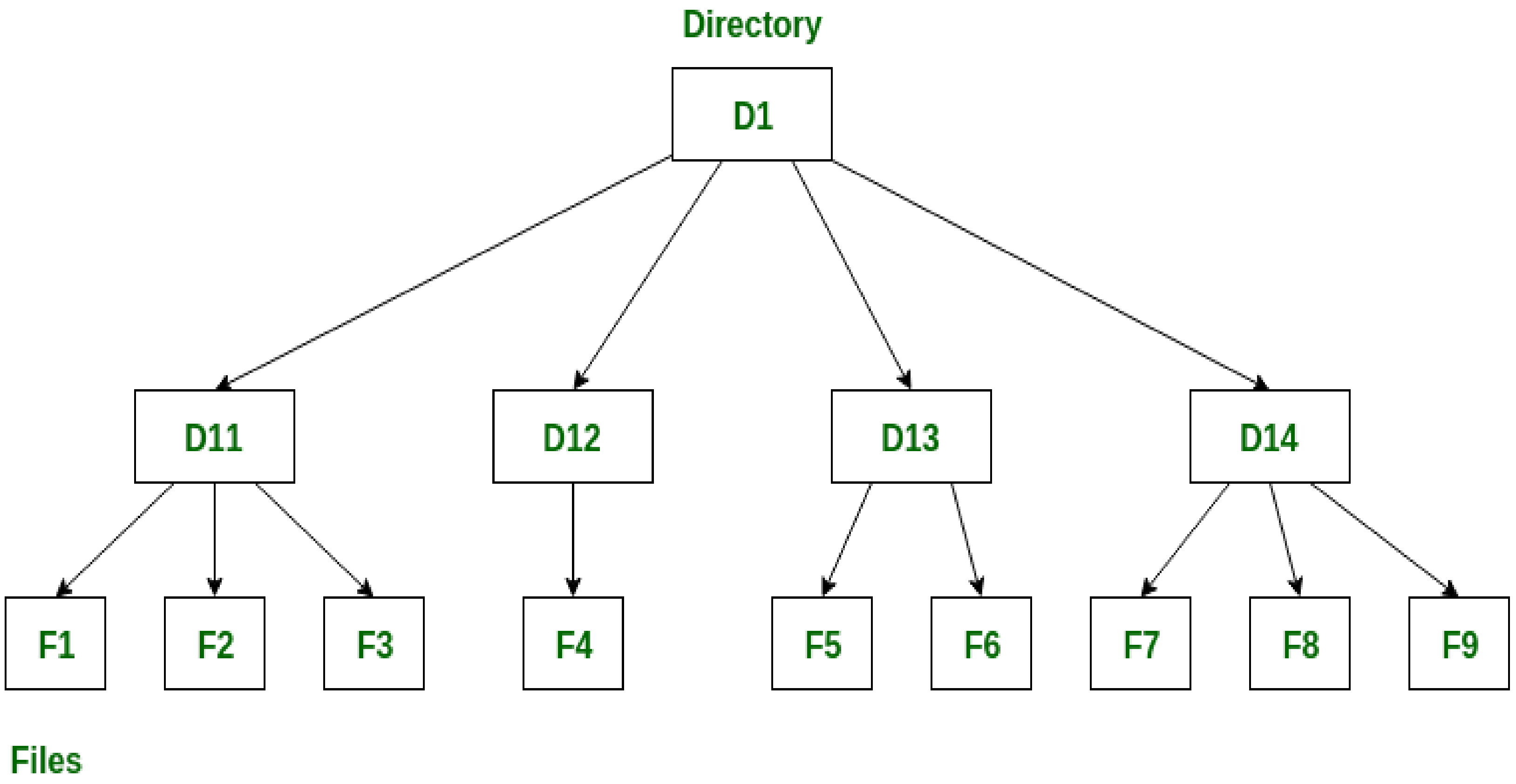
Structures of Directory in Operating System:

