Syllabus

24 September 2022 09:46 AM

UNIT 1 **Fundamentals of Operating System:-**
OS services and Components, Multitasking , Multiprogramming, Multiprocessing Time Sharing, Buffering, Spooling, Distributed OS

UNIT 2 **Process and Thread Management**
Concept of process and threads ,Process states ,Process management ,Context switching ,Interaction between processes and OS ,Multithreading, Example OS : Linux

UNIT 3 **Memory Management**
Memory partitioning, Swapping, Paging ,Segmentation ,Virtual memory Overlays ,Demand paging ,Performance of Demand paging ,Virtual memory concepts ,Page replacement algorithms ,Allocation algorithms ,Example OS : Linux

UNIT 4 **I/O Systems**
Secondary-Storage Structure, Disk structure ,Disk scheduling ,Disk management ,Swap-space management ,Disk reliability ,Stable storage implementation ,Introduction to clock ,Clock hardware ,Clock software

UNIT 5 **File systems**

4

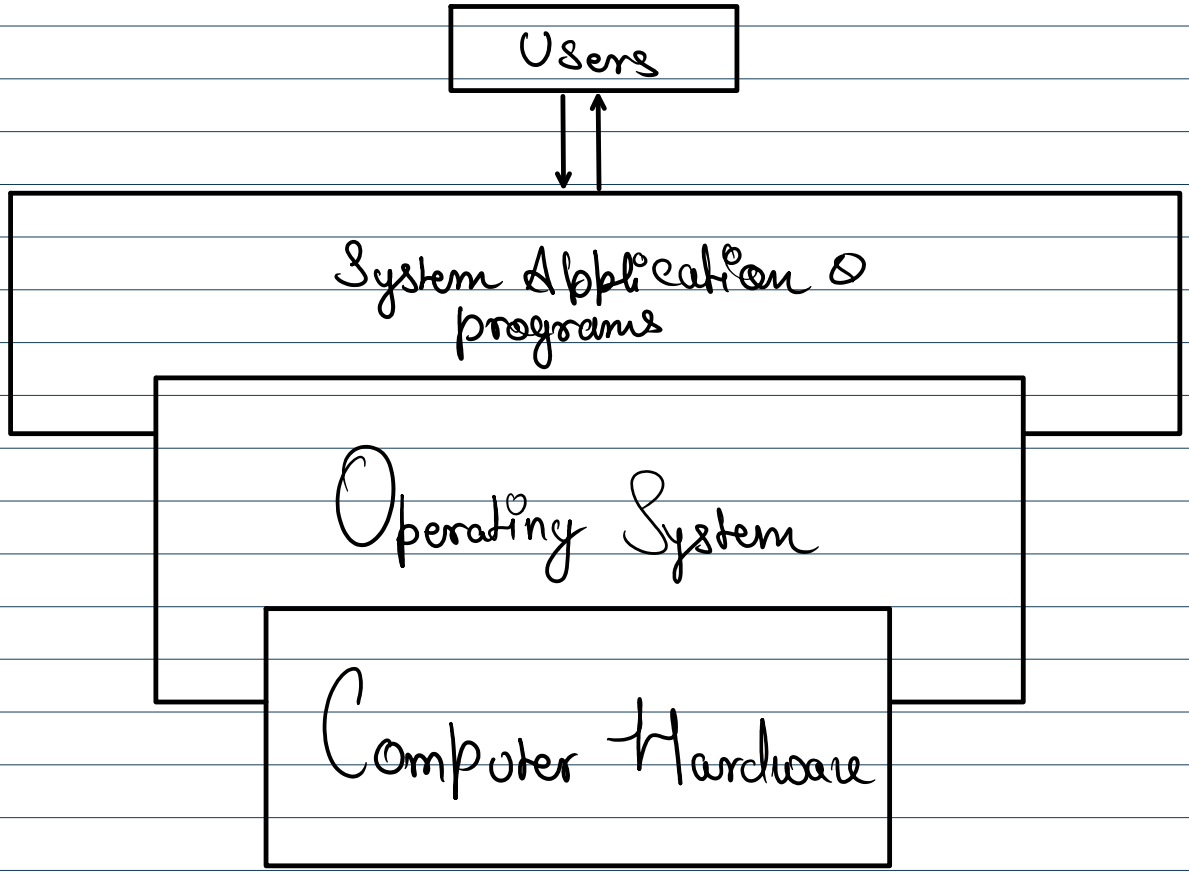
File concept ,File support ,Access methods ,Allocation methods ,Directory systems ,File protection ,Free space management
Example OS : Linux Case study : Android OS

Operating System

- A program that acts as an intermediary b/w a user of computer & computer hardware

  a user can be people, machine & other computers

- it utilizes computer hardware in efficient manner



Computer System Components

- Hardware : Provides basic computing resources (CPU, I/O, Memory)
- Operating System ; Controls & coordinates the use of hardware among various application

programs for various users

- Application programs ; These are the various programs which users compute / utilize
- Users ; People, machine, or Computers

A process is a program in Execution

It needs resources such as CPU, memory, Storage I/O device to accomplish its task

Process

26 September 2022 07:45 PM

An Operating System is responsible for

Process mgmt.

