Parul® University

Operating System

Prof. Swapnil Umbarkar, Assistant Professor
Computer Science & Engineering

Parul®
University

DIGITAL LEARNING CONTENT

CHAPTER-6.3

Disk Structure

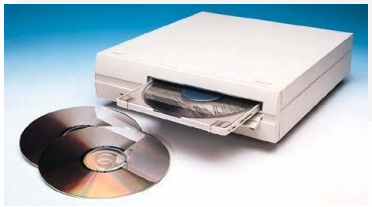
Disk

•A disk operating system (abbreviated DOS) is a computer operating system that resides on and can use a disk storage device, such as a floppy disk, hard disk drive, or optical disc.

•A disk operating system must provide a file system for organizing, reading, and writing files on the storage disk.

Storage Technologies

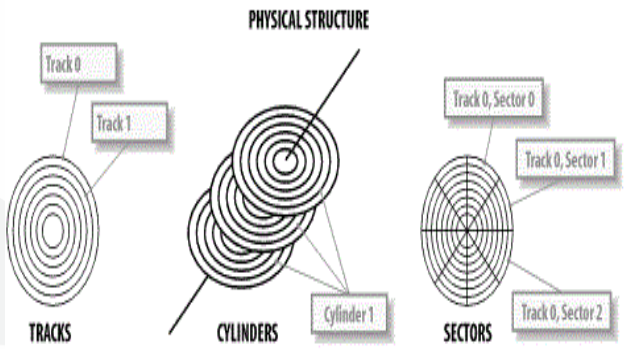
- Following are different example of storage technology



Storage Technologies (Source)

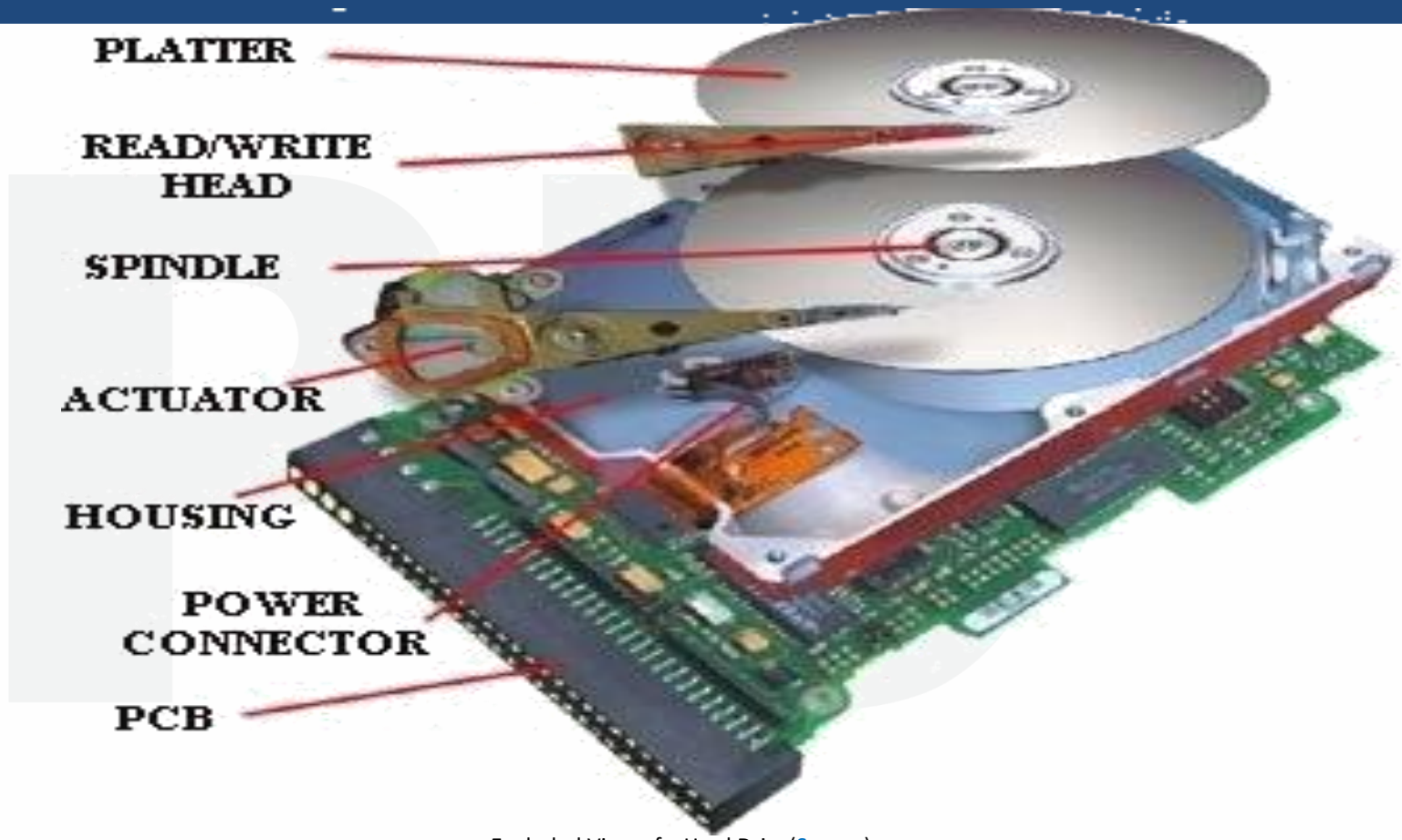
Disk Structure

- A Disk is usually divided into TRACKS, CYLINDERS AND SECTORS
- Hard disks drives are organized as a concentric stack of disks or 'platters'.
- Each platter has 2 surfaces and two read/write heads for each surface.
- Each platter has the same No. of tracks.
- Platter is made from aluminum, ceramic, or class, coated with a magnetic materials such as iron oxide.



Disk Structure(Source)

Exploded View of a Hard Drive



Exploded View of a Hard Drive(Source)

DISK GEOMETRY

Platters:

- Platters resemble the phonograph records found in an old-Fashioned Jukebox.
- Multiple platters increase storage without equivalent increase in cost.

Heads:

•Each platter is associated with the read/write Head.

•They are energy converters: I.e., they transform electric signals into magnetic(write the disk) and vice-versa(read the disk).

DISK GEOMETRY

Tracks:

- circular areas of the disk
- Length of a track one circumference of disk
- Over 1000 on a hard disk
- Data first written to outer most track

Sectors:

- Divide tracks sections

Disk Geometry

Cylinders:

- Logical groupings of the same track on each disk surface in a disk unit.
- OR
- All the tracks with the same radius are known as a CYLINDER.

