

File Concept

Operations on File

File Access Methods (Sequential Access and Direct Access)

Directory Systems File Management Functions.

File System and Directory Structure organization.

File Protection.

File Concept

What is File?

- A file is a named collection of related information that is recorded on secondary storage such as magnetic disks, magnetic tapes and optical disks.
- Files are mapped by the operating system on the physical devices. These devices are usually non-volatile.
- A file is logical storage unit of secondary storage. In general, a file is a sequence of bits, bytes, lines or records whose meaning is defined by the files creator and user.
- A file can be defined as a data structure which stores the sequence of records. Files are stored in a file system, which may exist on a disk or in the main memory. Files can be simple (plain text) or complex (specially-formatted).
- The collection of files is known as Directory. The collection of directories at the different levels, is known as File System.

File Attribute

A file attribute vary from one computer to another but typically consists of following:

- 1. Name:** Every file carries a name by which the file is recognized in the file system. One directory cannot have two files with the same name.
- 2. Identifier:** Along with the name, Each File has its own extension which identifies the type of the file. For example, a text file has the extension **.txt**, A video file can have the extension **.mp4**.
- 3. Type:** In a File System, the Files are classified in different types such as video files, audio files, text files, executable files, etc.
- 4. Location:** In the File System, there are several locations on which, the files can be stored. Each file carries its location as its attribute.
- 5. Size:** The Size of the File is one of its most important attribute. By size of the file, we mean the number of bytes acquired by the file in the memory.
- 6. Protection:** The Admin of the computer may want the different protections for the different files. Therefore each file carries its own set of permissions to the different group of Users.
- 7. Time and Date:** Every file carries a time stamp which contains the time and date on which the file is last modified.

Operations on File

- A file is an abstract data type.
 - To define file properly, we need to consider the operations that can be performed on files.
 - Operating system provide following 6 basic operation
1. Creating File
- Two steps are necessary to create File

- ✓ First, a space in the file system must be found for the file.
 - ✓ Second, an entry for the new file must be made in the directory. The directory entry records the name of the file and the location in the file system.
2. Writing File
 - ✓ To write a file, a system call is made that specifying the name of the file and the information to be written to the file.
 - ✓ Given the name of the file, the system searches the directory to find the location of the file.
 - ✓ The directory entry will need to store a pointer to the current block of the file (usually the beginning of the file). Using this pointer, the address of the next block can be computed where the information will be written.
 - ✓ The write pointer must be updated – in this way; successive writes can be used to write a sequence of blocks to the file.
 3. Reading File
 - ✓ To read a file, a system call is made that specifies the name of the file and where (in memory) the next block of the file should be put.
 - ✓ Again, the directory is searched for the associated directory entry, and the directory will need a pointer to the next block to be read. Once the block is read, the pointer is updated.
 4. Repositioning within file
 - ✓ The directory is searched for the appropriate entry, and the current-file position pointer is reposition to a given value.
 - ✓ Repositioning within file need not involve any actual I/O.
 - ✓ This file operation is also known as a file seeks.
 5. Deleting File
 - ✓ To delete a file, the directory is searched for the named file.
 - ✓ Having found the associated directory entry, the space allocated to the file is released (so it can be reused by other files) and remove from the directory entry
 6. Truncating File
 - ✓ The user wants to erase content of file for that its attribute unchanged.
 - ✓ Rather than forcing the user to delete file and then recreate it, this function allows all attributes to remain unchanged- except for file length but file be reset to length zero and its file space released.

Other common operations like appending new information to end of existing file and renaming an existing file

File Types

File type refers to the ability of the operating system to distinguish different types of file such as text files source files and binary files etc.

1. Text file
 - Sequence of character formed as line
 - Extension .txt, .doc
2. Source File
 - Sequence of subroutine and functions

Extension .c, .cpp, .java

3. Object File

Sequenced of byte formed as block.

Extension .obj, .o, .class

4. Executable File

Series of codes section which is in binary form that can bring into memory and execute it.

Extension .exe, .bin, .com

5. Batch File

A computer files containing a list of instructions to be carried out in turn.

