IDHAYA COLLEGE FOR WOMEN

KUMBAKONAM - 612 001

PG & RESEARCH DEPARTMENT

OF

COMPUTER SCIENCE

ACADEMIC YEAR	:	2019-2020
SEMESTER	:	VI
CLASS	:	III B. SC (CS)
SUBJECT INCHARGE	:	R. KANNIKA DEVI
SUBJECT NAME	:	OPERATING SYSTEMS
SUBJECT CODE	:	16SCCCS8

UNIT- V

FILE MANAGEMENT

The File Manager, Interacting with the File Manager, File Organization, Physical Storage Allocation, Access Methods, Levels in a File Management System, Access Control Verification Module.

File Management:

File:

- A file is a group of related records that contains information to be used by specific application programs to generate reports. This type of file contains data and is sometimes called a flat file because it has no connections to other files.
- File management is one of the basic and important features of operating system. Operating system is used to manage files of computer system. All the files with different extensions are managed by operating system.

  The following are some of the tasks performed by file management of operating system of any computer system:
- It helps to create new files in computer system and placing them at the specific locations.
- It helps in easily and quickly locating these files in computer system.
- It makes the process of sharing of the files among different users very easy and user friendly.
- It helps to stores the files in separate folders known as directories. These directories help users to search file quickly or to manage the files according to their types or uses.
- It helps the user to modify the data of files or to modify the name of the file in the directories.

  For example: The file system is like a library, with the File Manager playing the part of the librarian who performs the same four tasks.
- A librarian uses the catalog to keep track of each item in the collection; each entry lists the call number and the details that help patrons find the books they want.
- The library relies on a policy to store everything in the collection including oversized books, magazines, books-on-tape, DVDs, maps, and videos. And they must be physically arranged so people can find what they need.
- When it's requested, the item is retrieved from its shelf and the borrower's name is noted in the circulation records.
- When the item is returned, the librarian makes the appropriate notation in the circulation records and reshelve it.
- The computer system allocates a file by activating the appropriate secondary storage device and loading the file into memory while updating its records of who is using what file.
- The File Manager deallocates a file by updating the file tables and rewriting the file

(if revised) to the secondary storage device. Any processes waiting to access the file are then notified of its availability.

Field A	Field B	Field C	Field D	Record 19
Field A	Field B	Field C	Field D	Record 19
Field A	Field B	FieldC	Field D	Record 19
Field A	Field B	Field C	Field D	Record 19

- Directories are special files with listings of filenames and their attributes. Data collected to monitor system performance and provide for system accounting is collected into files.

Interaction with the file Manager:

- The user communicates with the File Manager, which responds to specific commands are OPEN, DELETE, RENAME, and COPY.
- Files can be created with other system-specific terms: for example, the first time a user gives the command to save a file, it's actually the CREATE command. In other operating systems, the OPEN NEW command within a program indicates to the File Manager that a file must be created.

(Menu of File options)

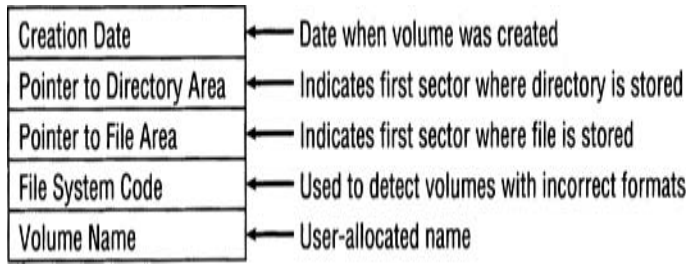
Example:

- when a user's program issues a command to read a record from a disk the READ instruction has to be decomposed into the following steps:

- Move the read/write heads to the cylinder or track where the record is to be found.
- Wait for the rotational delay until the sector containing the desired record passes under the read/write head.
- Activate the appropriate read/write head and read the record.
- Transfer the record to main memory.
- Set a flag to indicate that the device is free to satisfy another request.

Typical Volume Configuration:

- Each storage unit is considered a volume, and each volume can contain several files, so they're called "multifile volumes." However, some files are extremely large and are contained in several volumes; these are called "multivolume files."



- The Master File Directory (MFD) is stored immediately after the volume descriptor and lists the names and characteristics of every file contained in that volume. The filenames in the MFD can refer to program files, data files, and/or system files.
- The File Manager supports subdirectories, they're listed in the MFD as well. The remainder of the volume is used for file storage.
- The first operating systems supported only a single directory per volume. This directory was created by the File Manager and contained the names of files, usually organized in alphabetical, spatial, or chronological order.

Disadvantages:

- It would take a long time to search for an individual file, especially if the MFD was organized in an arbitrary order.
- If the user had more than 256 small files stored in the volume, the directory space (with a 256-filename limit) would fill up before the disk storage space filled up. The user would then receive a message of "disk full" when only the directory itself was full.
- Users couldn't create subdirectories to group the files that were related.

- Multiple users couldn't safeguard their files from other users because the entire directory was freely made available to every user in the group on request.