- Process creation & deletion
- Process suspension & resumption
- Provision of mechanism for
  - process synchronization
  - process communication

Memory mgmt.

- Keep track of which part of memory is used by & whom

OS Services

24 September 2022 06:25 PM

OS Services

Set of Services Useful to Users

- User Interface
- Program Execution
- I/O Operation
- File-System manipulation
- Communication
- Error detection

Set of Services for efficient System Operation

- Resource allocation
- Accounting
- Protection & Security

V/s Battle

24 September 2022 11:59 AM

Multiprogrammed : keeps multiple programs in main memory at the same time

- In this as soon as one job goes for an I/O task, this job is interrupted & another job is chosen from the job queue (waiting list)
- CPU starts executing new job while the previous job keeps doing its I/O task
- As soon as the prev. process's I/O job is done, immediately CPU is allocated to it

Thus no time is wasted

- Objective is to maximize CPU usage time

Multiprocessing : (Parallel Systems) A computer using more than one CPU at a time

- In this different processors execute different jobs
  - dual core processor can execute two processes simultaneously
- multiple physical processes needed

Parallel Systems

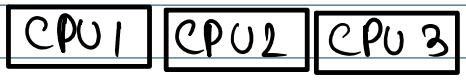
Symmetric multiprocessing

- Independent CPU (NO master-slave)

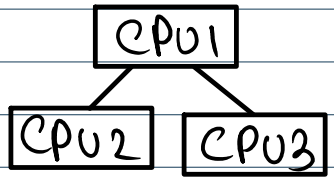
Asymmetric multiprocessing

- Dependent CPU (Master-slave relation)

CPU
(No master-slave)



(Master-Slave relation)



* (Time Sharing System)

Multitasking : Same as multiprogramming with Round-Robin Scheduling algo.

Execution of multiple task at the same time

In time sharing each process is assigned some specific quantum of time

CPU will share time slices with process accordingly

The content is that the switching occurs so fast that the user gets the illusion that multiple tasks are occuring simultaneously

Spooling And Buffering

24 September 2022 04:34 PM

Spooling: (Simultaneous peripheral operation Online)

- Similar to buffer it holds the job until the device is ready to accept the job
- Considers disk as a huge buffer

Buffering; The main memory has an area called buffer that is used to store data temporarily that is being transmitted either b/w devices or application

Buffer is applied to match the data stream between sender or reciver

The Basic difference b/w Spooling & buffering is that

- Spooling overlaps I/O of one job with execution of other job
- Buffering overlaps I/O of one job with execution of same job

Spooling

- Spooling is more efficient than buffering
- Huge disk space
- Can process data at remote places

Buffering

- Buffering is less efficient
- limited memory space
- doesn't support remote processing

Distributed OS are loosely coupled Systems

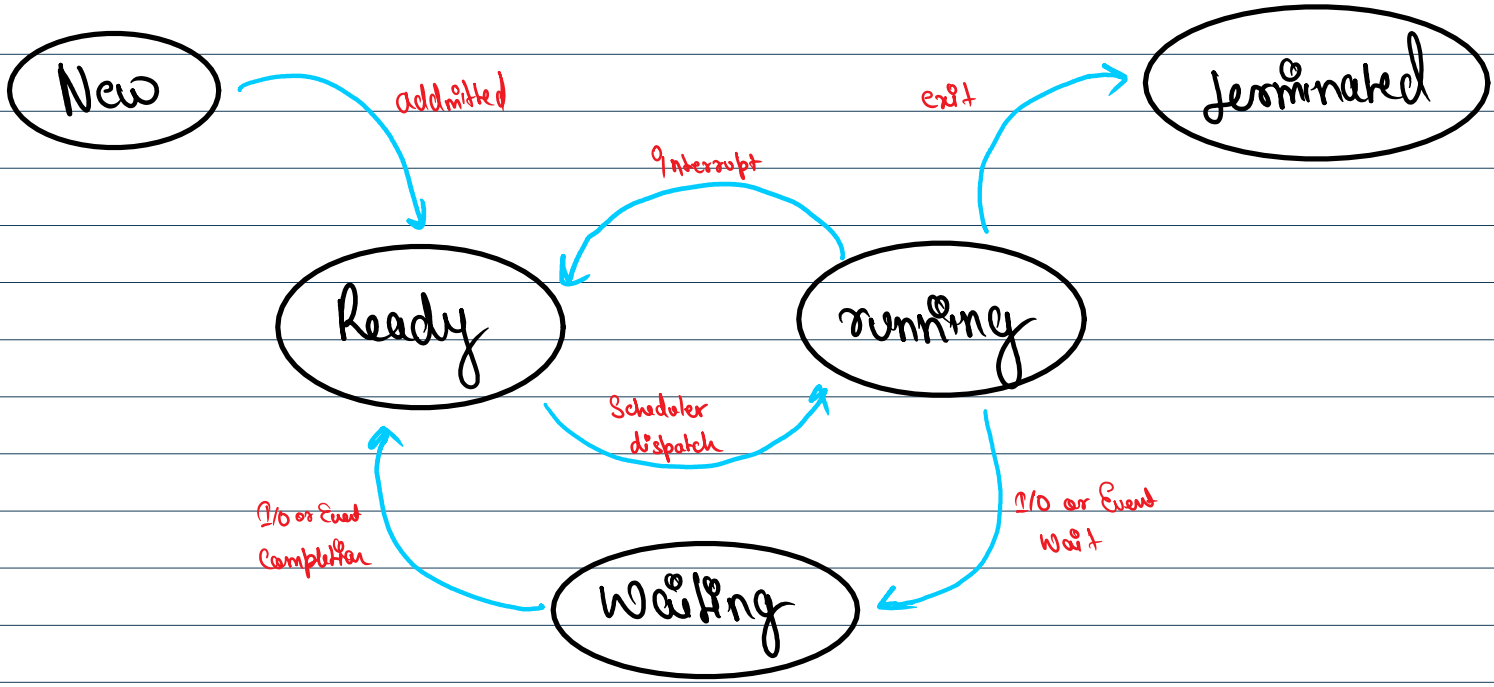
- In these system the processors differ in size & function
- It allows user to access files or Software which aren't available on users system

