Unit-4 File Management

❖ Files System

It is a part of OS which deals with files.
It manages all files, provides facilities to create, delete, read and write operation for files.
It provides directories to organize files.
It is responsibility of file system
To provide efficient access to information stored in files
To manage memory space on secondary storage devices when file is created, deleted or modified.
To provide protection mechanism to allow users to administer how other users can access the information in files.

❖ Files Attributes

Every file has its name and data. All file contains some other properties called attributes. Main attributes are as following:

1. **Name :**
   - A string of alphanumeric characters and some special characters like underscore (**_**).
   - We can refer file easily with its name.
2. **Identifier:**
   - It is unique tag number which uniquely identifies file within the system.
   - It is used by OS to refer files.
3. **Type :**
   - It identifies type of file.
   - It expressed generally in form of extension like **.c , .cpp , .txt**
4. **Location:**
   - It is the pointer to the device and location on that devices of the file
5. **size :**
   - It specifies current size of files (in bytes, words or blocks).
6. **Protection:**
   - This attribute defines who can read file, write files, executes files.
7. **Usage count:**
   - It indicates no. of processes that are currently using these files.
8. **Time , date and user identification:**
   - It specifies information regarding file creation, update and last access.

❖ File Operations

User can retrieve file information anytime. OS performs various operations on files.
OS provides a set of system call to perform such operations.
This system calls can be called using library functions provided by programming languages.

Basic file operations are as follows:

1. **Create:**
   - A new file can be created by system call embedded in a program or by an OS commands issued by an interactive users.
   - To create file two steps are performed; first required disk space is allocated to file; second, new entry for the created file is made in directory.
   - **The directory entry contains file name, identification number and some other information.**
2. **Delete:**
   - When file is not needed, it has been deleted to free up disk space.
   - Two steps for deletion:
     1. All the space on disk is released,
     2. Directory entry is erased.
3. **Open:**
   - To use file, it must be opened.
   - File attributes and data contents are fetched in main memory for fast access on later calls.
   - Memory space in main memory is allocated to store fetched information.
4. **Close:**
   - When use of file is finished, it should be closed to free up main memory space.
   - File attributes and contents are stored back on disk. This may contain modified information if file is updated.
5. **Read:** To Read the information which is Stored into the Files.

6. **Write: For** inserting some new Contents into a File.
7. **Append**: user can add data at the end of existing file.
8. **Seek:** if user wants to read random file data, then using this operation user can skip some lines and go forward to read.
9. **get attributes :** user can get attribute information
10. **set attribute :** user can set attribute information
11. **rename:** user can rename the file name.

❖ File Types

Regular files:

Regular File may belong to any type of Application for example notepad, paint, C Program, Songs etc.

- Ordinary Files are used for Storing the information about the user Programs. With the help of Ordinary Files we can store the information which contains text, database, any image or any other type of information.

Directory files:

The Files those are Stored into the Particular Directory or Folder. Then these are the Directory Files. Because they belongs to a Directory and they are Stored into a Directory or Folder. For Example a Folder Name Songs which Contains Many Songs So that all the Files of Songs are known as Directory Files.

Special Files. :

The Special Files are those which are not created by the user. Or The Files those are necessary to run a System. The Files those are created by the System. Means all the Files of an Operating System or Window, are refers to Special Files. There are Many Types of Special Files, System Files, or windows Files, Input output Files. All the System Files are Stored into the System by using. sys Extension.

❖ Directory Structures

1. Single level directory structure

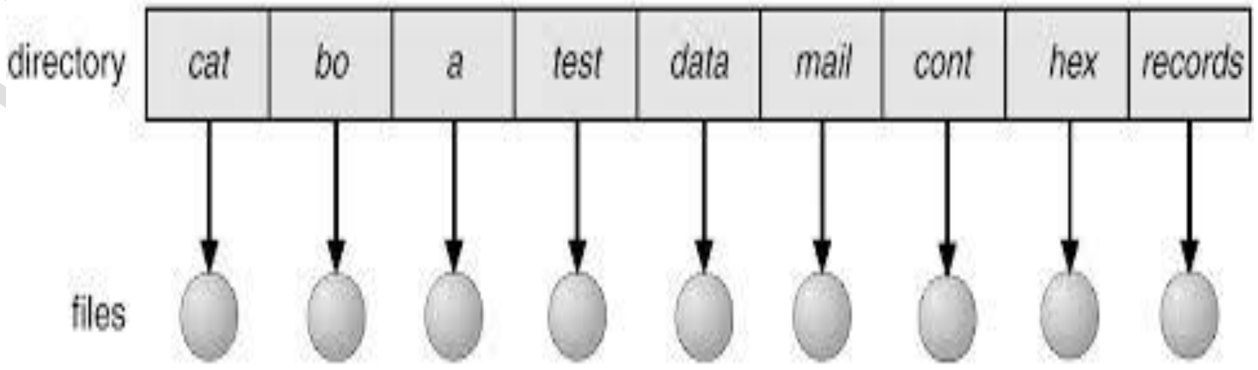
- It is simplest directory structure.
- In this type of structure all files are stored in same directory.
- Directory Contains only files, no sub directory.
- It was Used in Pc

