File Systems in Operating System

A file is a collection of related information that is recorded on secondary storage. Or file is a collection of logically related entities. From user's perspective a file is the smallest allotment of logical secondary storage.

Attributes	Types	Operations
Name	Doc	Create
Type	Exe	Open
Size	Jpg	Read
Creation Data	Xis	Write
Author	C	Append
Last Modified	Java	Truncate
protection	class	Delete
		Close

File type	Usual extension	Function
Executable	exe, com, bin	Read to run machine language program
Object	obj, o	Compiled, machine language not linked
Source Code	C, java, pas, asm, a	Source code in various languages
Batch	bat, sh	Commands to the command interpreter
Text	txt, doc	Textual data, documents
Word Processor	wp, tex, rrf, doc	Various word processor formats
Archive	arc, zip, tar	Related files grouped into one compressed file
Multimedia	mpeg, mov, rm	For containing audio/video information

FILE DIRECTORIES:
Collection of files is a file directory. The directory contains information about the files, including attributes, location and ownership. Much of this information, especially that is concerned with storage, is managed by the operating system. The directory is itself a file, accessible by various file management routines.
Information contained in a device directory are:

- Name
- Type
- Address
- Current length

- Maximum length
- Date last accessed
- Date last updated
- Owner id
- Protection information

Operation performed on directory are:

- Search for a file
- Create a file
- Delete a file
- List a directory
- Rename a file
- Traverse the file system

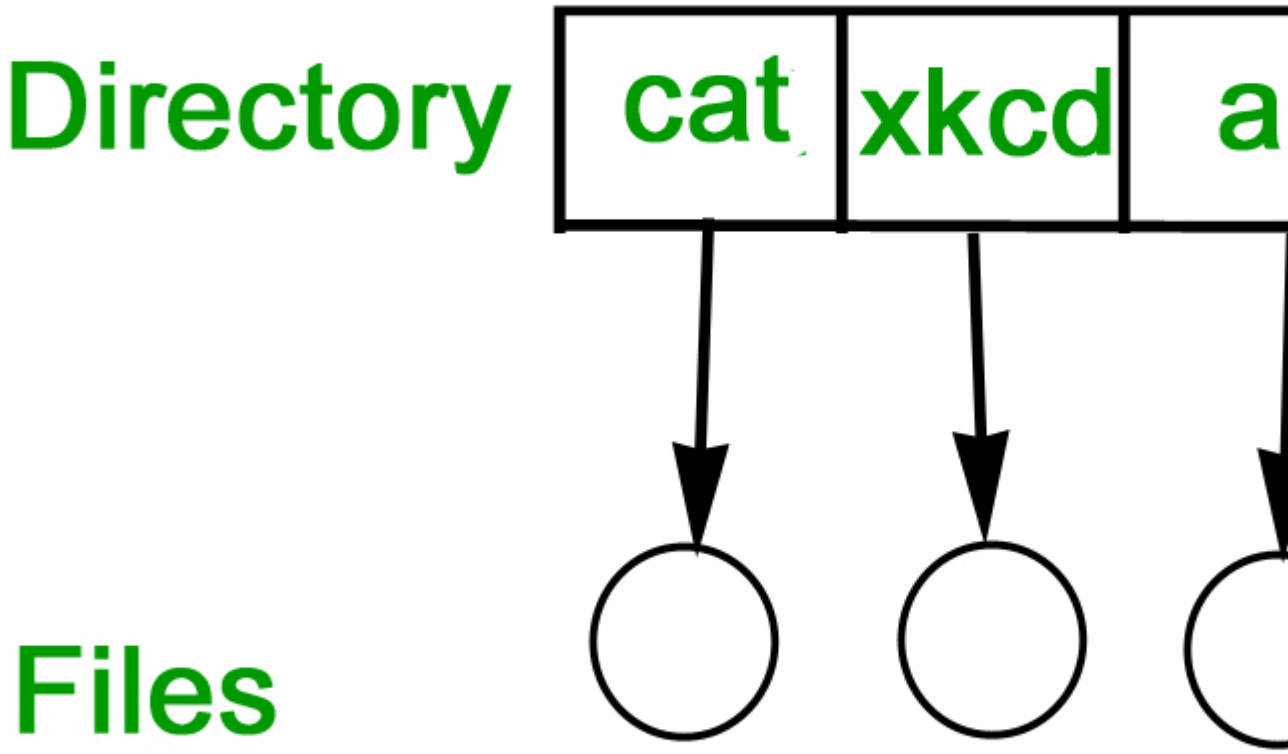
Advantages of maintaining directories are:

- **Efficiency:** A file can be located more quickly.
- **Naming:** It becomes convenient for users as two users can have same name for different files or may have different name for same file.
- **Grouping:** Logical grouping of files can be done by properties e.g. all java programs, all games etc.

SINGLE-LEVEL DIRECTORY

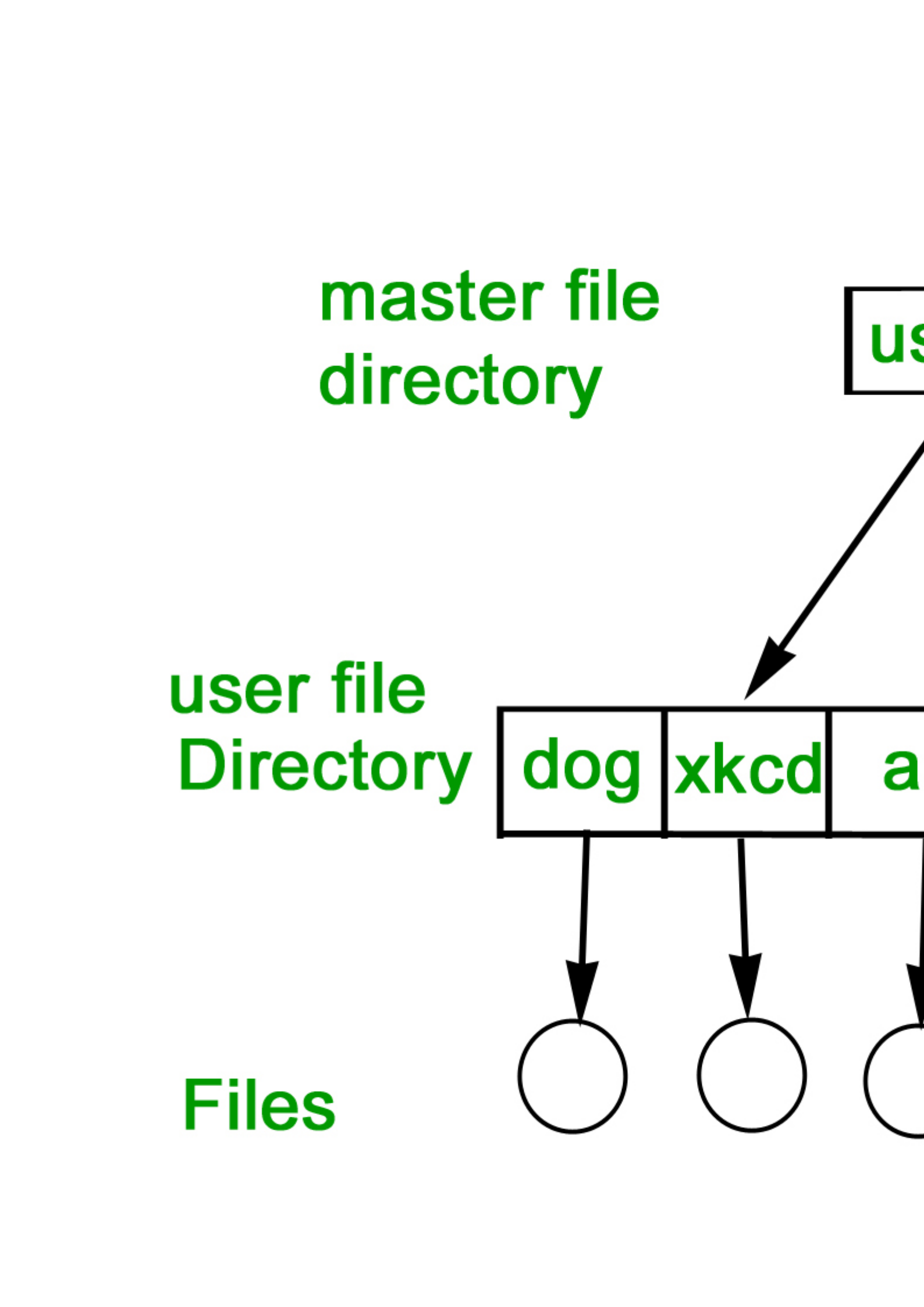
In this a single directory is maintained for all the users.