Extension .bat, .sh

6. Word Processor

Extension .wp, .rtf, .doc

7. Library

A library is a collection of precompiled sub routines that a program can use.

Extension .lib, .dll

8. Print or view

Ascii or binary file in format for printing or viewing.

Extension .pdf, .jpg

9. Archive

Related file grouped into one file, sometimes compressed, for archiving or storage

Extension .arc, .zip

10. Multi media

Binary file contains audio or video information.

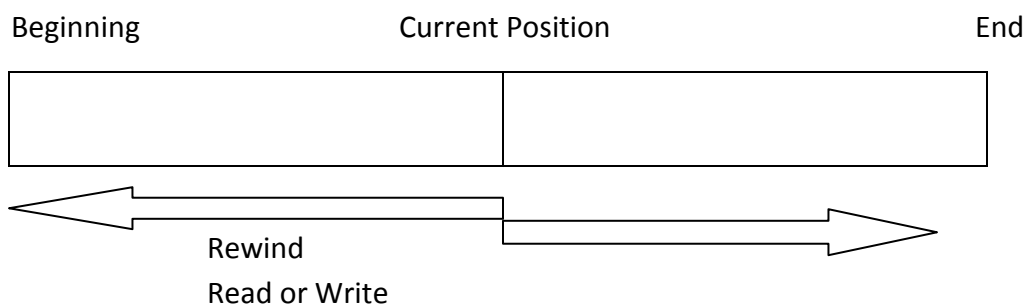
Extension .mp3, .avi

File Access Method

- ✓ File store information. When it is used, this information must be accessed and read into computer memory.
- ✓ The information in the file can be accessed in several ways.

1. Sequential Access

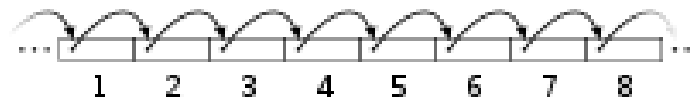
- ✓ The simplest access method is sequential access method.
- ✓ Information in the file is processed in order, one record after the other.



- ✓ Read and writes make up bulk of operation on a file.

- ✓ Read operation call “read next” i.e. it read file from beginning and then reads next portion of the file and automatically set file pointer, which tracks the I/O location.
- ✓ Similarly, Write operation call “write next” i.e. it appends to the end of file and set file pointer at end of newly written material.
- ✓ Reset to the beginning.
- ✓ Programmer can skip forward or backward n records.
- ✓ Example, editors and compilers usually access file in this fashion.
- ✓ Application of sequential access method is payroll employees and student record processing.

Sequential access



Advantage

- ✓ Here record maintain sequence so if we want to find specific record, we can find it easily.

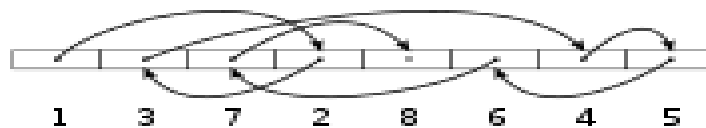
Disadvantage

- ✓ It is time consuming because to find specific record, whole file have to scan from beginning to matching record or end of file.

2. Direct access or random access

- ✓ A file made up of fixed length logical records that allow programmer to read or write records rapidly in no particular order.
- ✓ It is based on the disk model of file i.e. disk allow random access to any file block.
- ✓ Each record has its own address on the file which it can be directly accessed for reading or writing.
- ✓ The records need not be in any sequence within the file

Random access



- ✓ For Example, the file viewed as a numbered sequence of blocks or records. So we can read block 14 then block 53 then block 7 that means there is no restriction on reading or writing file.
- ✓ Direct access files are use for immediate access to large amounts of information. Databases are often using this type.
- ✓ Easy to read write and delete a record
- ✓ The file operation must be modified to include the block number as parameter. Thus we have read N where N is block number and L is length of record
- ✓ So address of N record is $L*(N-1)$

- ✓ Similarly we have write N where N is block number and L is length of record.
- ✓ Easy to simulate sequential access on direct access file.