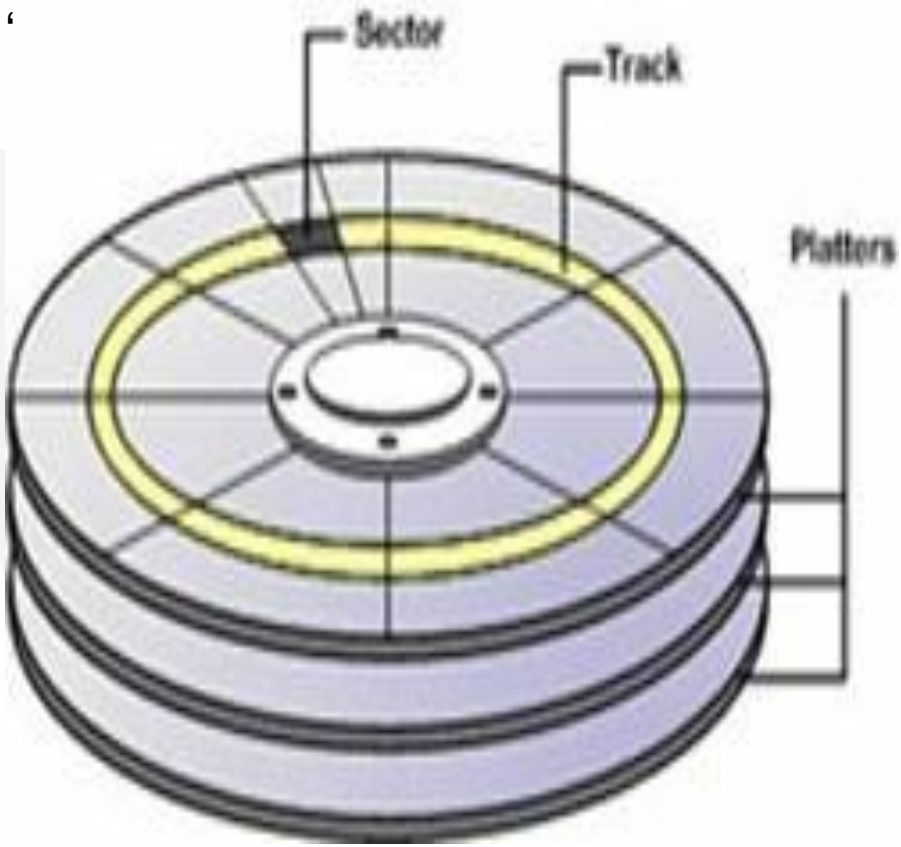
Clusters:

- Several sectors form a cluster.
- 64 sectors in one cluster
- Groups of sectors used by operating system.

Sectors

- A round magnetic plate that constitutes part of a Hard disk.
- Hard drives can contain a dozen of platters mounted on the same spindle.
- Platters require two read/write heads, one for each side and therefore can
- store information on both sides.

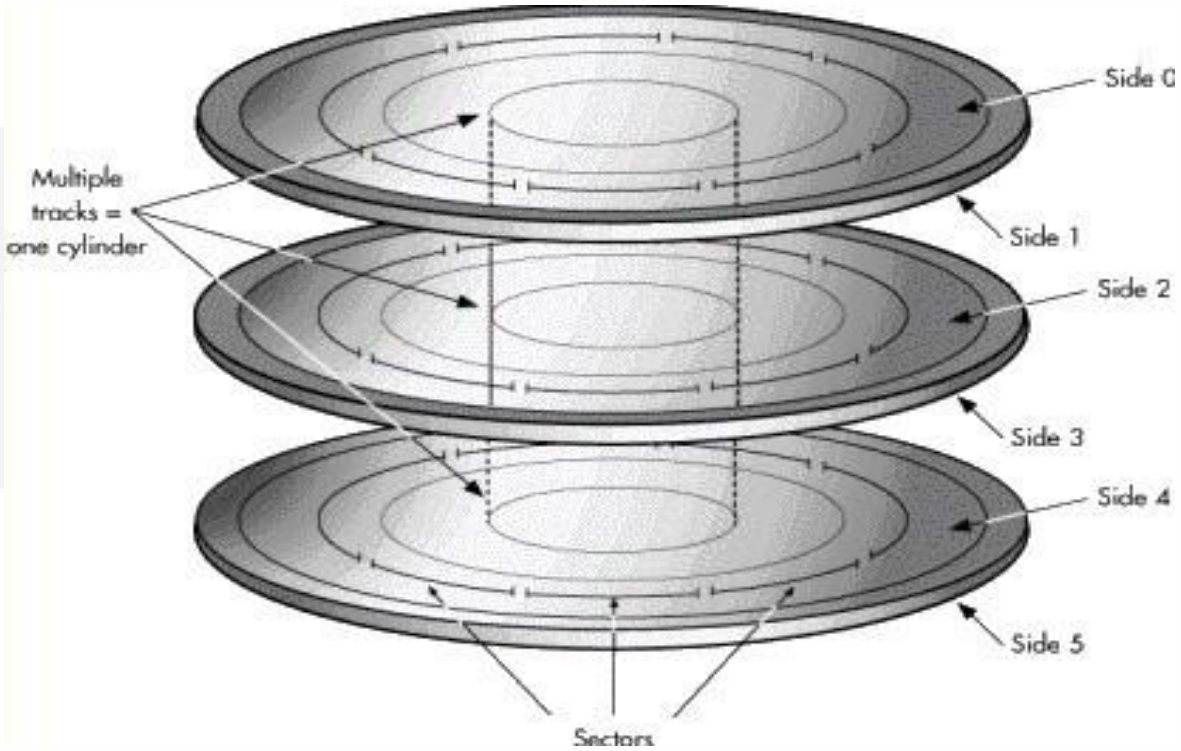
‘



Sectors(source)

Cylinders

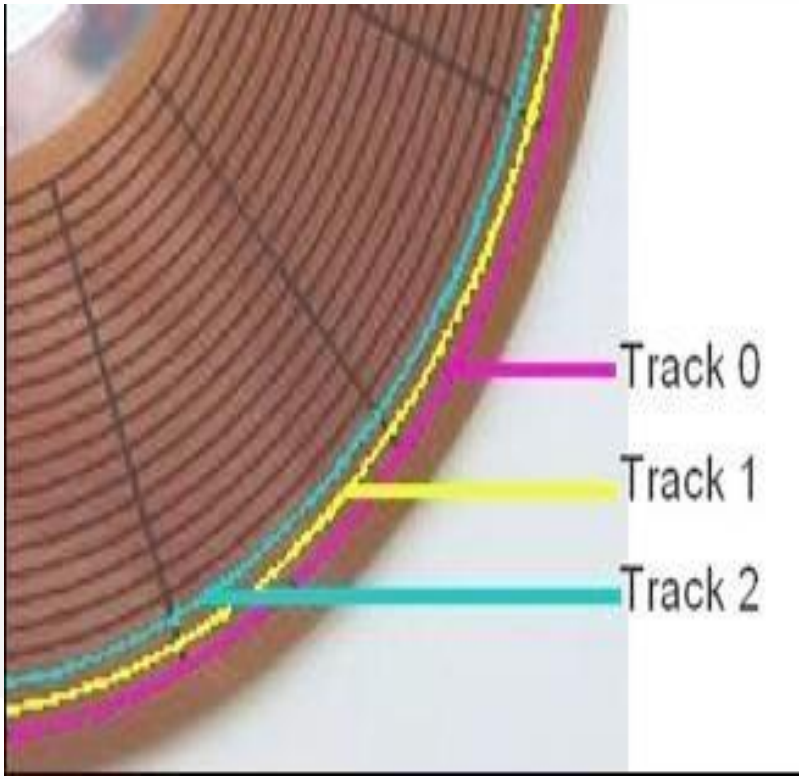
- Structure of cylinder



Structure of cylinder(Source)