Advantage

- Simple
- Easy to search

Disadvantage

- Not suitable for multiuser system
- No. of files becomes too large
- Different files can’t be grouped
- Since all files are in the same directory, they must have unique name.
- Single user may find it difficult to remember the names of all files as the number of file increases.
- Keeping track of so many file is tedious task.



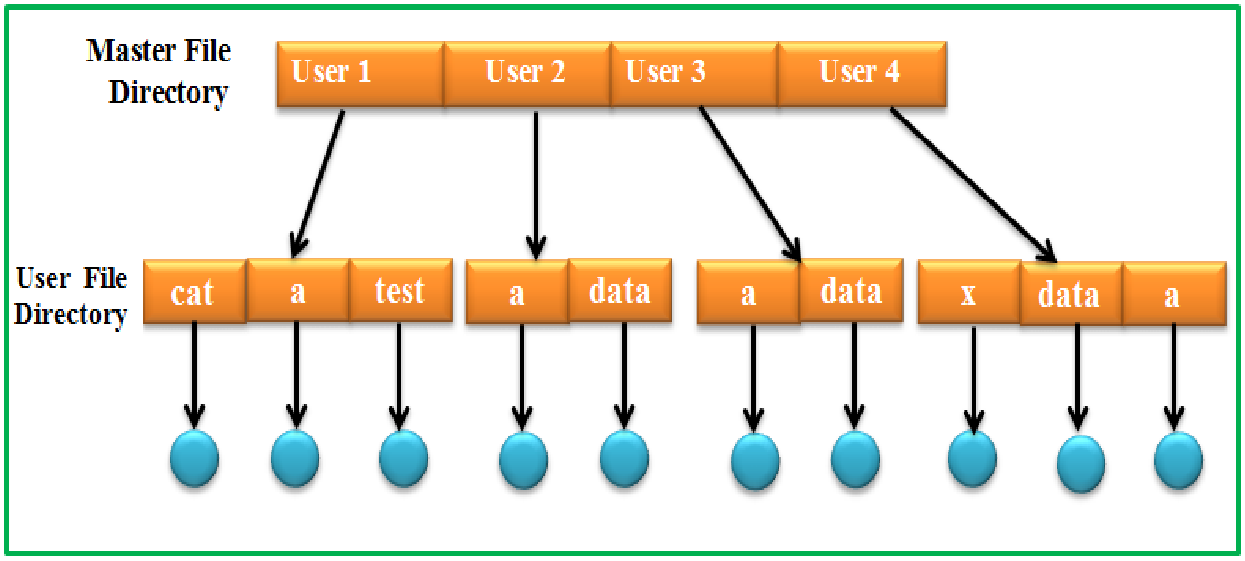
2. Two level directory structure

- Each user has private directory to avoid file collision
- Two level directory
    - Root directory(master file directory **MFD)**
    - User file directory(**UFD)**
- The root of a tree is Master File Directory (MFD). Its subdirectory is User File Directory (UFD).
- The descendents of UFD's are file themselves

Advantage

- Avoid file collision
- Used in multi user system
- Efficient searching

- **Disadvantage**
    - Not suitable for user having large no. of file
    - Different files cannot be grouped.



3. Tree structured directory structure

- User can create their own sub directory
- **It is also Known as hierarchical system**
- Each file has unique file path.
- **Advantage**
    - It Avoid file collision
    - It is Used in multi user system
    - Different files can be grouped

- **Disadvantage**
    - Sharing of files and directory is not possible.

