Unit 6 :
File Management and Disk Structure

Marwadi University

Computer Engineering
Diploma

Unit 6
File Management and
Disk Structure

Operating System
09CE2405

Prof. Foram Chovatiya

File

- **“Files are logical units of information”** created by processes.
- **“A file is a unit of storing data on a secondary storage device such as a hard disk or other external media”**.
- Information stored in files must be persistent(Continue), that is, not be affected by process creation and termination.
- Files are managed by the operating system.
- How they are structured, named, accessed, used, protected, implemented, and managed are major topics in operating system design.

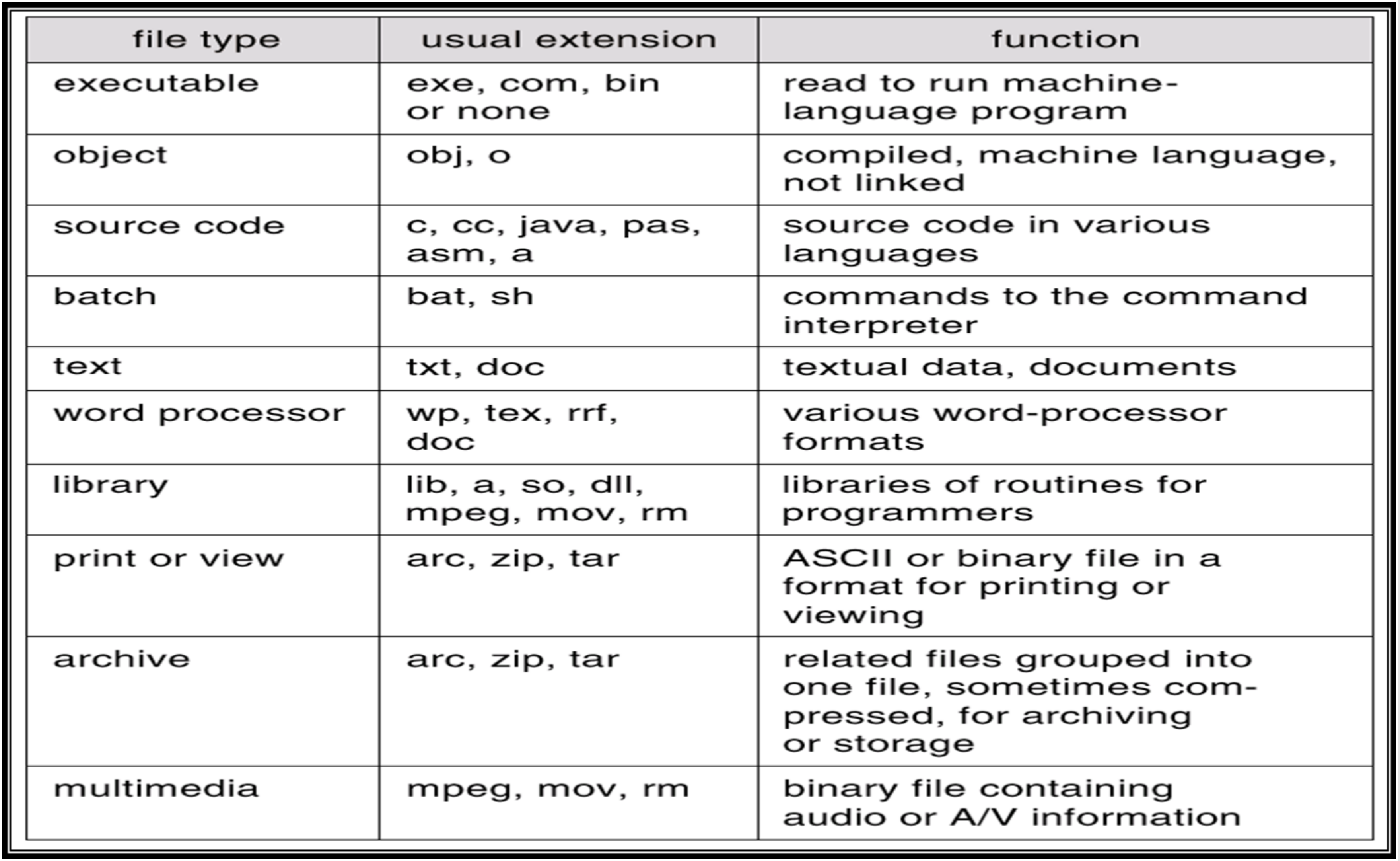
File

- As a whole, **“the part of the operating system dealing with files is known as the file system”**.

Directory:

- Collection of files is a file directory.
- The directory contains information about the files, including attributes, location and ownership.

File types names and extensions



file type	usual extension	function
executable	exe, com, bin or none	read to run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, 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
library	lib, a, so, dll, mpeg, mov, rm	libraries of routines for programmers
print or view	arc, zip, tar	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm	binary file containing audio or A/V information

Attributes of File

1) **Name** :- It is the only information which is in human-readable form.

2) **Identifier** :- The file is identified by a unique tag(number) within file system.

3) **Type** :- It is needed for systems that support different types of files.

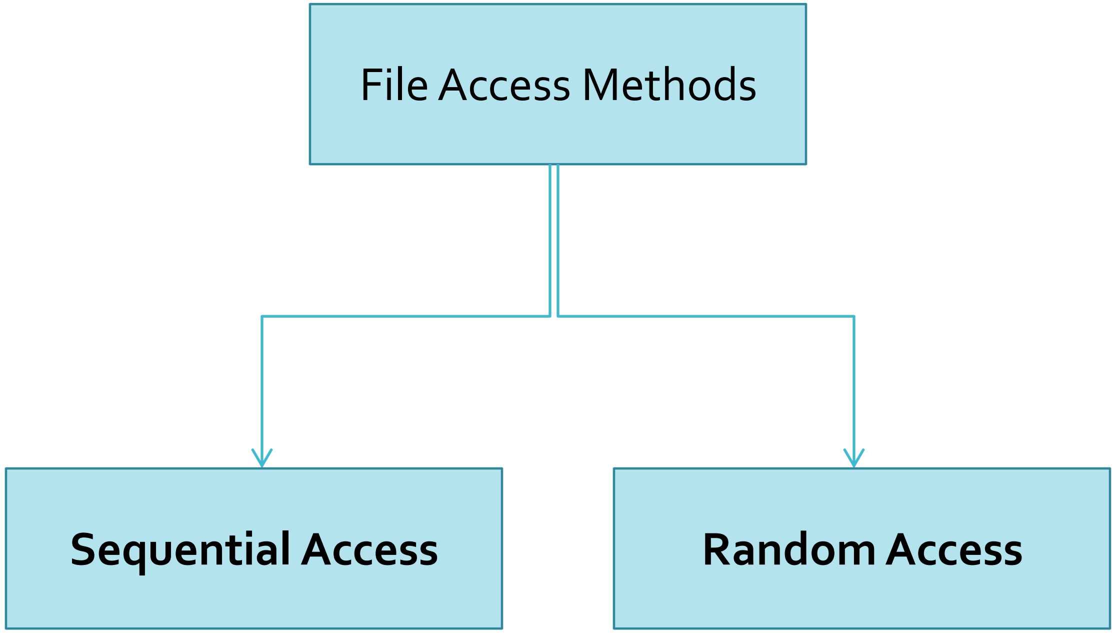
4) **Location** :- Pointer to file location on device.