TRACKS

- The data is stored on concentric circles on the surfaces known as tracks
- Corresponding tracks on all platter surfaces make up a cylinder
- Numbering starts with 0 at the outermost cylinder



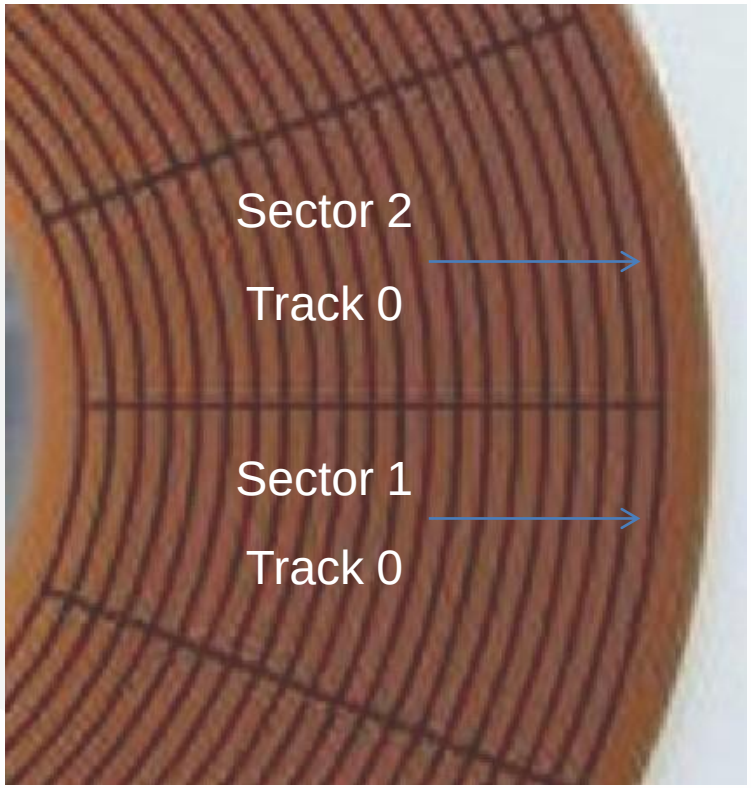
Tracks (Source)

Sector

A sector is a continuous linear stream of magnetized bits occupying a curved section of a track.

Sectors are the smallest physical storage units on a disk- Each sector stores 512 bytes of data

Numbering physical sectors within a track starts with 1.



Sectors(Source)

Cluster (Blocks)

- 1 or more contiguous sectors
- The smallest pieces of storage that an OS can place into data.
- The bytes in a cluster varies according to the size of the drive and the version of the OS.

DISK CAPACITY

- One method of calculating disk capacity is to multiply the number of cylinders, heads, and sectors (i.e. CHS) together, and then multiply by the block size of 512 Bytes:
- E.g. 12,495 cylinders * 16 heads * 63 sectors * 512 bytes = approx. 6GB

Logical Structure

•Basically, we can divide the logical structure of the hard disk in the following five logical terms:

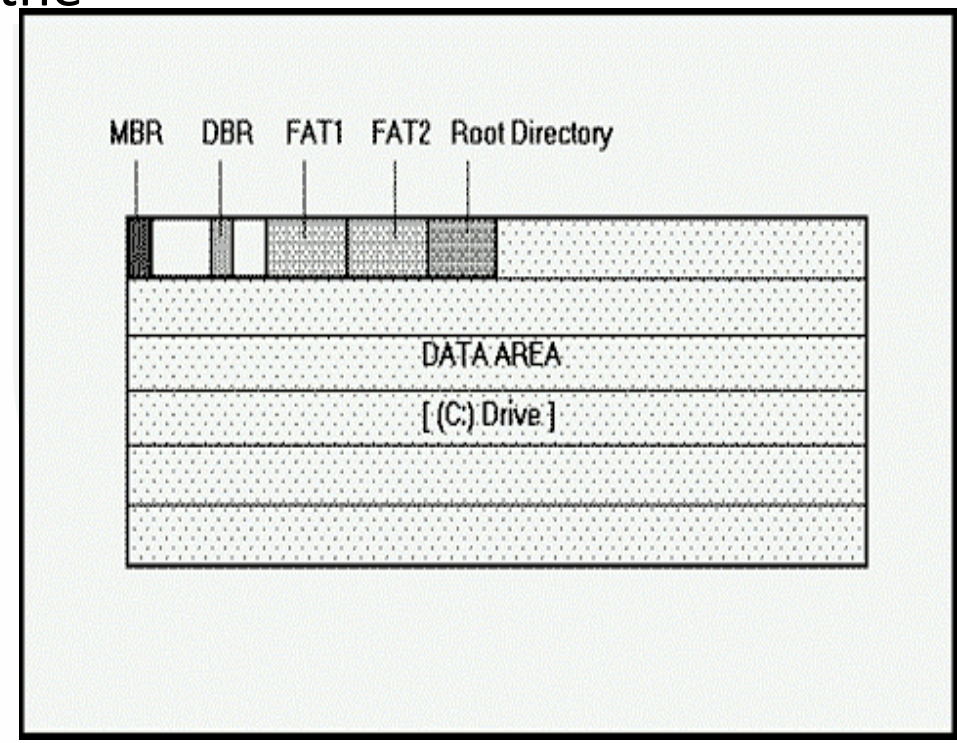
•MBR(Master Boot Record)

•DBR(DOS Boot Record)

•FAT(File Allocation Tables)

