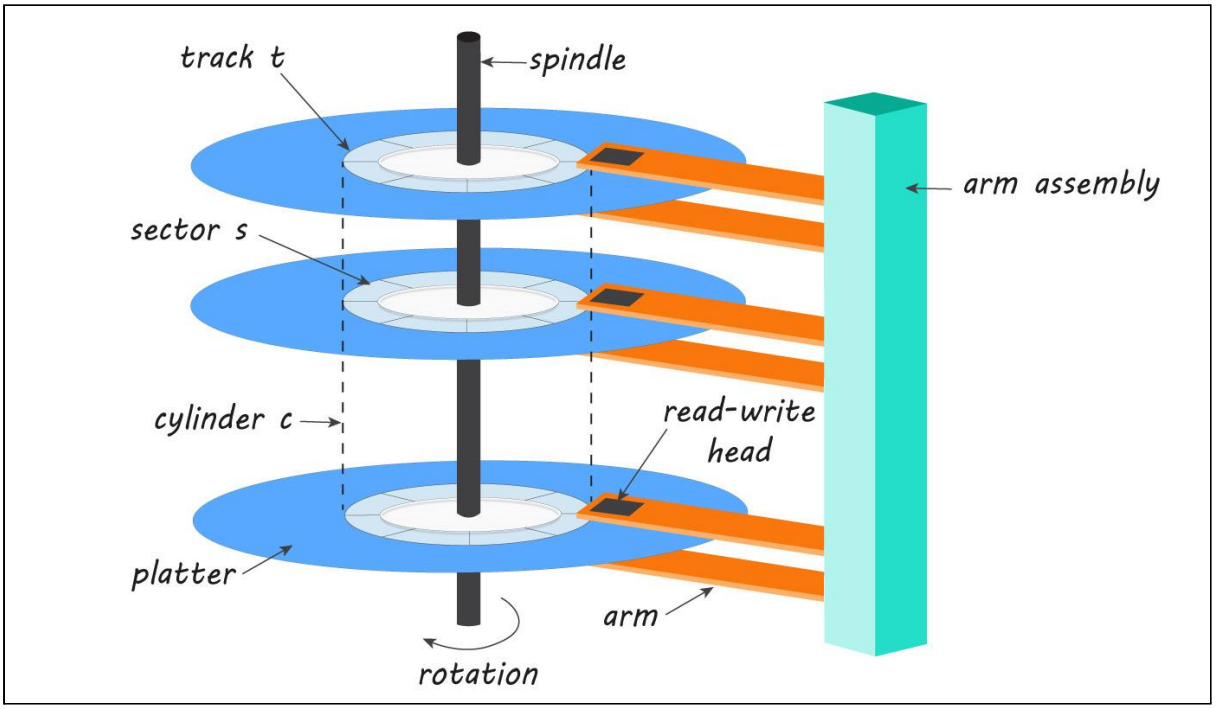
# Summary Note

## Hard Disk Drive and Solid State Drive

We have understood that secondary storage devices are used for storing both data and programs persistently and we have known about two most common non-detachable secondary storage devices, Hard Disk Drive (HDD) and Solid State Drive (SSD).

We focused more on HDD. The following diagram elaborately explains the components used in disk.




## Files and Directories

In this section, we understood that the operating system abstracts usage of secondary storage devices for the applications using **File System**. The File system provides the mechanism for:

1. Storage of data and programs on the disk
2. Access of data and programs on the disk

So, the file system defines the rules for storing and accessing data on disk. Now, OS uses two more abstractions:

- Files
- Directories

We went deep and learnt more details about Files and understood:

1. What are the common operations supported by OS on Files.
    a. Creating a file
    b. Writing a file
    c. Reading a file
    d. Repositioning within a file
    e. Deleting a file
    f. Truncating a file.
2. How does the OS gets to know about opened files? - using Open File Table
3. What are the locks used on files?
    a. A shared lock is for reading only.
    b. An exclusive lock is for writing as well as reading.
    c. An advisory lock is informational only, and not enforced. ( A "Keep Out" sign, which may be ignored. )
    d. A mandatory lock is enforced. ( A truly locked door. )
    e. UNIX uses advisory locks, and Windows uses mandatory locks.

Then we moved to learn about an important data structure that OS uses to store information of files and directories: **Inode Number**

So what information does inode store?