- In these failure of one system will not affect the other communication
- Single resources are being shared & Computation is highly fast & durable
- reduces load on host computer
- Easily Scalable network

- These type of Systems are expensive
- failure of main network will stop the entire communication
- Underlying Software is highly complex

Process Control Block

26 September 2022 09:10 PM



!Process Scheduling

26 September 2022 09:17 PM

Traps VS Interrupt

21 December 2022 07:51 AM

Trap

Comes from user program

(Synchronous)

↳ Software

{ program generates it }

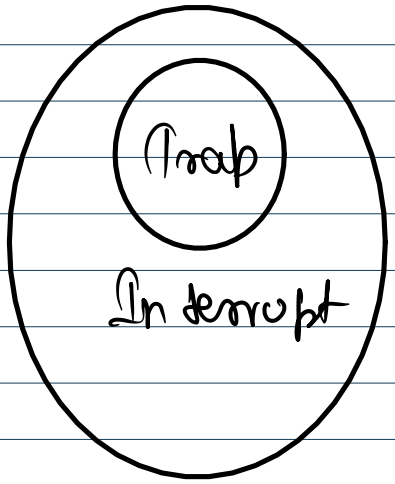
Interrupt

Comes from I/O → CPU

(Asynchronous)

↳ Software & hardware

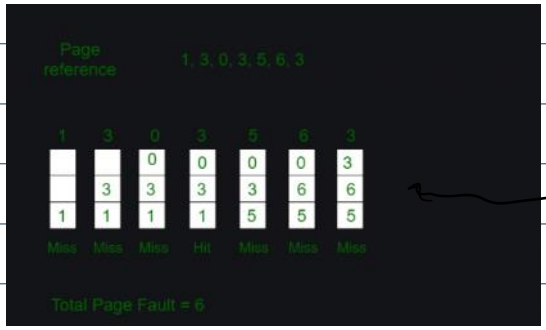
{ Hardware generates it }



!Memory Partitioning

21 December 2022 05:42 AM

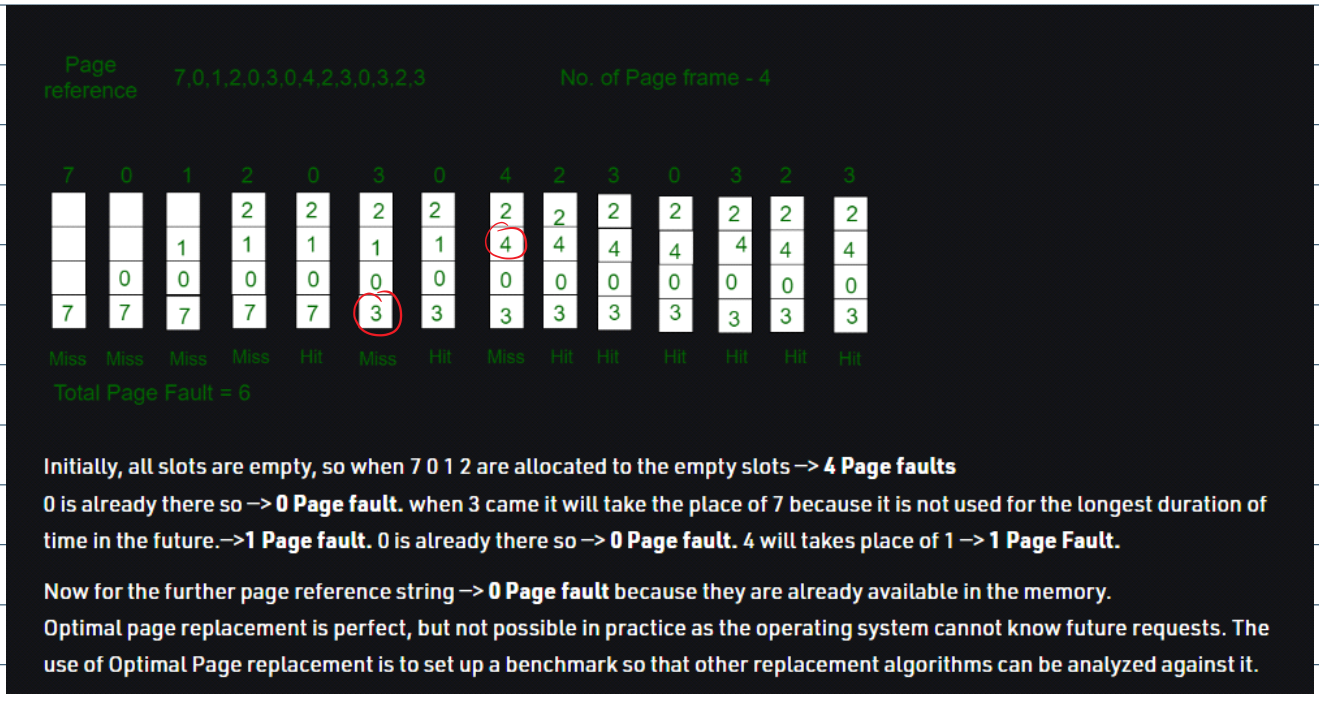
① FCFS :



frames

Belady's anamaly : page faults ↗ frames

② Optimal - page Replacement

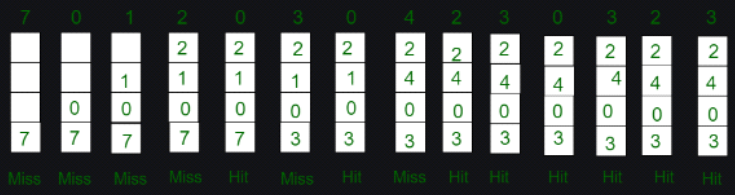


Initially, all slots are empty, so when 7 0 1 2 are allocated to the empty slots –> **4 Page faults**
0 is already there so –> **0 Page fault.** when 3 came it will take the place of 7 because it is not used for the longest duration of time in the future.–>**1 Page fault.** 0 is already there so –> **0 Page fault.** 4 will takes place of 1 –> **1 Page Fault.**

Now for the further page reference string –> **0 Page fault** because they are already available in the memory.
Optimal page replacement is perfect, but not possible in practice as the operating system cannot know future requests. The use of Optimal Page replacement is to set up a benchmark so that other replacement algorithms can be analyzed against it.

③ LRU

Page reference 7,0,1,2,0,3,0,4,2,3,0,3,2,3 No. of Page frame - 4



Total Page Fault = 6

Here LRU has same number of page fault as optimal but it may differ according to question.

Initially, all slots are empty, so when 7 0 1 2 are allocated to the empty slots –> **4 Page faults**

0 is already their so –> **0 Page fault.** when 3 came it will take the place of 7 because it is least recently used –>**1 Page fault**