5) **Size** :- The current size of the file.

6) **Protection** :- This controls and assigns the power of reading, writing, executing.

7) **Time, date, and user identification** :- This is the data for protection, security, and usage monitoring.

FILE access methods


File Access Methods

Sequential Access

Random Access

File Access Methods

1) Sequential File Access:-

- In Sequential access, process could read all the bytes or records from a file in order, starting from the beginning, but could not skip around and read them out of order.
- Sequential files could be rewound.
- These files were convenient when the storage medium was magnetic tape or CD-ROM.

File Access Methods

2) Random File Access:-

- Files whose bytes or records can be read **in any order** are called random access files.
- Random access files are essentials for many applications, for example, data base systems.
- If an airline customer calls up and wants to reserve a seat on a particular flight, the reservation program must be able to access the record for that flight without having to read the records for thousands of other flights.

File Operations

- File exists to store information and allow it to be retrieved later.
- Different system provides different operations to allow storage and retrieval.
- The most common system calls are shown below.

File Operations

- Create
- Delete
- Open
- Close
- Read
- Write
- Append
- Rename

Directory Structure

- To keep track of files, file systems normally have directories or folders.
- **Directories** are system files for maintaining the structure of the file system.

1. Single Level Directory
2. Two Level Directory
3. Hierarchical Directory

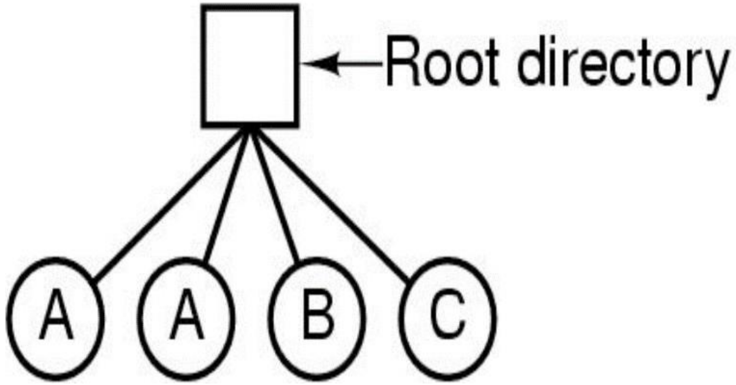
1. Single Level Directory:-

Single Level Directory

- Single level directory is simplest directory structure.
- In which all files are contained in same directory which make it easy to support and understand.
- A single level directory has a significant limitation, however, when the number of files increases or when the system has more than one user.
- Since all the files are in the same directory, they must have the unique name.
- if two users call their dataset test, then the unique name rule violated.

Single Level Directory

- The simplest form of directory system is having one directory containing all the files.
- This is some time called as root directory.
- The advantages of this scheme are its simplicity and the ability to locate files quickly there is only one directory to look, after all.



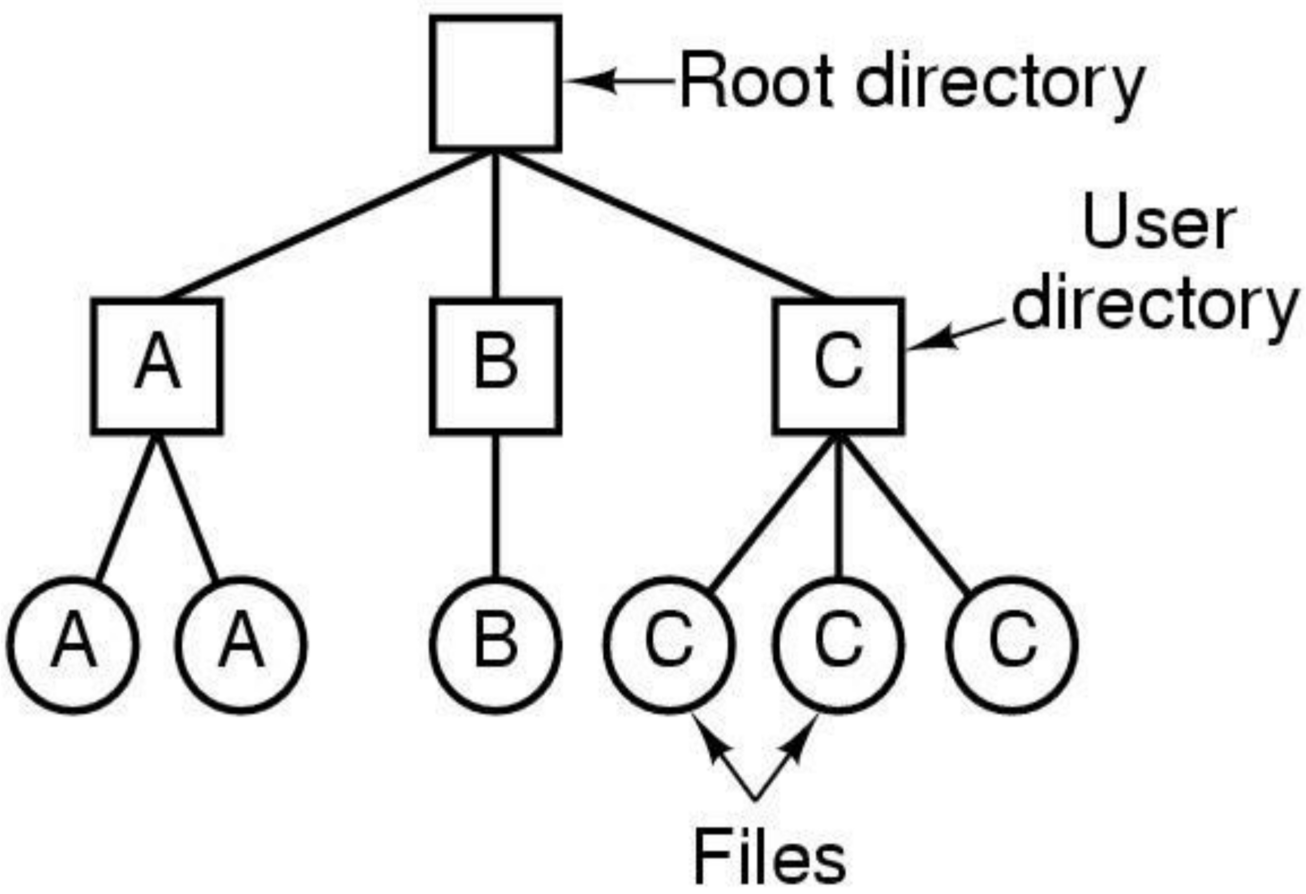
- Devices such as digital cameras and some portable music players use this structure.