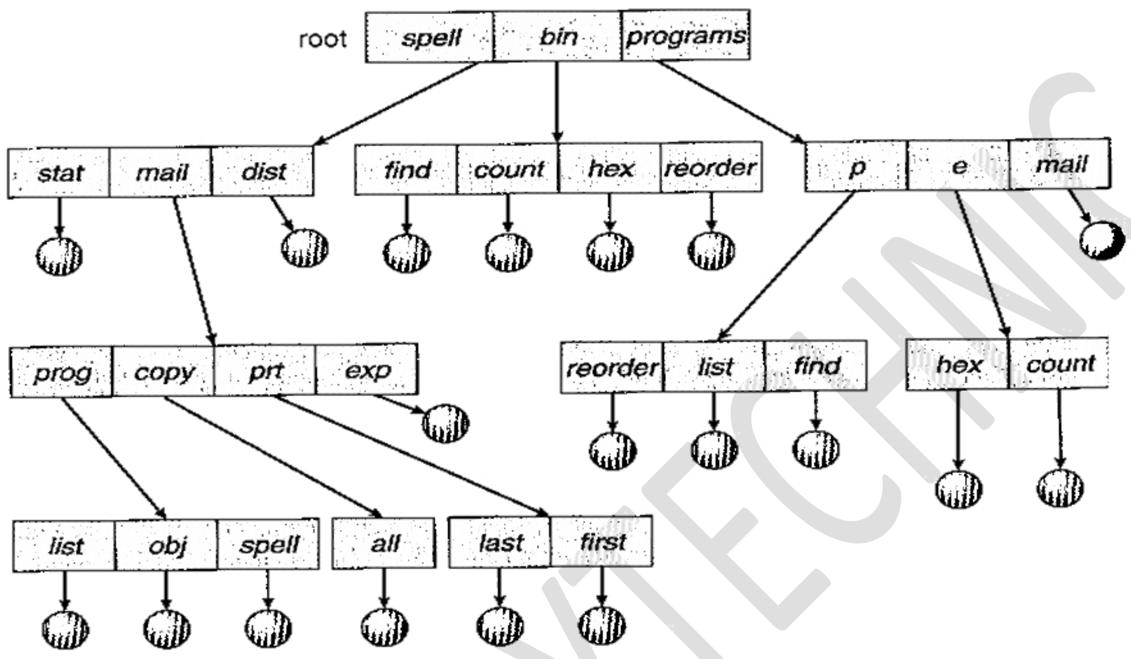


Figure 10.9 Tree-structured directory structure.

4. **Acyclic graph directory structure**

- When the same files need to be accessed in more than one place in the directory structure (e.g. because they are being shared by more than one user / process), it can be useful to provide an acyclic-graph structure. (Note the ***directed*** arcs from parent to child.)
- Permits user to create shared files and director
- shared files and directory created using links
- Directory structure does not contain cycle.
- **Advantage**
    - Sharing of files and directory
- **Disadvantage**
    - Deletion of files or dir. Is complex.
    - There are no cycle in directory
    - File may have multipath