0 is already in memory so –> **0 Page fault.**

4 will takes place of 1 –> **1 Page Fault**

Now for the further page reference string –> **0 Page fault** because they are already available in the memory.

→ No Belady's anamaly

&&Demand Paging || Page Fault || Thrashing

21 December 2022 08:00 AM

What is Demand Paging in OS (Operating System)?

According to the concept of Virtual Memory, in order to execute some process, only a part of the process needs to be present in the main memory which means that only a few pages will only be present in the main memory at any time.

However, deciding, which pages need to be kept in the main memory and which need to be kept in the secondary memory, is going to be difficult because we cannot say in advance that a process will require a particular page at particular time.

Therefore, to overcome this problem, there is a concept called Demand Paging is introduced. It suggests keeping all pages of the frames in the secondary memory until they are required. In other words, it says that do not load any page in the main memory until it is required.

Whenever any page is referred for the first time in the main memory, then that page will be found in the secondary memory.

After that, it may or may not be present in the main memory depending upon the page replacement algorithm which will be covered later in this tutorial.

What is a Page Fault?

If the referred page is not present in the main memory then there will be a miss and the concept is called Page miss or page fault.

The CPU has to access the missed page from the secondary memory. If the number of page fault is very high then the effective access time of the system will become very high.

What is Thrashing?

If the number of page faults is equal to the number of referred pages or the number of page faults are so high so that the CPU remains busy in just reading the pages from the secondary memory then the effective access time will be the time taken by the CPU to read one word from the secondary memory and it will be so high. The concept is called thrashing.

