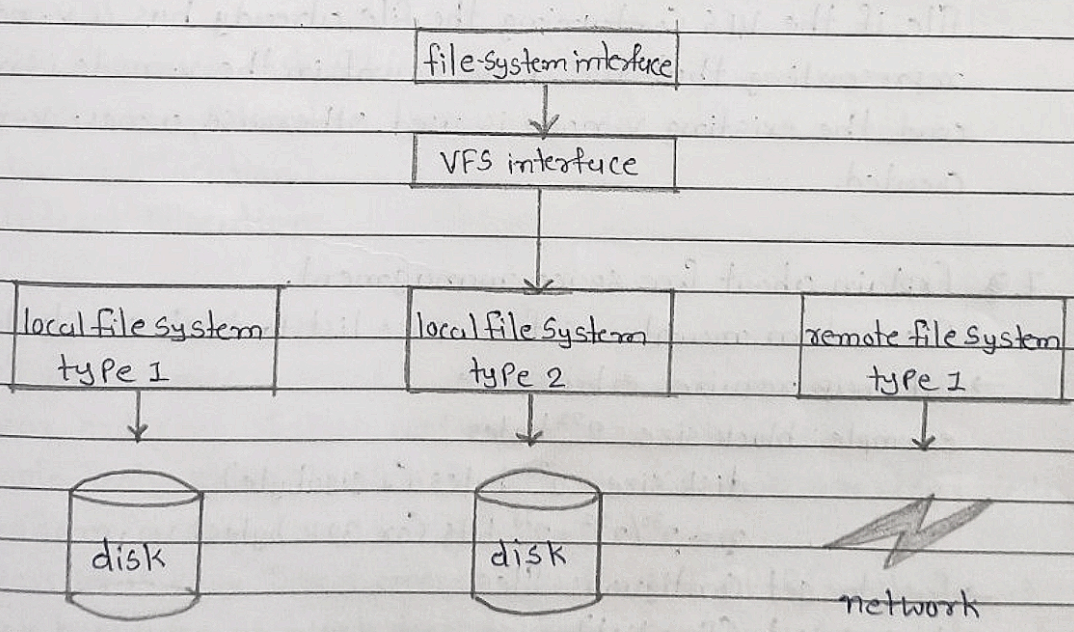
- The file control Block is a data structure that stores essential information about a file, including file name, file size, file location on disk, Permission and ~~order~~ ownership.

2. Draw the file access control list or file control block?

→



3. What do you mean by virtual file system. Draw and explain schematic view of VFS?



→ Virtual file system Provide an object-oriented way of implementing file system.

→ VFS allows the same system call interface to be used for different type of file systems.
→ The API is to the VFS interface, rather than any specific type of file system.

→ first layer is file sys. Interface based on read, write and close calls and file descriptors.
→ Second layer is VFS It serves 2 important functions.
1) It separates file sys-generic operations from their implementation by defining VFS interface.
2) VFS based on file representation structure called vnode that contains numerical file designator for network file.
→ So VFS distinguishes local files from remotes ones and it is further divided into their file types.
→ A virtual node (v-node) represents access to an object within a virtual file system... When a user attempts to open or create a file, if the VFS containing the file already has a v-node representing that file, a use count in the v-node is incremented and the existing v-node is used. Otherwise, a new v-node is created.

7. Explain about free space managment.
→ file system maintains free-space list to track available blocks.
→ Bit map requires extra space
example: block size = 2^{21} bytes
disk size = 2^{30} bytes (1 gigabyte)
$n = 2^{30}/2^{12} = 2^{18}$ bits (or 32K bytes)
→ Easy to get contiguous files
→ Linked list (free list)
• Cannot get contiguous space easily
• No waste of space
→ Grouping - free list has n free blocks, so large no. of free blocks can be easily found.

→ Counting - rather than keeping a list of n free block we can keep track of first free block and then n no. of free contiguous blocks follow the first block.

→ Need to Protect:

- Pointer to free list
- Bit map
  - must be kept on disk
  - Copy in memory and disk may differ
  - Cannot allow for block to have a situation where bit = 1 in memory and bit = 0 on disk.
- Solution
  - Set bit = 1 in disk
  - Allocate block
  - Set bit = 1 in memory

6. list out the allocation methods of disk blocks. Explain any one of them

→ Three Common disk allocation methods

1. Contiguous Allocation
2. linked Allocation
3. Indexed Allocation

→ Contiguous Allocation

- Each file occupies a set of contiguous blocks on the disk
- linear ordering of disk address.
- Simple - only starting location and length are required.
- Random access
- Access is easy. Direct access to a block i of file starts at block b will be at block b+i. So here sequential and direct access is easy.
- Wasteful of space how to satisfy the n size block from free blocks, i.e, first fit or best fit strategies can be used.
- files cannot grow. When file is created total amount of space is allocated to if so further if we extend the file it may not possible.

5. Differentiate blw hash table and linear implementation for directory.

Hash table	Linear list
Uses a hash function to map file names to specific entires for fast searching	Stores directory entries in a sequential list one after another
Fast search - O(1) in the best case, as hashing provides direct access	Slow search - O(n) in the worst case, since it requires scanning each entry.
Uses techniques like chaining or open addressing to resolve collisions.	No collision issue, but requires linear search for every lookup.
Fast and efficient as hashing Provides direct indexing	Slower insertion especially if entiries need to be sorted or reorganized.