2. Two Level Directory:-

Two Level Directory

- In the two-level directory structure, each user has there own user files directory (UFD).
- The UFDs has similar structures, but each lists only the files of a single user.
- Path name: Due to two levels there is a path name for every file to locate that file.
- We can have same file name for different user.
- Searching is efficient in this method

Two Level Directory

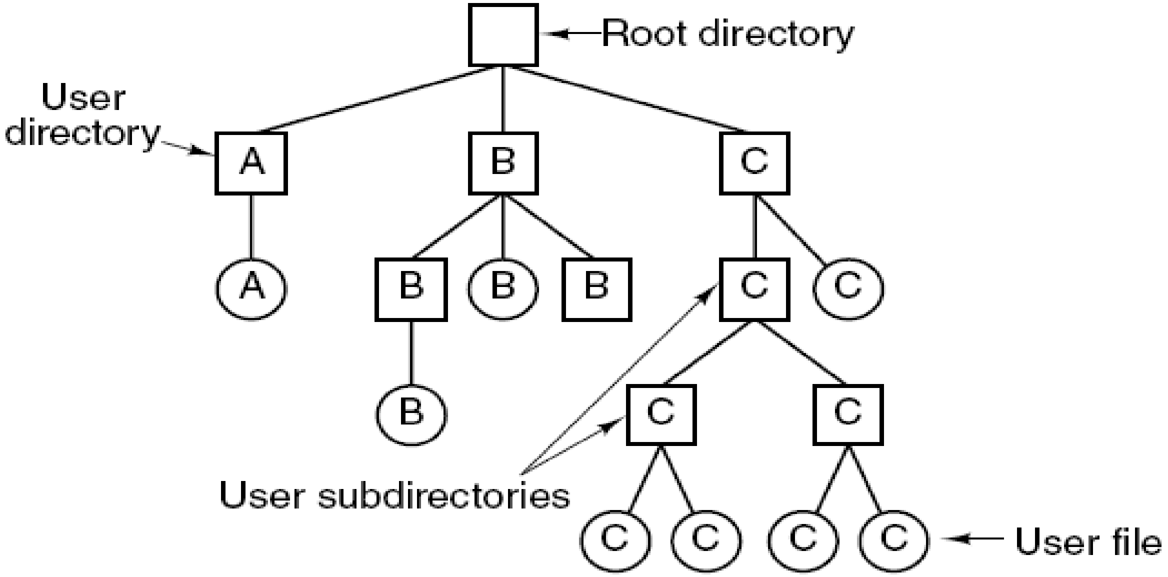


3. Hierarchical Level Directory:-

Hierarchical Directory

Also known as a TREE-STRUCTURED DIRECTORY

- Directory is maintained in the form of a tree.
- Generalization allows the user to create there own subdirectories and to organize on their files accordingly.
- Searching is efficient and also there is grouping capability.



Disk Management

Cylinder:

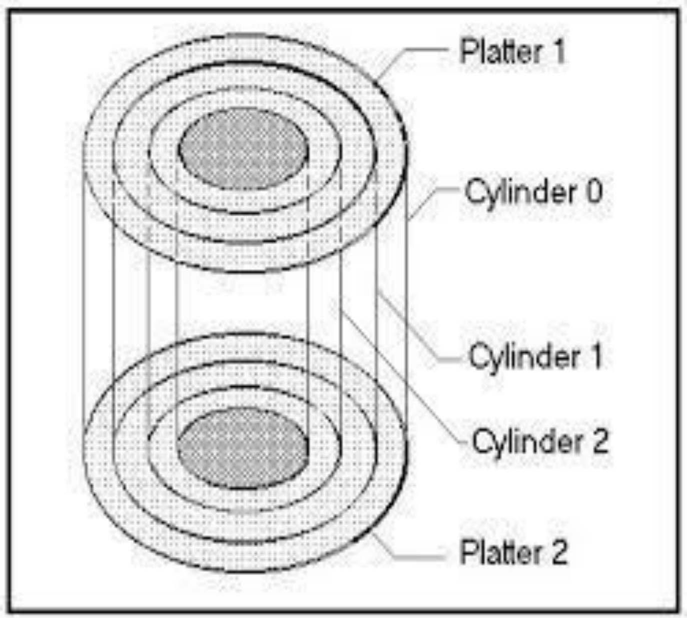
- A *cylinder* is any set of all of *tracks* of equal diameter in a hard disk drive (HDD).
- It can be visualized as a single, imaginary, circle that cuts through all of the *platters* (and both sides of each platter) in the drive.

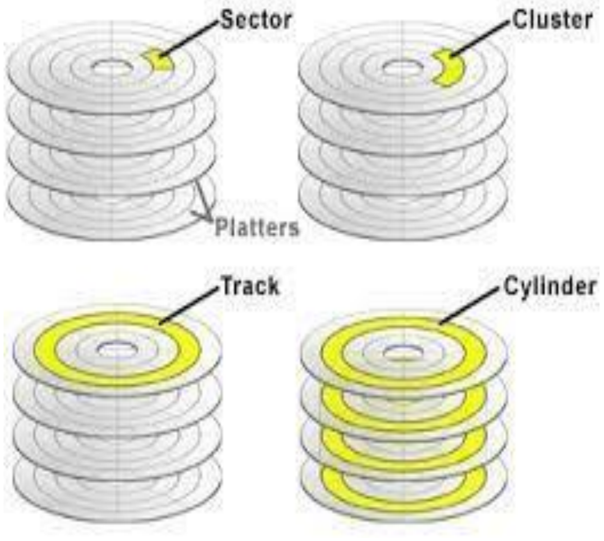
Head:

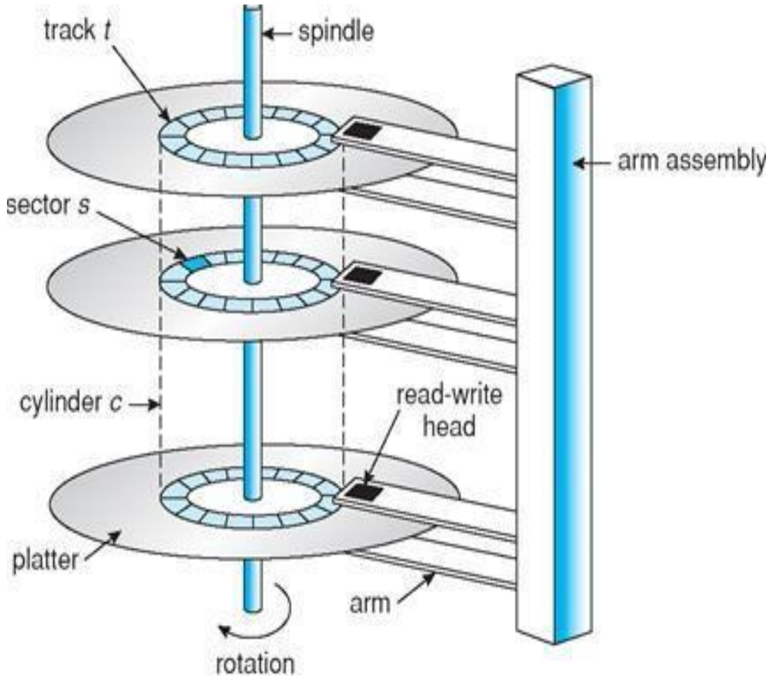
- Every hard drive consists of platters and read-write heads.
- Each platter surface has it's own dedicated read-write head.
- If a drive has four platters, it usually has eight read-write heads, one on the top and other on the bottom of each platter.

Disk Management

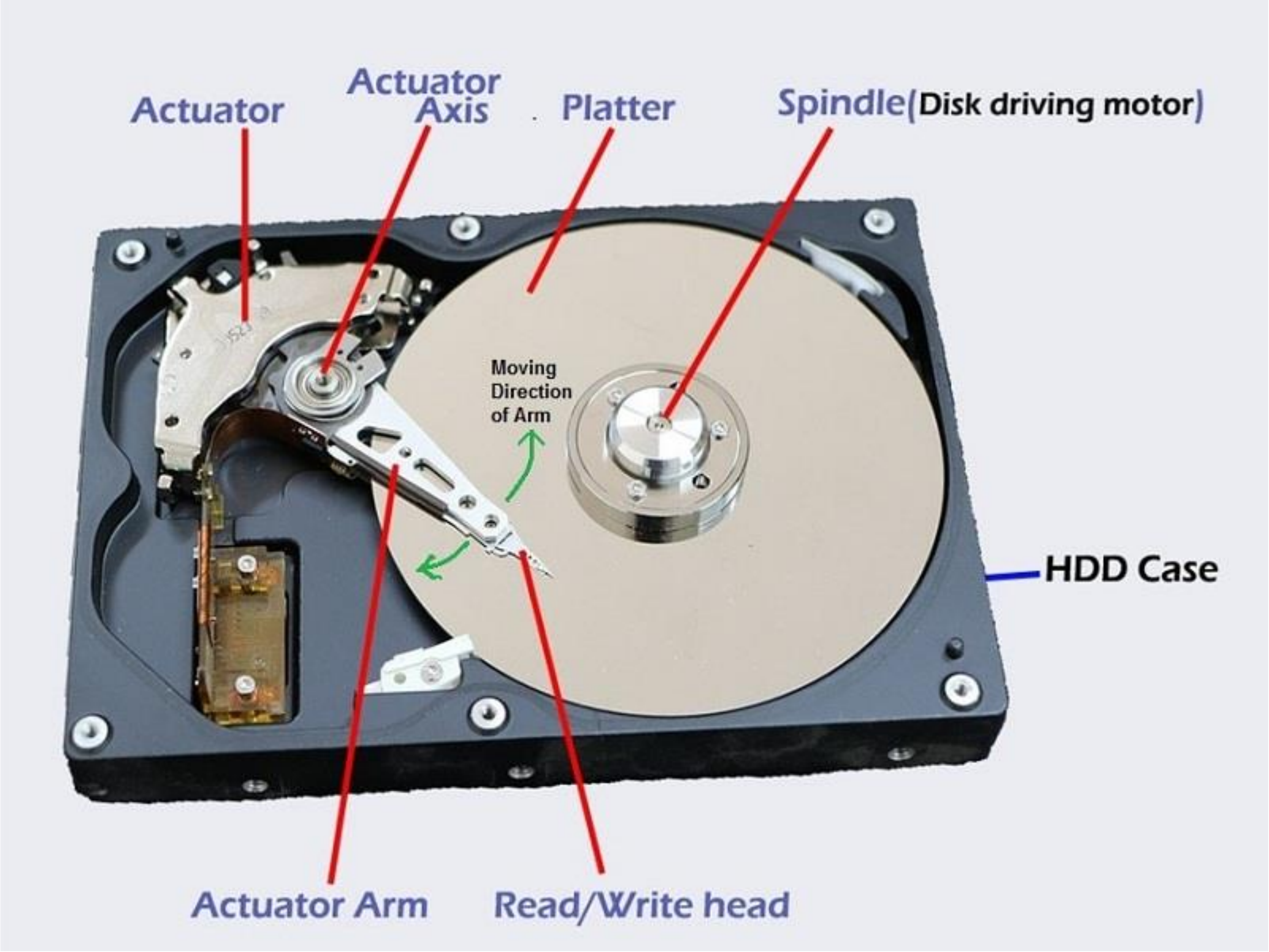
Hard Disk Structure







Disk Management



Disk Management

Sector :

- Each track is further broken down into smaller units called sectors. As sector is the basic unit of data storage on a hard disk.

Cluster:

- Sectors are often grouped together to form Clusters.

Disk Performance Parameters

(1) 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.

(2) Rotational delay or rotational latency:

- How long it takes the hard drive to move to the sector where the requested data is located.