If the page fault rate is PF %, the time taken in getting a page from the secondary memory and again restarting is S (service time) and the memory access time is ma then the effective access time can be given as;

&&Segmentation

21 December 2022 08:03 AM

Segmentation in OS (Operating System)

← Prev Next →

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.

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.

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.

Why Segmentation is required?

Till now, we were using Paging as our main memory management technique. Paging is more close to the Operating system rather than the User. It divides all the processes into the form of pages regardless of the fact that a process can have some relative parts of functions which need to be loaded in the same page.

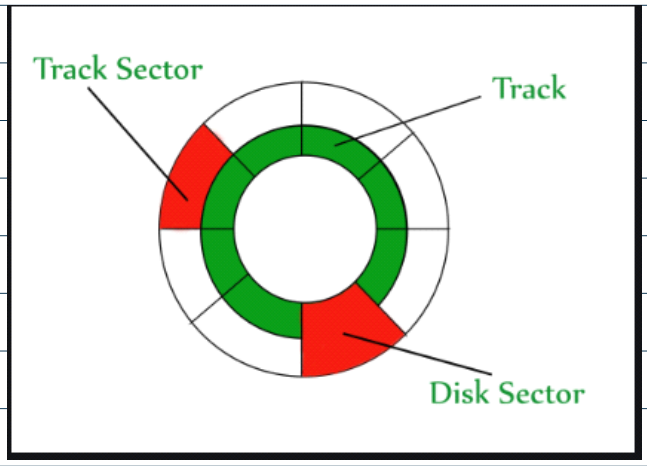
Paging || !Swapping

21 December 2022 08:03 AM

It is a mechanism used to retrieve process from secondary storage into main memory in form of pages

divide process → pages (n)

divide memory → frames (uniform) (same size)



disks are devided into tracks

Size (outer track) > Size (inner track)

but contain Same no. of sectors

& Same storage capacity

- Some space in sector is arranged for formatting

actual capacity (sector) < presented capacity (sector)

R/W head moves over the rotating disk

Seek time: The time taken to locate the disk arm to specified track where data is to be r/w

Rotational Latency: Time taken by desired sector of disk to rotate into position for r/w access head

Transfer Time: Time taken to tranfer the data

Time

!Disk Structure

20 December 2022 11:39 PM

IDK

Seek time: The time taken to locate the disk arm to specified track where data is to be r/w

Rotational latency: Time taken by desired sector of disk to rotate into position for r/w access head

Transfer time: Time taken to transfer the data

Sum of all three is **Disk access time**

Disk response time is D.A.T + wait time (queuing)

Disk Scheduling Algorithm

1. **FCFS**
   - Same as the name suggests
   - fair to every request & no indefinite postponement
   - unoptimized, slow

2. **SSTF** (Shortest Seek time first)
   Seeks closest location to current head pos^n
   - lower Avg response time & high throughput
   - high overhead & causes starvation

3. **SCAN** moves in one direction until the end of the disk then reverses the direction

(∩ Elevator)

requests @ midrange are serviced more & rest have to wait

∩ high throughput ∩ low variance of response time
∩ long long waiting time for request

(1) CSCAN : It is same as SCAN but after reaching end on one direction it goes to other then start service

(∩ Circular Scan) ∩ provides uniform wait time as compared to SCAN

(5) LOOK : Similar to SCAN except if goes upto last entry & not the end. then reverses the direction

∩ prevents extra delay due to unnecessary traversal

(6) CLOOK : Basically {CSCAN} ∪ {LOOK}

(7) RSS (Random Scan Scheduling)

(8) LIFO ∩ xD

& 2 others not in syllabus

Disk Management

21 December 2022 05:34 AM

Disk Management in OS

1. **Disk format**
   - divides the disk into sector before storing data
   - divides into different cylinder grp @ as logical data
2. **Booting from disk**
   - creating bootloader, ----
   - sets up 1st partition as boot block
3. **Bad Block Recovery**
   - maintains list of bad blocks
   - ↓
   - blocks with error due to external or internal factors

!Clock

21 December 2022 07:30 AM

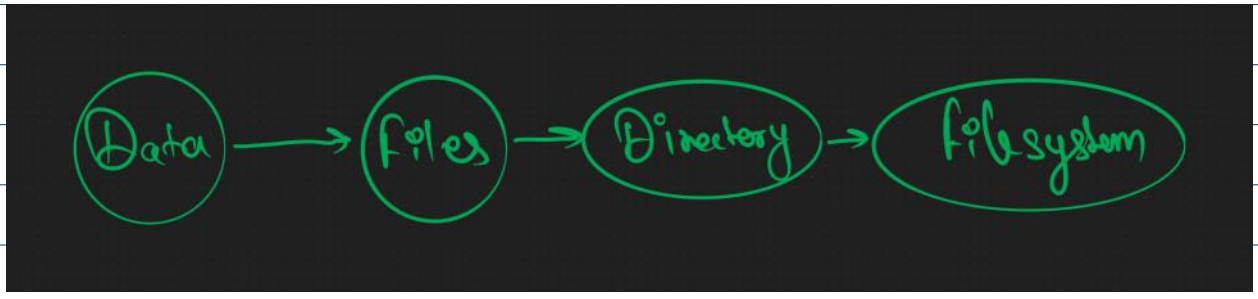
File Concept

20 December 2022 09:25 AM

A file is a named collection of related information that is recorded on secondary storage

