Operating System with Linux as case study

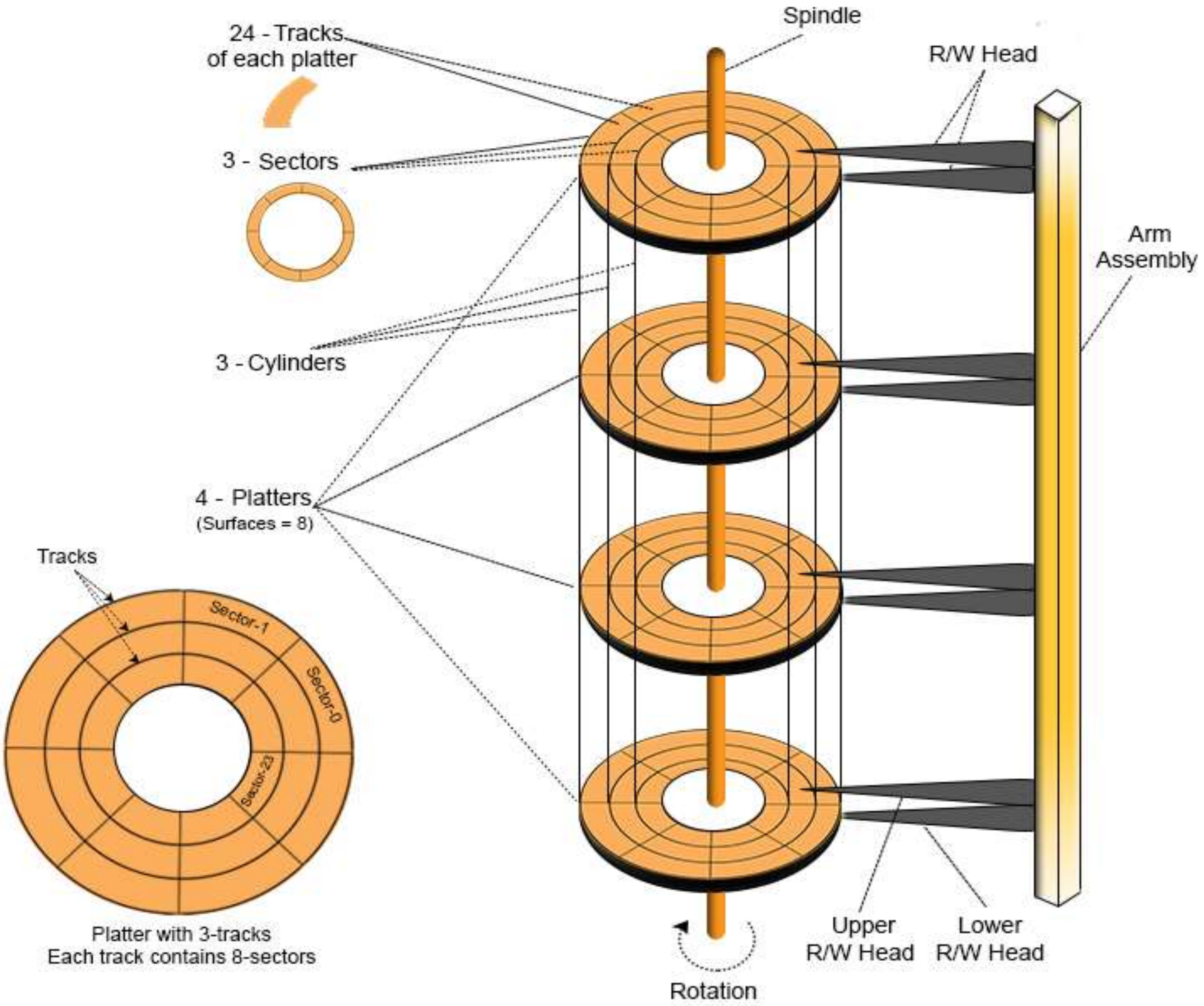
Module 1.3

Disk Management

Disk Structure

- Hard disk is secondary storage which stores the large amount of data.
- It is a type of electromechanical device.
- The Hard disk drive contains a dozens of disks. These **disks are** also known as **platters**.
- These platters are coated with magnetic material and mount over the spindle which rotates in any direction i.e. clockwise or anti-clockwise.

Disk Structure



Platter with 3-tracks
Each track contains 8-sectors

Platter

- The Platter is made of aluminum or iron oxide.
- Platter diameter range is 1.8 inches to 5.25 inches
- Each platter contains 2 surfaces and 2 Read/Write Head. One Read/Write head requires for one surface of the platter. So, other R/W head use for other surface to store the information's.
- Every platter holds the same no. of tracks.
- Multiple platters increase the storage capacity.

R/W Head

- R/W Heads moves forth and back over the platter surfaces to Read or Write the data on sectors.
- Read/Write heads does not touch to platters surface. The data written over the platter surface is done through magnetic field. If R/W Head touches over the surface of platter then bad sectors may creates. Hard disk may damage due to these bad sectors.

Tracks

- Circular areas of disk are known as tracks.
- There may be more than a 1000 tracks on a 3.5 inch hard disk and sector size. Track Numbering start with zero at outermost track.

Sectors

- Tracks are further divided into number of small units; these small units are known as sectors.
- Sectors are the smallest physical storage units on disk. Size of each sector is almost always **512** Bytes.
- Sector Numbering start with 1 at outermost tracks.

Cylinder

- All Corresponding tracks with same radius of all platters in the Hard disk are known as cylinders.
- So, Number of tracks in platter is always equal to number of cylinders.
- For example, a hard disk, where each platter contains 600 tracks then the number of cylinders will also be 600 in the hard disk.
- Cylinder Numbering start with zero at outermost cylinder.

Cluster

- Cluster is also known as blocks.
- **Group of sectors makes a cluster.**

Disk Capacity

- As we know there are number of platters in the hard disk. Each platter contains two R/W heads.
- There are number of cylinder/tracks in the hard disk.
- Each track is divided into number of sectors.
- And each sector has some size but mostly size of sector is 512 Bytes.

Question 01: Consider a disk pack with the following specifications- 16 surfaces, 128 tracks per surface, 256 sectors per track and 512 bytes per sector.

What is the capacity of disk pack?

Solution

- Number of surfaces = 16
- Number of tracks per surface = 128
- Number of sectors per track = 256
- Number of bytes per sector = 512 bytes

Capacity of disk pack

= Total number of surfaces x Number of tracks per surface x Number of sectors per track x Number of bytes per sector

= 16 x 128 x 256 x 512 bytes

= 2^{28} bytes

= 256 MB

Disk Management

- **Disk management** is the process of organizing and maintaining the storage on a computer's hard disk.
- It involves dividing the hard disk into partitions, formatting these partitions to different file systems, and regularly maintaining and optimizing disk performance.
- The goal of disk management is to provide a convenient and organized storage system for users to store and access their data, as well as to ensure that the computer runs smoothly and efficiently.

The operating system's disk management includes:

- Disk Formatting(or low-level format or physical format)
- Boot Block -Booting from disk
- Bad Block

Disk Formatting

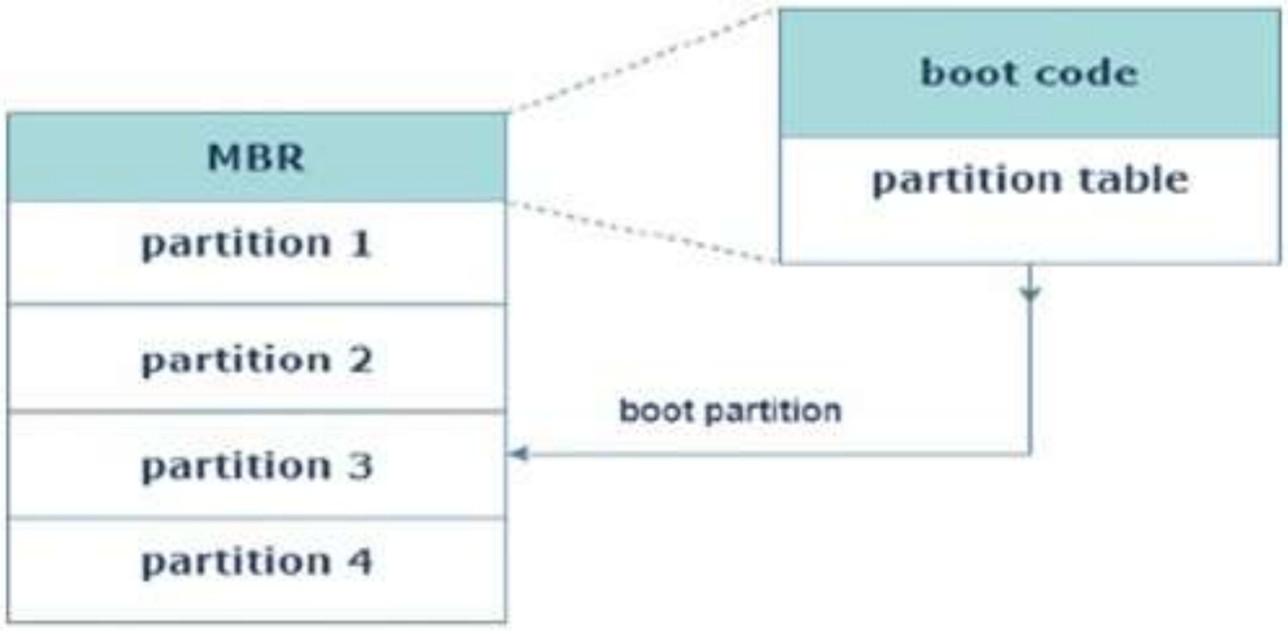
- A new magnetic disk is mainly a blank slate. It is platters of the magnetic recording material.
- Before a disk may hold data, it must be partitioned into sectors that may be read and written by the disk controller.
- It is known as **physical formatting** and **low-level formatting.**
- **Low-level formatting** creates a unique data structure for every sector on the drive.
- A data structure for a sector is made up of a header, a data region, and a trailer.
- The disk controller uses the header and trailer to store information like an error-correcting code (ECC) and a sector number.