•Root Directory

•Data Area

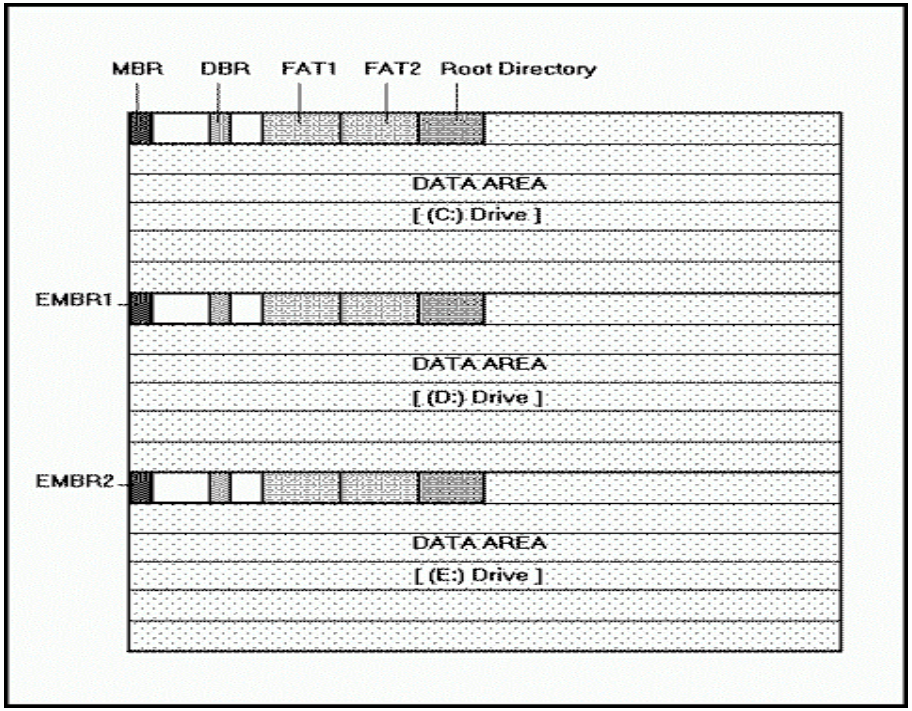


Logical Structure(source)

Master Boot Record

- It is sometimes referred as the master partition table(MPT).
- It contains a small program to load and start the active partition from the hard disk.
- The MBR is created on the hard disk drive by executing FDISK.EXE command of DOS.(If there is no MBR) like C:\>FDISK.EXE/MBR
- It is located at absolute sector 0 or we can say at cylinder 0,head 0 and sector 1(The MBR).
- If we have more than one partition, then there are Extended Master Boot Records, located at the beginning of each extended partition volume.

Master Boot Record



Master Boot Record(Source)

Dos Boot Record

•After the partition table , the DOS Boot Record(DBR) or sometimes called DOS Boot Sector is the second most important information on your hard disk.

•It contains some important information about disk geometry like:

- •Bytes Per Sector
- •Sectors per cluster
- •Reserved Sectors etcetra.

•The DBR is created by the FORMAT command of DOS.

•The job of DBR is to load the operating system from the hard disk drive into the main memory of computer and give the systems control to the loaded program.

•For doing this, the DBR contains a small program which is executed by the MBR Executable program. All DOS partitions contain the program code to boot the machine, but only that partition is given control by the MBR which is specified as active partition.

File Allocation Table

- The FAT was introduced in 1977 to store data and has been modified several times to accommodate expanding needs.
- It was developed to fulfill the requirements of a fast and flexible system for managing data on both removable and fixed media.
- FAT keeps a map of the complete surface of the disk drive such that, which area is free, which area is taken up by which file etc. When some data stored on the disk is to be accessed, the DOS consults the FAT to find out the areas of the hard disk which contains the data.
- The FAT manages the disk area in a group of sectors called “CLUSTER”.

FAT entries can contain values that indicate

- The next cluster in a FAT chain for a given file.
- The free clusters I.e., the clusters which are not in use by any file.
- The information of Bad sectors i.e., the clusters containing one or more sectors that are physically damaged and should not be used.

FAT entries can contain values that indicate

Number(Hex.)	Description
0	Free cluster
????	Cluster in use, next cluster in chain.
FF0- FF6/FFF0- FFF6	Cluster is reserved
FF7/FFF7	Cluster contains bad sectors i.e., it is not used for data storage.
FF8- FFF/FFF8- FFFF	End of file.

FAT entries(Source)

Root Directory

- The Root Directory is like a table of contents for the information stored on the hard disk drive.
- The directory area keeps the information about the file name, date and time of the file creation, file attribute, file size and starting cluster of the particular file.
- The number of files that one can store on the root directory depends on the FAT type being used.

The following table lists the limits of root entries for

Media and File System Description	Maximum Root Directory Entries
Single-sided 5¼ Inch 180k FDD(Floppy DiskDrive)	64
Double-sided 5¼ Inch 320kFDD	64
Double-sided 5¼ Inch 360kFDD	112
Double-sided 3½ Inch 720kFDD	112
Double-sided 5¼ Inch 1.2-megabyte FDD	224
Double-sided 3½ Inch 1.44-megabyte FDD	224
Double-sided 3½ Inch 2.88-megabyte FDD	240
Hard Drives(FAT12 &FAT16)	512
Hare Drives with FAT32(As it treats the root directory as a file)	65536

Data Area / Files Area

- The remainder of the volume after Root Directory is the Data Area.
- The data area contains the actual data stored on the disk surfaces.
- When we format a hard disk the FORMAT command of DOS does not destroy or overwrite the data on the data area. The FORMAT command only removes the directory entry and FAT entries and it does not touch the actual data area. This makes the recovery of accidentally formatted hard disk drive possible.

Disk Scheduling Algorithms

Overview of Mass Storage Structure

- **Magnetic disks** provide bulk of secondary storage of modern computers Drives which rotate at a speed of 60 to 250 times per second
- **Transfer rate** is rate at which data flow between drive and computer
- **Positioning time** (**random-access time**) is time to move disk arm to desired cylinder (**seek time**) and time for desired sector to rotate under the disk head (**rotational latency**)
- **Head crash** results from disk head making contact with the disk surface

Disk Structure

- Disk drives are addressed as large 1-dimensional arrays of logical blocks, where the logical block is the smallest unit of transfer.

- The 1-dimensional array of logical blocks is mapped into the sectors of the disk sequentially.
- Sector 0 is the first sector of the first track on the outermost cylinder.
- Mapping proceeds in order through that track, then the rest of the tracks in that cylinder, and then through the rest of the cylinders from outermost to innermost.

Disk Scheduling

- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth.
- Access time has two major components
- Seek time is the time for the disk are to move the heads to the cylinder containing the desired sector.
- Rotational latency is the additional time waiting for the disk to rotate the desired sector to the disk head.
- Minimize seek time
- Seek time ∝ seek distance
- Disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.