name of file is devided into two parts

- Name
- extension



File attributes

(1) Name : Name by which System recognizes the file

(2) Identifier : Extension of the file which identifies the type of the file

↓

Type

(3) Location : In file system there are several locations where files can be stored each file has its own file locations

(4) Size : It is the number of bytes aquired by the file in memory

(5) Protection : It is the level of authority / Set of permissions for different user group ex Admin

(6) Time & Date : It is the time stamp of when the file was created & modified.

File operations

1. Create
2. Open
3. Write
4. Read
5. Seek
6. Delete
7. Truncate
8. Close
9. Append
10. Rename

(1) Create operation : It is creation of file on which a new file of a particular type is allocated space.
@ apropriate directory / loc^n

(2) Open operation : After file is created, it must be opened before performing file processing

When user wants to open a file, it tells the operating system to invoke open system call & pass file name to file system.

(3) Write operation : It is used to write information onto file,

With system call write is issued the specifies the name

of file & length of data that has to be written after that file pointer is repositioned after last written byte

④ Read operation : Reads the content of a file with the help of Read pointer of OS which positions upto which data has been read.

⑤ Seek operation : It repositions the file pointers from current position to specific place in file as per user's requirement

⑥ Delete operation : It deletes all the data of the file & frees the disk space occupied by it

In order to delete specified file in directory, after locating directry entry, associated file space & directory entry is released

⑦ Truncate operation : It is deleting files except for its attributes

Only info stored inside is deleted / replaced

⑧ Close operation : After processing is complete files are closed to make all changes permanent & release occupied resource

(9) Append operation : Adds data @ end of file

(10) Rename operation : renames existing file.

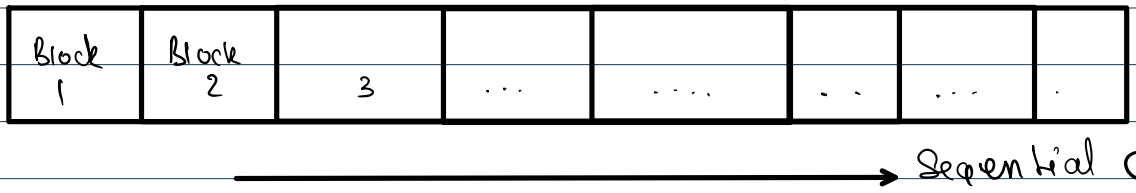
File Access method

① Sequential access ② Direct access ③ Indexed access

① **Sequential Access :** In this the OS reads the file word by word

- A pointer is maintained which initially points towards base address of file
- If the user wants to read first word, then pointer provides that word & increases its value by one word, this continues till the end of file.

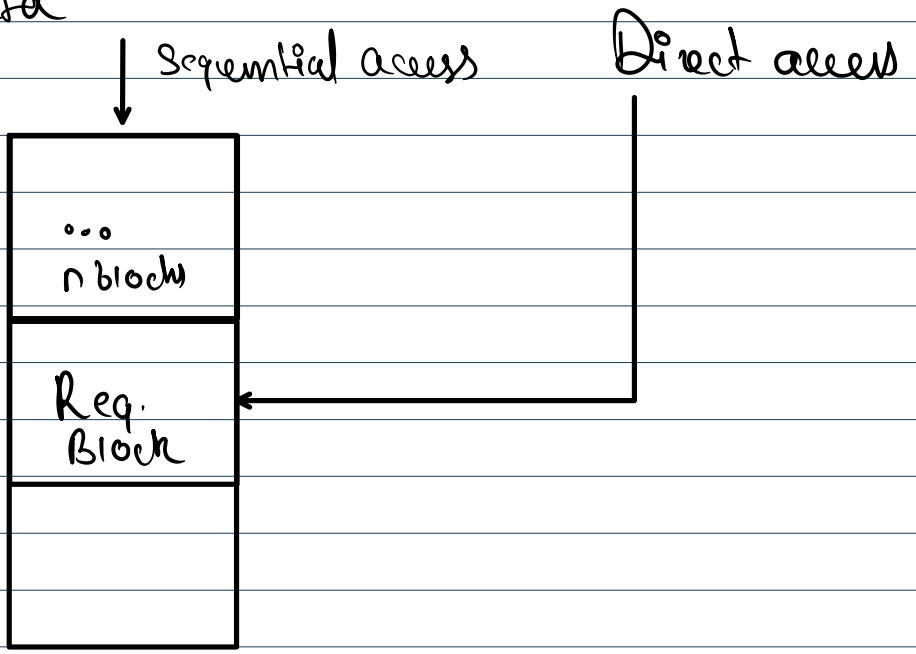
~ Most of the file access method are sequential some files such as audio, video, text are needed to be accessed sequentially



② **Direct Access :** Mostly required in database System

Unlike Sequential access which can take a lot of time to get to a piece of data inside the file,

Direct access goes directly to the required block of data



Suppose the data required is on 10th block then Sequential access will take time, however in direct access it will go directly to that block through some complex bedition.