- The OS must require recording its own data structures on the disk drive to utilize it as a storage medium for files.
- **It accomplishes this in two phases.**
- The initial step is to divide the disk drive into one or more cylinder groups.
- The OS may treat every partition as it were a separate disk. For example, one partition could contain a copy of the OS executable code, while another could contain user files.
- The second stage after partitioning is logical formatting. The os stores the initial file system data structure on the disk drive in this second stage.

Boot Block

- The boot block process is the sequence of events that occur when a computer starts up.
- It starts with the power-on of the computer, which triggers the boot block to start the boot loader.
- The **bootstrap loader** reads the boot block, which contains information about the location of the operating system and other necessary files. It then loads these files into memory and starts the operating system.
- The operating system, in turn, performs various initialization tasks and prepares the computer for use. This includes loading drivers, setting up system settings, and starting user applications.

- The boot block process is a critical component of the computer's start-up process and must complete successfully for the computer to startup and run smoothly.
- Any errors or problems during the boot block process can prevent the computer from starting or cause it to start with errors.



Bad Block

- A bad block, also known as a bad sector, is a sector on a hard disk that is unable to store or retrieve data due to physical or logical damage.
- A bad block can be caused by a variety of factors, including disk wear and tear, power surges, software errors, and other factors.
- Bad blocks can result in data loss and can negatively impact the performance of the hard disk.
- In some cases, they can also cause the hard disk to fail. To detect and repair bad blocks, disk management tools may perform a disk scan, which locates and identifies bad blocks on the hard disk.

- Some disk management tools may also attempt to repair bad blocks by remapping them to a spare sector on the disk.
- It is important to regularly check for and repair bad blocks as part of disk management best practices.
- Failure to address bad blocks can result in further data loss and can increase the likelihood of hard disk failure.

Logical structure

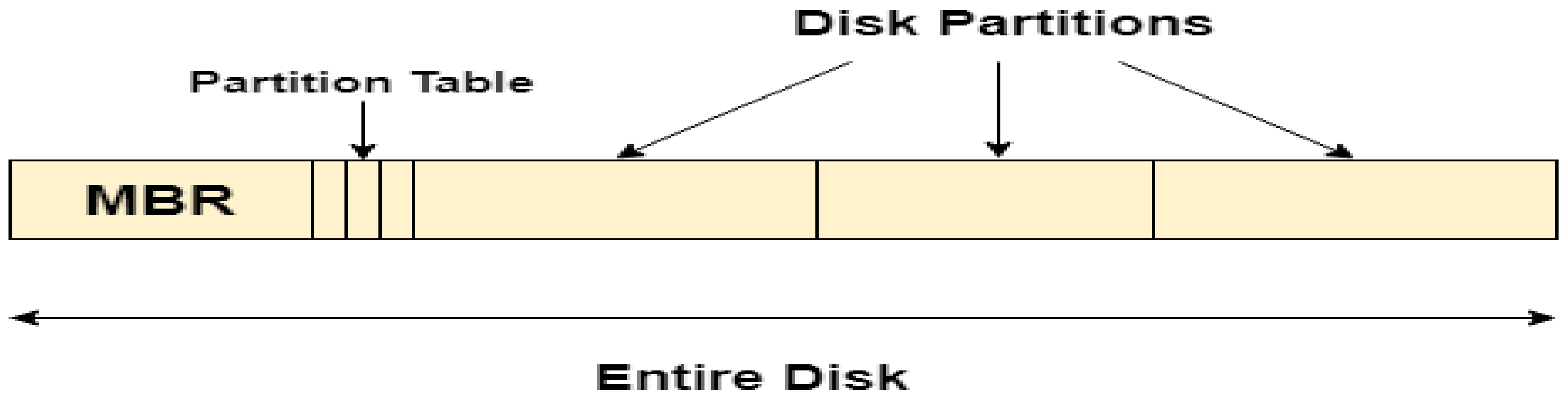
Logical division of hard disk are :

- MBR (Master Boot Record)
- FAT (File Allocation Tables)
- Root Directory
- Data Area

Master Boot Record (MBR)

- Master boot record is the information present in the first sector of any hard disk. It contains the information regarding how and where the Operating system is located in the hard disk so that it can be booted in the RAM.
- MBR is sometimes called master partition table because it includes a partition table which locates every partition in the hard disk.
- Master boot record (MBR) also includes a program which reads the boot sector record of the partition that contains operating system.

Master Boot Record (MBR)



Data Area

The data area contains the **sync bytes, user data and an error-correcting code (ECC)** that is used to check and possibly correct errors that may have been introduced into the data.

Disk Scheduling

Disk Scheduling Algorithm

- Disk scheduling is an important process in operating systems that determines the order in which disk access requests are serviced.
- The objective of disc scheduling is to minimize the time it takes to access data on the disk and to minimize the time it takes to complete a disk access request.
- Disk scheduling algorithms are an essential component of modern operating systems and are responsible for determining the order in which disk access requests are serviced.
- The primary goal of these algorithms is to minimize disk access time and improve overall system performance.

Seek Time

Seek time is the time taken in locating the disk arm to a specified track where the read/write request will be satisfied.

Rotational Latency

It is the time taken by the desired sector to rotate itself to the position from where it can access the R/W heads.

Transfer Time

It is the time taken to transfer the data.

Disk Access Time

Disk access time is given as,

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

Disk Response Time

It is the average of time spent by each request waiting for the IO operation.

Purpose of Disk Scheduling

The main purpose of disk scheduling algorithm is to select a disk request from the queue of IO requests and decide the schedule when this request will be processed.

Goal of Disk Scheduling Algorithm

- Fairness
- High throughout
- Minimal traveling head time

Types of Disk scheduling Algorithms

- FCFS scheduling algorithm
- SSTF (shortest seek time first) algorithm
- SCAN scheduling
- C-SCAN scheduling
- LOOK Scheduling
- C-LOOK scheduling

FCFS scheduling algorithm

- The First-Come-First-Served (FCFS) disk scheduling algorithm is one of the simplest and most straightforward disk scheduling algorithms used in modern operating systems.
- It operates on the principle of servicing disk access requests in the order in which they are received.
- In the FCFS algorithm, the disk head is positioned at the first request in the queue and the request is serviced.
- The disk head then moves to the next request in the queue and services that request.
- This process continues until all requests have been serviced.