- Inode number
- Mode information which determines the file type and how the file's owner, its group, and others can access the file.
- Number of links to the file
- UID of the owner
- Group ID (GID) of the owner
- Size of the file
- Actual number of blocks that the file uses
- Time last modified
- Time last accessed
- Time last change

On a unix system we can get the inode of a file r a directory using the ls command with -i option

```
# ls -i /bin/ls
1048630 /bin/ls
```

We can query this inode number to get the details of that file or the directory using stat or debugfs command.

Here is how the inode for a file looks on a Unix based systems

```
Inode: 1715035   Type: regular    Mode:  0644   Flags: 0x80000
Generation: 1035150724    Version: 0x00000000:00000001
User:     0   Group:     0   Size: 227404
File ACL: 0    Directory ACL: 0
Links: 1   Blockcount: 448
Fragment:  Address: 0    Number: 0    Size: 0
 ctime: 0x607e55b2:df434ac8 -- Mon Apr 19 21:16:50 2021
 atime: 0x607e6b2f:0c0bd568 -- Mon Apr 19 22:48:31 2021
 mtime: 0x607e55b2:df434ac8 -- Mon Apr 19 21:16:50 2021
crtime: 0x607e549b:b97003b4 -- Mon Apr 19 21:12:11 2021
Size of extra inode fields: 32
EXTENTS:
(0-55):6895285-6895340
```

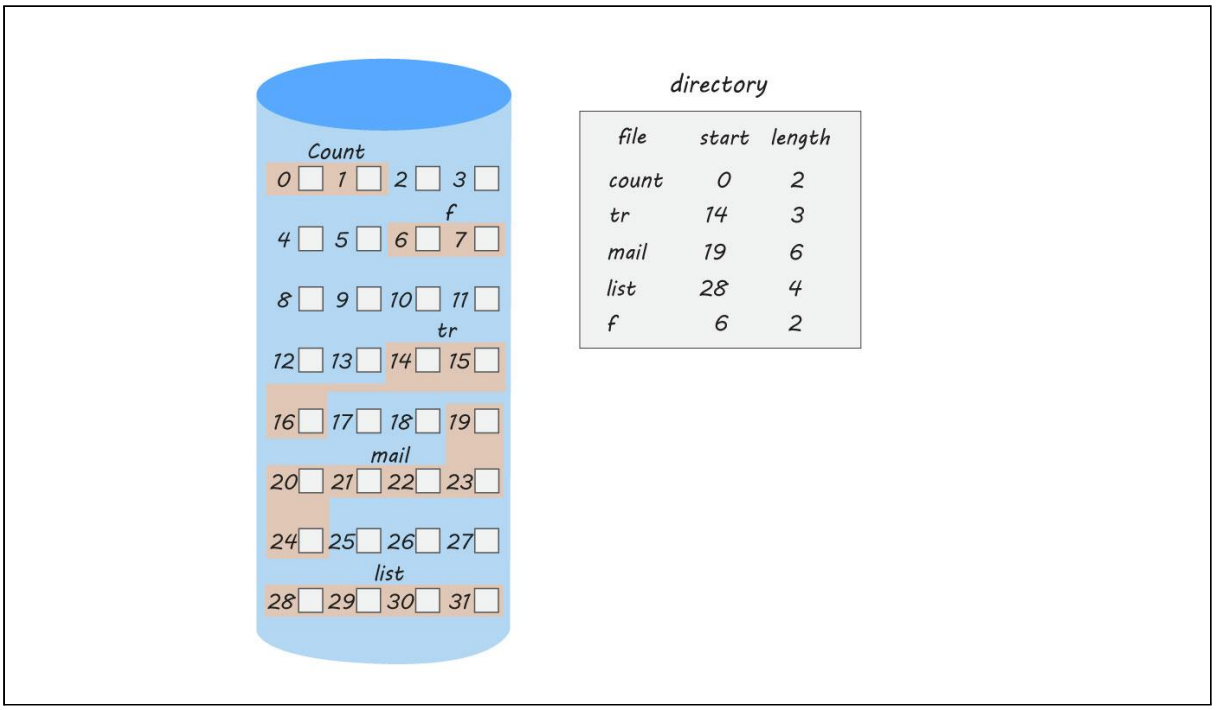
Here is how the inode for a directory looks on a Unix based systems

```
Inode: 1048577   Type: directory    Mode:  0755   Flags: 0x80000
Generation: 3641990193    Version: 0x00000000:000003d6
User:     0   Group:     0   Size: 4096
File ACL: 0    Directory ACL: 0
Links: 2   Blockcount: 8
Fragment:  Address: 0    Number: 0    Size: 0
 ctime: 0x6023697c:94bb9a94 -- Tue Feb  9 21:05:00 2021
 atime: 0x607e7ce8:87d0b3c0 -- Tue Apr 20 00:04:08 2021
 mtime: 0x6023697c:94bb9a94 -- Tue Feb  9 21:05:00 2021
crtime: 0x5abe6d9a:45ff046c -- Fri Mar 30 10:02:18 2018
Size of extra inode fields: 32
EXTENTS:
(0):4202528
```