FCFS Scheduling

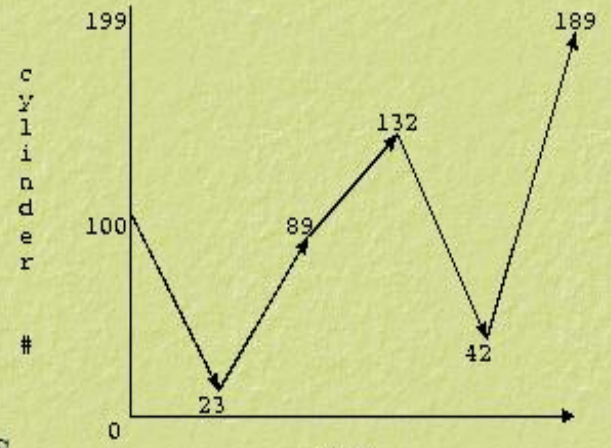
- The operating system is responsible for using hardware efficiently — for the disk drives, this means having a fast access time and disk bandwidth.
- Access time has two major components
- Seek time is the time for the disk are to move the heads to the cylinder containing the desired sector.
- Rotational latency is the additional time waiting for the disk to rotate the desired sector to the disk head.
- Minimize seek time
- Seek time seek distance
- Disk bandwidth is the total number of bytes transferred, divided by the total time between the first request for service and the completion of the last transfer.

FCFS Scheduling

- FCFS is the simplest disk scheduling algorithm. As the name suggests, this algorithm entertains requests in the order they arrive in the disk queue.
- The algorithm looks very fair and there is no starvation (all requests are serviced sequentially) but generally, it does not provide the fastest service.

FCFS Scheduling

- Work Queue: 23, 89, 132, 42, 187
- there are 200 cylinders numbered from 0 - 199
- the diskhead stars at number 100



1. **FCFS**

- total time is estimated by total arm motion

$$|100-23|+|23-89|+|89-132|+|23-132|+|132-42|+|42-187|=77+66+43+90$$

SSTF(Shortest Sick Time First)

SSTF Disk Scheduling Algorithm-

•SSTF stands for Shortest Seek Time First.
•This algorithm services that request next which requires least number of head movements from its current position regardless of the direction.
•It breaks the tie in the direction of head movemen

SSTF(Shortest Sick Time First)

Advantages-

- It reduces the total seek time as compared to FCFS.
- It provides increased throughput.
- It provides less average response time and waiting time.
-

SSTF(Shortest Sick Time First)

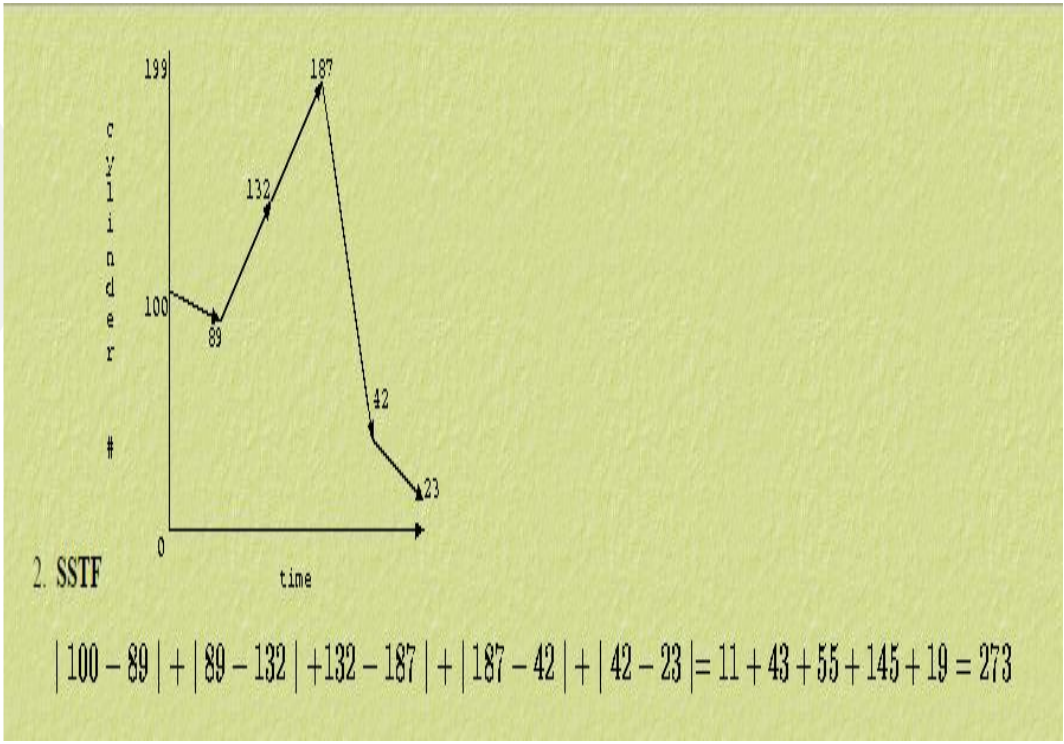
Disadvantages-

- There is an overhead of finding out the closest request.
- The requests which are far from the head might starve for the CPU.
- It provides high variance in response time and waiting time.
- Switching the direction of head frequently slows down the algorithm.

Shortest Sick Time First

Features

- (Shortest Seek Time First)
- Service all the request close to the current head position before moving the head far away
- May cause starvation of some requests
- Not an optimal algorithm



SCAN Scheduling

SCAN Disk Scheduling Algorithm-

- As the name suggests, this algorithm scans all the cylinders of the disk back and forth.
- Head starts from one end of the disk and move towards the other end servicing all the requests in between.
- After reaching the other end, head reverses its direction and move towards the starting end servicing all the requests in between.
- The same process repeats.

NOTE-

- SCAN Algorithm is also called as **Elevator Algorithm.**
- This is because its working resembles the working of an elevator.

CSCAN

Advantages-

- It is simple, easy to understand and implement.
- It does not lead to starvation.
- It provides low variance in response time and waiting time.