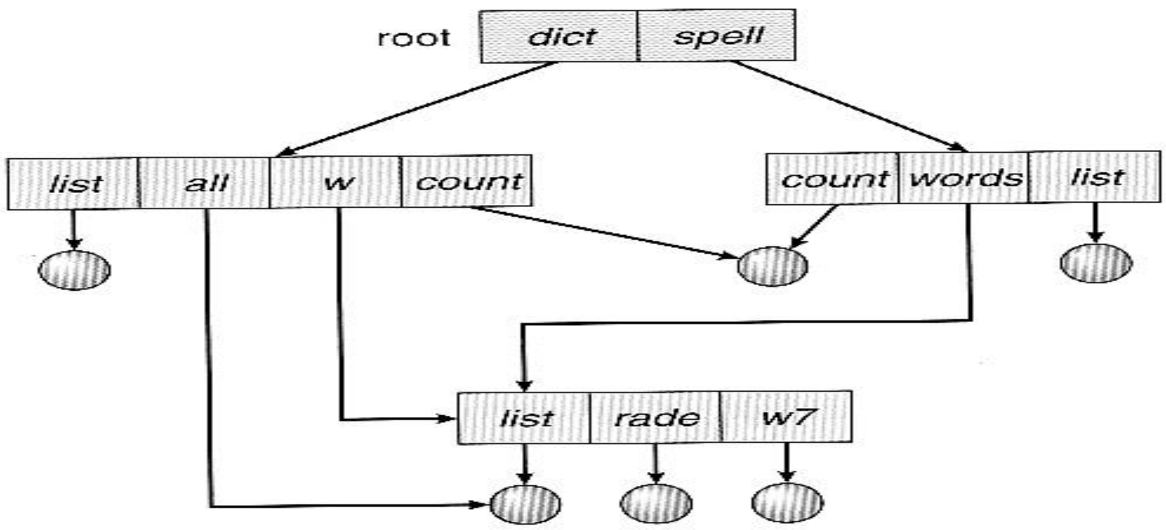


Figure 10.10 Acyclic-graph directory structure.

5. General graph_directory structure

- Same as acyclic graph but allows cycles in directory
- **Advantage**
  - Allows sharing of file
- **Disadvantage**
  - Searching should be done careful ,it may get infinite.

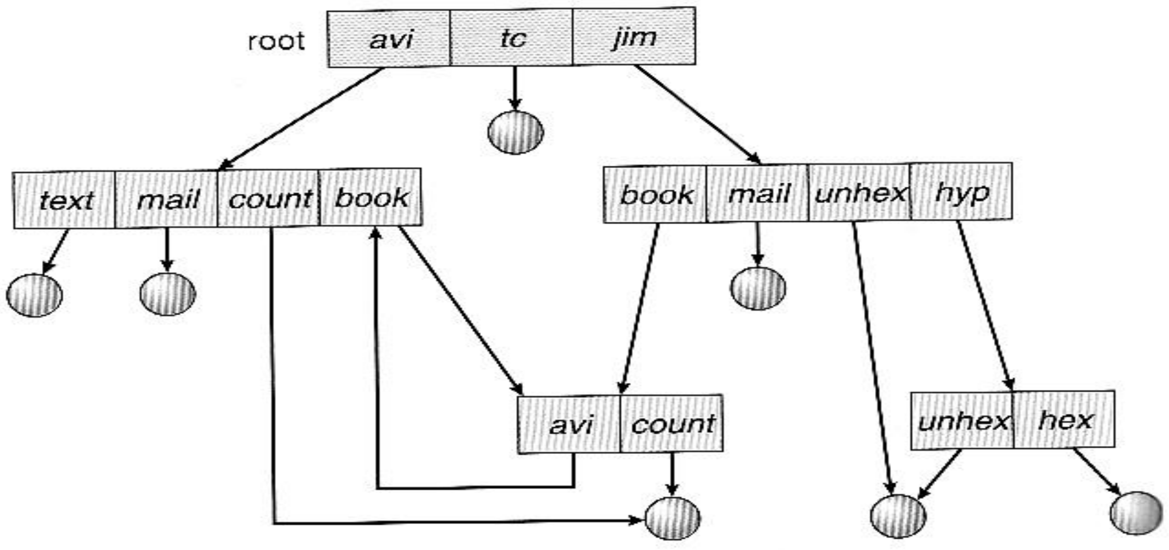


Figure 10.11 General graph directory.

❖ Protection

Protection is especially important in a multiuser environment when multiple users use computer resources such as CPU, memory, etc. It is the operating system's responsibility to offer a mechanism that protects each process from other processes. In a multiuser environment, all assets that require protection are classified as objects, and those that wish to access these objects are referred to as subjects. The operating system grants different 'access rights' to different subjects.

A mechanism that controls the access of programs, processes, or users to the resources defined by a computer system is referred to as protection. You may utilize protection as a tool for multi-programming operating systems, allowing multiple users to safely share a common logical namespace, including a directory or files.

It needs the protection of computer resources like the software, memory, processor, etc. Users should take protective measures as a helper to multiprogramming OS so that multiple users may safely use a common logical namespace like a directory or data. Protection may be achieved by maintaining confidentiality, honesty and availability in the OS. It is critical to secure the device from unauthorized access, viruses, worms, and other malware.