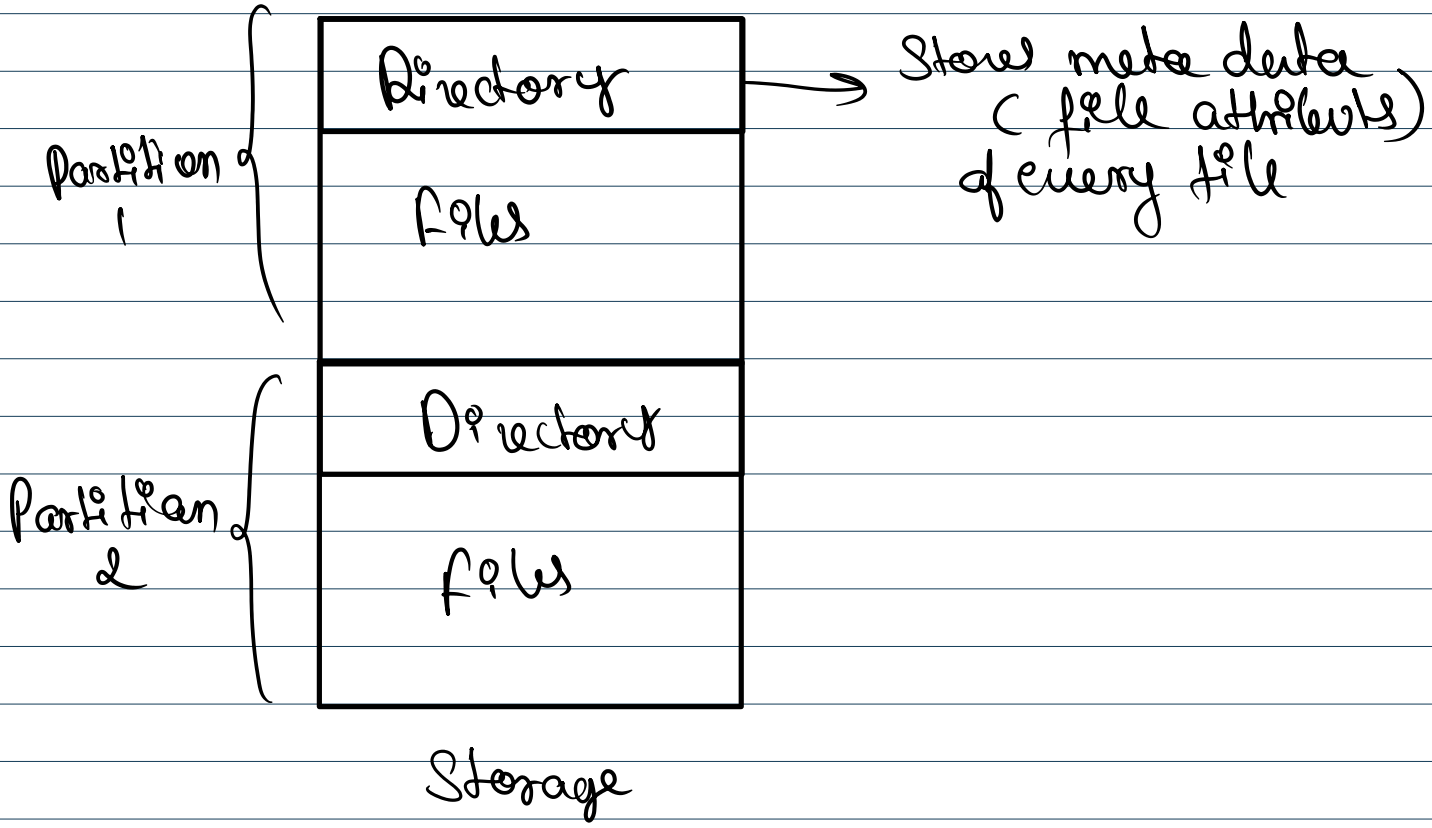
③ Indexed Access : In this the files are stored & assigned index to a group of certain records

A particular record can be accessed by its index

It optimizes searching in large files, quick but needs some extra space for indexes