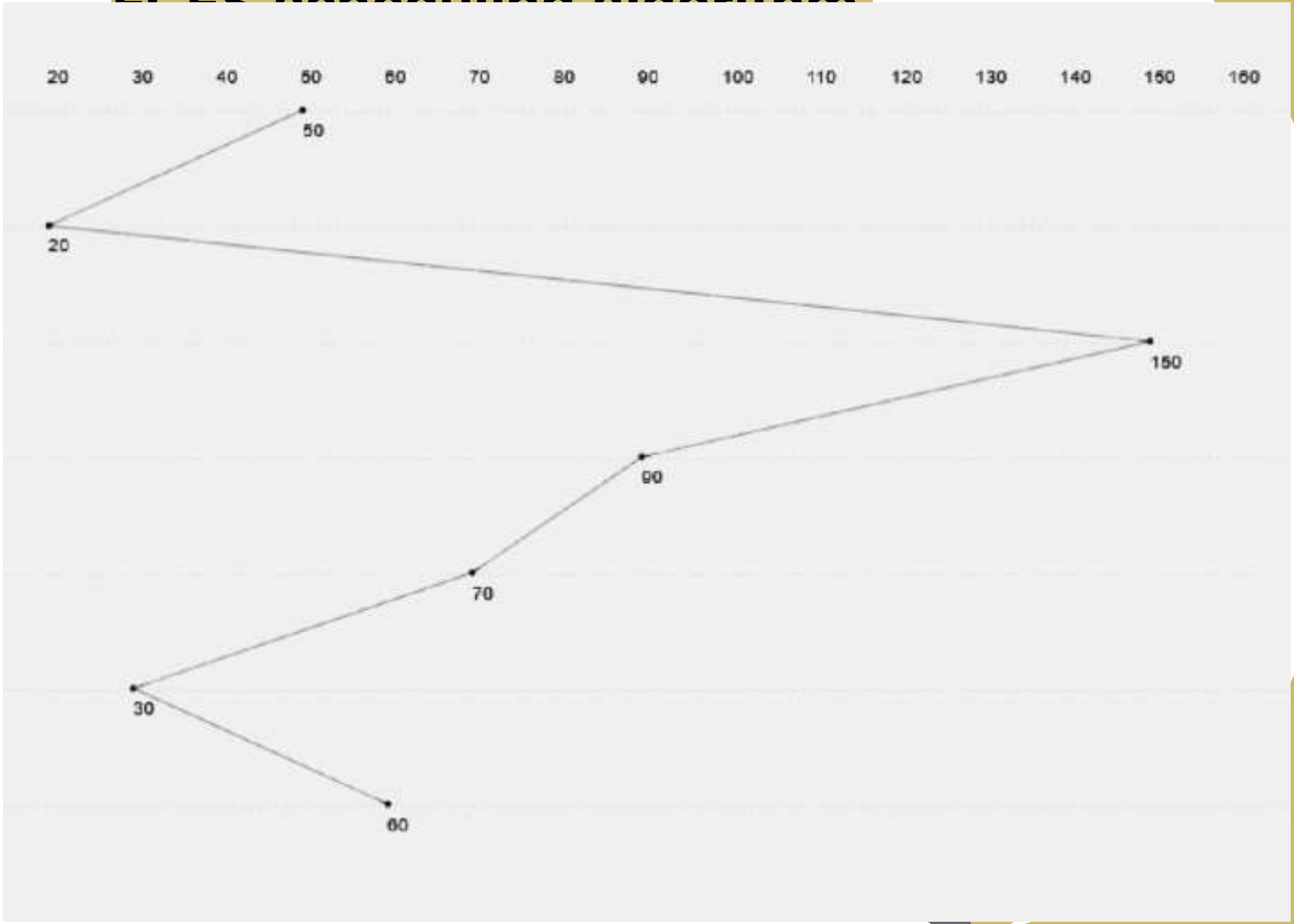
- **Example : Problem**

Suppose we have an order of disk access requests:

20 150 90 70 30 60. The disk header initially at track

50. Find the total seek time?

FCFS scheduling algorithm



- **Example : Solution**

  Disk access requests: 20 150 90 70 30 60.

  Disk header initially at track : 50

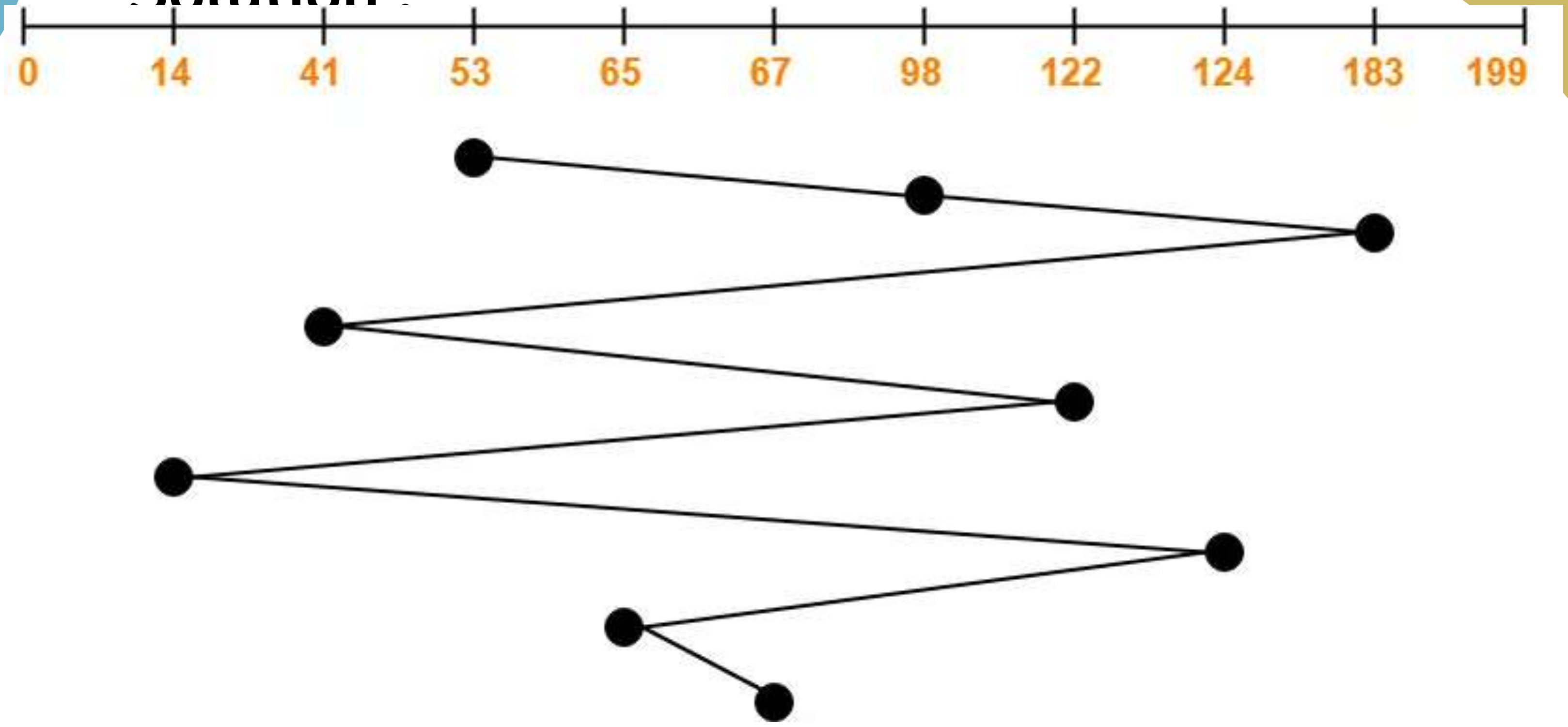
  The total seek time = (50-20) + (150-20) + (150-90) + (90-70) + (70-30) + (60-30) = **310**

- **Disadvantages**
  - The scheme does not optimize the seek time.
  - The request may come from different processes therefore there is the possibility of inappropriate movement of the head.

Example :

Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65, 67. The FCFS scheduling algorithm is used. The head is initially at cylinder number 53. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is _______.

Solution :



Total head movements incurred while servicing these requests

= (98 – 53) + (183 – 98) + (183 – 41) + (122 – 41) + (122 – 14) + (124 – 14) + (124 – 65) + (67 – 65)

= 45 + 85 + 142 + 81 + 108 + 110 + 59 + 2

= 632

Shortest Seek Time First (SSTF)

- The algorithm selects the disk I/O request which requires the least disk arm movement from its current position regardless of the direction.
- It reduces the total seek time as compared to FCFS.
- It allows the head to move to the closest track in the service queue.