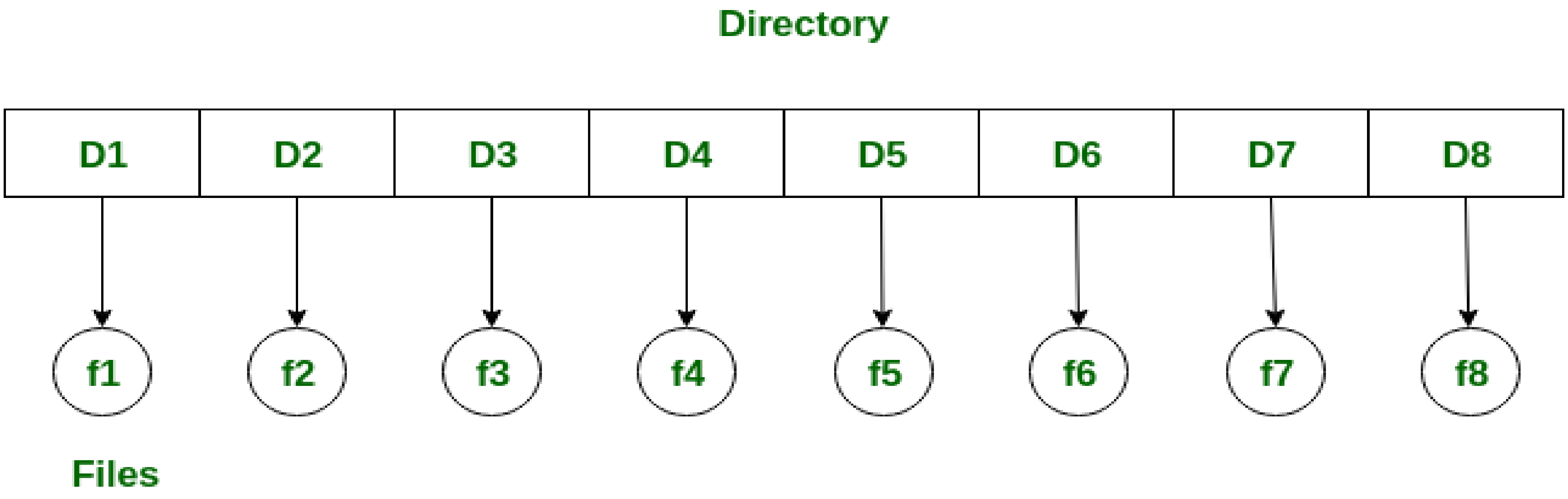
A **directory** is a container that is used to contain folders and file. It organizes files and folders into a hierarchical manner.



There are several logical structures of a directory, these are given here.

1. Single-level directory –

- **Single level directory structure** has only one directory which is called the **root** directory.
- The users are not allowed to create subdirectories under the root directory.
- All the files created by the several users are present in the root directory only.



Advantages:

- Since it is a single directory, so its implementation is very easy.
- If the files are smaller in size, searching will become faster.

- The operations like file creation, searching, deletion, updating are very easy in such a directory structure.