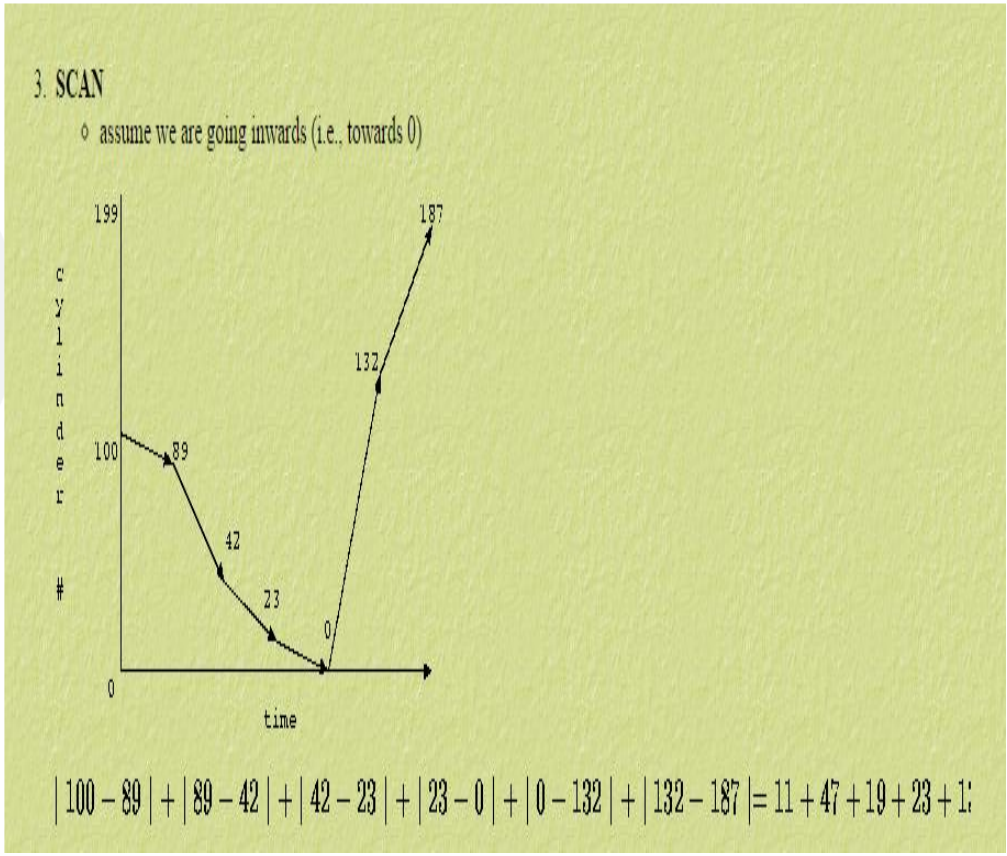
Disadvantages-

- It causes long waiting time for the cylinders just visited by the head.
- It causes the head to move till the end of the disk even if there are no requests to be serviced.

SCAN

Features

- Also known as elevator algorithm
- Disk arm starts at one end of the disk and moves toward the other end, servicing request
- Head scans back and forth across the disk
- Totaltime=|100-89|+|89-42|+|42-23|+|23-0|+|0-132|+|132-187|=11+47+19+23+132+55=287



CSCAN

C-SCAN Disk Scheduling Algorithm-

- Circular-SCAN Algorithm is an improved version of the **SCAN Algorithm**
- Head starts from one end of the disk and move towards the other end servicing all the requests in between.
- After reaching the other end, head reverses its direction.
- It then returns to the starting end without servicing any request in between.
- The same process repeats.

CSCAN

Advantages-

- The waiting time for the cylinders just visited by the head is reduced as compared to the SCAN Algorithm.
- It provides uniform waiting time.
- It provides better response time.

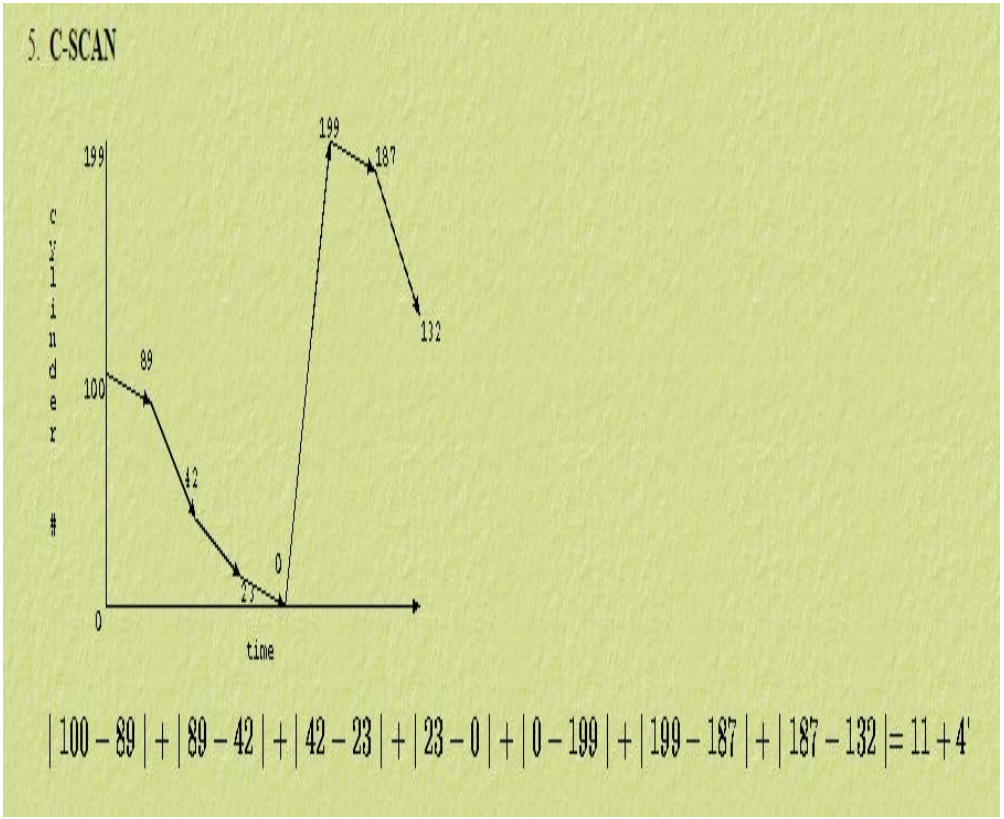
CSCAN

Disadvantages-

- It causes more seek movements as compared to SCAN Algorithm.
- It causes the head to move till the end of the disk even if there are no requests to be serviced.