- **Naming problem:** Users cannot have same name for two files.
- **Grouping problem:** Users cannot group files according to their need.

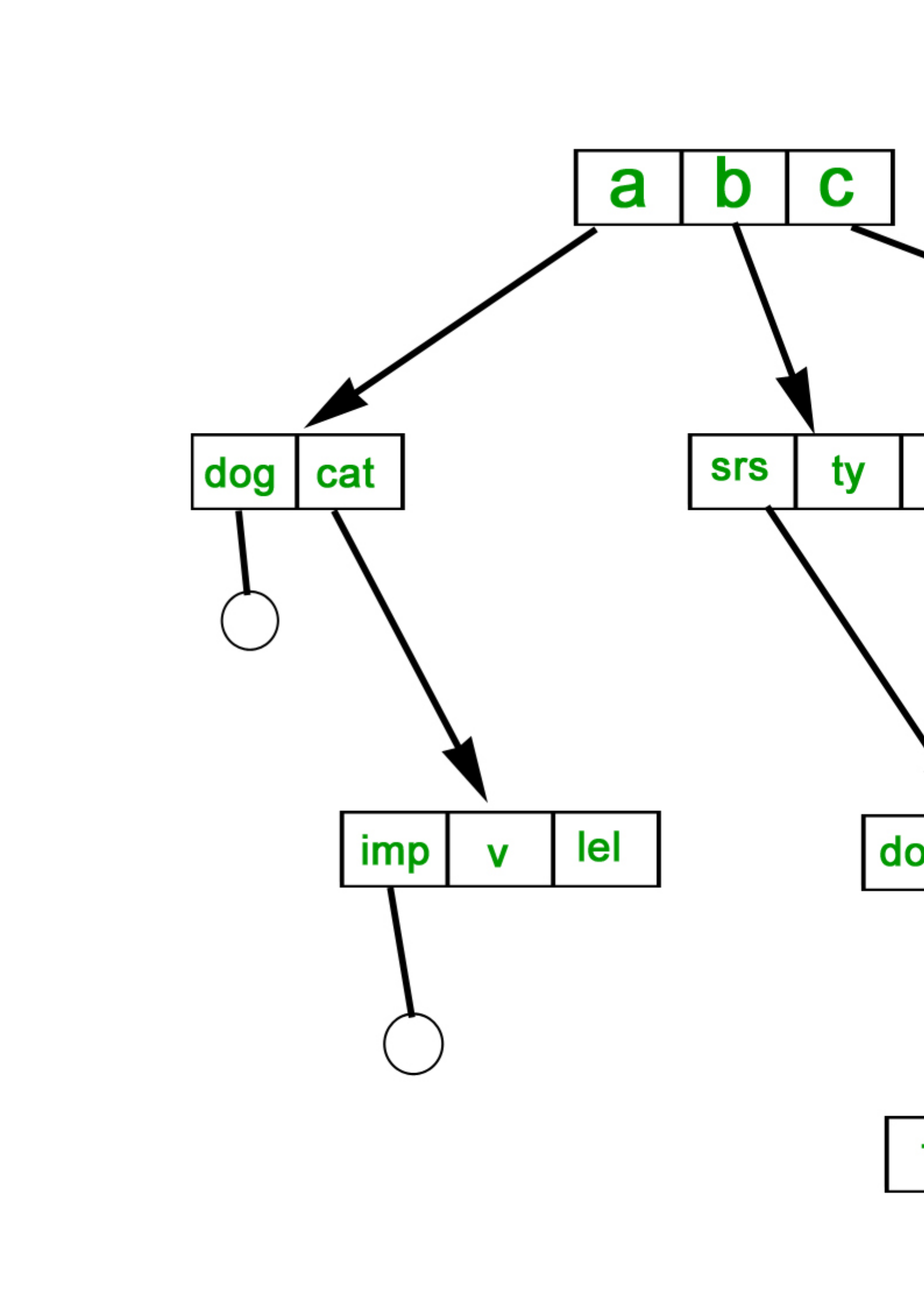


TWO-LEVEL DIRECTORY
In this separate directories for each user is maintained.

- Path name:Due to two levels there is a path name for every file to locate that file.
- Now,we can have same file name for different user.
- Searching is efficient in this method.