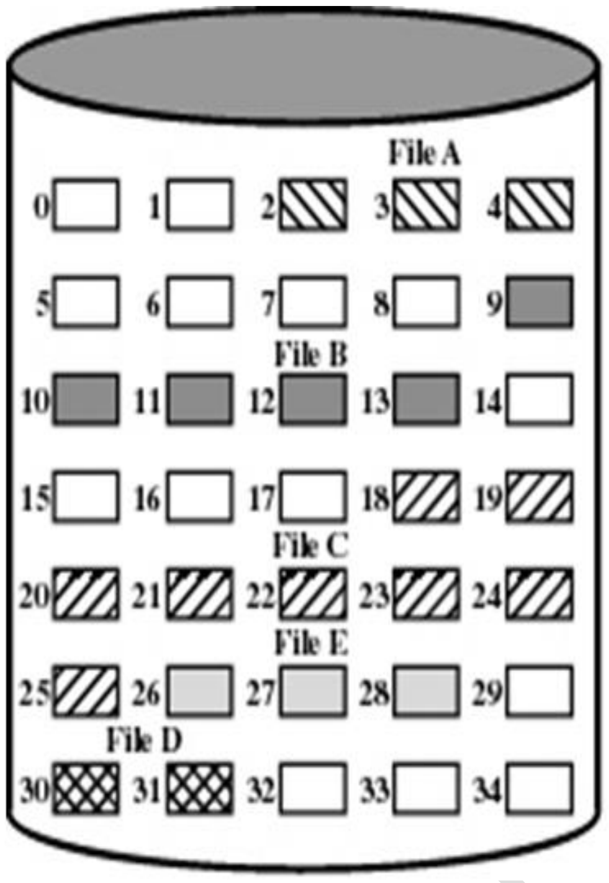
❖ Allocation Methods – Contiguous, Linked

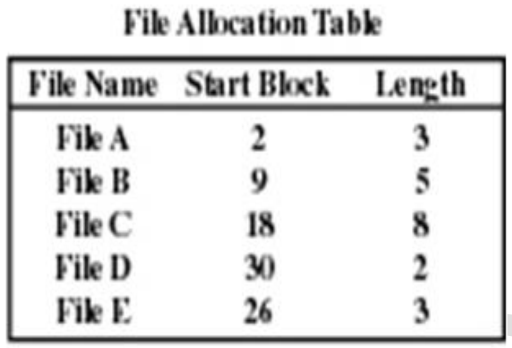
<u>Contiguous file Allocations method</u>

- Each fie occupies a set of contiguous blocks on disk.
- When file is created ,a disk is searched to find out a group of free memory having enough size to store a file. once chunk is found, required memory is allocated.
- Directory entry contains file name, starting block no. and length of a file.

File name	Starting block number	length

- This method used on CD-ROMs where all file sizes are known in advance.they never change during subsequent uses.