Disadvantages

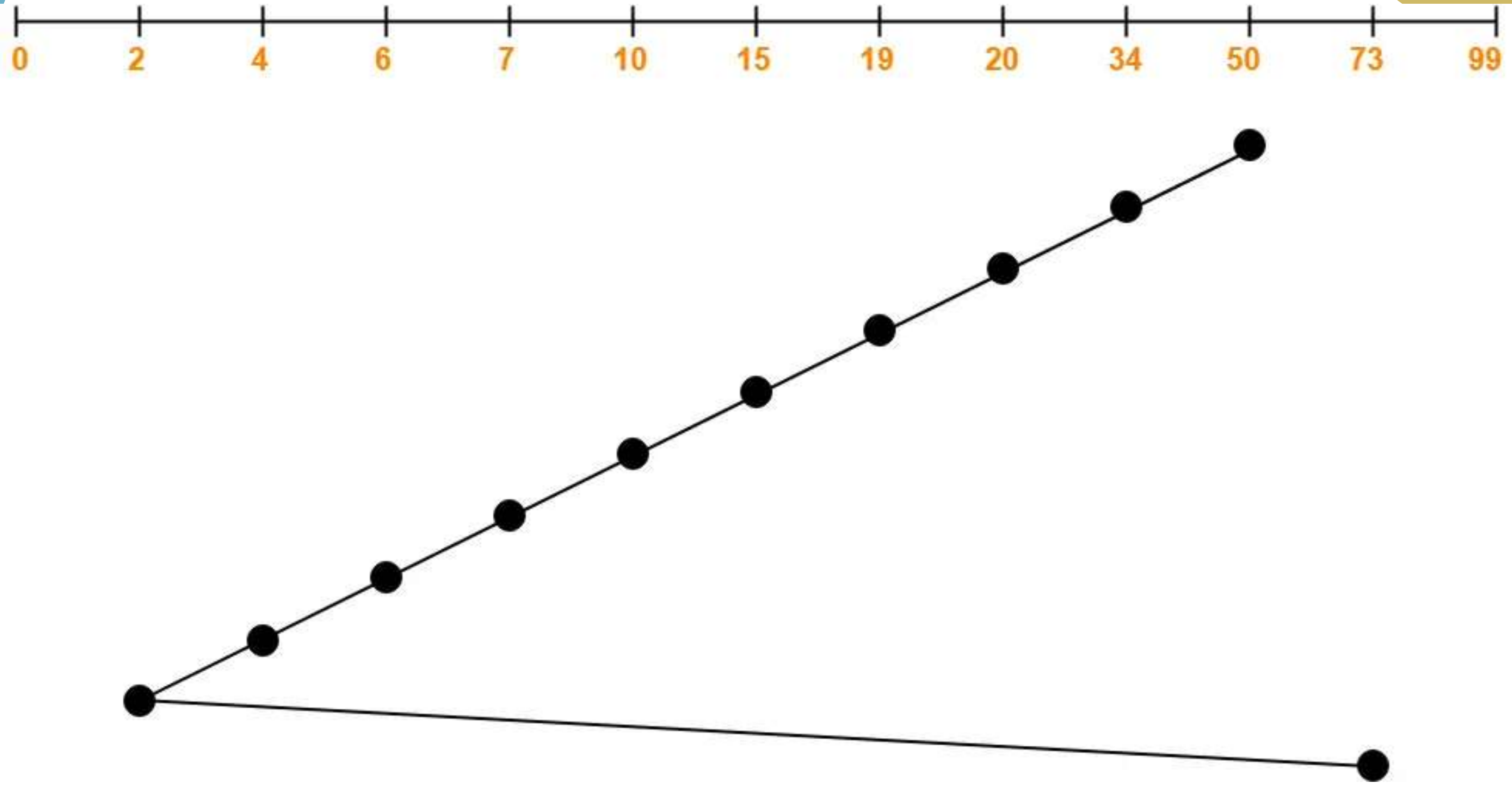
- It may cause starvation for some requests.
- There is an overhead of finding out the closest request.
- Switching direction on the frequent basis slows the working of algorithm.
- It is not the most optimal algorithm.

Example

Consider a disk system with 100 cylinders. The requests to access the cylinders occur in following sequence-

4, 34, 10, 7, 19, 73, 2, 15, 6, 20

Assuming that the head is currently at cylinder 50, what is the time taken to satisfy all requests if it takes 1 ms to move from one cylinder to adjacent one and shortest seek time first policy is used?

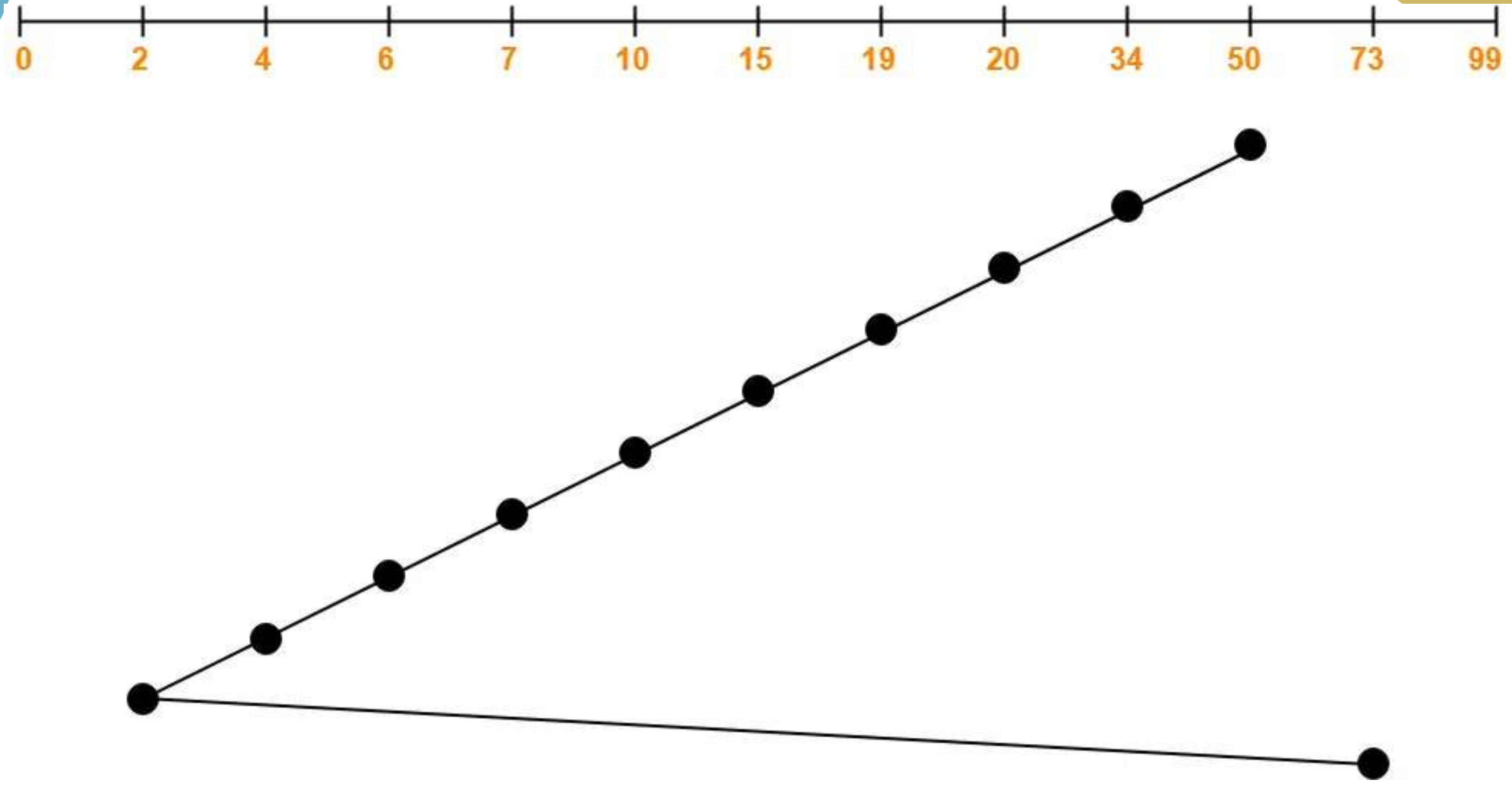


Total head movements incurred while servicing these requests

= (50 – 34) + (34 – 20) + (20 – 19) + (19 – 15) + (15 – 10) + (10 – 7) + (7 – 6) + (6 – 4) + (4 – 2) + (73 – 2)

= 16 + 14 + 1 + 4 + 5 + 3 + 1 + 2 + 2 + 71

= 119



Time taken for one head movement = 1 msec. So,

Time taken for 119 head movements

= 119 x 1 msec

= 119 msec

Scan Algorithm

- It is also called as Elevator Algorithm.
- In this algorithm, the disk arm moves into a particular direction till the end, satisfying all the requests coming in its path,and then it turns back and moves in the reverse direction satisfying requests coming in its path.
- It works in the way an elevator works, elevator moves in a direction completely till the last floor of that direction and then turns back.

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.