Advantage

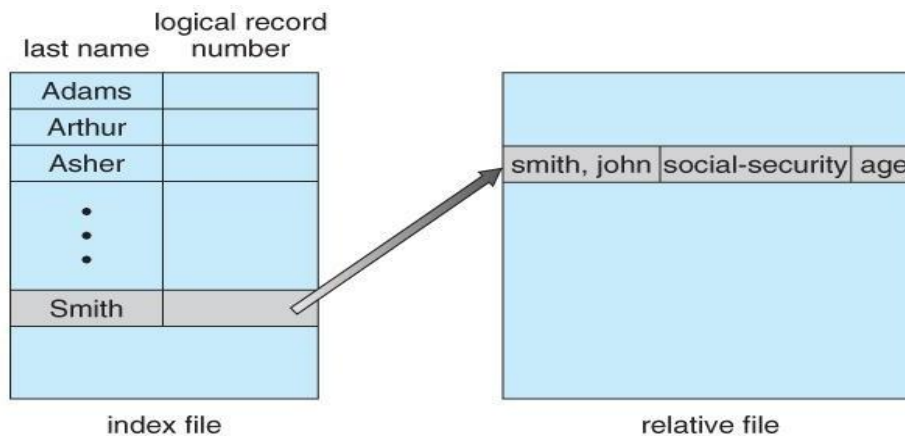
- ✓ It retrieves information faster than Sequential.

Disadvantage

- ✓ It is very difficult to find particular record.

3. Index sequential access

- ✓ This method generally involves construction of an index of file.
- ✓ The index contain pointer to the various blocks.



- ✓ To find a record in the file we first search an index and then use the pointer to access a file directly and to find a desire record.
- ✓ With large file the index may become large to keep in memory.
- ✓ One solution is to create an index for the index file; the primary index file would contain a pointer to the secondary index file which would point to actual data items.
- ✓ It is the combination of sequential and direct access method.

Advantage

- ✓ We can get accurate result.

Disadvantage

- ✓ To find a particular item, we first make a binary search of master index which provides the block number of secondary index. The block is read and again binary search is used to find the block containing desire records. Finally this block search sequentially.

Directory System File Management Functions

Information about files is maintained by Directories. A directory can contain multiple files. It can even have directories inside of them. In Windows we also call these directories as folders.

The directory can be view as a symbol table that translates file name into their directory entries.

1. Search for a file

- ✓ We need to be able to search a directory structure to find the entry for a particular file.

2. Create a file

- ✓ New file need to be created and added to the directory

3. Delete a file

- ✓ When a file no longer needed, we want able to remove it from directory.

4. List a directory

- ✓ We need to be able to list the files in a directory and contents of the directory entry for each file in the list.

5. Rename a file

- ✓ Because the name of file represents its content to its user, the name must be changeable when the content or use of the file changes.
- ✓ Renaming a file may also allow its position within the directory structure to be changed.

6. Traverse file

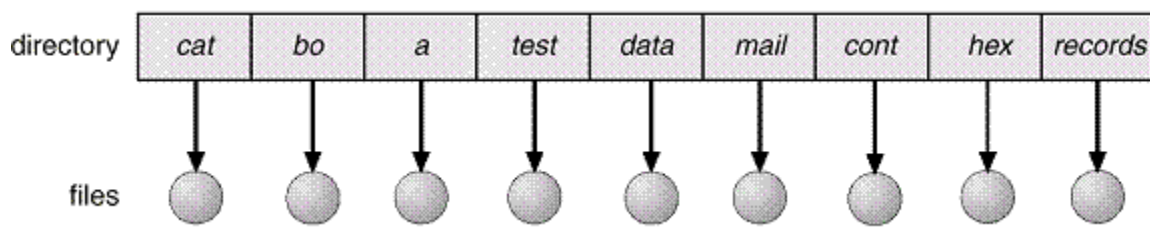
- ✓ It is useful to be able to access every directory and every file within a directory structure.

File System and Directory Structure Organization.

- ✓ A file is saving on secondary storage to manage all these data so we need to organize them.
- ✓ This organization involves the use of directory.