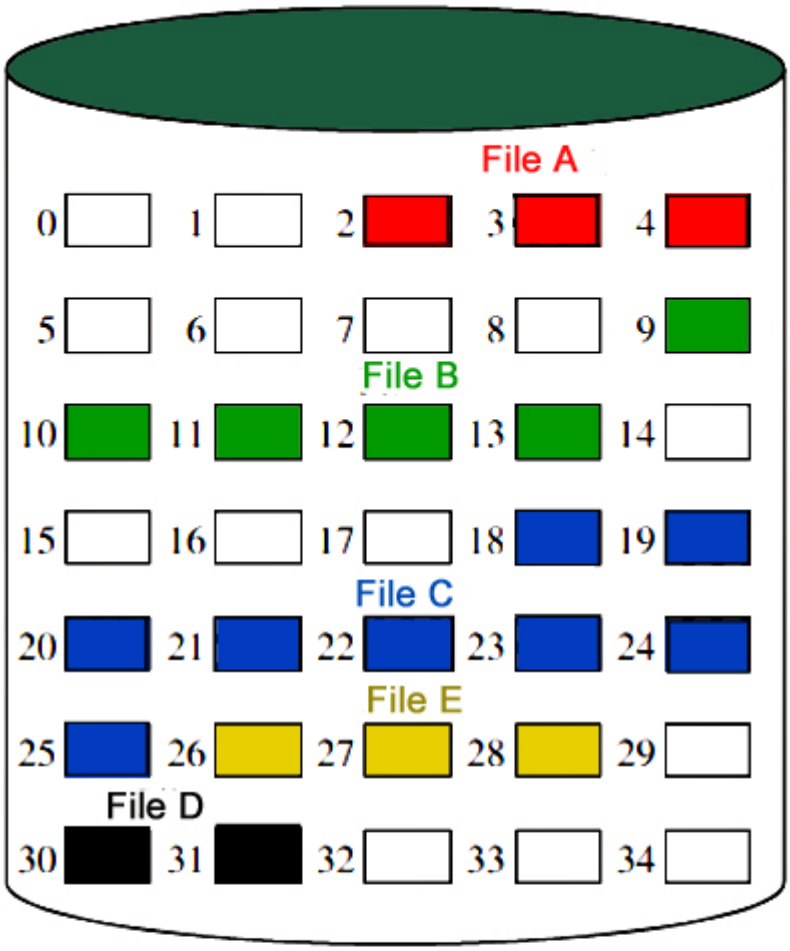

master file
directory
us
user file
Directory
dog
xkcd
a
Files

TREE-STRUCTURED DIRECTORY :
Directory is maintained in the form of a tree. Searching is efficient and also there is grouping capability. We have absolute or relative path name for a file.


a b c
dog cat
srs ty
imp v lel
do

FILE ALLOCATION METHODS

1. Continuous Allocation: A single continuous set of blocks is allocated to a file at the time of file creation. Thus, this is a pre-allocation strategy, using variable size portions. The file allocation table needs just a single entry for each file, showing the starting block and the length of the file. This method is best from the point of view of the individual sequential file. Multiple blocks can be read in at a time to improve I/O performance for sequential processing. It is also easy to retrieve a single block. For example, if a file starts at block b, and the ith block of the file is wanted, its location on secondary storage is simply b+i-1.



File allocation table

File name	Start block	Length
File A	2	3
File B	9	5
File C	18	8
File D	30	2
File E	26	3

Disadvantage

- External fragmentation will occur, making it difficult to find contiguous blocks of space of sufficient length. Compaction algorithm will be necessary to free up additional space on disk.
- Also, with pre-allocation, it is necessary to declare the size of the file at the time of creation.

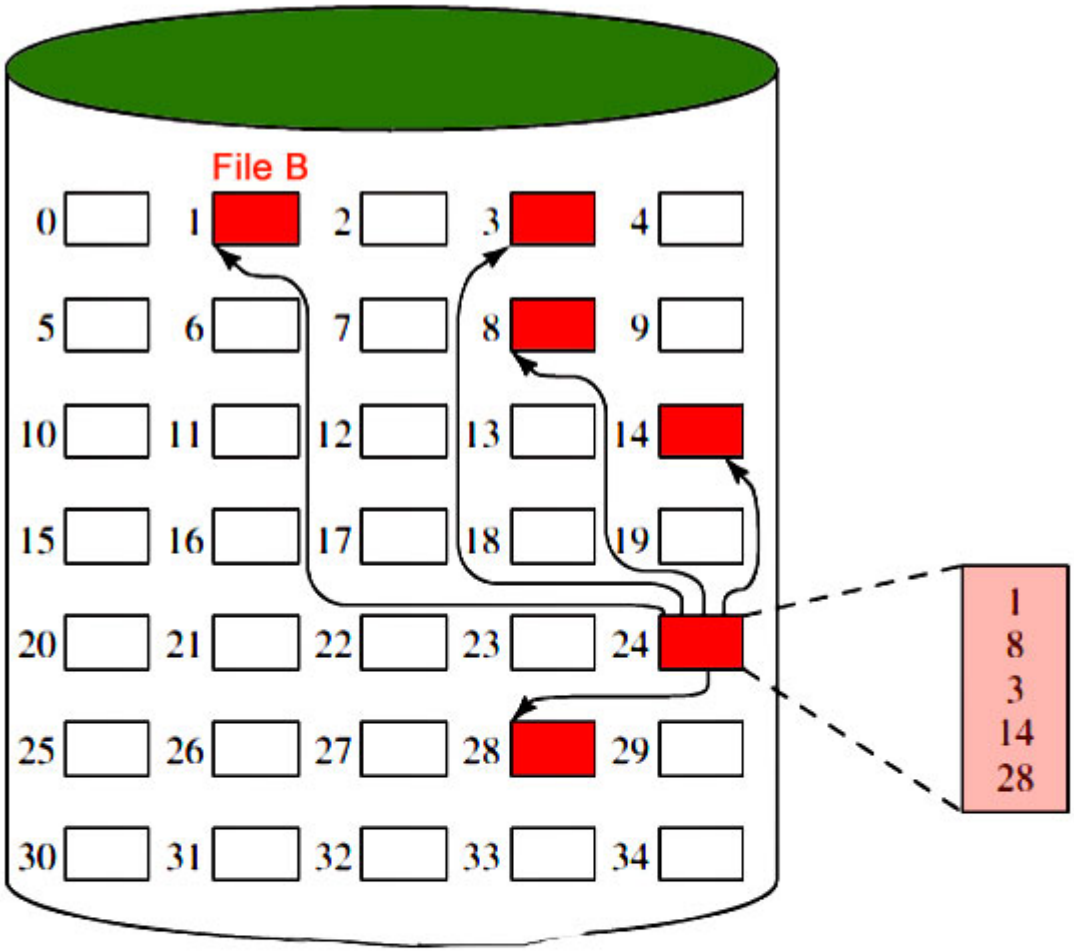
2. Linked Allocation(Non-contiguous allocation) : Allocation is on an individual block basis. Each block contains a pointer to the next block in the chain. Again the file table needs just a single entry for each file, showing the starting block and the length of the file. Although pre-allocation is possible, it is more common simply to allocate blocks as needed. Any free block can be added to the chain. The blocks need not be continuous. Increase in file size is always possible if free disk block is available. There is no external fragmentation because only one block at a time is needed but there can be internal fragmentation but it exists only in the last disk block of file.

Disadvantage:

- Internal fragmentation exists in last disk block of file.
- There is an overhead of maintaining the pointer in every disk block.
- If the pointer of any disk block is lost, the file will be truncated.
- It supports only the sequential access of files.

3. Indexed Allocation:

It addresses many of the problems of contiguous and chained allocation. In this case, the file allocation table contains a separate one-level index for each file: The index has one entry for each block allocated to the file. Allocation may be on the basis of fixed-size blocks or variable-sized blocks. Allocation by blocks eliminates external fragmentation, whereas allocation by variable-size blocks improves locality. This allocation technique supports both sequential and direct access to the file and thus is the most popular form of file allocation.