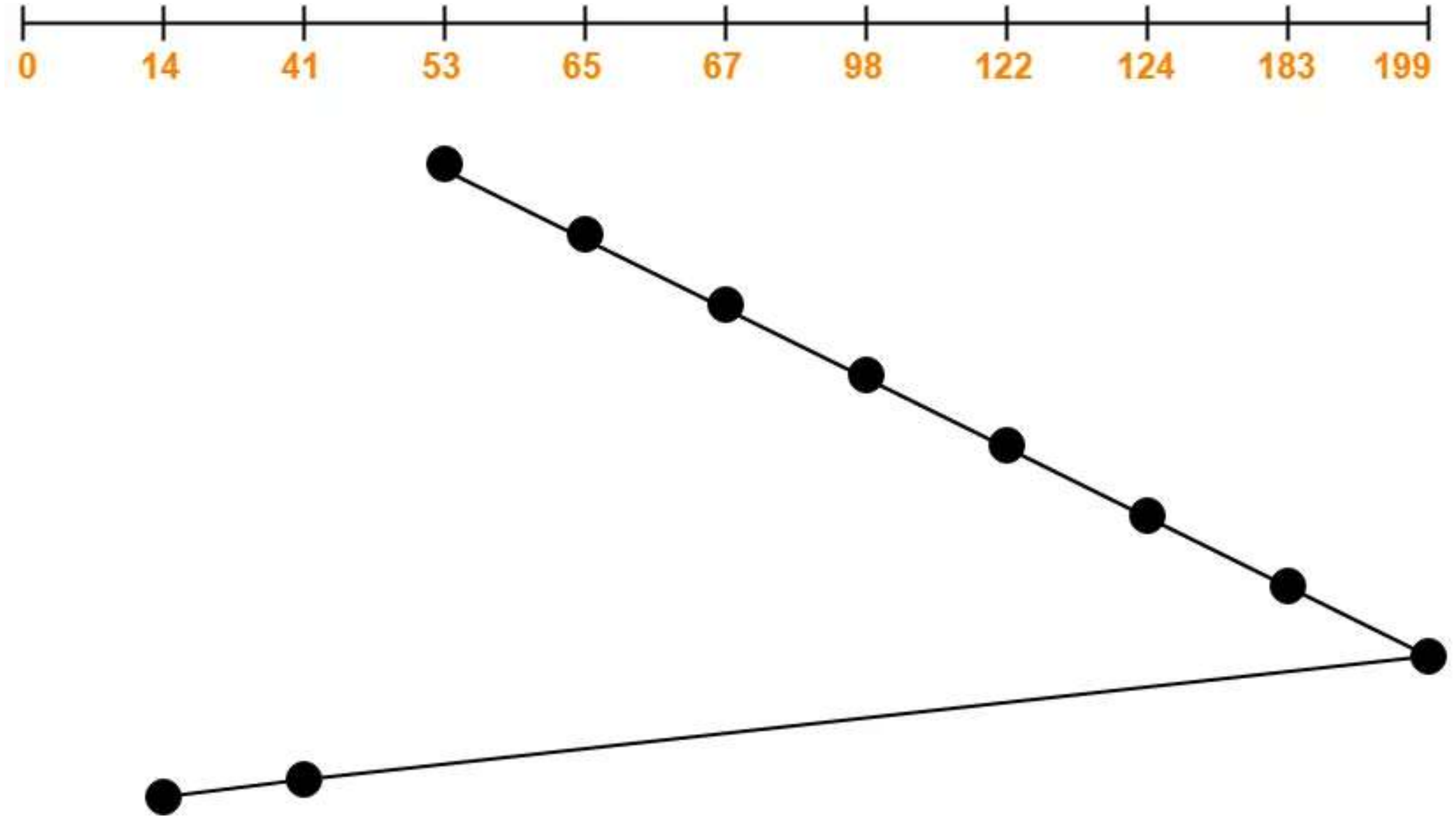
Disadvantages

- 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

Problem :

Consider a disk queue with requests for I/O to blocks on cylinders **98, 183, 41, 122, 14, 124, 65, 67**. The SCAN scheduling algorithm is used. The head is initially at cylinder number **53 moving towards larger cylinder** numbers on its servicing pass. The cylinders are numbered from **0 to 199**. The total head movement (in number of cylinders) incurred while servicing these requests is

Solution



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) + (199 – 183) + (199 – 41) + (41 – 14)

= 12 + 2 + 31 + 24 + 2 + 59 + 16 + 158 + 27

= 331

Alternatively,

Total head movements incurred while servicing these requests

= (199 – 53) + (199 – 14)

= 146 + 185

= 331

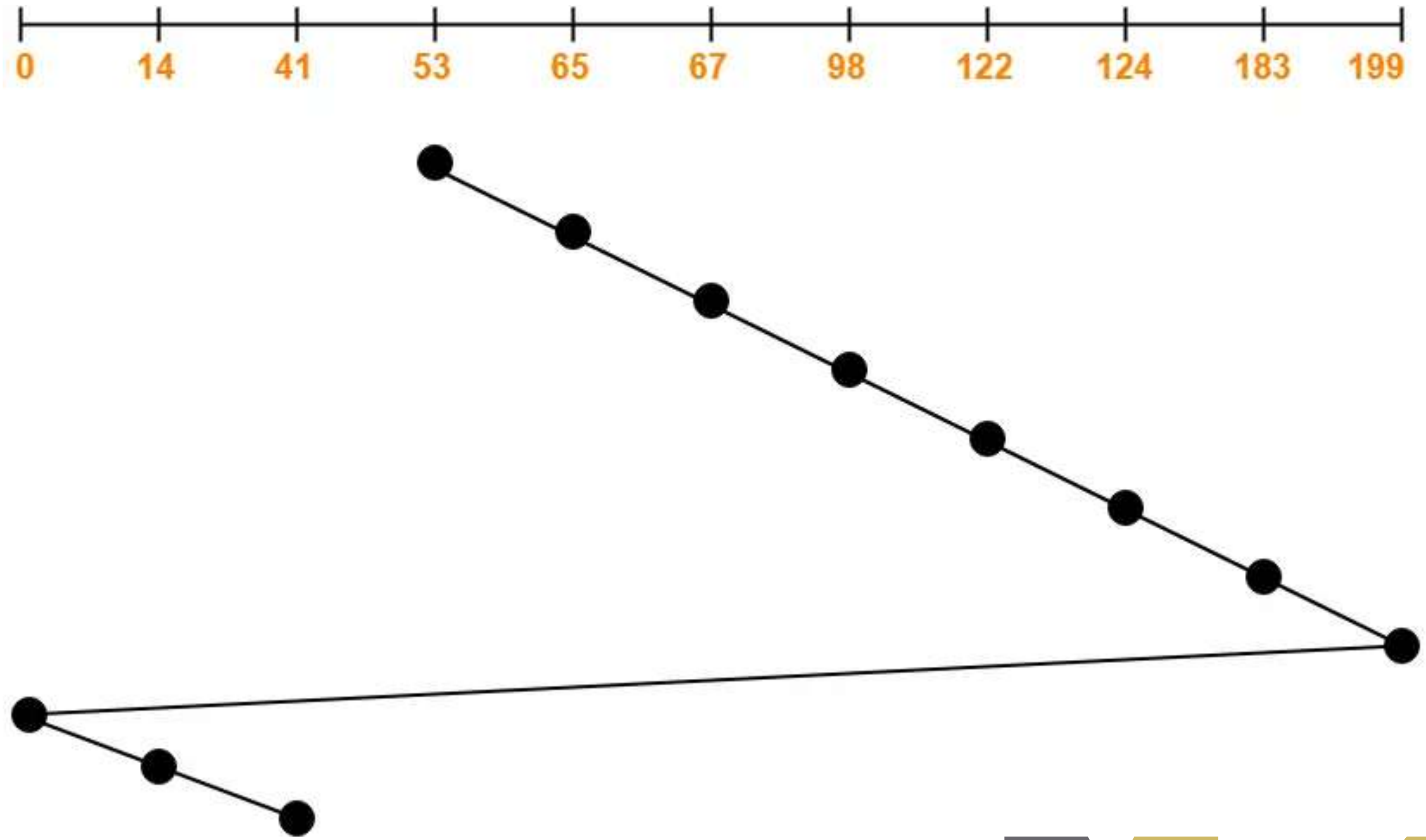
C-SCAN algorithm

- 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.

Problem:

Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65, 67. The C-SCAN scheduling algorithm is used. The head is initially at cylinder number 53 moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is ________.

Solution :



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) + (199 – 183) + (199 – 0) + (14 – 0) + (41 – 14)

= 12 + 2 + 31 + 24 + 2 + 59 + 16 + 199 + 14 + 27

= 386

Alternatively,

Total head movements incurred while servicing these requests

= (199 – 53) + (199 – 0) + (41 – 0)

= 146 + 199 + 41

= 386

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.

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.

LOOK Algorithm

- It is is an improved version of the SCAN Algorithm.
- Head starts from the first request at one end of the disk and moves towards the last request at the other end servicing all the requests in between.
- After reaching the last request at the other end, head reverses its direction.
- It then returns to the first request at the starting end servicing all the requests in between.
- The same process repeats.