Directory: It is defined as the listing of the related files on a disk

→ It stores some or entire file attributes



Every directory supports a number of common operations

1. File creation
2. Search for file
3. File deletion
4. Rename file
5. Traverse file
6. Listing of files

?Types of Directory/Directory Strucutres

20 December 2022 01:59 PM

Types of directory
- Single level directory
- Two level directory
- Tree Structured Directory
- Acrylic graph Structured directory

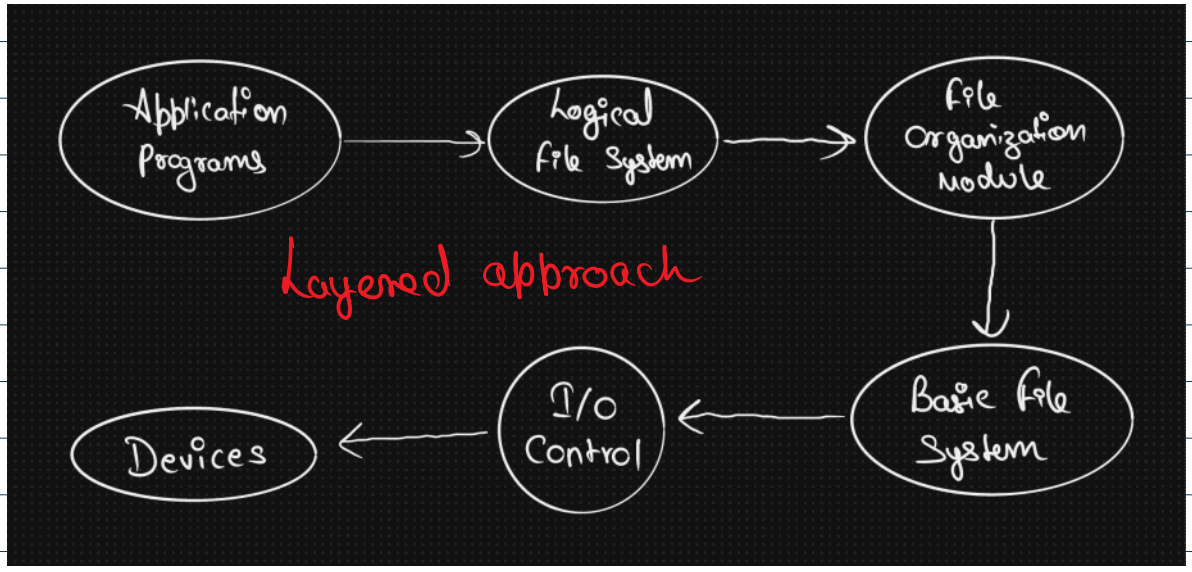
File systems are the part of OS which is responsible for file management

- It is responsible for storing and accessing data to user's programs

It is responsible for :

- → File Structure
- → Recovering free Space
- → Disk space assignment to files
- → Tracking data locations

- It provides an efficient way/access to disk
- Layering approach is used.



When an application/program asks for a file

↓ request directed to Logical file System

logical file system

The logical file system contains the metadata of file & directory structure

If the application doesn't have required permission, this layer will throw an error

generally files are divided into various logical blocks whereas the hard disk is divided into various tracks & sectors.
The logical blocks are mapped to physical blocks by file organization module

↓

After file organization module has decided which physical block the application program needs
Then it passes that information to basic file system & basic file system issues command to I/O control in order to fetch those block

↓

I/O controls contains codes (device drivers) by which it accesses hard disk
(It is responsible for handling Interrupts)

?File Control Block (in Memory Data Structure)

20 December 2022 04:55 PM

On Disk Data Structures

← Prev Next →

There are various on disk data structures that are used to implement a file system. This structure may vary depending upon the operating system.

1. **Boot Control Block**

   Boot Control Block contains all the information which is needed to boot an operating system from that volume. It is called boot block in UNIX file system. In NTFS, it is called the partition boot sector.

2. **Volume Control Block**

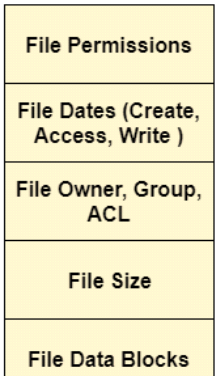
   Volume control block all the information regarding that volume such as number of blocks, size of each block, partition table, pointers to free blocks and free FCB blocks. In UNIX file system, it is known as super block. In NTFS, this information is stored inside master file table.

3. **Directory Structure (per file system)**

   A directory structure (per file system) contains file names and pointers to corresponding FCBs. In UNIX, it includes inode numbers associated to file names.

4. **File Control Block**

   File Control block contains all the details about the file such as ownership details, permission details, file size,etc. In UFS, this detail is stored in inode. In NTFS, this information is stored inside master file table as a relational database structure. A typical file control block is shown in the image below.



?File Allocation Methods

21 December 2022 05:06 AM

File allocation methods

1. Contigious Allocation
2. Linked Allocation
3. Indexed Allocation