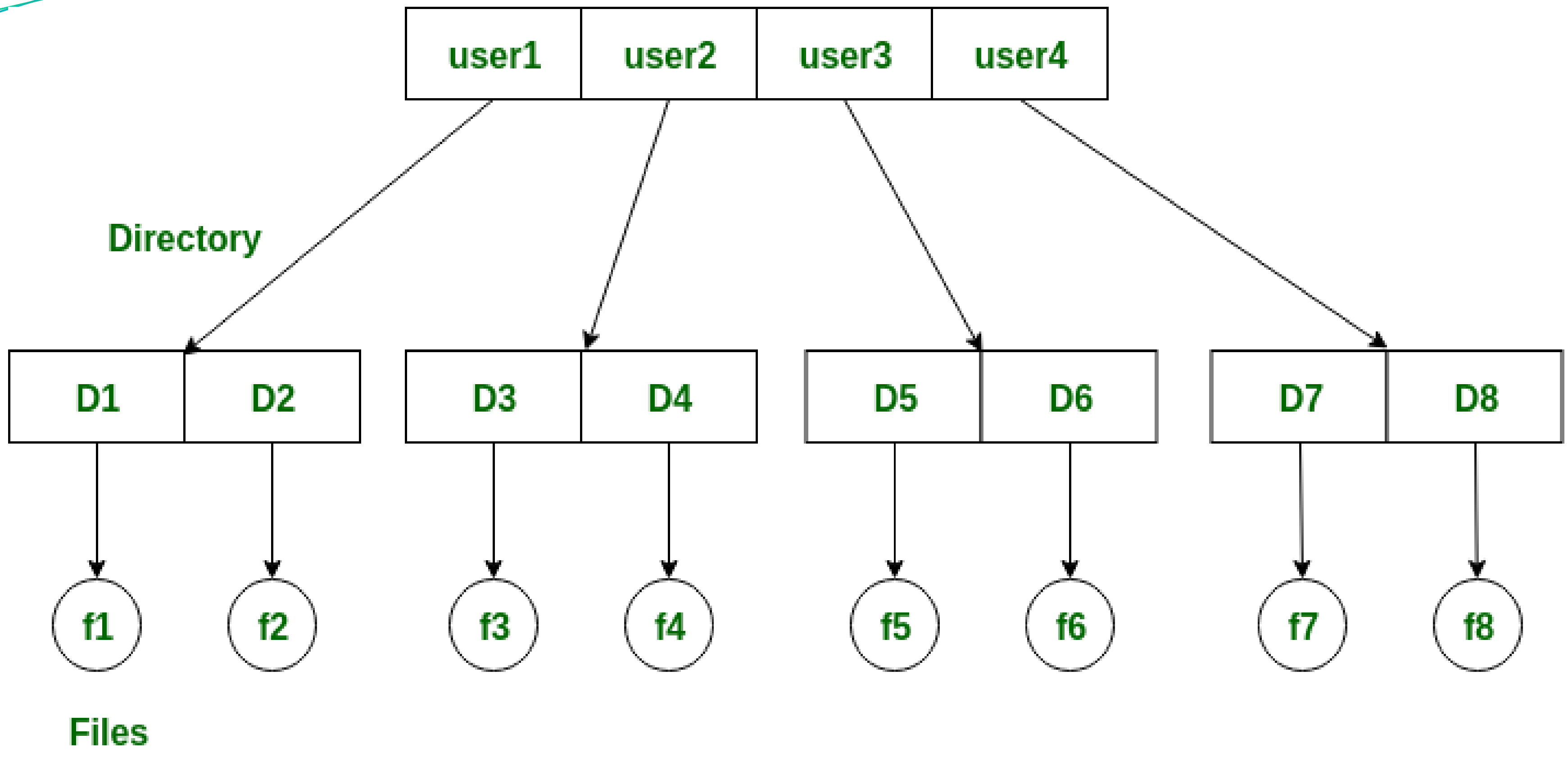
Disadvantages:

- There may chance of name collision because two files can not have the same name.
- Searching will become time taking if the directory is large.
- In this can not group the same type of files together.

1. Two-level directory –

As we have seen, a single level directory often leads to confusion of files names among different users. the solution to this problem is to create a separate directory for each user.

- In **Two-level directory structure,** the users create directory directly inside the root directory.
- But once a user creates such directory, further he cannot create any subdirectory inside that directory.
- This two-level structure allows each user to keep their files separately inside their own directory. This structure allows to use the same name for the files but under different user directories.



Advantages:

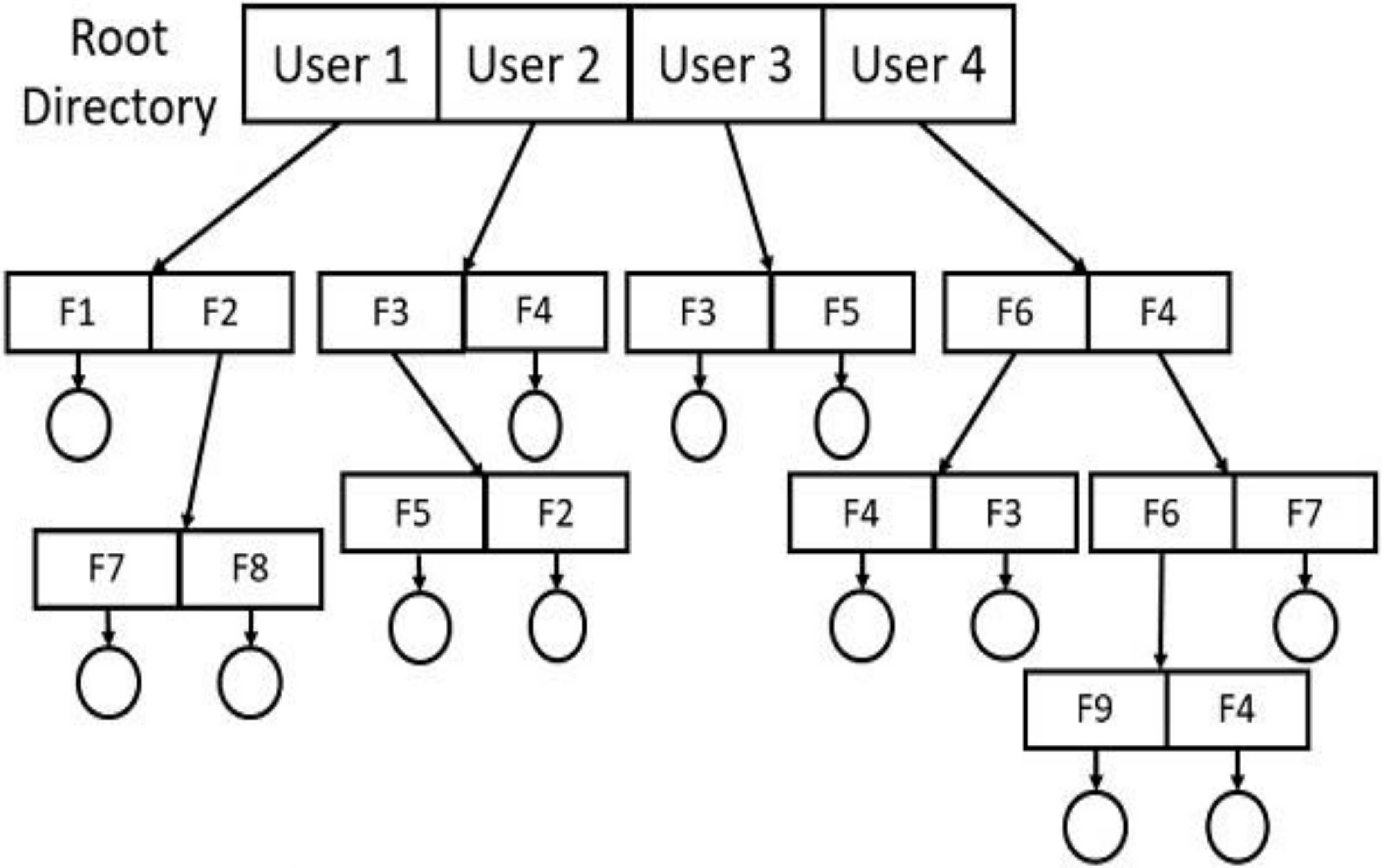
- We can give full path like /User-name/directory-name/.
- Different users can have same directory as well as file name.
- Searching of files become more easy due to path name and user-grouping.

Disadvantages:

- A user is not allowed to share files with other users.
- Still it not very scalable, two files of the same type cannot be grouped together in the same user.

3. Tree-structured directory –

- In a tree directory structure, except root directory, every directory or file has only **one parent directory**.
- So, there is a total separation between the users which provide complete naming freedom.
- Here, if a user wishes to access another users file, it has to go through two or more directories.
- The tree directory structure provides an asymmetric way for the user to access the shared files of a different user.
- For **example**, a user can access a file of its own user directory with a shorter path than the other user.



Tree Directory Structure

A tree structure is the most common directory structure. The tree has a root directory, and every file in the system have a unique path.

Advantages:

- Very generalize, since full path name can be given.
- Very scalable, the probability of name collision is less.
- Searching becomes very easy, we can use both absolute path as well as relative.