NOTE-

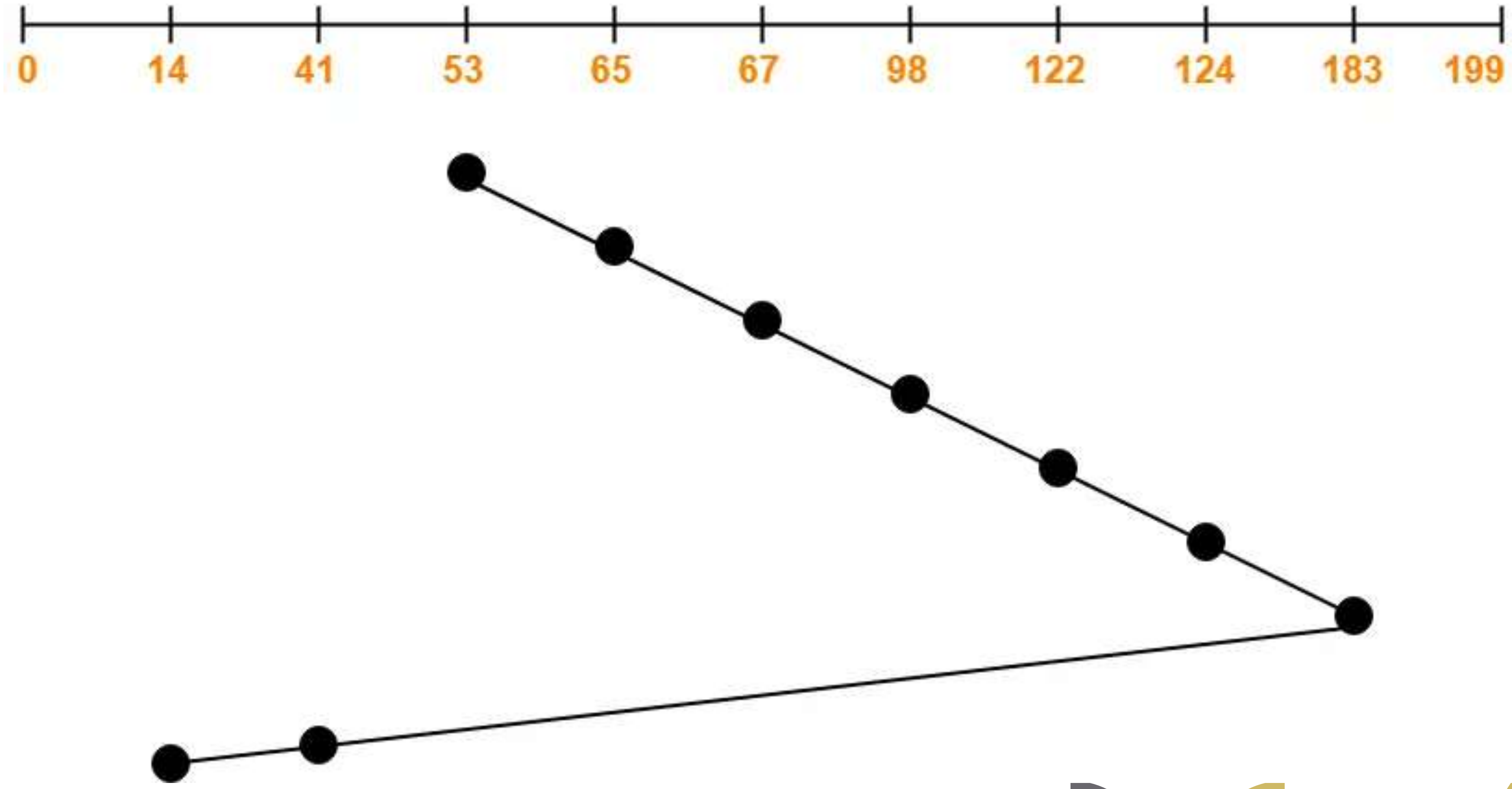
The main difference between SCAN Algorithm and LOOK Algorithm is-

- SCAN Algorithm scans all the cylinders of the disk starting from one end to the other end even if there are no requests at the ends.
- LOOK Algorithm scans all the cylinders of the disk starting from the first request at one end to the last request at the other end.

Problem:

Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65, 67. The LOOK scheduling algorithm is used. The head is initially at cylinder number 53 moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is _______.

Solution :



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) + (183 – 41) + (41 – 14)

= 12 + 2 + 31 + 24 + 2 + 59 + 142 + 27

= 299

Alternatively,

Total head movements incurred while servicing these requests

= (183 – 53) + (183 – 14)

= 130 + 169

= 299

Advantages

- It does not causes the head to move till the ends of the disk when there are no requests to be serviced.
- It provides better performance as compared to SCAN Algorithm.
- It does not lead to starvation.
- It provides low variance in response time and waiting time.

Disadvantages

- There is an overhead of finding the end requests.
- It causes long waiting time for the cylinders just visited by the head.

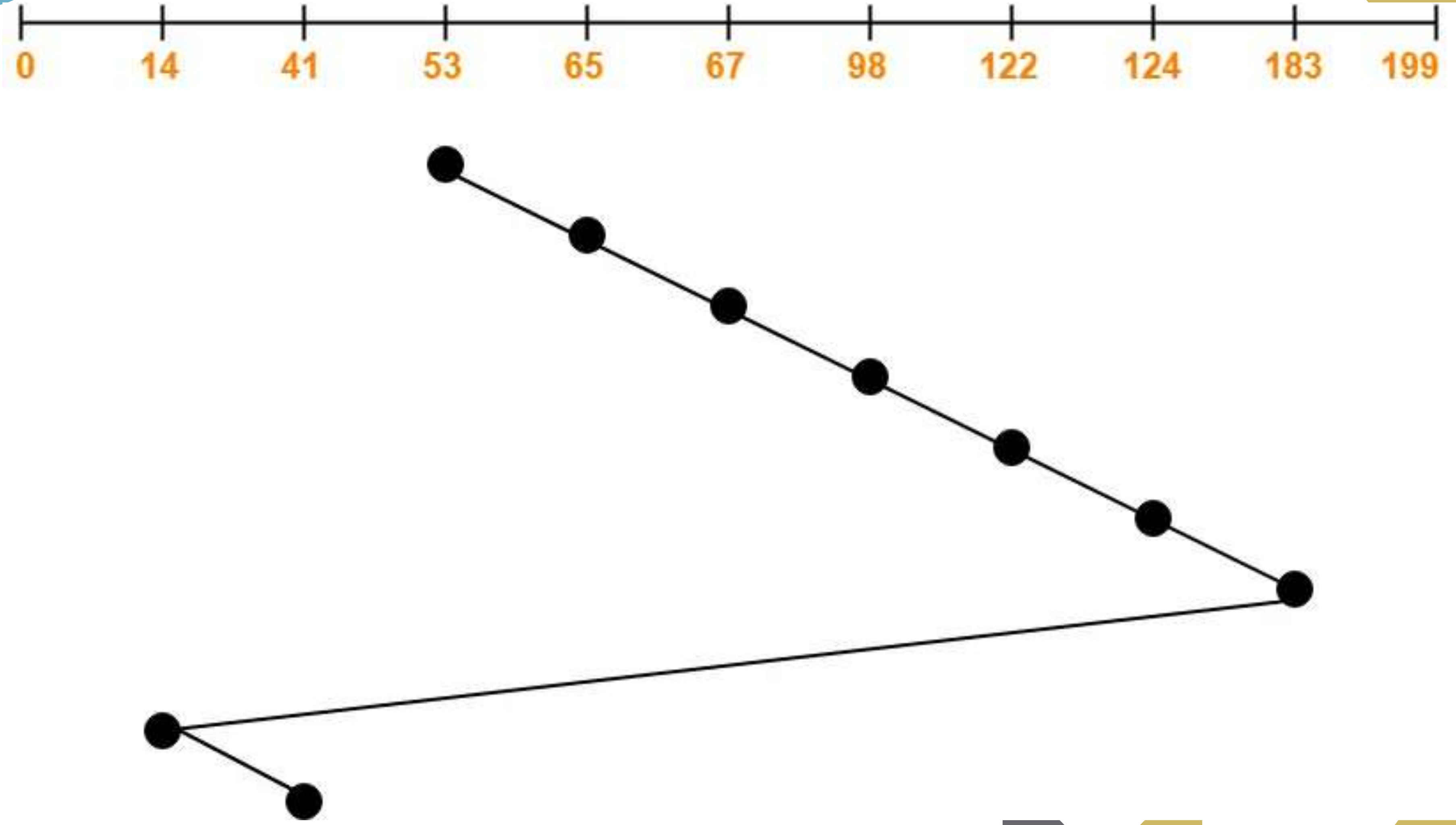
C-LOOK Algorithm

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

Problem :

Consider a disk queue with requests for I/O to blocks on cylinders 98, 183, 41, 122, 14, 124, 65, 67. The C-LOOK scheduling algorithm is used. The head is initially at cylinder number 53 moving towards larger cylinder numbers on its servicing pass. The cylinders are numbered from 0 to 199. The total head movement (in number of cylinders) incurred while servicing these requests is _______.

Solution:



Total head movements incurred while servicing these requests

= (65 – 53) + (67 – 65) + (98 – 67) + (122 – 98) + (124 – 122) + (183 – 124) + (183 – 14) + (41 – 14)

= 12 + 2 + 31 + 24 + 2 + 59 + 169 + 27

= 326

Alternatively,

Total head movements incurred while servicing these requests

= (183 – 53) + (183 – 14) + (41 – 14)

= 130 + 169 + 27

= 326

Advantages

- It does not causes the head to move till the ends of the disk when there are no requests to be serviced.
- It reduces the waiting time for the cylinders just visited by the head.
- It provides better performance as compared to LOOK Algorithm.
- It does not lead to starvation.
- It provides low variance in response time and waiting time.

Disadvantages-

- There is an overhead of finding the end requests.

Disk Reliability

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).

- Disk reliability refers to the measure of how dependable and fault-tolerant a storage device, such as a hard disk drive (HDD) or solid-state drive (SSD), is in preserving data integrity and availability over time.
- It encompasses various aspects of a disk's performance, including Mean Time Between Failures (MTBF), error correction mechanisms, bad block management, power failure protection, and monitoring capabilities.
- A reliable disk is essential to ensure that data is safely stored and can be accessed without encountering data loss or corruption.

What is RAID?

RAID (redundant array of independent disks) is a way of storing the same data in different places on multiple hard disks or solid-state drives (SSDs) to protect data in the case of a drive failure. There are different RAID levels, however, and not all have the goal of providing redundancy.