1. Single level Directory

- ✓ It is very common on single user operating system.
- ✓ It is simple of all directory structures.
- ✓ In single level directory structure all files are under one directory.

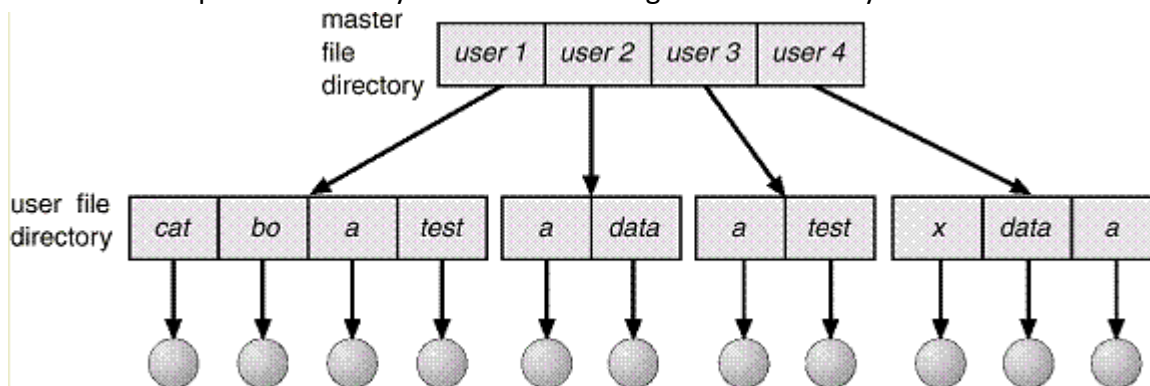


Limitations:

- ✓ As there is only one directory and number of files is there then it becomes difficult for the user to remember all the names.
- ✓ If there is more than one user then this problem occurs.
- ✓ All files must have unique names and it may quite possible that different users give same file name, and then the problem of name collision occurs.
- ✓ We can't give descriptive name to the file because the identifiers length is limited.

2. Two level directory

- ✓ The major disadvantage of single level directory is name collision so to overcome from this we can create a separate directory for each user using 2 level directory structures.



- ✓ It has one master file directory (MFD) and in that user file directory (UFD).
- ✓ All UFDs have similar structure, but each directory (UFD) lists files of a single user.
- ✓ When a user job starts or user logs in, the system master file directory (MFD) is searched.
- ✓ MFD is indexed by username and account number and each entry point to the UFD for that user.
- ✓ When user read to particular file, only his/her own UFD searched.
- ✓ To create a file for a user, operating system searches only that user's UFD to check whether another file of that name exists or not.
- ✓ To delete a file, operating system limits its search to the local UFD so that it cannot accidentally delete another user's file.
- ✓ Every file in system has a path which is combination of user name and file name.

Advantages:

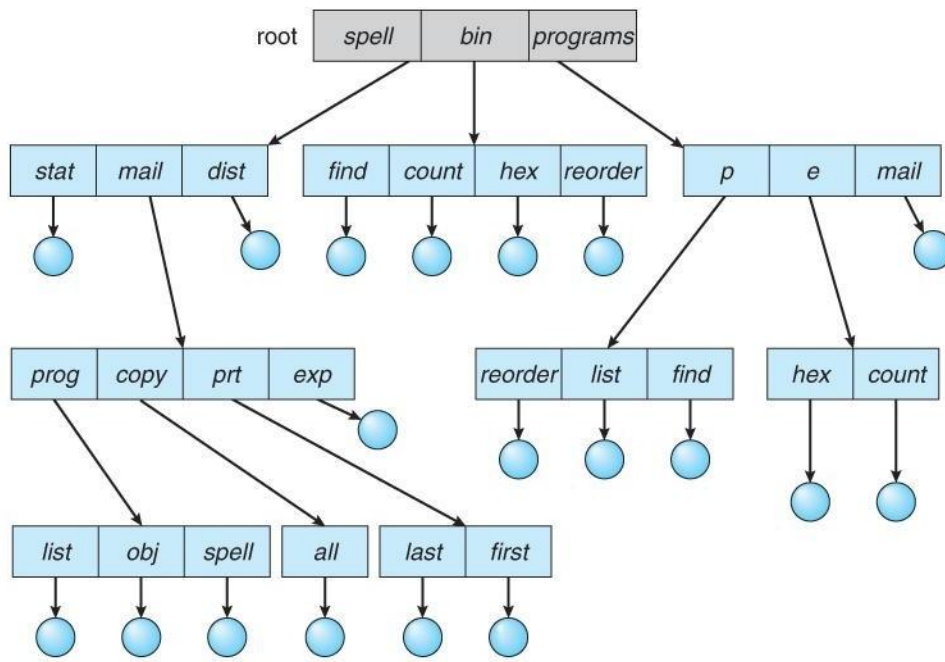
- ✓ It solves the problem of name collision at user level.
- ✓ Isolates one user from another, when user is creating or naming a file.