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

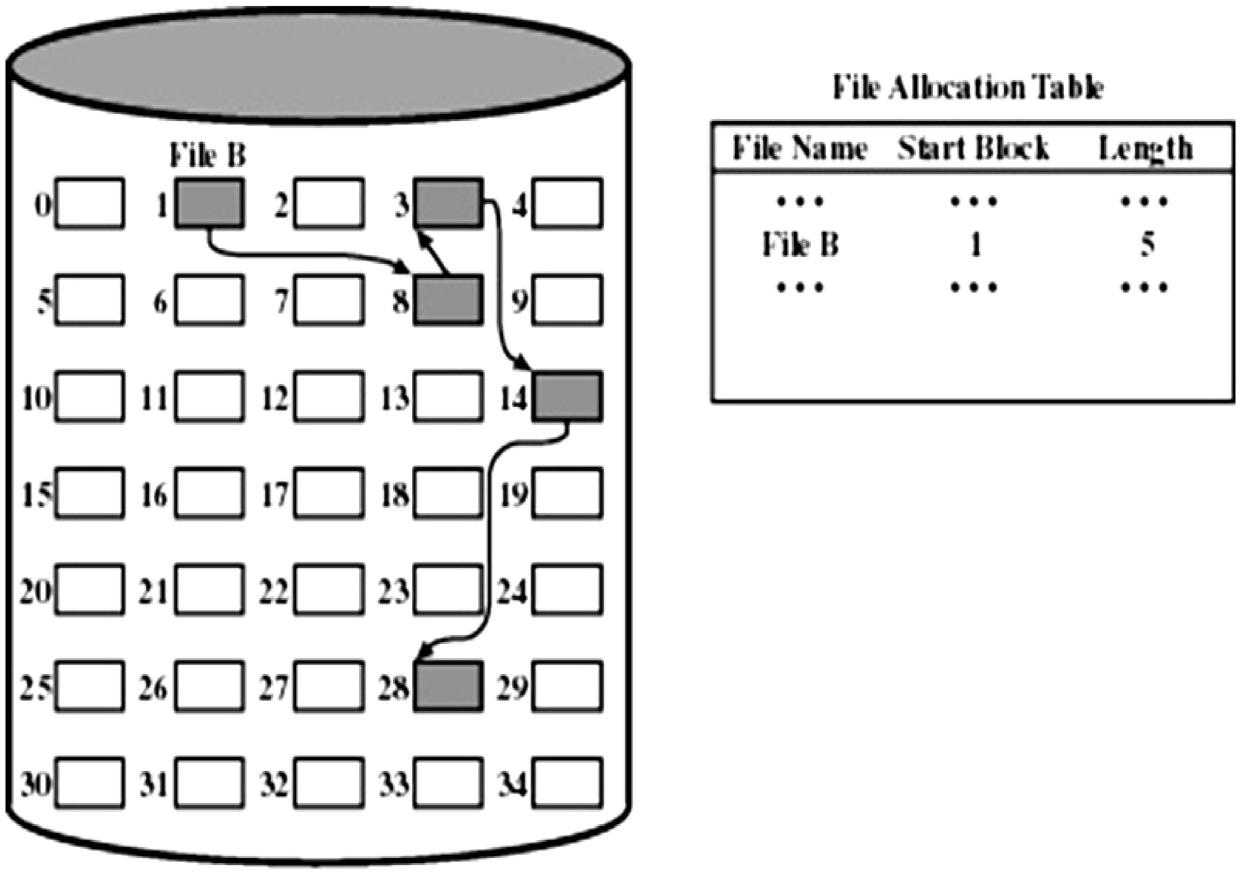
- **Advantage**
  - Simple to implement
  - File access is quick
- **Disadvantage :**
  - Finding free space for a new file is time consuming
  - If size of an existing file is increases ,it may not be possible to accommodate such extension.
  - External fragmentation may be generated

Linked or chained allocation

- Allocation on basis of individual block
- Disk blocks are scattered any where on the disk.
- Each block contains a pointer to the next block in the chain
- Only single entry in the file allocation table
  - Starting block and length of file

File name	Starting block number	Last block number

- **Advantage :**
  - No external fragmentation
  - Disk space can be utilized effectively
  - Best for sequential files



- Linked scheme
- Multilevel index
- Combined scheme

❖ Disk Structure

- Hard disk is a sealed circuit. it contains one or more platters.
- **Platters :**
  - Each platter contains ***magnetic coating on both of its surfaces.***Platters are stacked one on top of another.
  - Each platter contains two surfaces.
  - They rotate together round a **central spindle.**
  - ***Rotation speed*** is normally 3600, 5400 or 7200 rpm (Rotation per Minute).
  - Disk contains 1 to 8 platters.

- Data is read from and written to these platters using a number of ***read /write heads.***