How RAID works

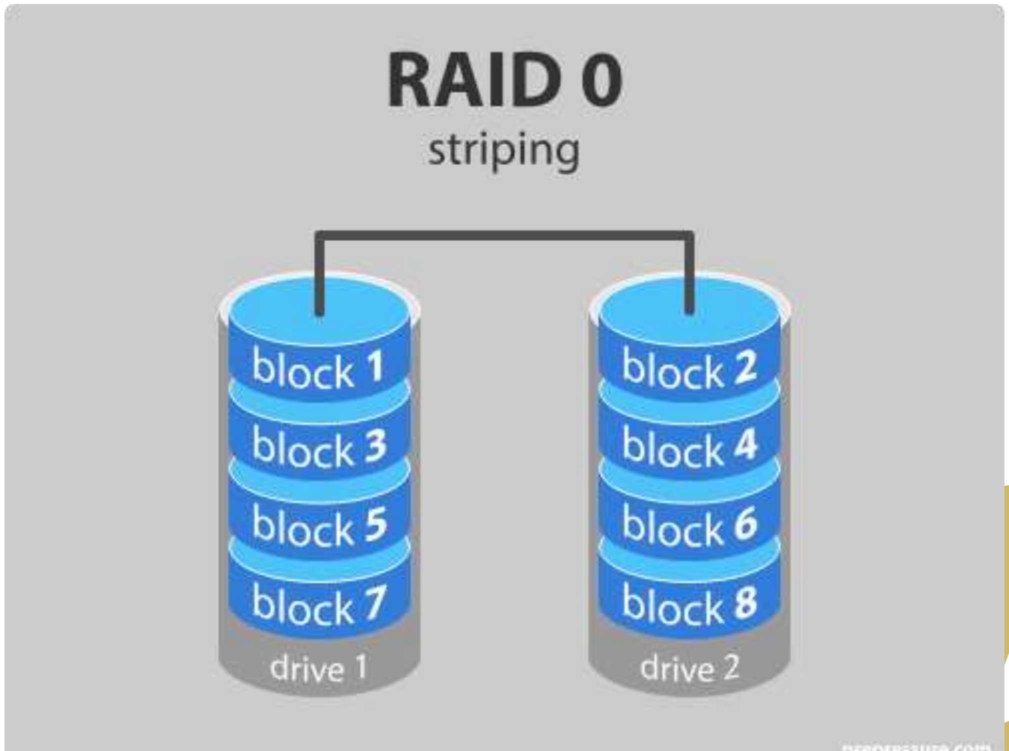
RAID works by placing data on multiple disks and allowing input/output (I/O) operations to overlap in a balanced way, improving performance. Because using multiple disks increases the mean time between failures, storing data redundantly also increases fault tolerance.

- RAID arrays appear to the operating system (OS) as a single logical drive.
- RAID employs the techniques of disk mirroring or disk striping.
- Mirroring will copy identical data onto more than one drive.
- Striping partitions help spread data over multiple disk drives.
- Each drive's storage space is divided into units ranging from a sector of 512 bytes up to several megabytes.
- The stripes of all the disks are interleaved and addressed in order.
- Disk mirroring and disk striping can also be combined in a RAID array.

RAID Levels

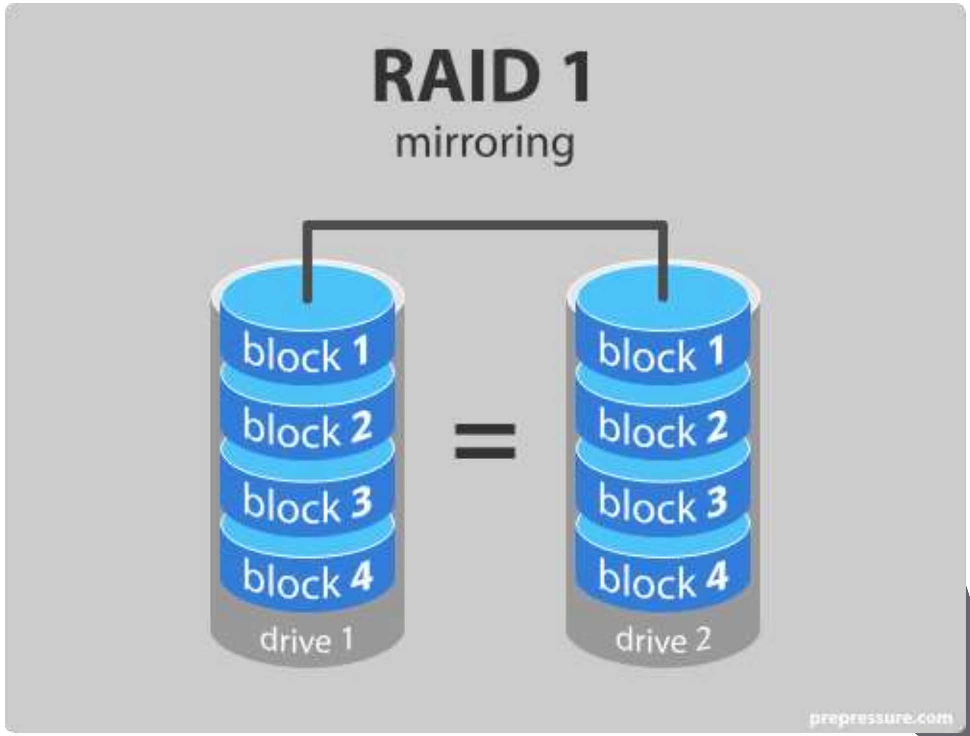
RAID 0

- Provides **data striping** across multiple drives to enhance read/write performance.
- However, it does not offer any data redundancy, meaning a single drive failure can result in complete data loss.



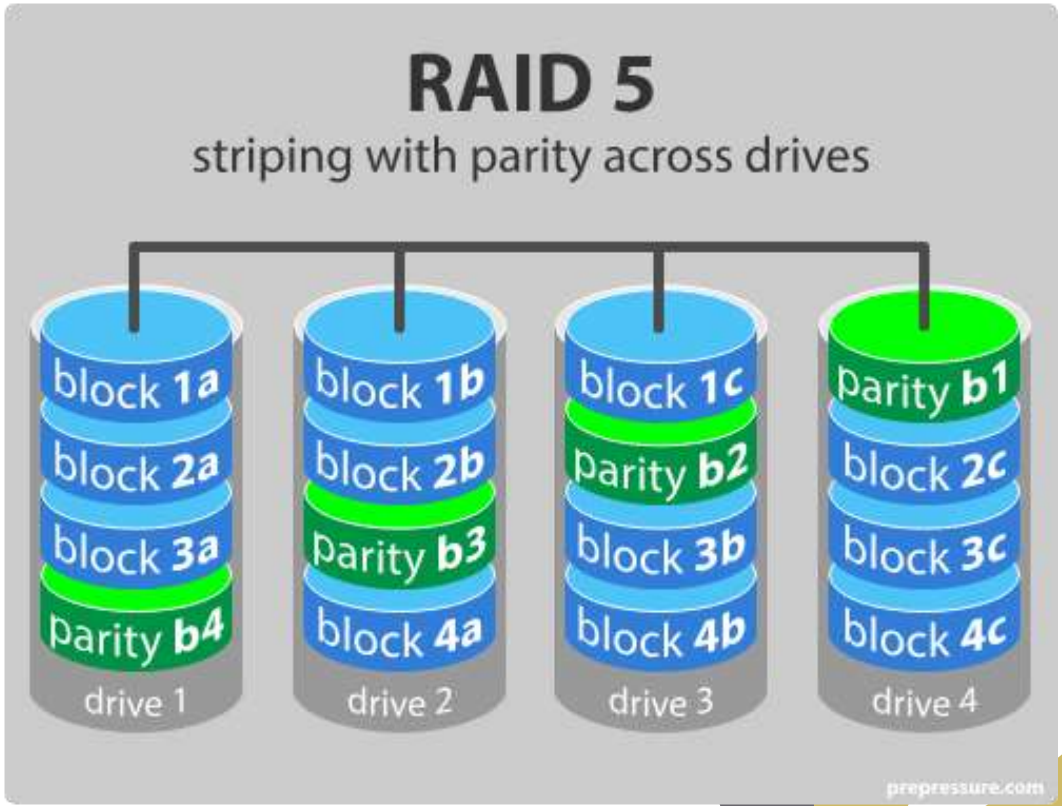
RAID 1

- Uses **data mirroring**, where each disk has an exact copy of the data.
- RAID 1 offers excellent data redundancy, but it reduces the total usable storage capacity to 50% because one drive is a mirror of the other.