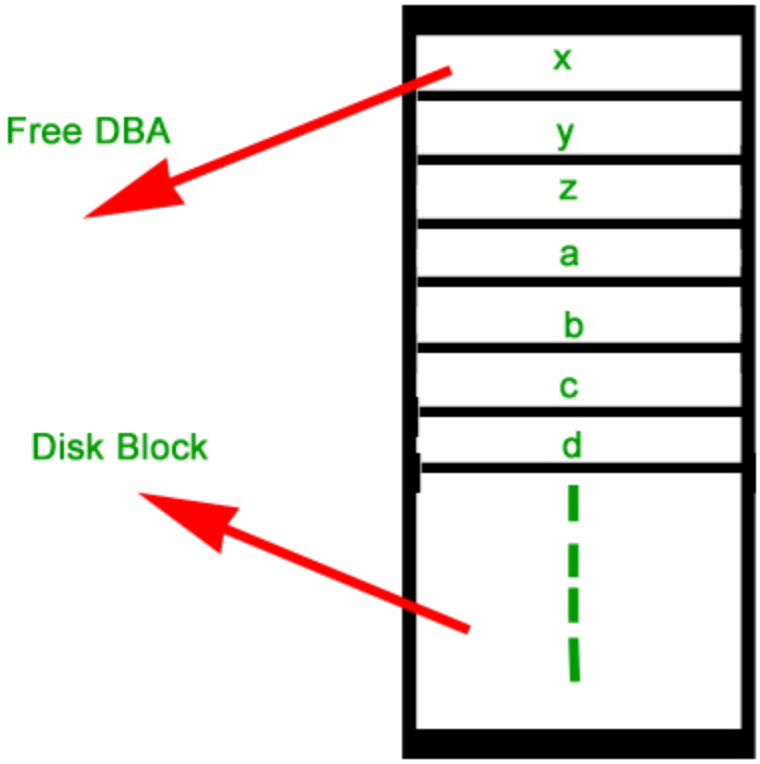
File allocation table

File name	Index block
•••	•••
File B	24
•••	•••

Disk Free Space Management

Just as the space that is allocated to files must be managed ,so the space that is not currently allocated to any file must be managed. To perform any of the file allocation techniques,it is necessary to know what blocks on the disk are available. Thus we need a disk allocation table in addition to a file allocation table.The following are the approaches used for free space management.

1. **Bit Tables** : This method uses a vector containing one bit for each block on the disk. Each entry for a 0 corresponds to a free block and each 1 corresponds to a block in use.
   For example: 00011010111100110001
   In this vector every bit correspond to a particular block and 0 implies that, that particular block is free and 1 implies that the block is already occupied. A bit table has the advantage that it is relatively easy to find one or a contiguous group of free blocks. Thus, a bit table works well with any of the file allocation methods. Another advantage is that it is as small as possible.
2. **Free Block List** : In this method, each block is assigned a number sequentially and the list of the numbers of all free blocks is maintained in a reserved block of the disk.



DISK MANAGEMENT

Disk Scheduling Algorithms

- Difficulty Level : Easy
- Last Updated : 28 Jun, 2021

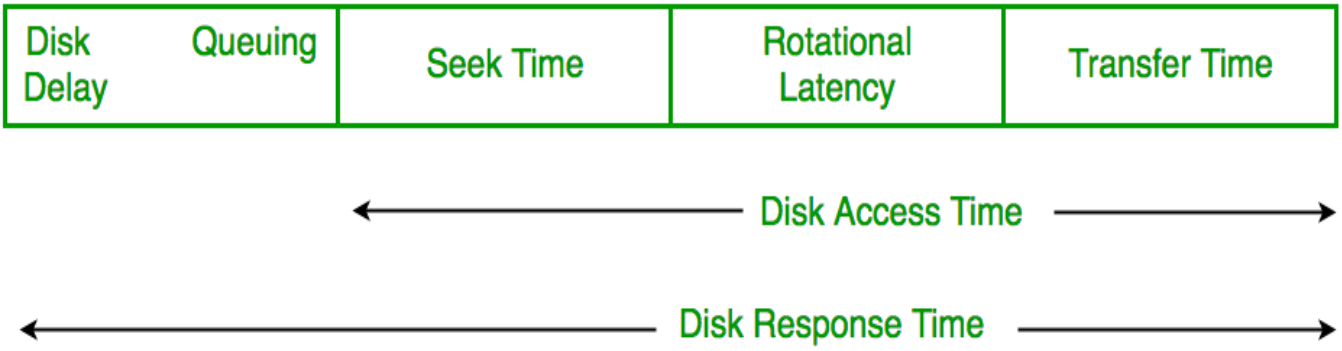
Disk scheduling is done by operating systems to schedule I/O requests arriving for the disk. Disk scheduling is also known as I/O scheduling.
Disk scheduling is important because:

- Multiple I/O requests may arrive by different processes and only one I/O request can be served at a time by the disk controller. Thus other I/O requests need to wait in the waiting queue and need to be scheduled.
- Two or more request may be far from each other so can result in greater disk arm movement.
- Hard drives are one of the slowest parts of the computer system and thus need to be accessed in an efficient manner.

There are many Disk Scheduling Algorithms but before discussing them let's have a quick look at some of the important terms:

- **Seek Time:**Seek time is the time taken to locate the disk arm to a specified track where the data is to be read or write. So the disk scheduling algorithm that gives minimum average seek time is better.
- **Rotational Latency:** Rotational Latency is the time taken by the desired sector of disk to rotate into a position so that it can access the read/write heads. So the disk scheduling algorithm that gives minimum rotational latency is better.
- **Transfer Time:** Transfer time is the time to transfer the data. It depends on the rotating speed of the disk and number of bytes to be transferred.
- **Disk Access Time:** Disk Access Time is:

```
Disk Access Time = Seek Time +
                   Rotational Latency +
                   Transfer Time
```