(3) Access time:

- seek time + rotational delay
- The time it takes to get in position to read or write

Disk Arm Scheduling Algorithms

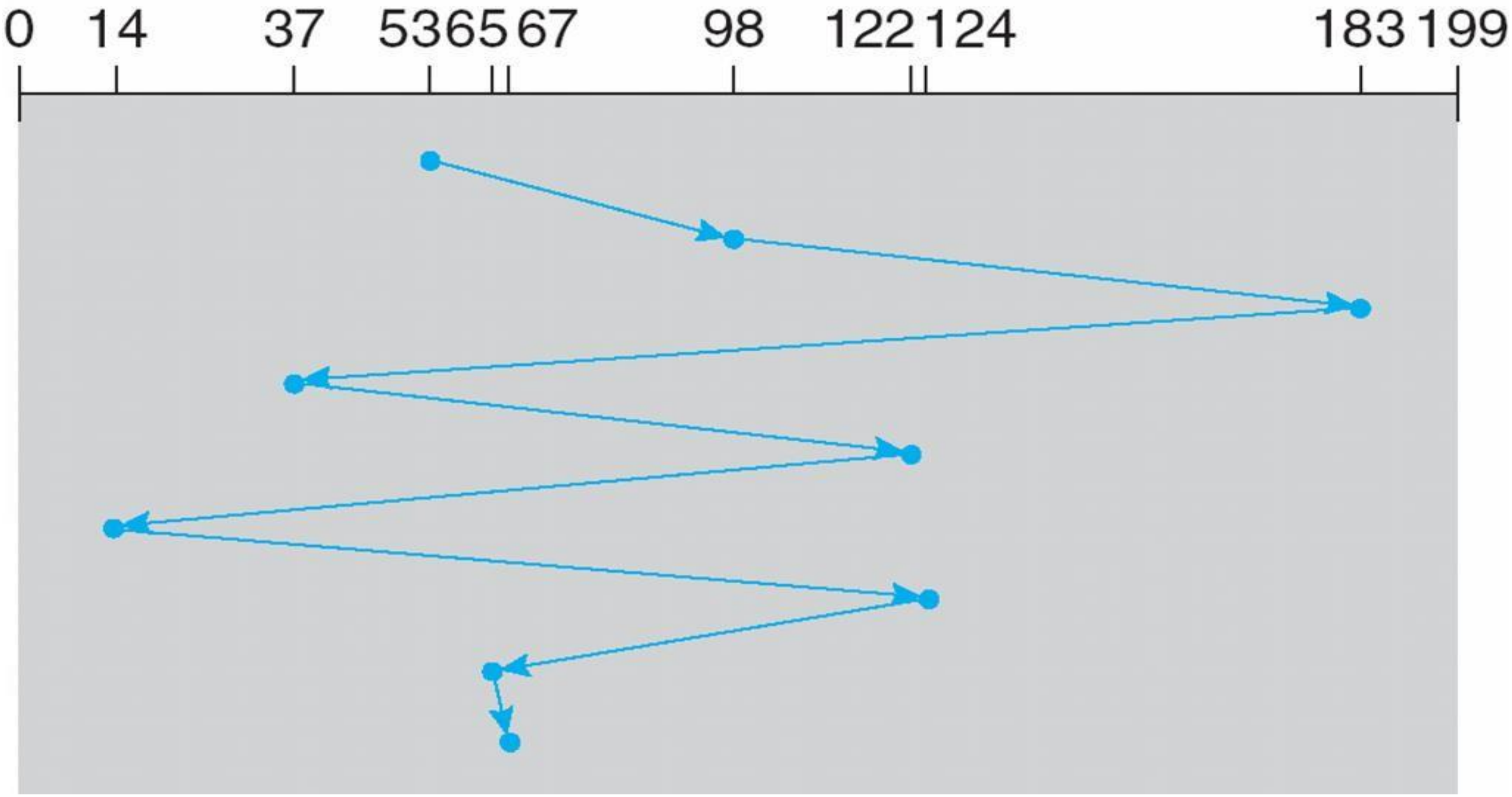
- FCFS(FIFO)
- SSTF (Shortest Seek Time First)
- SCAN
- CIRCULAR SCAN(C-SCAN)
- LOOK
- C-LOOK

FIFO

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.

FIFO

queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53



FIFO

- Total Distance that disk arm moved

= |53 - 98| + |98 – 183| + |183 – 37| + |37 – 122| + |122 – 14| + |14 – 124| + |124 – 65| + |65 – 67|

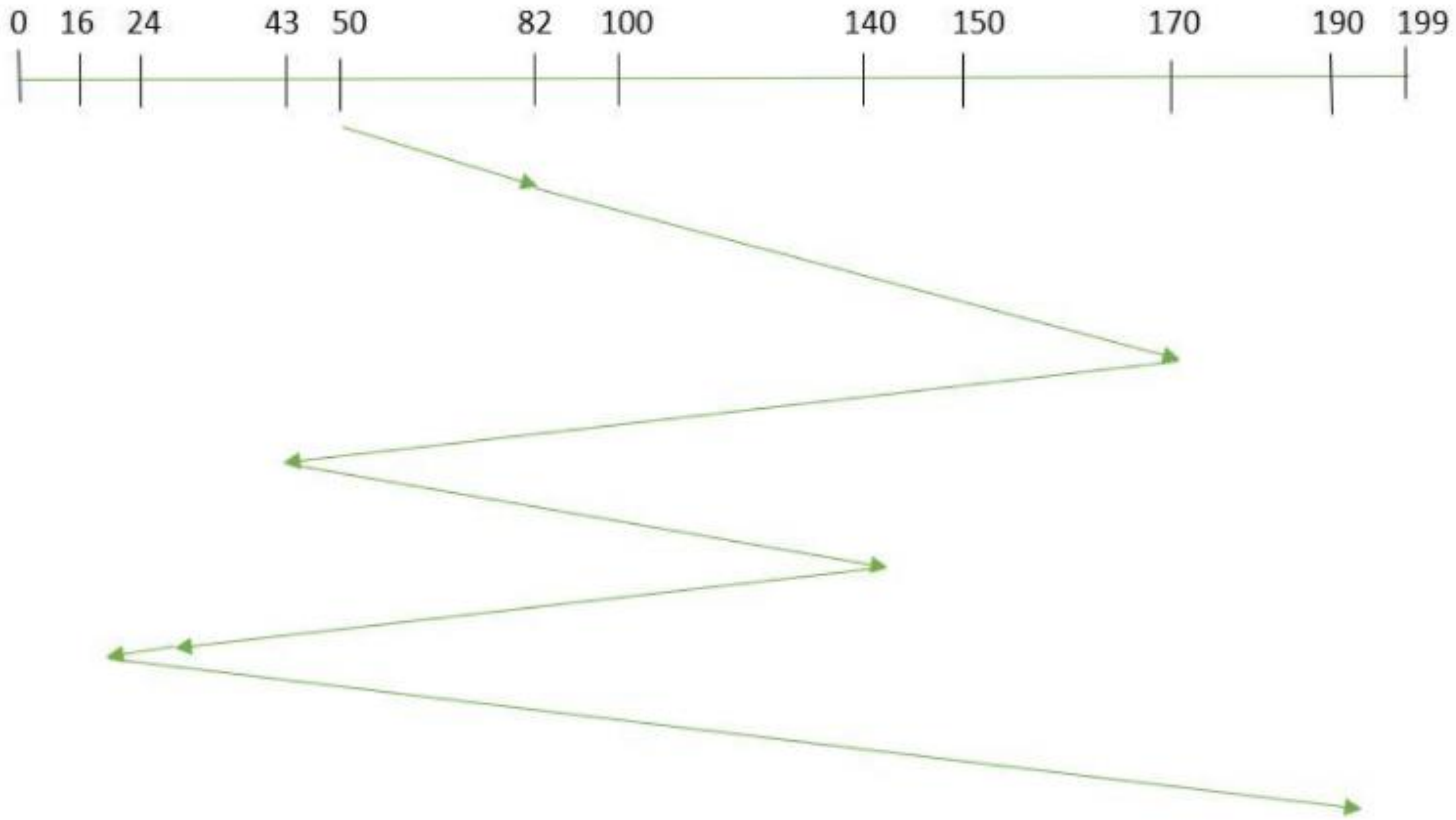
= 45 + 85 + 146 + 85 + 108 + 110 + 59 + 2