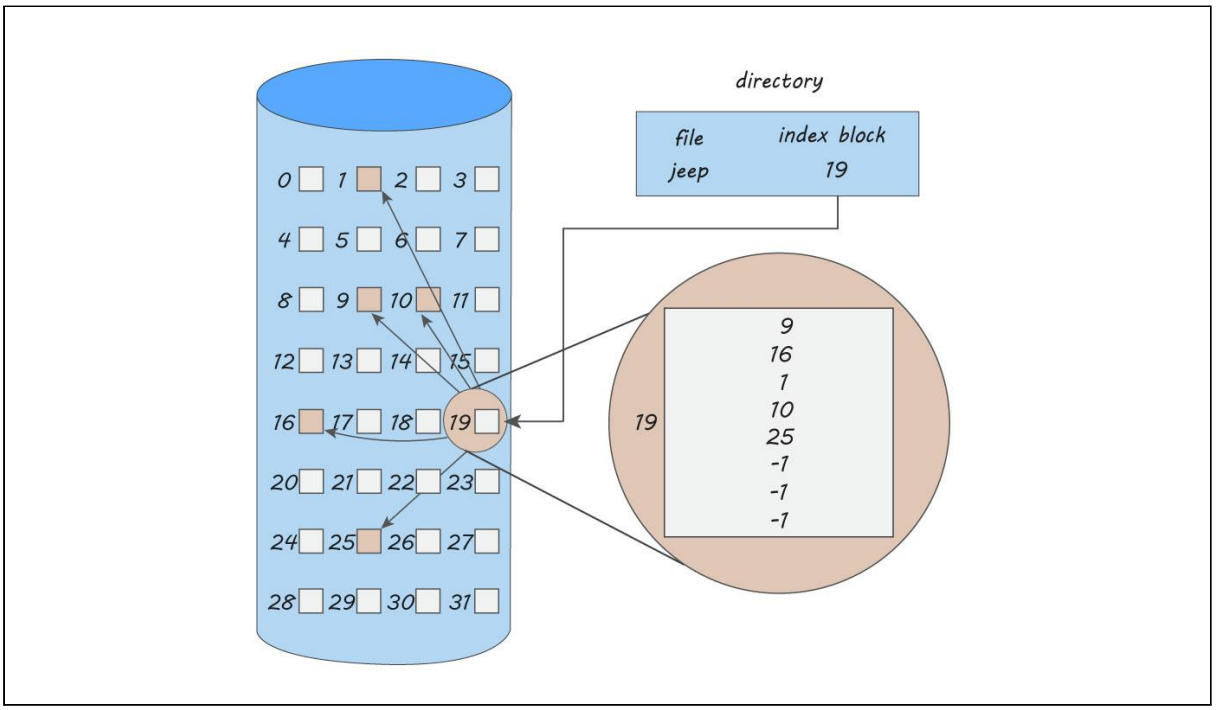
## Disk Space Allocation Methods

After learning about the structure of Disk and how OS abstracts usage of Disk for users, we moved on to understand how disk space is allocated to files by OS. We discussed following algorithms:

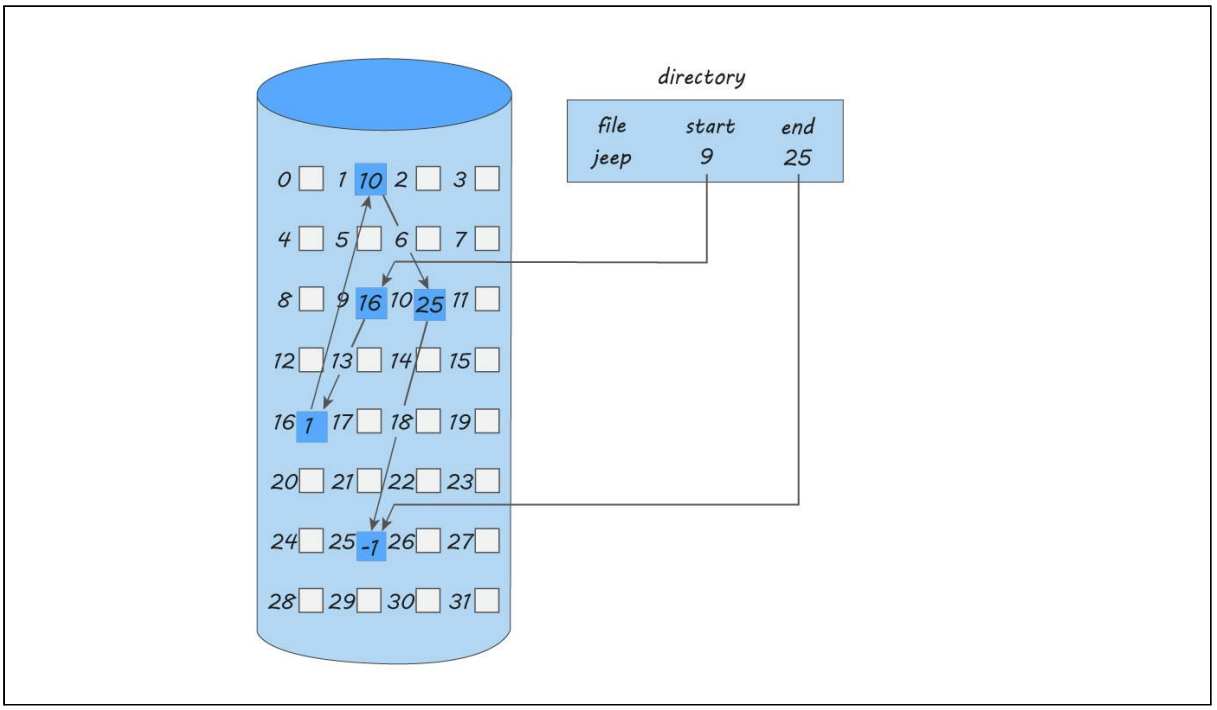
1. Contiguous Allocation Method




2. Linked Allocation Method



3. Indexed Allocation Method

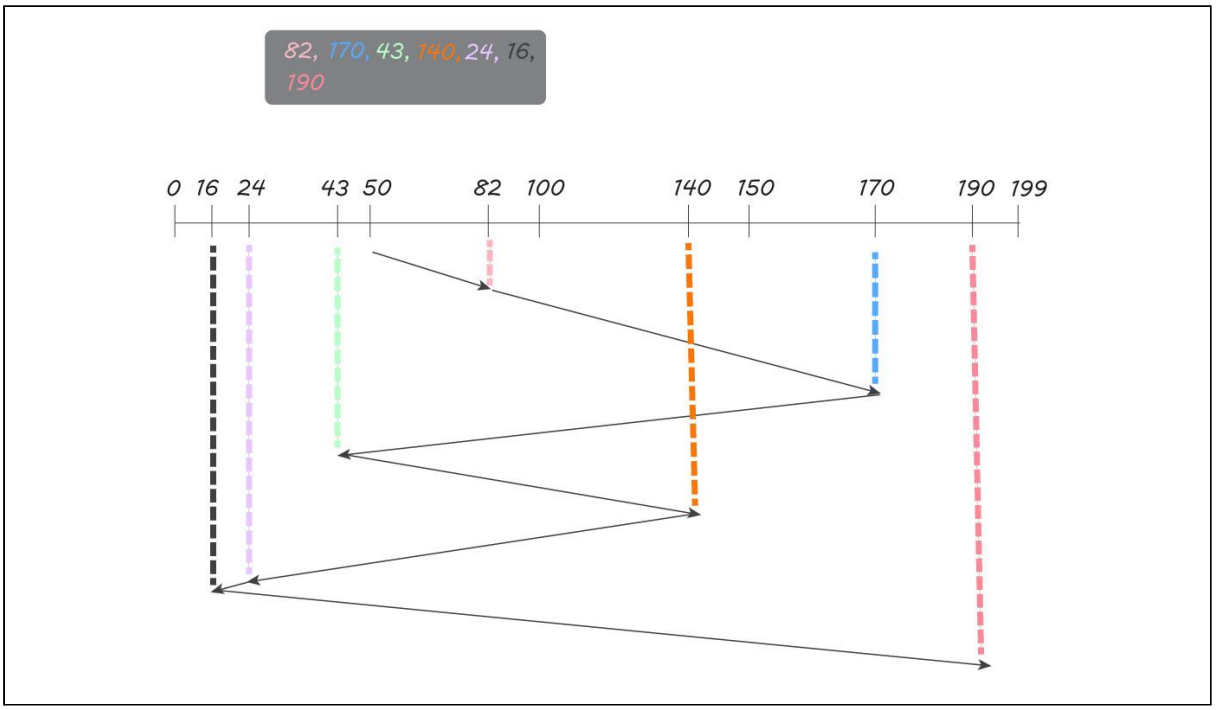



We also learnt about an interesting fact that Linux internally uses indexed allocation method.