Disadvantages:

- ✓ To name a file uniquely, user must know the desire path.
- ✓ When user want to share file another users it is not possible.

3. Tree level directory

- ✓ The Most common example of tree level directory structure is MS-DOS (Micro Soft Disk Operating System)
- ✓ It has root directory which containing sub-directories and files.



- ✓ Every file in this system has unique path name.
- ✓ Path name- It is start from root directory to all subdirectories and file name which was specified file location.
- ✓ All directories have some internal format that is if sub-directory is created entry will be 1 bit and if file is created entry will be 0 bit.
- ✓ If we want to search particular file, it will first search in current directory and then it is not found then search will go to the path and current directory will be change to where the file is found.
- ✓ Absolute path :
 - Path name begin at the root and follow the path down to specific file.
 - E.g. d:\sem2\c\abc.c
- ✓ Relative path :
 - Define path from current directory.
 - E.g. .\c\abc.c
 - Where your current directory is sem2.

Advantages:

- ✓ It solves the problem of name collision

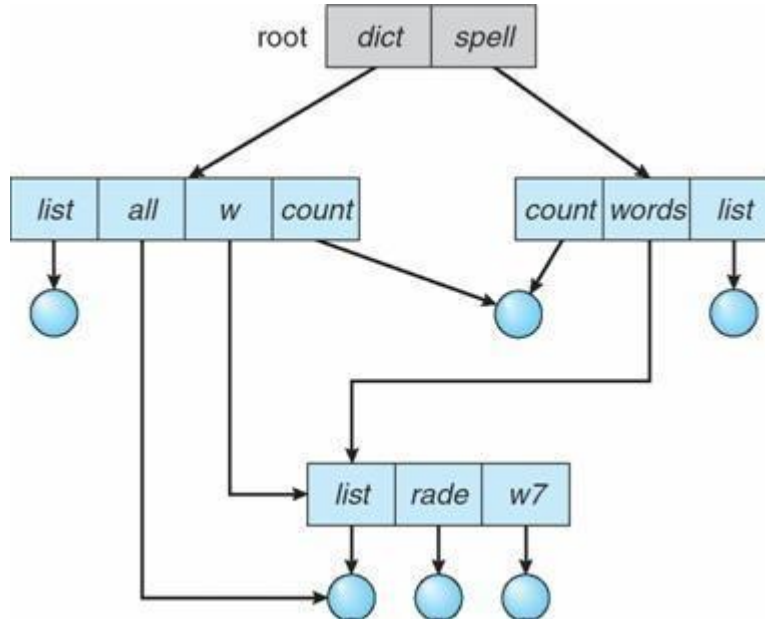
Disadvantages:

- ✓ If we delete a directory then a directory should be empty.
- ✓ We cannot share file and directories.

4. Acyclic graph directory structure

- ✓ It is a generalization form of tree structure directories.
- ✓ The common sub-directories or file should be shared ie. A shared file or directory will exists more than one place.

- ✓ A tree structure does not provide sharing of files or sub-directories where acyclic graph allows directory to share sub-directories or file with no cycle.