=640

FIFO

Suppose the order of request is- (82,170,43,140,24,16,190)

And current position of Read/Write head is : 50



FIFO

So, total seek time:

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

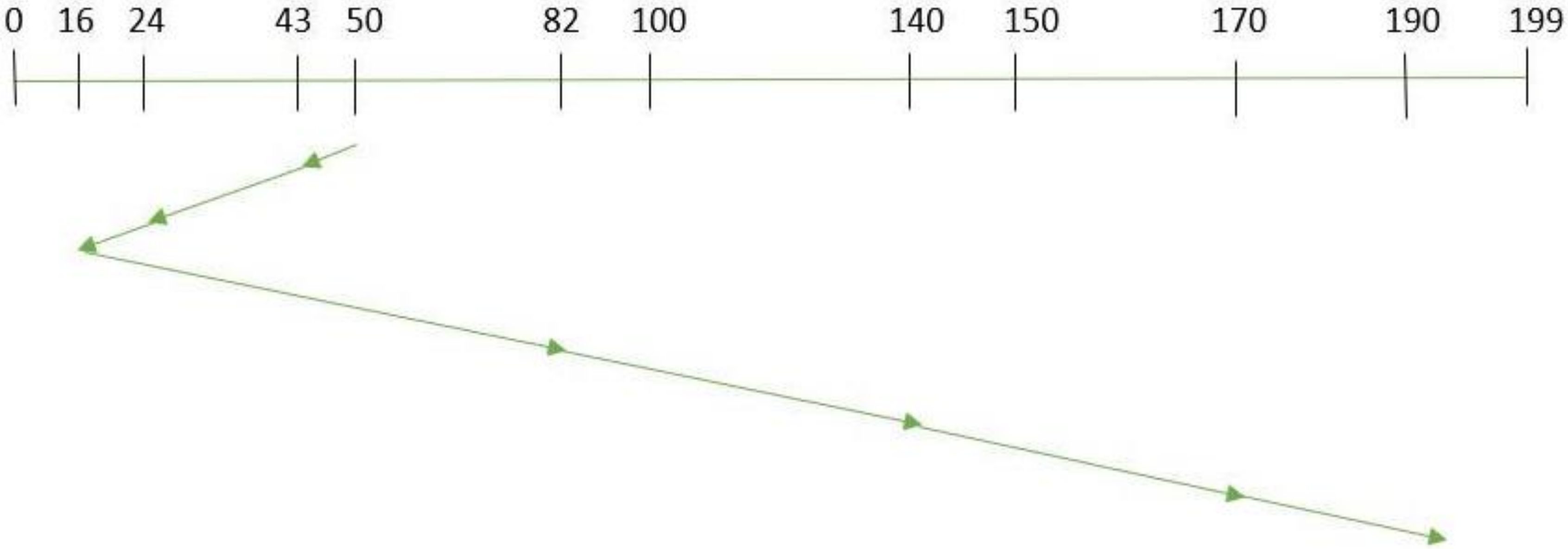
=642

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

- 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