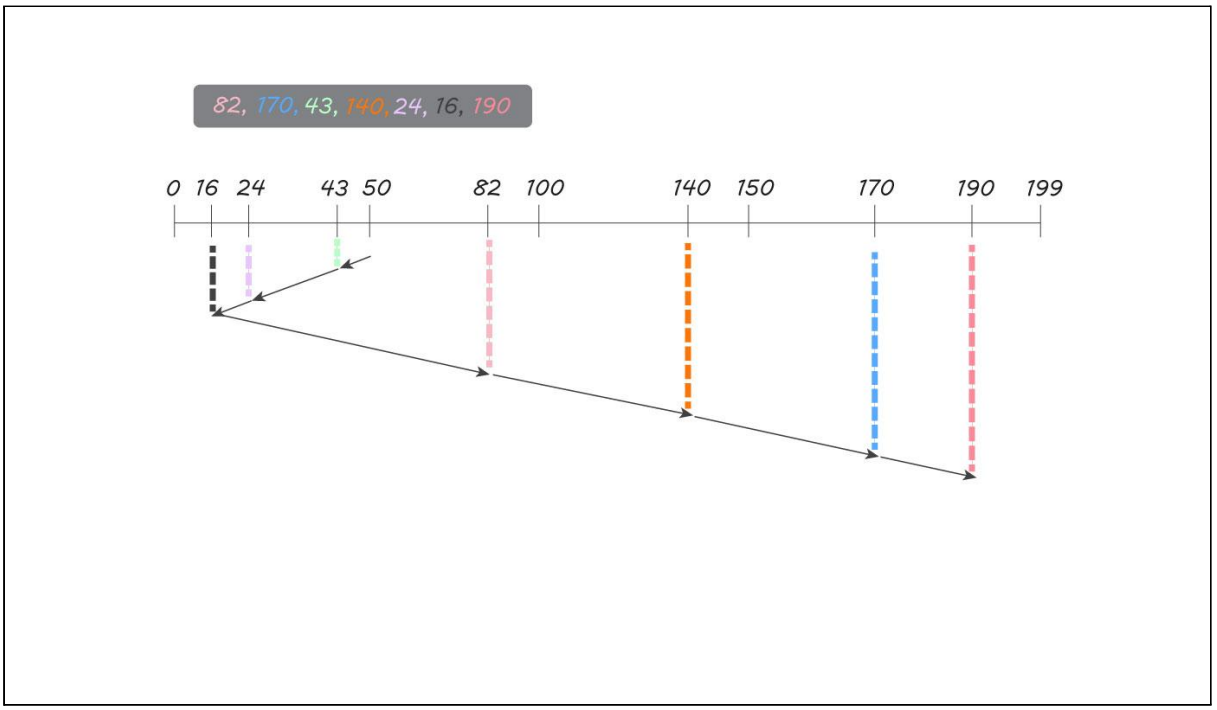
## Disk Scheduling Algorithms

In the last section we discussed Disk Scheduling algorithms, whose objective is to reduce the seek time for disk. The following algorithms were discussed:

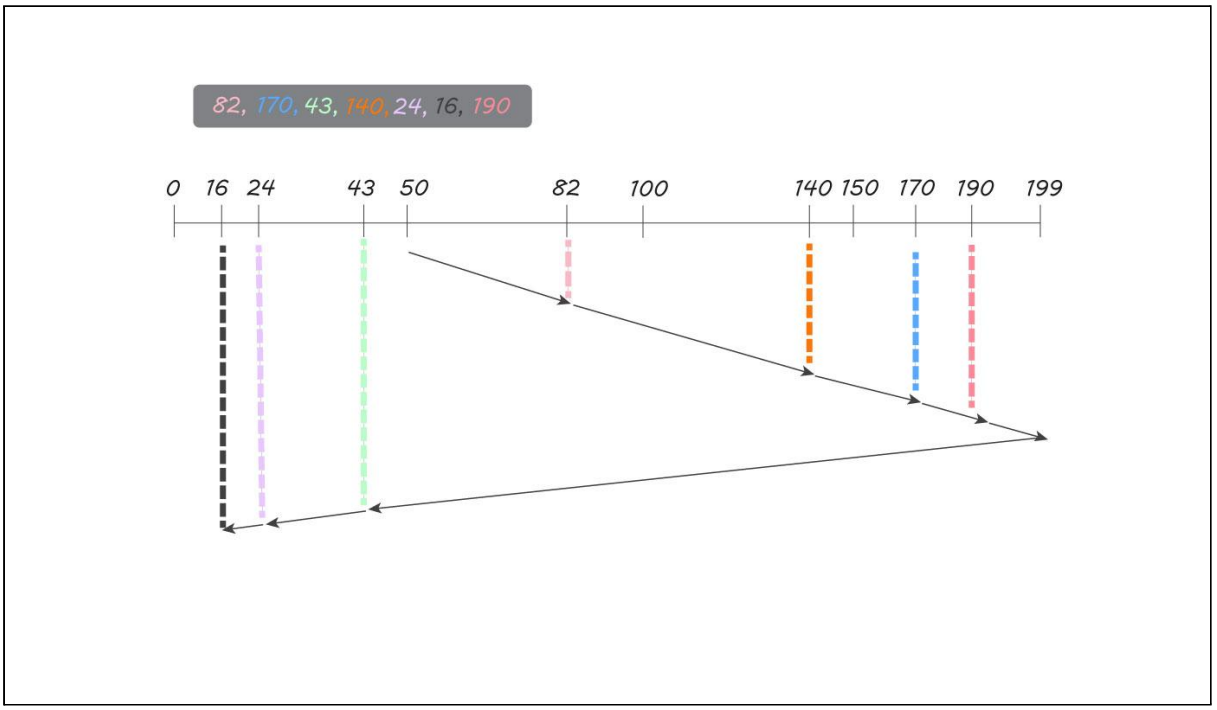
1. FCFS (First Come First Served)