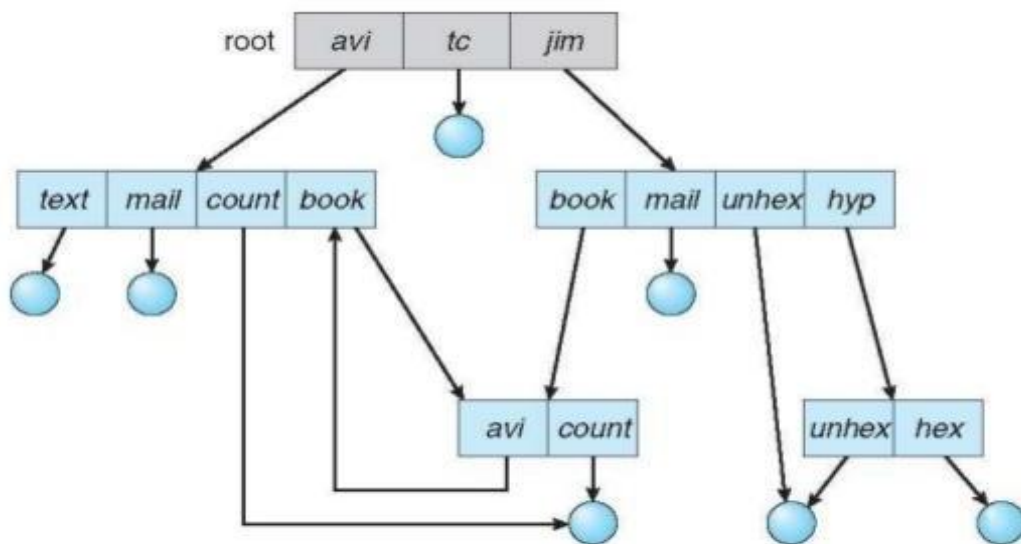
- ✓ Sharing means the same file or directory may be in 2 different directories. But same file or directory is not same as 2 copy of file or directory.
- ✓ With two copies :
 - Each user view only copy, not original
 - If a user changes the file, the changes will not appear in the other's copy.
- ✓ With share File
 - Only one actual file Exists
 - If user changes the file, another user immediately visible that changes.
- ✓ A common way to create a new directory entry calls **Link**.
- ✓ A link is effectively a pointer to another file or sub directory.
- ✓ When a reference to a file is made, we search the directory. If the directory entry is marked as a link then name of real file include in the link information.
- ✓ When resolve the link by using that path name to locate the real file links are easily identify by their format in the directory entry and are effectively named indirect pointers.
- ✓ Another common approach to implementing a share file to duplicate all the information about their in both sharing directories thus both entries are identical and equal.
- ✓ A link is clearly different from original directory entry. Thus two are not equal.
- ✓ A major problem with duplicate directory entries is maintaining consistency when a file is modified.
- ✓ Through this approach, it involve problem at deletion :
 - When one removes a file which is original, this action may leave dangling pointer to the non-existent files.
 - If the remaining file pointer contain actual disk address, the space is use for another file.
- ✓ For a shared file the directory entry is marked as a link
- ✓ A link may be implemented as an absolute path name or relative path name (a symbolic link)

- ✓ When a shared file is referenced, the link is resolved
- ✓ Links are identified by their special format or by their having a special type
- ✓ Share file system is implemented by symbolic links, this method is easier to handle
 - Here, we no need to keep entire list, we need to keep only number of link or references.
 - Adding a new link entry, increment link count.
 - Deletion of link need not to affect original file, only link is deleted.
 - If the last entry of file is deleted then only space for the file is reallocated.

5. General graph directory structure

- ✓ When links are added to an existing tree-structured directory, a general graph structure general graph structure can be created.

General Graph Directory



- ✓ One serious problem with using an acyclic graph structure –how to ensure that there are no cycles.
 - Can happen only when linking a directory.
 - Every time a link is added to directory- use a cycle detection algorithm.
- ✓ If cycles are allowed to exists in the directory, following problems occur
 - We want to avoid **searching** any component twice for correction and performance.
 - Poorly designed algorithms create **infinite loopie** continuously searching through cycles and never terminating.
 - It is more costly and need a garbage collection.
- ✓ One solution is to limit the number of directory which will be access during a search.
- ✓ Garbage collector involves traversing the entire file system. Marking everything that can be accessed.
- ✓ Garbage collector is necessary only because of possible cycle in a graph.
- ✓ How do we guarantee no cycles?
 - Allow only links to files not subdirectories.
 - Use Garbage collection.