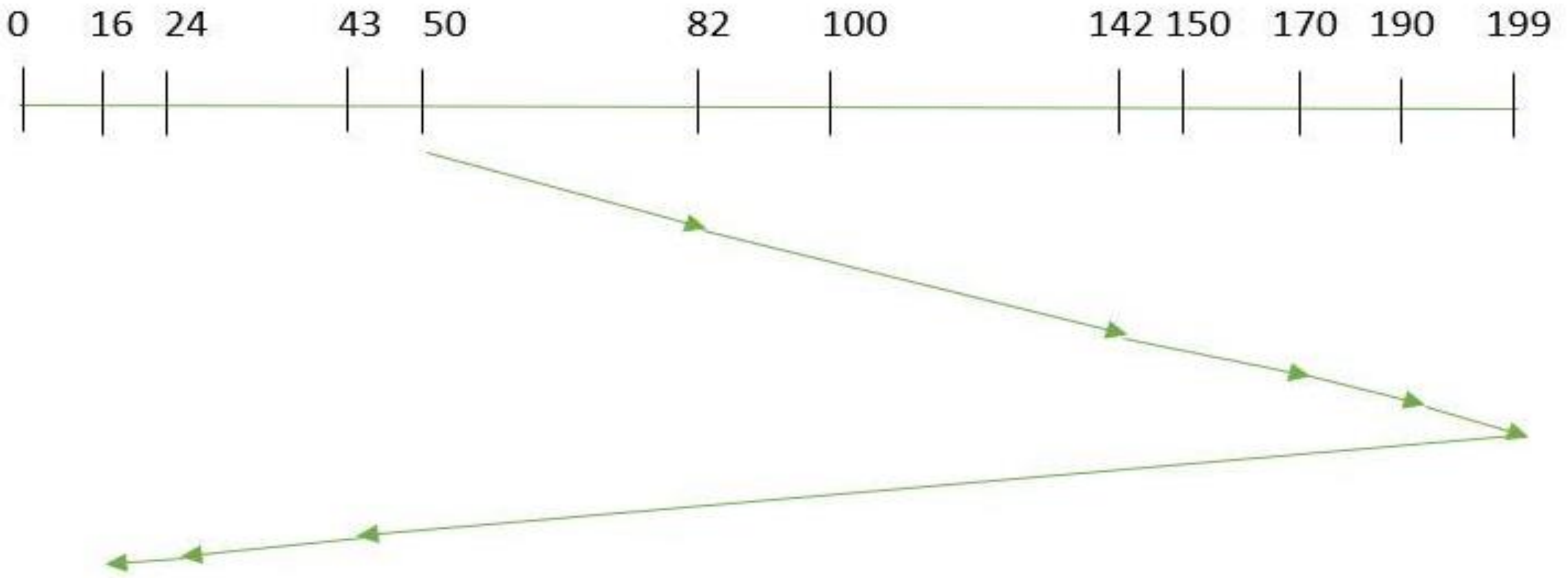
SCAN

- 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**

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” or Right direction.**



SCAN

- Therefore, the seek time is calculated as:
  - =(199-50)+(199-16)

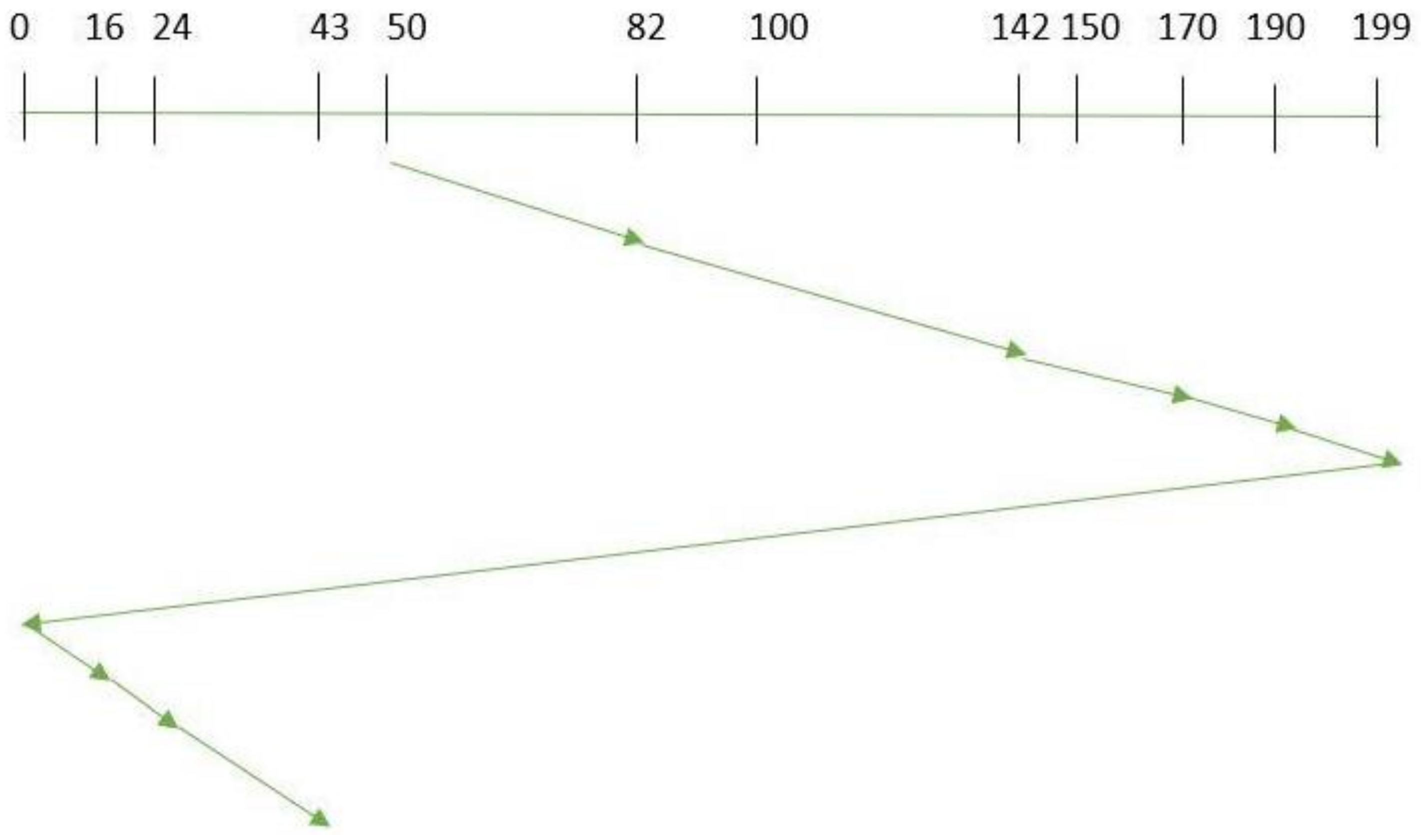
    =332

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)