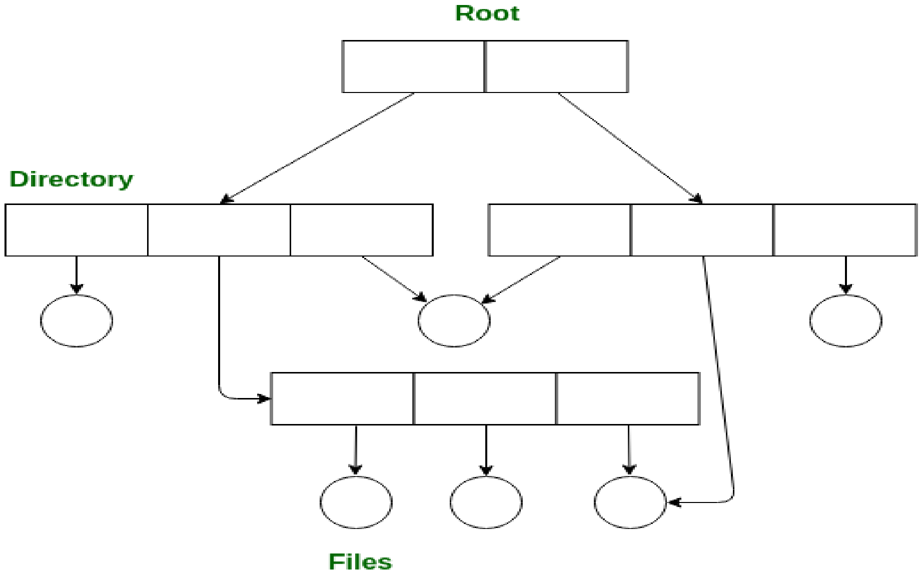
Disadvantages:

- Every file does not fit into the hierarchical model, files may be saved into multiple directories.
- We can not share files.
- It is inefficient, because accessing a file may go under multiple directories.

4. Acyclic graph directory –

An acyclic graph is a graph with no cycle and allows to share subdirectories and files. The same file or subdirectories may be in two different directories. It is a natural generalization of the tree-structured directory.

- It is used in the situation like when two programmers are working on a joint project and they need to access files.
- The associated files are stored in a subdirectory, separating them from other projects and files of other programmers, since they are working on a joint project so they want the subdirectories to be into their own directories.
- The common subdirectories should be shared. So here we use Acyclic directories.
- It is the point to note that shared file is not the same as copy file . If any programmer makes some changes in the subdirectory it will reflect in both subdirectories.



- Shared files and subdirectories can be implemented in several ways. A common way is to create a new directory entry called a link. A **link** is a **pointer** to another file or subdirectory.

Advantages:

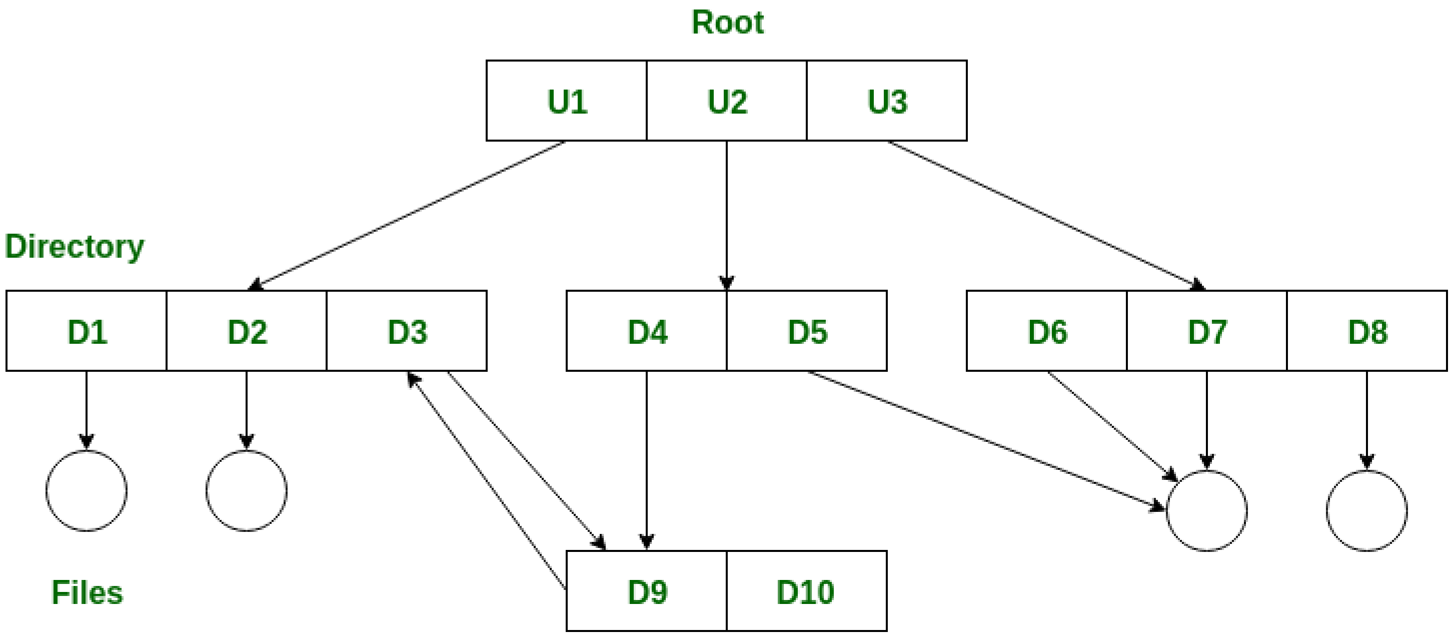
- We can share files.
- Searching is easy due to different-different paths.

Disadvantages:

- We share the files via linking, in case of deleting it may create the problem,

5. General graph directory structure –

- In general graph directory structure, cycles are allowed within a directory structure where multiple directories can be derived from more than one parent directory.
- The main problem with this kind of directory structure is to calculate total size or space that has been taken by the files and directories.



Advantages:

- It allows cycles.
- It is more flexible than other directories structure.

Disadvantages:

- It is more costly than others.
- It needs garbage collection.