- Every time a new link is added, use a cycle detection algorithm to determine whether a cycle exists.

File Protection

- When information is kept in a computer system, a major concern is its protection from physical damage (reliability) and improper access
- Reliability is generally provided by duplicate copies of files.
- In multiuser system, there are 3 ways of protection

1. Types of access

- ✓ The need for protection a file is a direct result of the ability to access the files.
- ✓ Protection mechanism provides control access by limiting the types of file access that can be made.
- ✓ Access is permitted or denied depending on several factors.
- ✓ Several different types of operations may be controlled:
 1. Read : read from a file
 2. Write: write or rewrite a file.
 3. Execute: Load the file into memory and execute it.
 4. Append: write new information at the end of file.
 5. Delete: Delete the file and free its space for reuse.
 6. List: List the name and attributes of the file.

Other operations can be renaming, copying, editing the file may also be controlled.

Protection is provided at only lower level (system level).

2. Access List and Groups

- ✓ The most common approach to the protection problem is to make access dependant on the identity of the user.
- ✓ Various users may need different types of access to a file or directory.
- ✓ To implement this mechanism is to associate with each file and directory access list, specifying the user name and types of access allowed for each user.
- ✓ When user request access to a particular file the operating system checks the access list associate with that file.
- ✓ If the user is listed for the requested access, the access is allowed. Otherwise the protection violation occurs.
- ✓ The main problem with access list is length. If we want to allow everyone to read a file, we must list all the users with read access.
- ✓ To solve the length of access list, many systems recognize this classification of users in connection with each file
 1. Owner: the user who creates a file is owner.
 2. Group: A set of users who are sharing the file and need similar access is a group or work group.
 3. Universe: All other users in the system.

3. Other Protection Approach

- ✓ There is another approach to the protection problem which is to associate with a password to each file. To access a computer system itself often controlled by a password; access to each file can be controlled by a password.
- ✓ **There are several Disadvantages for this schemas**
 1. If we associated a separate password with each file, the number of passwords that a user needs to remember may become large.
 2. If only one password is used for all the files then once it is discovered, all files are accessible.
 3. Some system allows a user to associate a password with a subdirectory rather than with an individual file to deal with this problem.

Differentiate between Access Method

No.	Sequential	Direct	Index Sequential
1	Sequential access must begin at the beginning and access each element in order, one after the other.	Direct access allows the access of any record directly by locating it by its address.	Access of any record first search a record in index and then use pointer to access record directly.
2	Duplicate Data is allowed	Duplicate Data is not allowed	Duplicate data is not allowed in index
3	Record is not sorted in order	Record is in sorted order based on Relative address	Record is in sorted order based on key.
4	Access is slow	Access is faster than Index sequential.	Access is faster but slower than Direct.
5	Records length can be very	Records length must be same	Records length can be very
6	This method is used to access data from Tap	This method is used to access data from CD-DVD	This method is used to access data from DBMS system

Differentiate between Allocation Method

No.	Contiguous	Linked	Index
1	All blocks of the file are consecutive on disk	Linked list of file blocks	File block addresses are stored in an array which is stored in a disk block
2	Sequential and direct access method well supported but much more effective for sequential access	Effective for sequential access but Problematic with random access	Sequential and direct access methods well supported but much more effective for direct access
3	External Fragmentation Problem Occur	External Fragmentation Problem Resolve	External Fragmentation Problem Resolve
4	Difficult to grow File	Easily File can be expanded	Easily File can be expanded up to max size of index
5	File name, start and length field store for each file entry	File name, start and End field store for each file entry	File name and Index block field store for each file entry
6	Major problem For creating file, How much space allocate to file	No need to declare the size of file when file create but each block occupy pointer reference of next block	Major problem with how long index block when file create.
7	Efficient for small file	Efficient for any size of file	Inefficient for small file