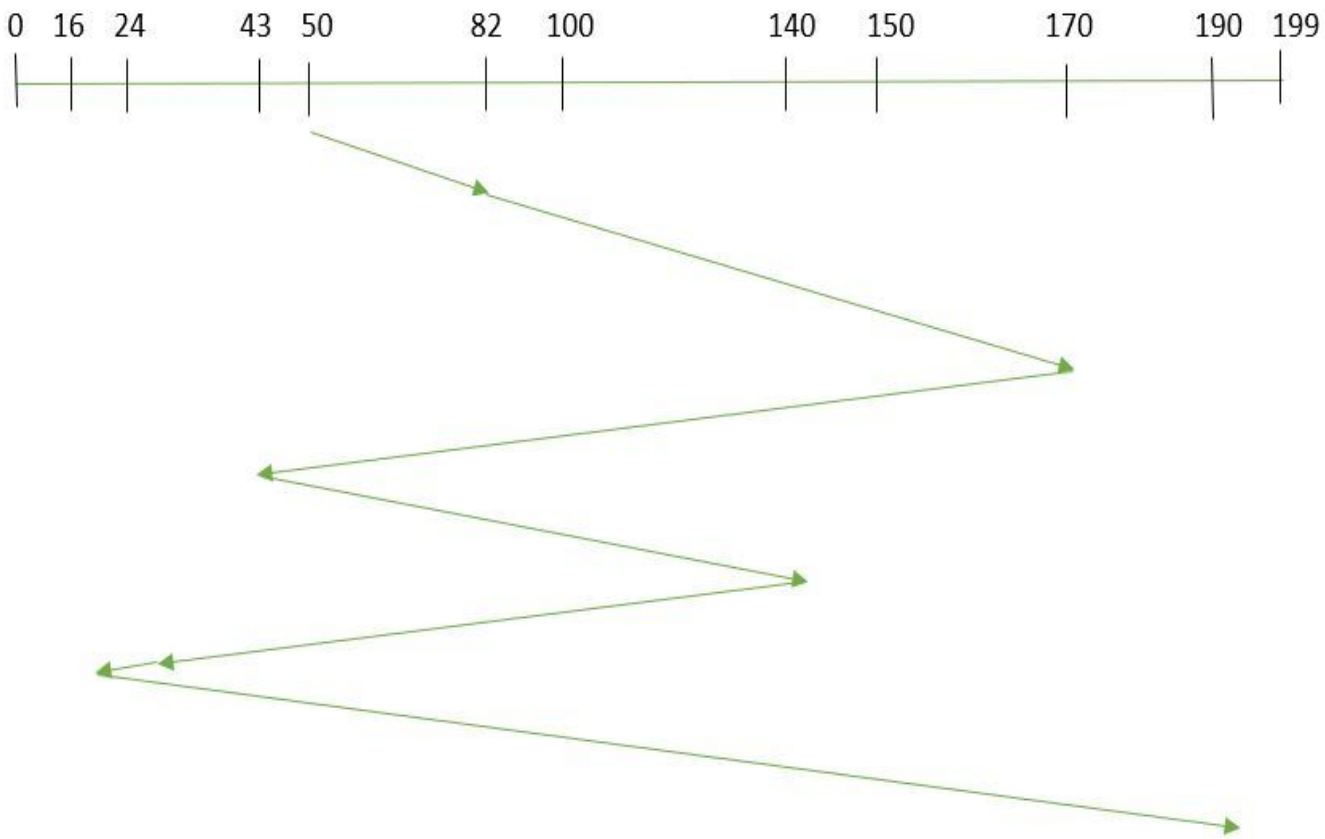


- **Disk Response Time:** Response Time is the average of time spent by a request waiting to perform its I/O operation. *Average Response time* is the response time of the all requests. *Variance Response Time* is measure of how individual request are serviced with respect to average response time. So the disk scheduling algorithm that gives minimum variance response time is better.

Disk Scheduling Algorithms

1. **FCFS:** FCFS is the simplest of all the Disk Scheduling Algorithms. In FCFS, the requests are addressed in the order they arrive in the disk queue.Let us understand this with the help of an example.

   Example:
   Suppose the order of request is- (82,170,43,140,24,16,190)
   And current position of Read/Write head is : 50



So, total seek time:
=(82-50)+(170-82)+(170-43)+(140-43)+(140-24)+(24-16)+(190-16)
=642

Advantages:

- Every request gets a fair chance
- No indefinite postponement

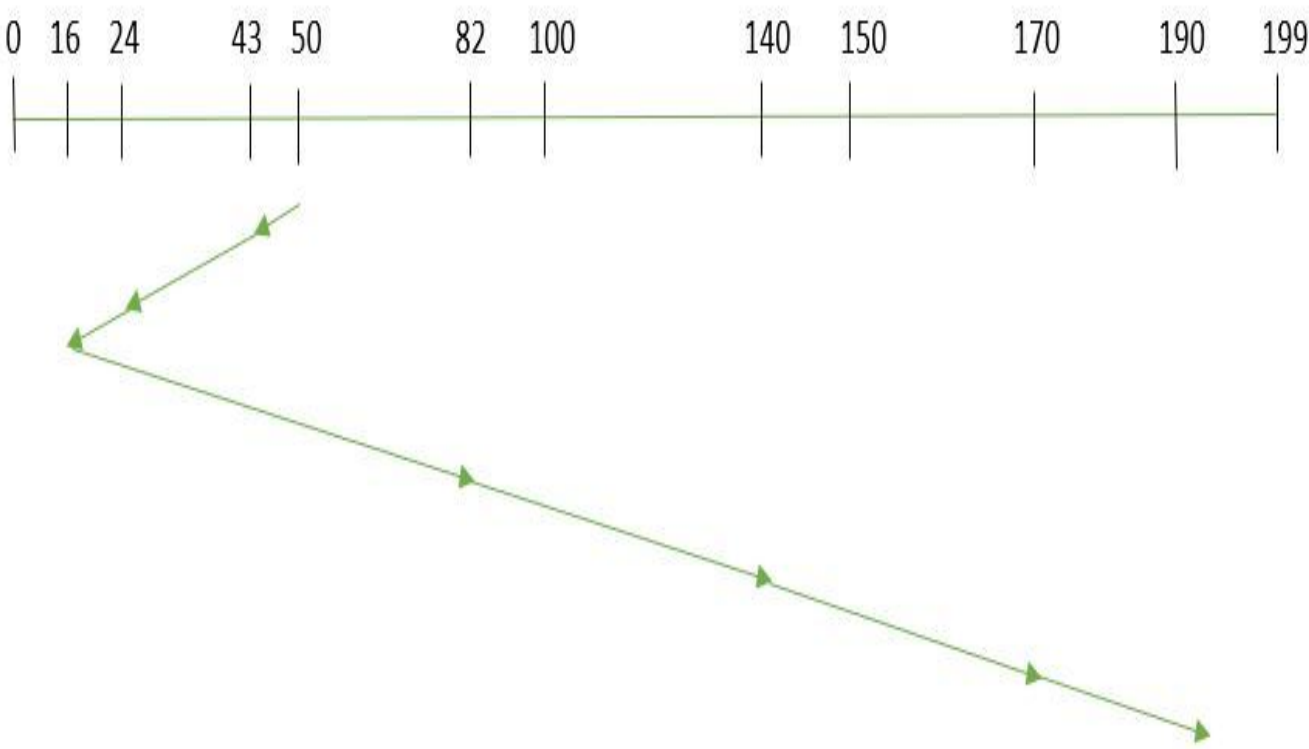
Disadvantages:

- Does not try to optimize seek time
- May not provide the best possible service

2. **SSTF:** In SSTF (Shortest Seek Time First), requests having shortest seek time are executed first. So, the seek time of every request is calculated in advance in the queue and then they are scheduled according to their calculated seek time. As a result, the request near the disk arm will get executed first. SSTF is certainly an improvement over FCFS as it decreases the average response time and increases the throughput of system.Let us understand this with the help of an example.