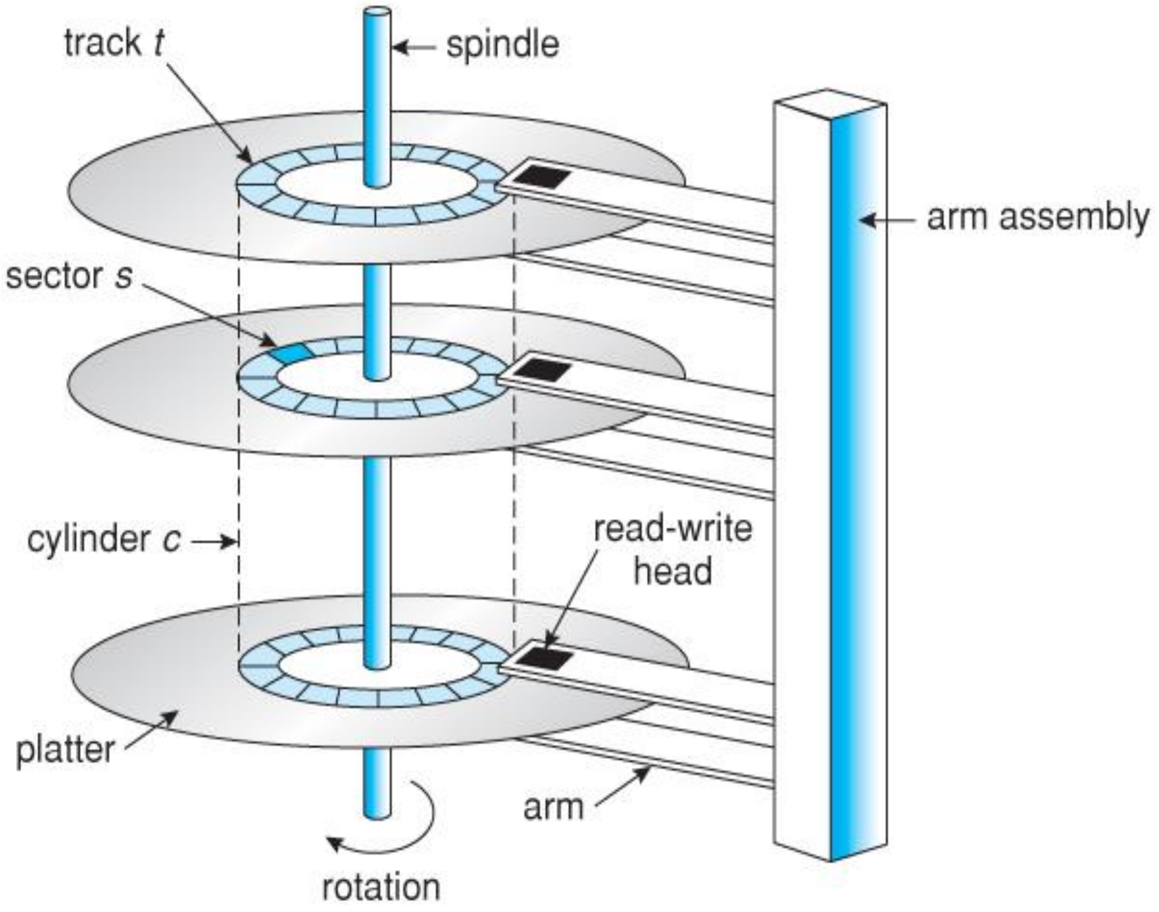
SECONDARY STORAGE STRUCTURE

Secondary storage devices are those devices whose memory is non volatile, meaning, the stored data will be intact even if the system is turned off. Here are a few things worth noting about secondary storage.

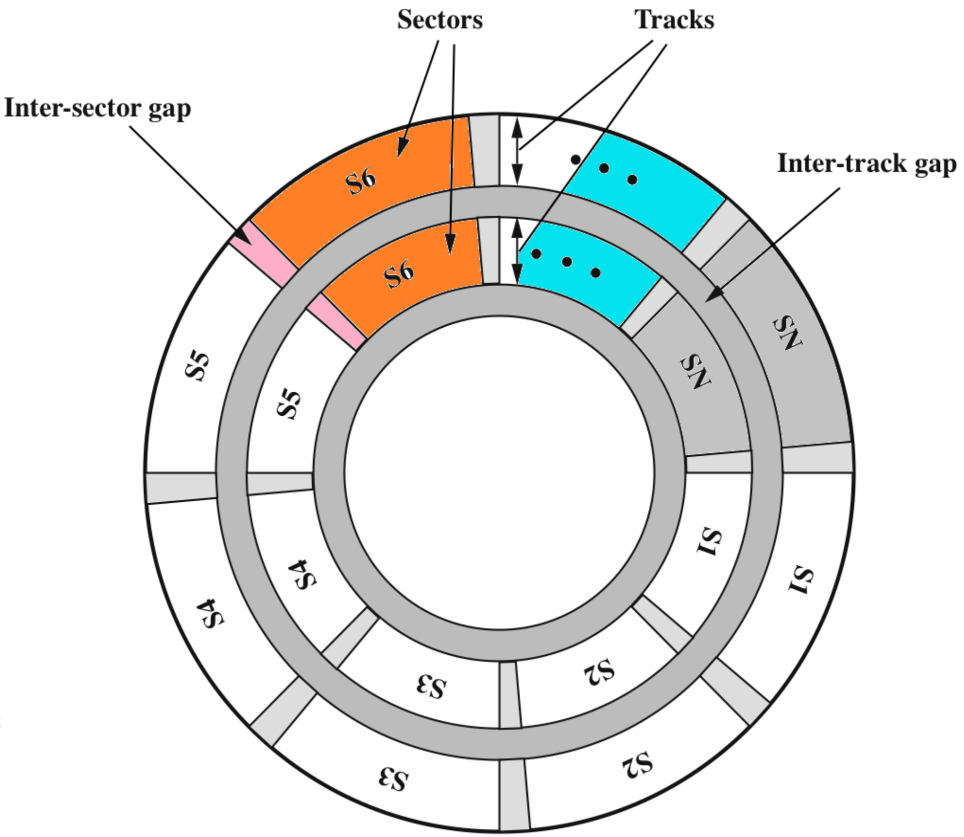
- Secondary storage is also called auxiliary storage.
- Secondary storage is less expensive when compared to primary memory like RAMs.
- The speed of the secondary storage is also lesser than that of primary storage.
- Hence, the data which is less frequently accessed is kept in the secondary storage.
- A few examples are magnetic disks, magnetic tapes, removable thumb drives etc.