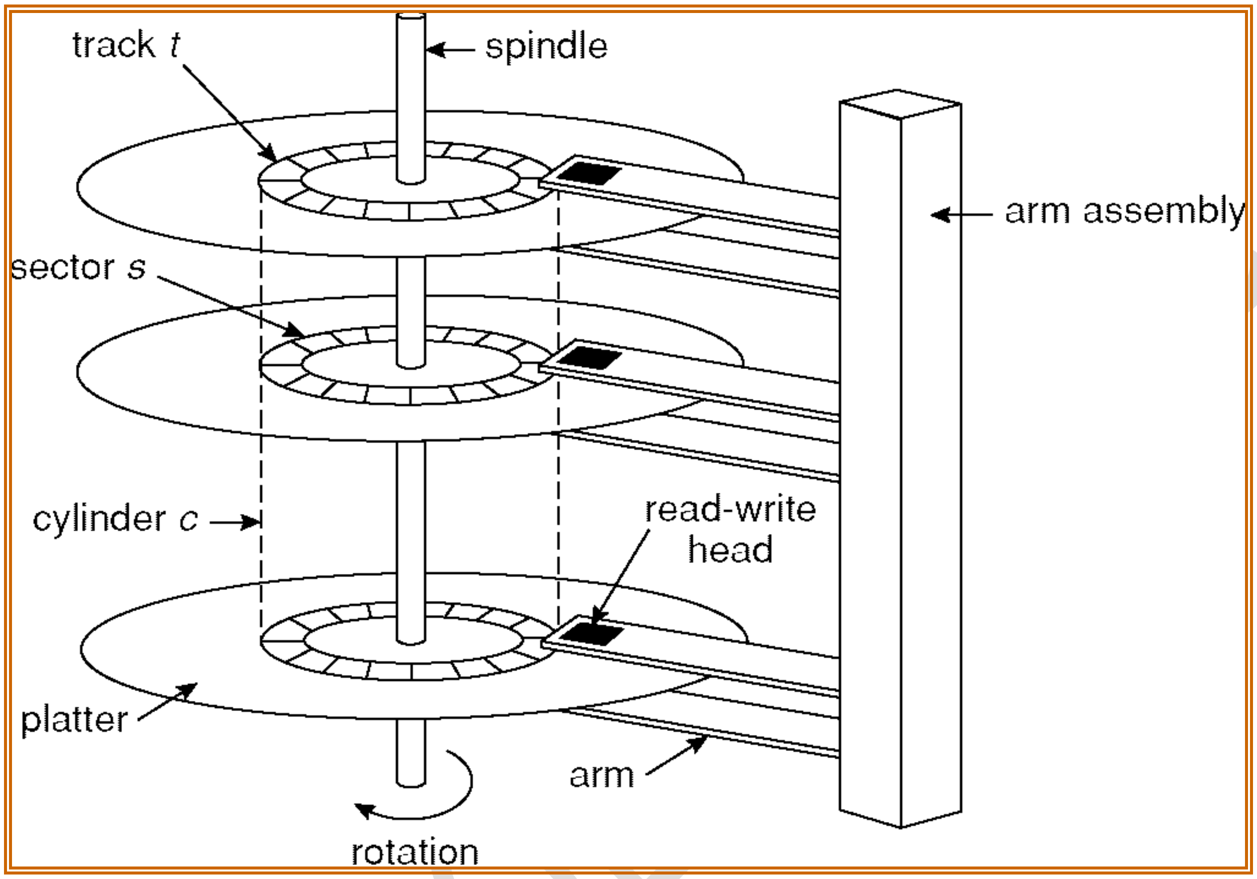


Figure :physical structure of hard disk

- There are many heads; each of head is attached to an *arm.*
- ***An arm*** can be moved in order to access different parts of the platters.
- A platter is a collection of circular, concentric, flat rings. These *rings* are called ***tracks.***
- *D*ata is stored on surface of a platter inside these tracks in form bits (*0 to 1).*there are more than thousands f tracks on a single surface.
- Each track is divided into a fixed size blocks, called *sector*. Sector size is of 512 bytes and there are *hundreds of sectors on a single track.*
- A group of tracks with the same radius are called ***cylinders.*** Each track is on different platter.

Logical address

- Disk is considered as large one dimensional array of fixed size logical **block .**
- Size of blocks is varying from system to system.
- Unique no. is used to uniquely identify on a disk.
- Numbering starts from '0' or '1' to some maximum number.
- Total number of blocks depends upon a storage capacity of a disk and the size of block.

❖ 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 arm 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.
- Several algorithms exist to schedule the servicing of disk I/O requests.
- We illustrate them with a request queue (0-199).

SCAN and C-SCAN algorithm

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 backand moves in the reverse direction satisfying requests coming in its path.

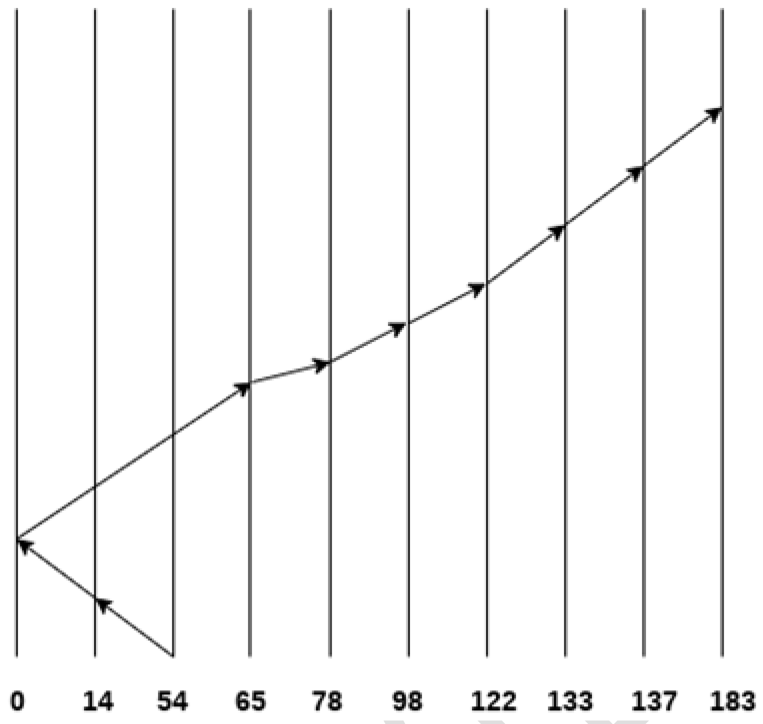
It works in the way an elevator works, elevator moves in a direction completely till the last floor of that direction and then turns back.