Example:

Suppose the order of request is- (82,170,43,140,24,16,190)
And current position of Read/Write head is : 50



So, total seek time:

=(50-43)+(43-24)+(24-16)+(82-16)+(140-82)+(170-40)+(190-170)
=208

Advantages:

- Average Response Time decreases
- Throughput increases

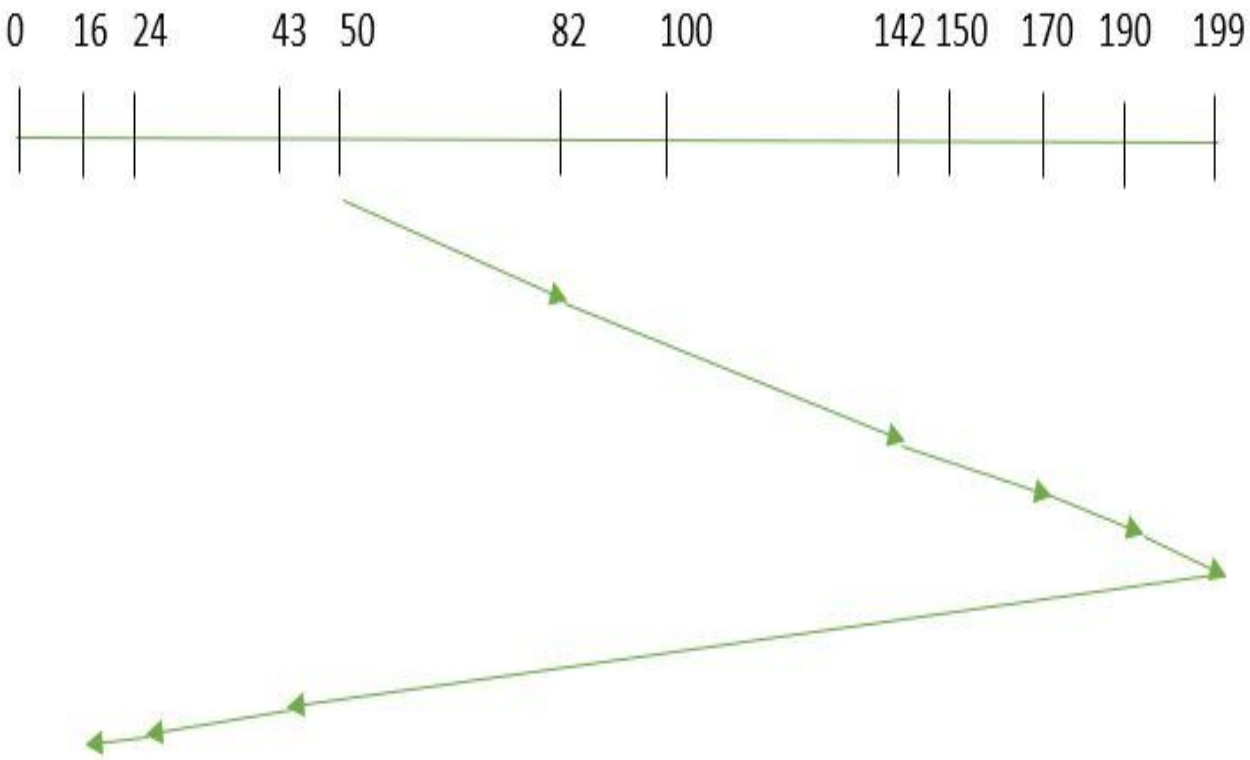
Disadvantages:

- Overhead to calculate seek time in advance
- Can cause Starvation for a request if it has higher seek time as compared to incoming requests
- High variance of response time as SSTF favours only some requests

3. **SCAN:** In SCAN algorithm the disk arm moves into a particular direction and services the requests coming in its path and after reaching the end of disk, it reverses its direction and again services the request arriving in its path. So, this algorithm works as an elevator and hence also known as **elevator algorithm.** As a result, the requests at the midrange are serviced more and those arriving behind the disk arm will have to wait.

Example:

Suppose the requests to be addressed are-82,170,43,140,24,16,190. And the Read/Write arm is at 50, and it is also given that the disk arm should move **"towards the larger value".**



Therefore, the seek time is calculated as:

=(199-50)+(199-16)
=332

Advantages:

- High throughput
- Low variance of response time
- Average response time

Disadvantages:

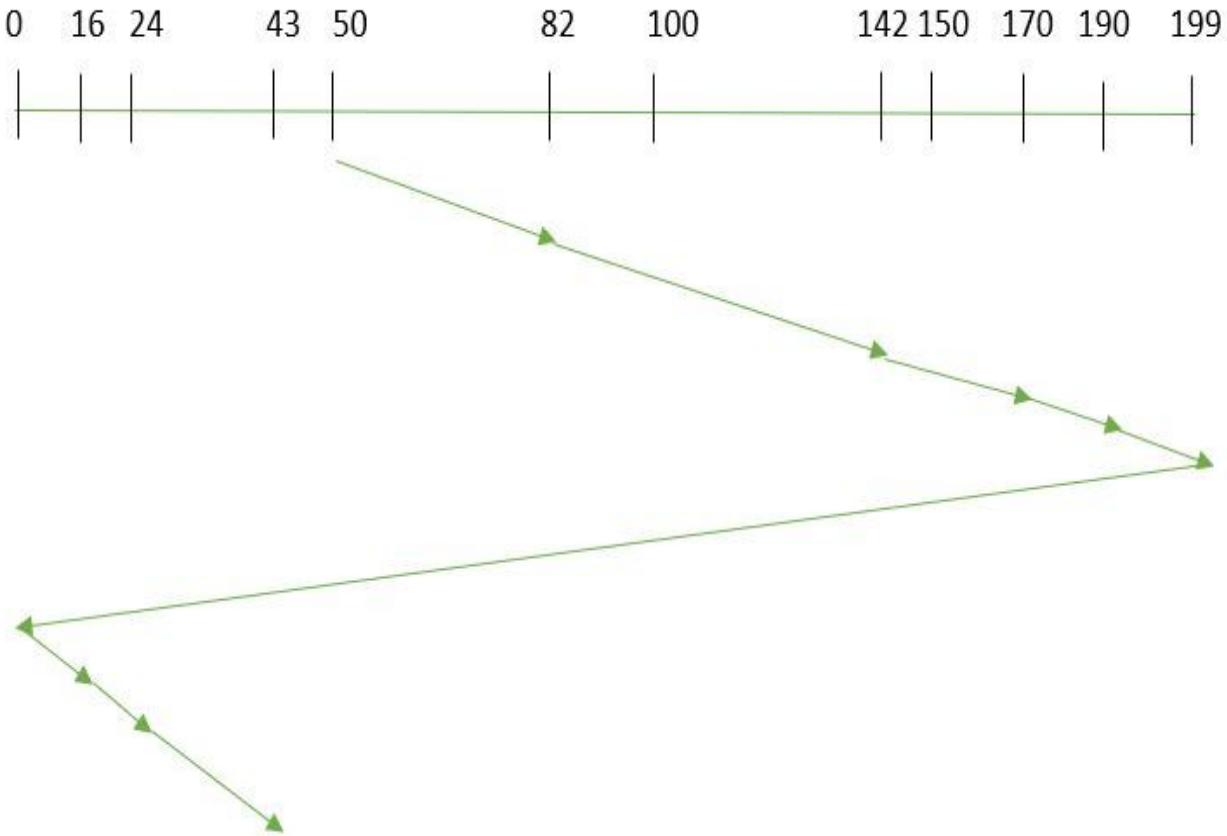
- Long waiting time for requests for locations just visited by disk arm

4. **CSCAN**: In SCAN algorithm, the disk arm again scans the path that has been scanned, after reversing its direction. So, it may be possible that too many requests are waiting at the other end or there may be zero or few requests pending at the scanned area.