CSCAN

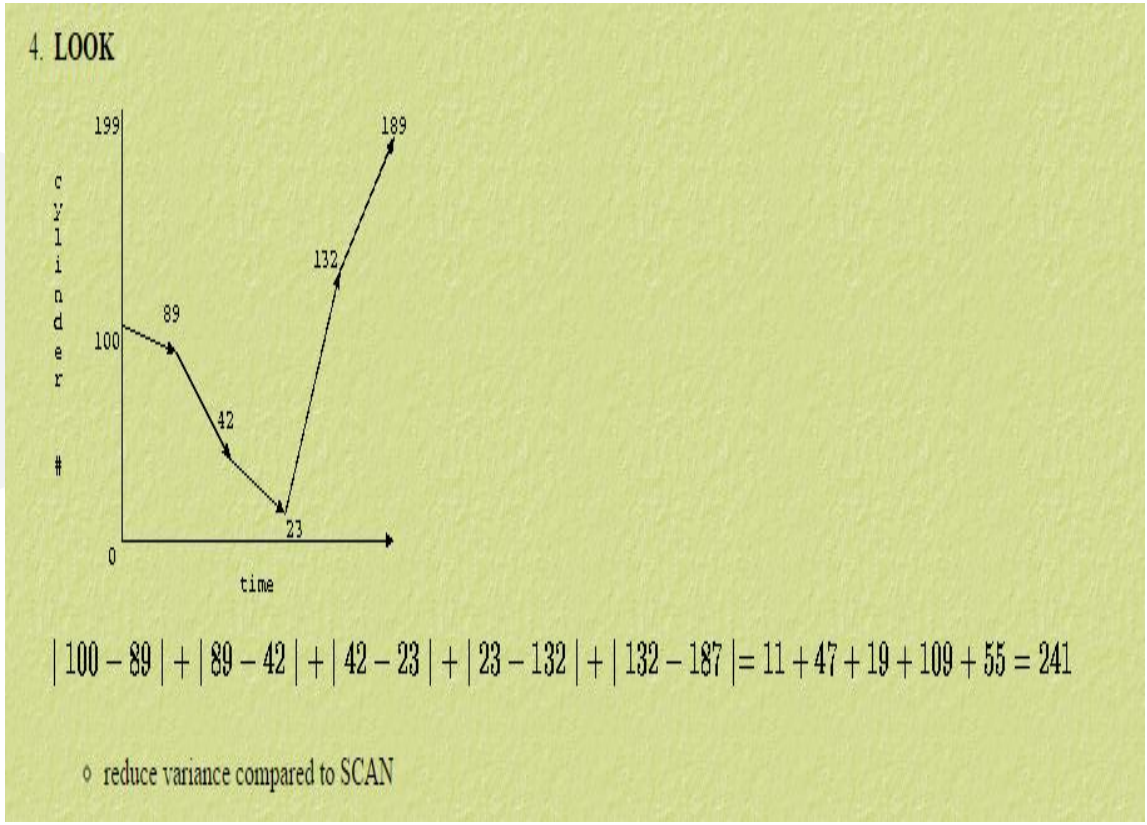
- Features
- Variant of SCAN designed to provide more uniform wait time
- Like SCAN, C-SCAN moves the head from one end of the disk to other, servicing request along the way.
- When the head reaches the other end, it immediately returns to the beginning of the disk without servicing any requests on the return trip.
- Time taken: |100-89|+|89-42|+|42-23|+|23-0|+|0-199|+|199-187|+|187-132|= 11+47+19+23+199+12+55=366



LOOK

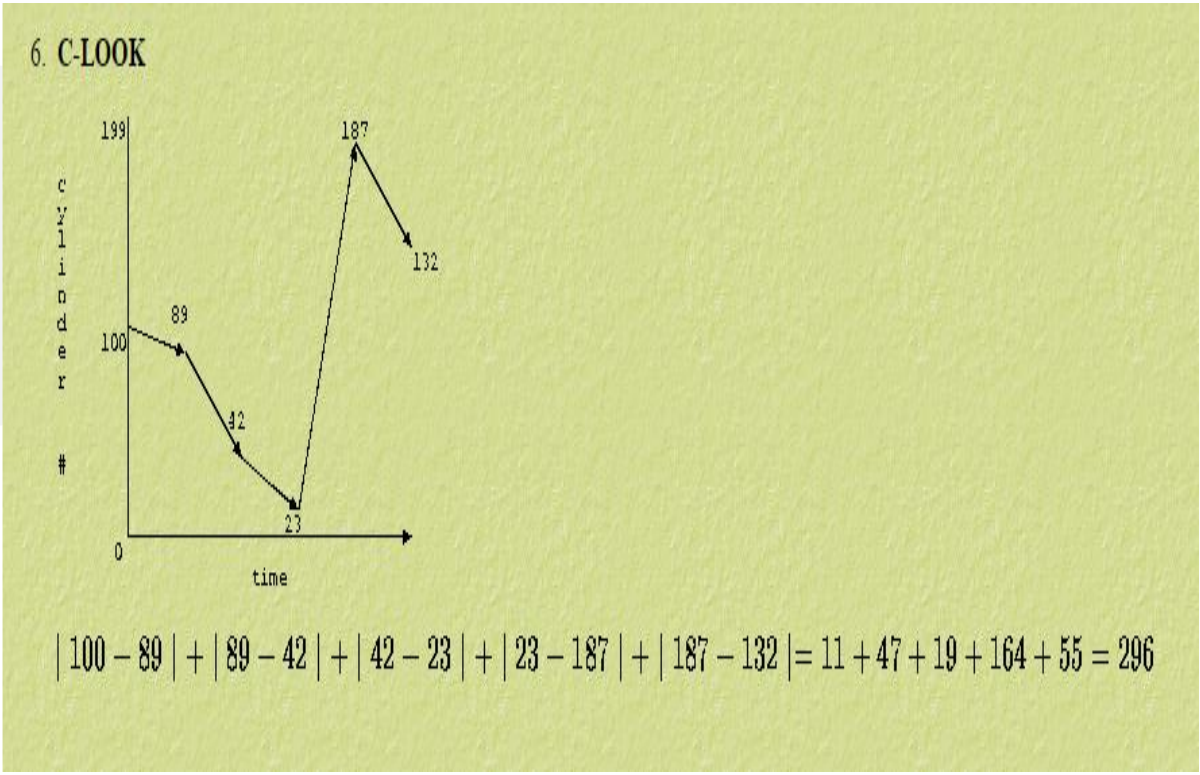
- Features

1. The arm goes only as far as the final request in each direction
2. Called as LOOK because it looks for a request before continuing to move in a given direction



CLOOK

CLOOK same as look only difference is it will cover only last point and returns back



Selecting a disk scheduling algorithm

- SSTF is common and has a natural appeal
- SCAN and C-SCAN perform better for systems that place a heavy load on the disk
  - Less starvation

- Performance depends on the number and types of requests
- Requests for disk service can be influenced by the file-allocation method and metadata layout

- The disk-scheduling algorithm should be written as a separate module of the operating system, allowing it to be replaced with a different algorithm if necessary
- Either SSTF or LOOK is a reasonable choice for the default algorithm

Disk Scheduling Algorithm

Table 11.3 Disk Scheduling Algorithms

Name	Description	Remarks
Selection according to requestor		
RSS	Random scheduling	For analysis and simulation
FIFO	First in first out	Fairest of them all
PRI	Priority by process	Control outside of disk queue management
LIFO	Last in first out	Maximize locality and resource utilization
Selection according to requested item		
SSTF	Shortest service time first	High utilization, small queues
SCAN	Back and forth over disk	Better service distribution
C-SCAN	One way with fast return	Lower service variability
N-step-SCAN	SCAN of N records at a time	Service guarantee
FSCAN	N-step-SCAN with N = queue size at beginning of SCAN cycle	Load sensitive

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 need 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).

- This location is good for storage because this place doesn't require initialization and moreover location here it is fixed so that processor can start executing when powered up or reset.
- 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.

Boot Block

- Full bootstrap program is stored in the boot blocks at a fixed location on the disk.
- A disk which 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.