Example

Consider the following disk request sequence for a disk with 100 tracks

98, 137, 122, 183, 14, 133, 65, 78

Head pointer starting at 54 and moving in left direction. Find the number of head movements in cylinders using SCAN scheduling.



Number of Cylinders = 40 + 14 + 65 + 13 + 20 + 24 + 11 + 4 + 46 = 237

C-SCAN algorithm

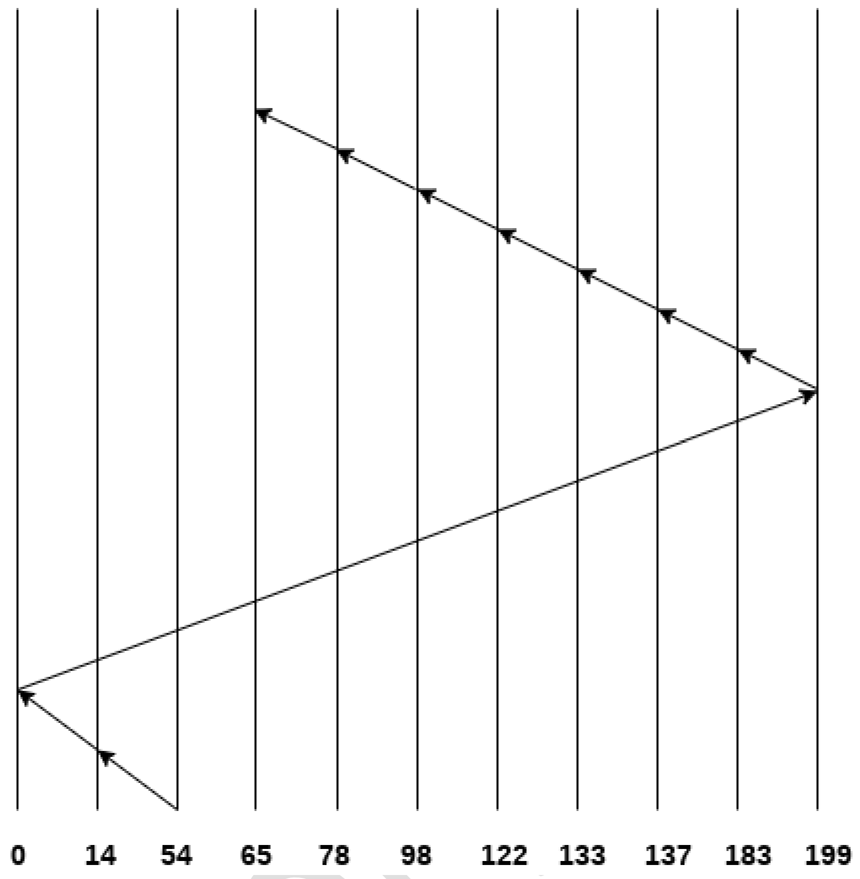
In C-SCAN algorithm, the arm of the disk moves in a particular direction servicing requests until it reaches the last cylinder, then it jumps to the last cylinder of the opposite direction without servicing any request then it turns back and start moving in that direction servicing the remaining requests.

Example

Consider the following disk request sequence for a disk with 100 tracks

98, 137, 122, 183, 14, 133, 65, 78

Head pointer starting at 54 and moving in left direction. Find the number of head movements in cylinders using C-SCAN scheduling.



No. of cylinders crossed = 40 + 14 + 199 + 16 + 46 + 4 + 11 + 24 + 20 + 13 = 387