CSCAN

- 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 - Right Side”.**



CSCAN

- Seek time is calculated as:
- =(199-50)+(199-0)+(43-0)
- =391

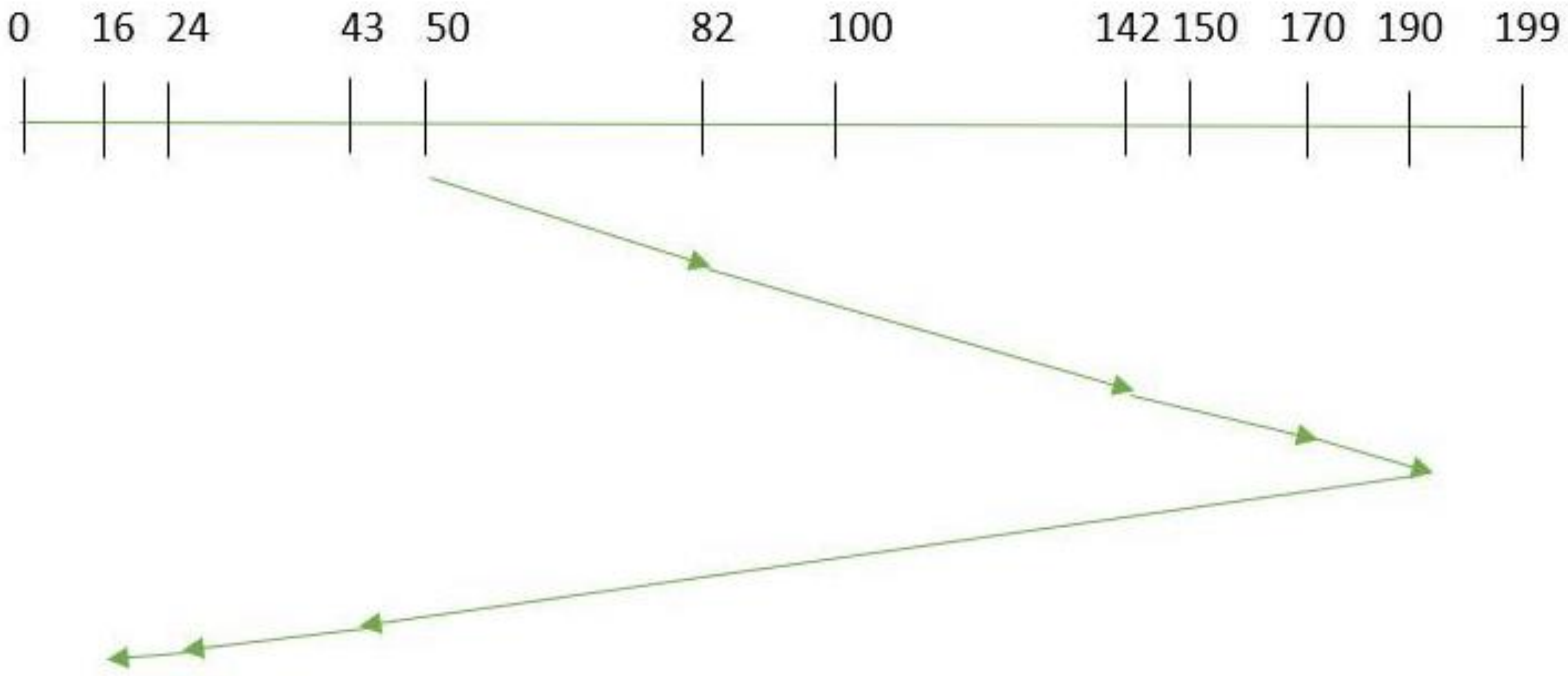
Look

It is similar to the SCAN disk scheduling algorithm except for the difference that the disk arm in spite of going to the end of the disk goes only to the last request to be serviced.

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 Right Side”.**

Look



So, the seek time is calculated as:

=(190-50)+(190-16)

=314

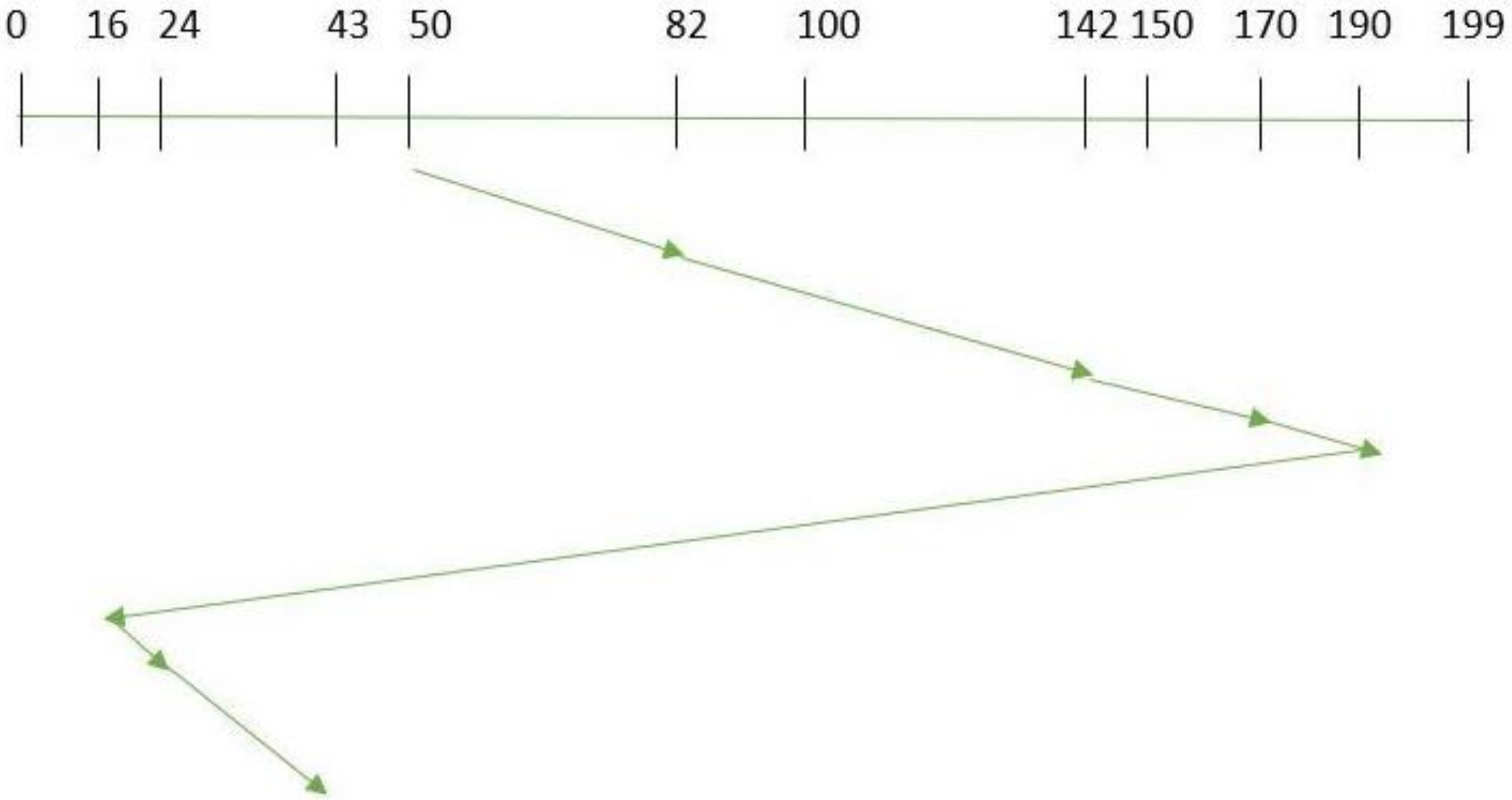
CLOOK

As LOOK is similar to SCAN algorithm, in similar way, CLOOK is similar to CSCAN disk scheduling algorithm. In CLOOK, the disk arm in spite of going to the end goes only to the last request to be serviced.

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 - Right Side”**

CLOOK



So, the seek time is calculated as:

=(190-50)+(190-16)+(43-16)

=341

Thank you…