Example

- Let us try to understand this using an example of the boot process in Windows 2000.
- The Windows 2000 basically stores its boot code in the first sector on the hard disk. Moreover, Windows 2000 allows the hard disk to be divided into one or more partition. In this one partition is basically identified as the boot partition which basically contains the operating system and the device drivers.
- Booting in Windows 2000 starts by running the code that is placed in the system's ROM memory. This code basically directs the system to read code directly from MBR.
- In addition to this boot code also contain the table which lists the partition for the hard disk and also a flag which basically indicates which partition is to be boot from the system. Once the system identifies the boot partition it reads the first sector from the memory which is basically known as boot sector and continue the process with the remainder of the boot process which basically includes loading of various system services.

Increasing disk reliability with RAID

•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).

Types of 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.

RAID 0

- **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

- **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

RAID 10

- **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.

RAID 5

- RAID 5 — RAID 5 has the following characteristics:
- High reliability — Provides good failure protection
- Low performance — Performance is poor for writes due to the parity's construction
- Absorbed state — Running in an absorbed state provides diminished performance throughout the application because the information must be reconstructed from parity
- Caution: Progress Software Corporation recommends not using RAID 5 for database systems.
- It is possible to have both high reliability and high performance. However, the cost of a system that delivers both of these characteristics is higher than a system that is only delivers one of the two.

Bad Block

- A bad block is an area of storage media that is no longer reliable for storing and retrieving data because it has been physically damaged or corrupted. Bad blocks are also referred to as bad sectors.
- There are two types of bad blocks: A physical, or hard, bad block comes from damage to the storage medium. A soft, or logical, bad block occurs when the operating system (OS) is unable to read data from a sector. Examples of a soft bad block include when the cyclic redundancy check (CRC), or error correction code (ECC), for a particular storage block does not match the data read by the disk.

Causes

- Storage drives can ship from the factory with defective blocks that originated in the manufacturing process. Before the device leaves the factory, these bad blocks are marked as defective and remapped to the drive's extra memory cells.
- A bad block can also result from physical damage to a device that makes it impossible for the OS to access data. On HDDs, mishaps, such as dropping a laptop, can cause the drive head to damage the platter. Dust and natural wear can also damage HDDs.
- Damage to a solid-state drive (SSD) can occur when a memory transistor fails. Storage cells can also become unreliable over time, as the NAND flash substrate in a cell becomes unusable after a certain number of program-erase cycles.

What Bad Block Do

- When a block is damaged or corrupted, it can make the data stored there inaccessible. If operating system or application files are stored in a damaged block, that can cause OS issues or result in an application failing to run. As the number of bad blocks on a disk increases, they can diminish the drive's capacity and performance and eventually cause hardware failure.
- Disk utility software, such as CHKDSK on Microsoft Windows systems or badblocks on Linux systems, can scan storage media and mark the failed sectors so that the OS doesn't use them. The firmware on an HDD controller can also identify and mark a bad block as unusable. This usually happens when a block is being overwritten with new data. The controller automatically remaps bad blocks to a different sector. Once it is identified as bad, that sector is not used in future operations.

Bad Block Management

- Bad block management is critical to improving NAND flash drive reliability and endurance. Unlike magnetic storage media, flash can't be overwritten at the byte level; all changes must be written to a new block and the data in the original block must be marked for deletion.
- Once a flash drive fills up, the controller must start clearing out blocks marked for deletion before it can write new data. To do this, it consolidates good data by copying it to a new block. This process requires extra writes to consolidate the good data and results in write amplification where the number of actual writes exceeds the number of writes requested. Write amplification can decrease the performance and life span of a flash drive.
- Flash vendors use a number of techniques to control write amplification. One, known as garbage collection involves proactively consolidating data by freeing up blocks that were written to previously. Done properly, these reallocated sectors can reduce the need to erase entire blocks of data for every write operation.

References

[1] Silberschatz, A., Galvin, P. B., & Gagne, G. (2005). Operating system concepts. Hoboken, NJ: J. Wiley & Sons.

[2] Stallings, W. (2018). Operating systems: Internals and design principles.

Prentice-Hall

[3] Tanenbaum, A. (2014). Modern operating systems. Harlow: Pearson.

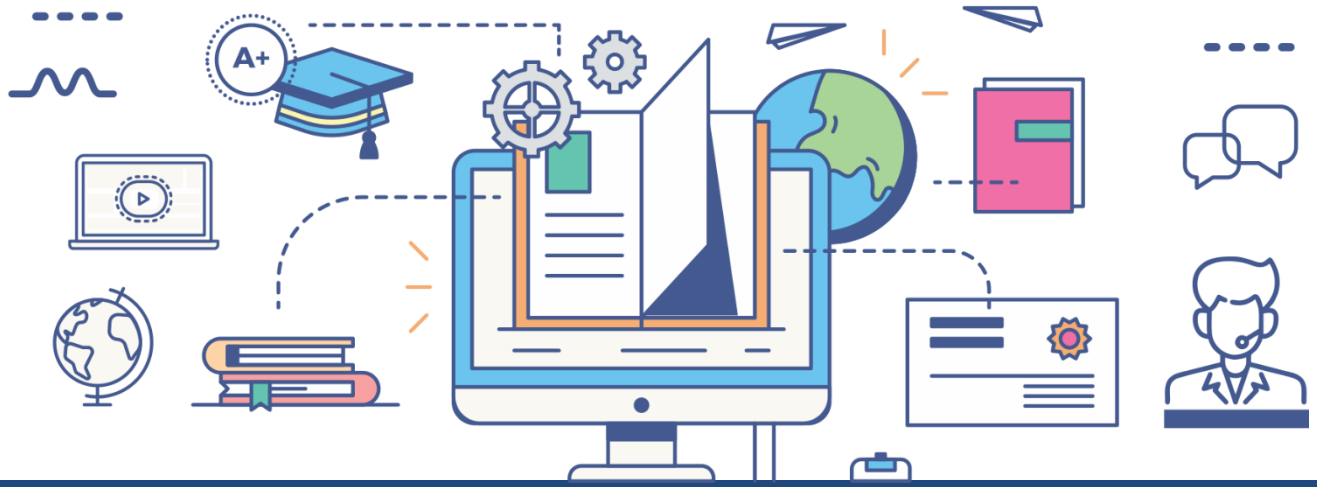
[4] Nutt, G. J. (2004). Operating systems: A modern perspective. Boston: Pearson/Addison Wesley.

[5] Bower T. Operating System Structure. K–State Polytechnic. http://faculty.salina.k-state.edu/tim/ossg/Introduction/struct.html

[6] Bower T. Basic Operating System Concepts. K–State Polytechnic. http://faculty.salina.k-state.edu/tim/ossg/Introduction/OSrole.html

[7] Operating System Generations. Tutorialspoint. https://www.tutorialspoint.com/operating-system-generations

DIGITAL LEARNING CONTENT



Parul® University

www.paruluniversity.ac.in