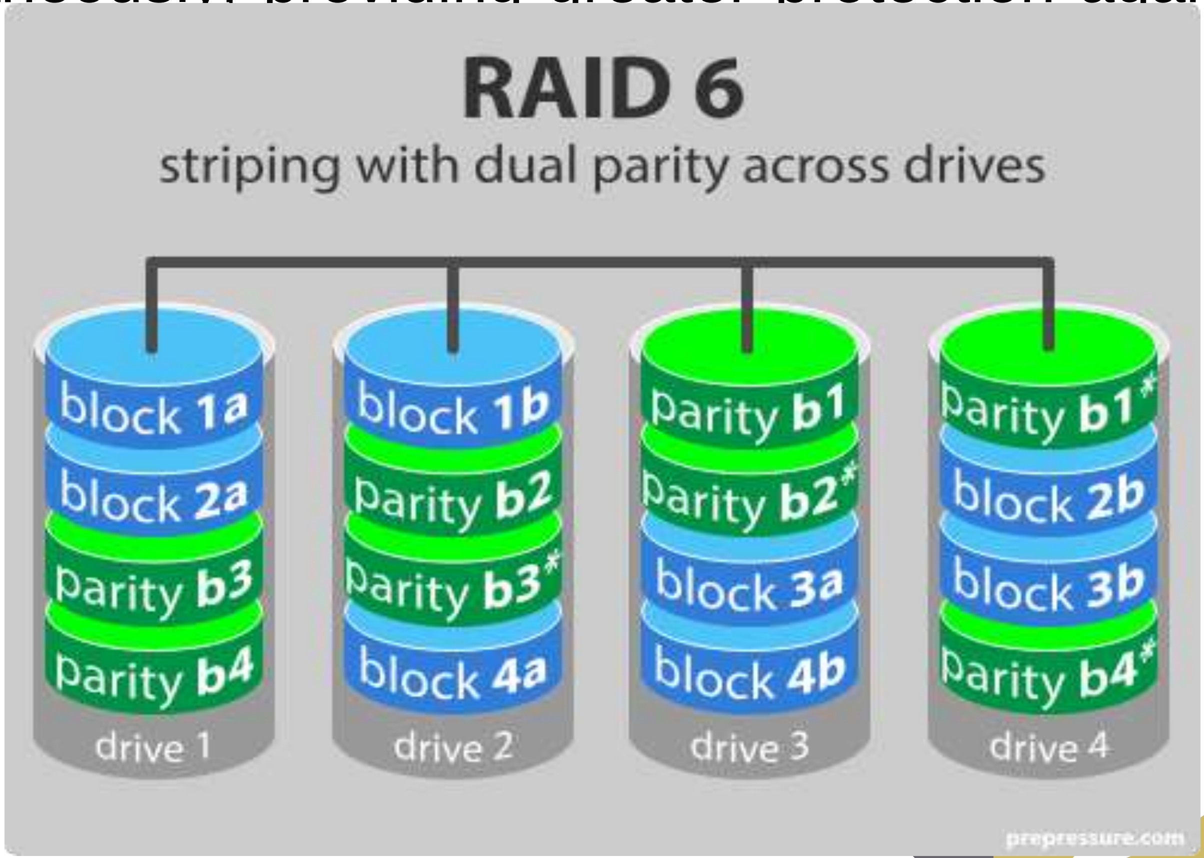
RAID 5

- Combines data **striping and parity,** distributing data and parity information across multiple drives.
- RAID 5 requires at least three drives and offers a good balance of performance and data redundancy.
- If one drive fails, the lost data can be rebuilt using the parity information.



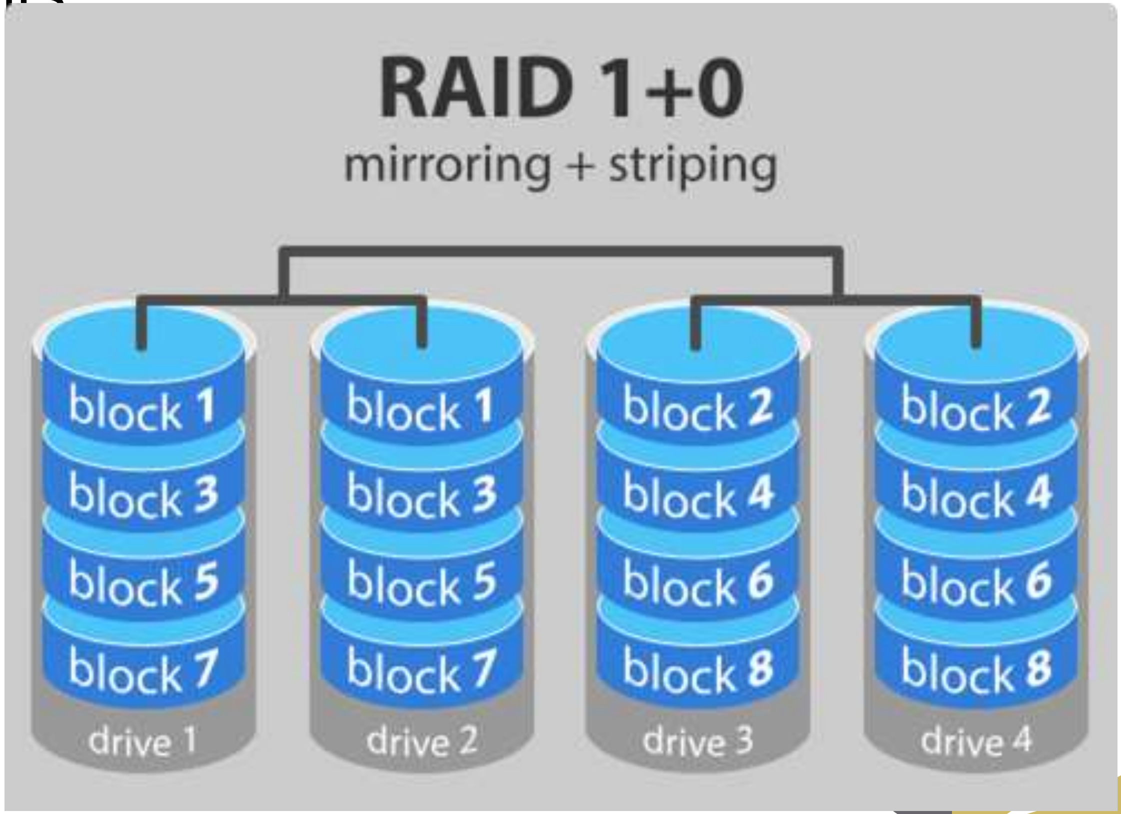
RAID 6

- Similar to **RAID 5**, but **with dual parity** for enhanced fault tolerance.
- RAID 6 can withstand the failure of two drives simultaneously, providing greater protection against data loss.



RAID 10 (or RAID 1+0)

- **Combines mirroring (RAID 1) and striping (RAID 0).** RAID 10 requires at least four drives and offers excellent data redundancy and performance.
- It can survive multiple drive failures within certain constraints.



- **RAID 0** – striping
- **RAID 1** – mirroring
- **RAID 5** – striping with parity
- **RAID 6** – striping with double parity
- **RAID 10** – combining mirroring and striping

Advantages of RAID 0

- RAID 0 offers great performance, both in read and write operations. There is no overhead caused by parity controls.
- All storage capacity is used, there is no overhead.
- The technology is easy to implement.

Disadvantages of RAID 0

- RAID 0 is not fault-tolerant. If one drive fails, all data in the RAID 0 array are lost. It should not be used for mission-critical systems.

Advantages of RAID 1

- RAID 1 offers excellent read speed and a write-speed that is comparable to that of a single drive.
- In case a drive fails, data do not have to be rebuild, they just have to be copied to the replacement drive.
- RAID 1 is a very simple technology.

Disadvantages of RAID 1

- The main disadvantage is that the effective storage capacity is only half of the total drive capacity because all data get written twice.
- Software RAID 1 solutions do not always allow a hot swap of a failed drive. That means the failed drive can only be replaced after powering down the computer it is attached to. For servers that are used simultaneously by many people, this may not be acceptable. Such systems typically use hardware controllers that do support hot swapping.

Advantages of RAID 5

- Read data transactions are very fast while write data transactions are somewhat slower (due to the parity that has to be calculated).
- If a drive fails, you still have access to all data, even while the failed drive is being replaced and the storage controller rebuilds the data on the new drive.

Disadvantages of RAID 5

- Drive failures have an effect on throughput, although this is still acceptable.
- This is complex technology. If one of the disks in an array using 4TB disks fails and is replaced, restoring the data (the rebuild time) may take a day or longer, depending on the load on the array and the speed of the controller. If another disk goes bad during that time, data are lost forever.

Advantages of RAID 6

- Like with RAID 5, read data transactions are very fast.
- If two drives fail, you still have access to all data, even while the failed drives are being replaced. So RAID 6 is more secure than RAID 5.

Disadvantages of RAID 6

- Write data transactions are slower than RAID 5 due to the additional parity data that have to be calculated. In one report I read the write performance was 20% lower.
- Drive failures have an effect on throughput, although this is still acceptable.
- This is complex technology. Rebuilding an array in which one drive failed can take a long time.

Advantages of RAID 10

- If something goes wrong with one of the disks in a RAID 10 configuration, the rebuild time is very fast since all that is needed is copying all the data from the surviving mirror to a new drive. This can take as little as 30 minutes for drives of 1 TB.

Disadvantages of RAID 10

- Half of the storage capacity goes to mirroring, so compared to large RAID 5 or RAID 6 arrays, this is an expensive way to have redundancy.