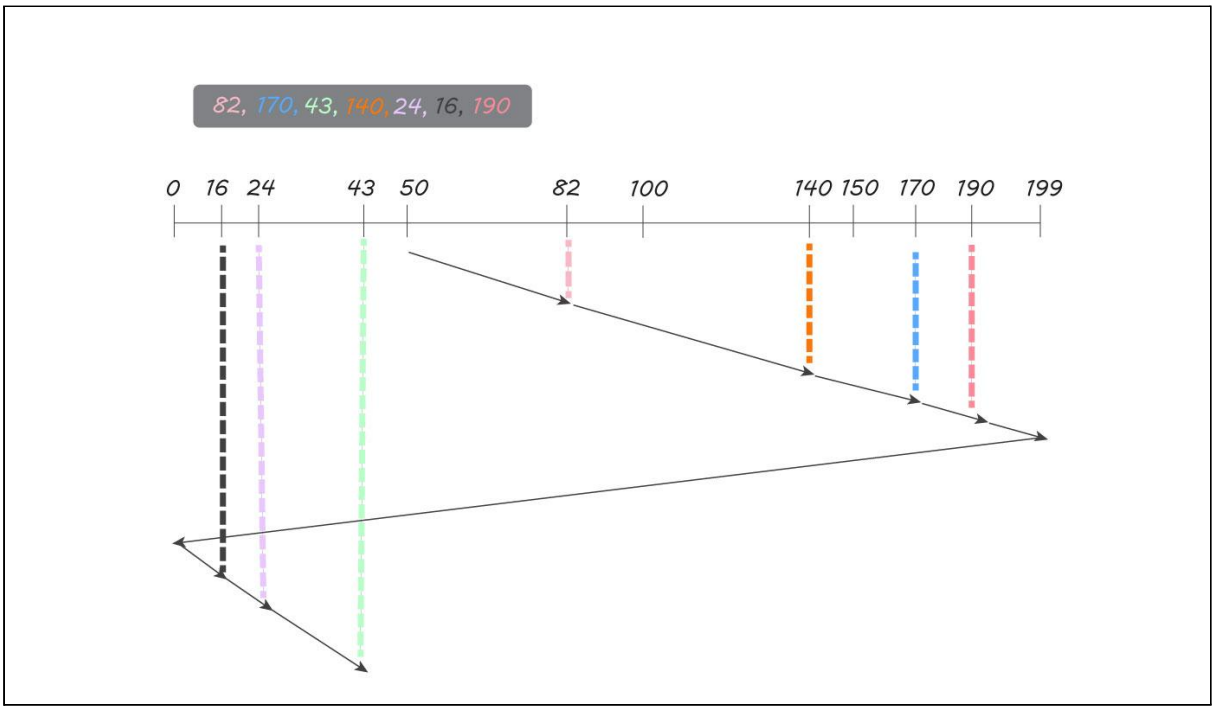

2. SSTF (Shortest Seek Time First)



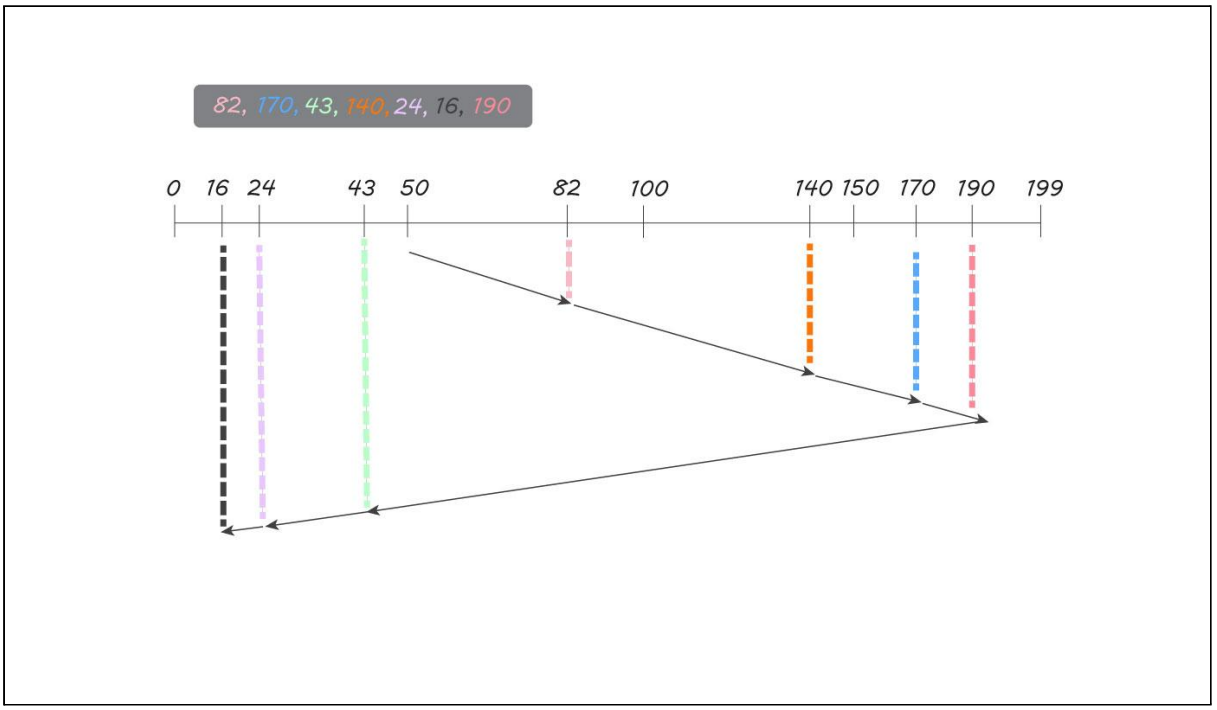
3. SCAN




4. C-SCAN



5. LOOK




6. C-LOOK