These situations are avoided in *CSCAN* algorithm in which the disk arm instead of reversing its direction goes to the other end of the disk and starts servicing the requests from there. So, the disk arm moves in a circular fashion and this algorithm is also similar to SCAN algorithm and hence it is known as C-SCAN (Circular SCAN).

Example:

Suppose the requests to be addressed are-82,170,43,140,24,16,190. And the Read/Write arm is at 50, and it is also given that the disk arm should move **"towards the larger value".**



Seek time is calculated as:

=(199-50)+(199-0)+(43-0)
=391

Advantages:

- Provides more uniform wait time compared to SCAN

Disk Formatting

Disk formatting is a process to configure the data-storage devices such as hard-drive, floppy disk and flash drive when we are going to use them for the very first time or we can say initial usage. Disk formatting is usually required when new operating system is going to be used by the user. It is also done when there is space issue and we require additional space for the storage of more data in the drives. When we format the disk then the existing files within the disk is also erased.
We can perform disk formatting on both magnetic platter hard-drives and solid-state drives.

When we are going to use hard-drive for initial use it is going to search for virus. It can scan for virus and repair the bad sectors within the drive. Disk formatting has also the capability to erase the bad applications and various sophisticated viruses.

As we know that disk formatting deletes data and removes all the programs installed with in the drive. So it can be done with caution. We must have the backup of all the data and applications which we require. No-doubt disk formatting requires time. But the frequent formatting of the disk decreases the life of the hard-drive.

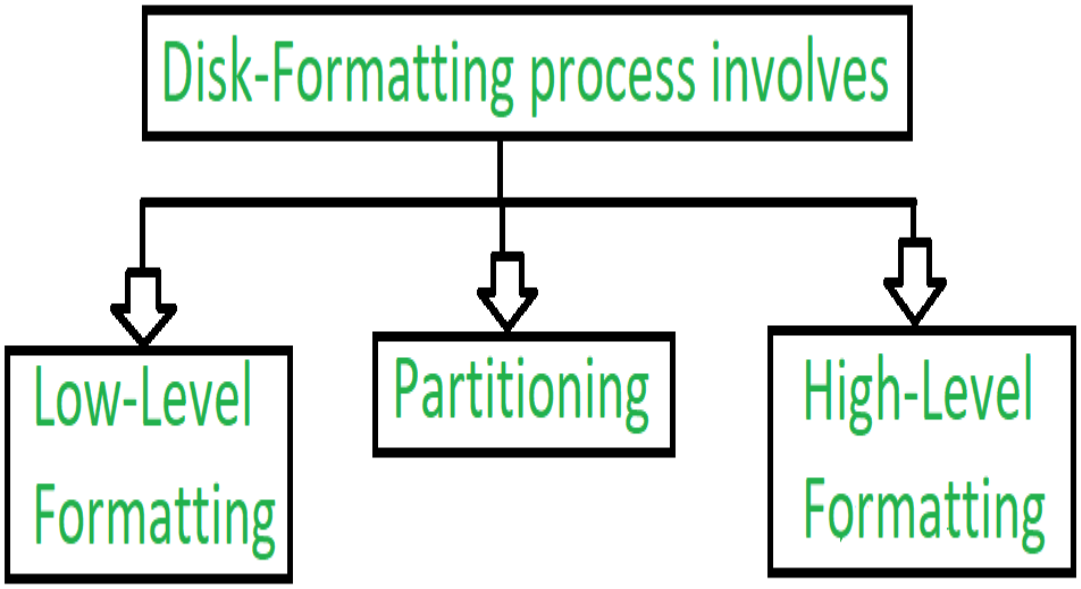


Figure – Formatting process of disk

1. Low-level Formatting :
Low level formatting is a type of physical formatting. In is the process of marking of cylinders and tracks of the blank hard-disk. After this there is the division of tracks into sectors with the sector markers. Now-a-days low-level formatting is performed by the hard-disk manufactures themselves.

We have data in our hard-disks and when we perform low-level formatting in the presence of data in the hard-disk all the data have been erased and it is impossible to recover that data. Some users make such a format that they can avoid their privacy leakage. Otherwise low-level will cause damage to hard-disk shortens the service-life.

Therefore, this formatting is not suggested to users.

2. Partitioning :
As suggesting from the name, partitioning means divisions. Partitioning is the process of dividing the hard-disk into one or more regions. The regions are called as partitions.
It can be performed by the users and it will affect the disk performance.

3.High-level Formatting :
High-level formatting is the process of writing. Writing on a file system, cluster size, partition label, and so on for a newly created partition or volume. It is done to erase the hard-disk and again installing the operating system on the disk-drive.

Boot Block in Operating System

Basically for a computer to start running to get an instance when it is powered up or rebooted it need to have an initial program to run. And this initial program which is known as **bootstrap** needs to be simple. It must initialize all aspects of the system, from CPU registers to device controllers and the contents of the main memory, and then starts the operating system.
To do this job the bootstrap program basically finds the operating system kernel on disk and then loads the kernel into memory and after this, it jumps to the initial address to begin the operating-system execution.

Why ROM:
For most of today's computer bootstrap is stored in Read Only Memory (ROM).

1. This location is good for storage because this place doesn't require initialization and moreover location here is fixed so that processor can start executing when powered up or reset.
2. ROM is basically read-only memory and hence it cannot be affected by the computer virus.

The problem is that changing the bootstrap code basically requires changes in the ROM hardware chips. Because of this reason, most system nowadays has the tiny bootstrap loader program in the boot whose only job is to bring the full bootstrap program from the disk. Through this now we are able to change the full bootstrap program easily and the new version can be easily written onto the disk.

The full bootstrap program is stored in the **boot blocks** at a fixed location on the disk. A disk that has a boot partition is called a boot disk. The code in the boot ROM basically instructs the read controller to read the boot blocks into the memory and then starts the execution of

code. The full bootstrap program is more complex than the bootstrap loader in the boot ROM, It is basically able to load the complete OS from a non-fixed location on disk to start the operating system running. Even though the complete bootstrap program is very small.

Bad Block in Operating system

Bad Block is an area of storing media that is no longer reliable for the storage of data because it is completely damaged or corrupted.
We know disk have moving parts and have small tolerances, they are prone to failure. In case when the failure is complete, then the disk needs to be replaced and its contents restored from backup media to the new disk. More frequently, one or more sectors become defective. More disks even come from the factory named **Bad blocks**.
This is also referred to as Bad Sector.