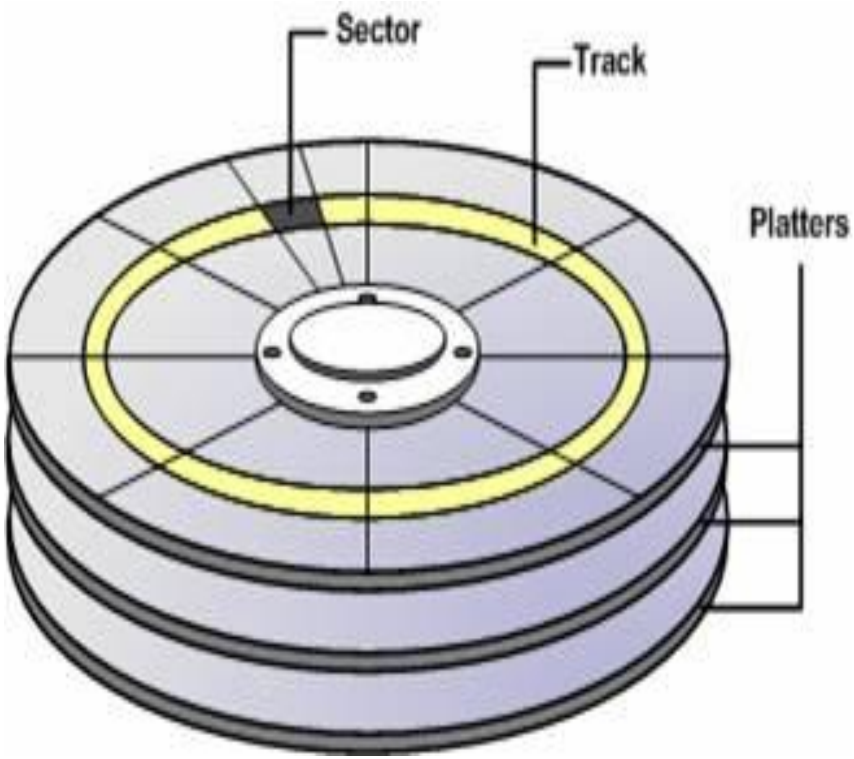
Magnetic Disk Structure

In modern computers, most of the secondary storage is in the form of magnetic disks. Hence, knowing the structure of a magnetic disk is necessary to understand how the data in the disk is accessed by the computer.



- A magnetic disk contains several **platters**. Each platter is divided into circular shaped **tracks**. The length of the tracks near the centre is less than the length of the tracks farther from the centre. Each track is further divided into **sectors**, as shown in the figure.


Sectors
Tracks
Inter-sector gap
Inter-track gap
S6
S6
S5
S5
S4
S4
S3
S3
S2
S2
S1
S1
SN
SN


Sector
Track
Platters

Tracks of the same distance from centre form a cylinder. A read-write head is used to read data from a sector of the magnetic disk.

The speed of the disk is measured as two parts:

- **Transfer rate:** This is the rate at which the data moves from disk to the computer.
- **Random access time:** It is the sum of the seek time and rotational latency.

Seek time is the time taken by the arm to move to the required track. **Rotational latency** is defined as the time taken by the arm to reach the required sector in the track.

Even though the disk is arranged as sectors and tracks physically, the data is logically arranged and addressed as an array of blocks of fixed size.

The size of a block can be **512** or **1024** bytes. Each logical block is mapped with a sector on the disk, sequentially. In this way, each sector in the disk will have a logical address.