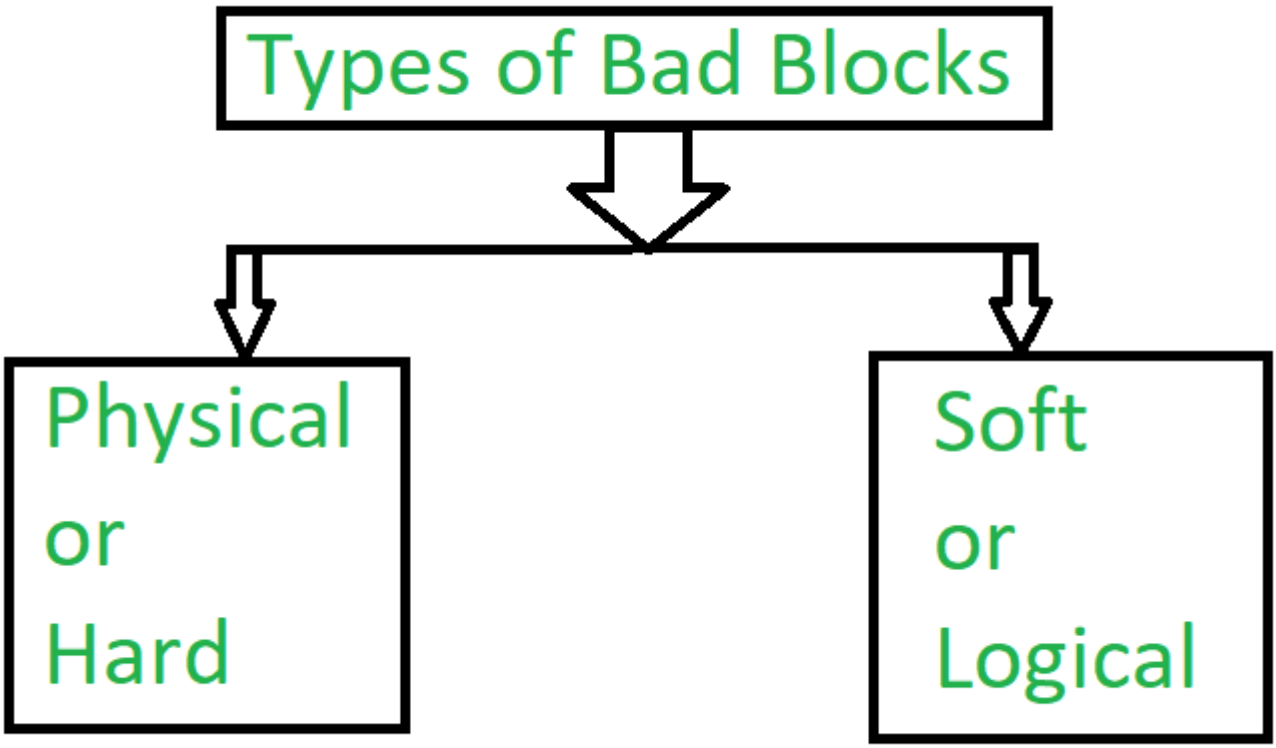
Cause of Bad Block :
Storage drives can ship from the factory with defective blocks that originated in the manufacturing process. The device with **bad-blocks** are marked as defective before leaving the factory. These are remapped with the available extra memory cells.
A physical damage to device also makes a device as bad block because sometimes operating system does not able to access the data. Dropping a laptop will also cause damage to the platter of the HDD's. Sometimes dust also cause damage to HDD's.

When the memory transistor fails it will cause damage to the solid-state drive. Storage cells can also become unreliable over time, as NAND flash substrate in a cell becomes unusable after a certain number of program-erase cycles.

For the erase process on the solid-state drive it requires a huge amount of electrical charge through the flash cards. This degrades the oxide layer that separates the floating gate transistors from the flash memory silicon substrate and the bit error rates increase. The drive's controller can use error detection and correction mechanisms to fix these errors. However, at some point, the errors can outstrip the controller's ability to correct them and the cell can become unreliable.

Soft bad sectors are caused by software problems. For instance, if a computer unexpectedly shuts down, due to this, hard drive also turn of in the middle of writing to a block. Due to this, the data contain in the block doesn't match with the CRC detection error code and it would marked as bad sector.

Types of Bad Blocks :
There are two types of bad blocks –



1. **Physical or Hard bad block :**
   It comes from damage to the storage medium.
2. **Soft or Logical bad block :**
   A soft, or logical, bad block occurs when the operating system (OS) is unable to read data from a sector.

DISK RELIABILITY

It is important to understand the terms reliability and performance as they pertain to disks. Reliability is the ability of the disk system to accommodate a single- or multi-disk failure and still remain available to the users. Performance is the ability of the disks to efficiently provide information to the users.

Adding redundancy almost always increases the reliability of the disk system. The most common way to add redundancy is to implement a Redundant Array of Inexpensive Disks (RAID).

There are two types of RAID:

- **Hardware** — The most commonly used hardware RAID levels are: RAID 0, RAID 1, RAID 5, and RAID 10. The main differences between these RAID levels focus on reliability and performance as previously defined.
- **Software** — Software RAID can be less expensive. However, it is almost always much slower than hardware RAID, because it places a burden on the main system CPU to manage the extra disk I/O.
The different hardware RAID types are as follows:

- **RAID 0 (Striping)** — RAID 0 has the following characteristics:
  - **High performance** — Performance benefit for randomized reads and writes
  - **Low reliability** — No failure protection
  - **Increased risk** — If one disk fails, the entire set fails

  The disks work together to send information to the user. While this arrangement does help performance, it can cause a potential problem. If one disk fails, the entire file system is corrupted.

- **RAID 1 (Mirroring)** — RAID 1 has the following characteristics:
  - **Medium performance** — Superior to conventional disks due to "optimistic read"
  - **Expensive** — Requires twice as many disks to achieve the same storage, and also requires twice as many controllers if you want redundancy at that level
  - **High reliability** — Loses a disk without an outage
  - **Good for sequential reads and writes** — The layout of the disk and the layout of the data are sequential, promoting a performance benefit, provided you can isolate a sequential file to a mirror pair

  In a two disk RAID 1 system, the first disk is the primary disk and the second disk acts as the parity, or mirror disk. The role of the parity disk is to keep an exact synchronous copy of all the information stored on the primary disk. If the primary disk fails, the information can be retrieved from the parity disk.

  Be sure that your disks are able to be hot swapped so repairs can be made without bringing down the system. Remember that there is a performance penalty during the resynchronization period of the disks.

  On a read, the disk that has its read/write heads positioned closer to the data will retrieve information. This data retrieval technique is known as an optimistic read. An optimistic read can provide a maximum of 15 percent improvement in performance over a conventional disk. When setting up mirrors, it is important to consider which physical disks are being used for primary and parity information, and to balance the I/O across physical disks rather than logical disks.

- **RAID 10 or 1+0** — RAID 10 has the following characteristics:
  - **High reliability** — Provides mirroring and striping
  - **High performance** — Good for randomized reads and writes.
  - **Low cost** — No more expensive than RAID 1 mirroring

  RAID 10 resolves the reliability problem of striping by adding mirroring to the equation.

CASE STUDY ON UNIX AND LINUX LINKS....

https://www.geeksforgeeks.org/difference-between-unix-and-windows-operating-system/

https://www.geeksforgeeks.org/difference-between-unix-and-windows-operating-system/