Subdirectories:

- A subdirectory is created when a user opens an account to access the computer system. Although this user directory is treated as a file, its entry in the MFD is flagged to indicate to the File Manager that this file is really a subdirectory and has unique properties.
- Today's File Managers encourage users to create their own subdirectories, so related files can be grouped together. Many computer users and some operating systems call these subdirectories "folders."
- Tree structures allow the system to efficiently search individual directories. Information typically included in a file descriptor includes the following:

Filename—Within a single directory, filenames must be unique; in some operating systems, the filenames are case sensitive

File type— The organization and usage that are dependent on the system (for example, files and directories)

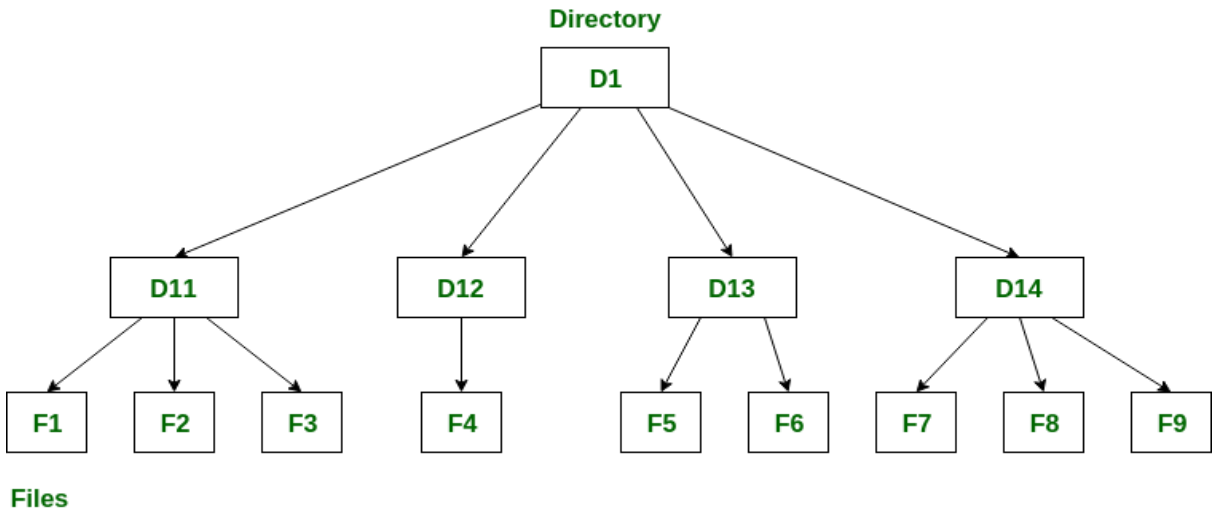
File size— It could be computed from other information, the size is kept here for convenience

File location—Identification of the first physical block (or all blocks) where the file is stored

Date and time of creation, Owner

Protection information— Access restrictions, based on who is allowed to access the file and what type of access is allowed

Record size— Its fixed size or its maximum size, depending on the type of record



File-Naming Conventions:

- The two components common to many file- names are a relative filename and an extension.
- Relative filename" to indicate the name without path information that appears in directory listings and folders. The relative filename is the name that differentiates it from other files in the same directory.
- The examples can include DEPARTMENT ADDRESSES, TAXES_PHOTOG, or AUTOEXEC. Generally, the relative filename can vary in length from one to many char- acters and can include letters of the alphabet, as well as digits.

Here's how a file named INVENTORY_COST.DOC is identified by different operating systems:

- Using a Windows operating system and a personal computer with three disk drives, the file's complete name is composed of its relative name and extension, preceded by the drive label and directory name:

C:\IMFST\FLYNN\INVENTORY_COST.DOC

File Organization:

Record Format:

- All files are composed of records. When a user gives a command to modify the contents of a file, it's actually a command to access records within the file.

- Within each file, the records are all presumed to have the same format: they can be of fixed length or of variable length.

(a)

Dan	Whitesto	1243 Ele	Harrisbu	PA	412 683-

(b)

Dan	Whiteston e	1243 Elementary Ave.	Harrisbur g	PA

Fixed-length records:

- It is the most common because they're the easiest to access directly. The critical aspect of fixed-length records is the size of the record.
- If it's too smaller than the number of characters to be stored in the record, the leftover characters are truncated. But if the record size is too larger than the number of characters to be stored
- storage space is wasted.

Variable-length records:

- It has not leave empty storage space and don't truncate any characters, thus eliminating the two disadvantages of fixed-length records.
- They can easily be read (one after the other)
- They are difficult to access directly because it's hard to calculate exactly where the record is located.

Physical File Organization:

- The physical organization of a file has to do with the way records are arranged and the characteristics of the medium used to store it.
- Files can be organized in one of several ways: sequential, direct, or indexed sequential.

Sequential record organization:

- It is easiest to implement because records are stored and retrieved serially, one after the other. To find a specific record, the file is searched from its beginning until the requested record is found.

- To speed the process, some optimization features may be built into the system. One is to select a key field from the record and then sort the records by that field before storing them.
- This technique aids the search process, it complicates file maintenance because the original order must be preserved every time records are added or deleted.

Direct record organization:

- It uses direct access files, which, of course, can be implemented only on direct access storage devices. These files give users the flexibility of accessing any record in any order without having to begin a search from the beginning of the file to do so. It's also known as "random organization," and its files are called "random access files."
- Records are identified by their relative addresses—their addresses relative to the beginning of the file.
- These logical addresses are computed when the records are stored and then again when the records are retrieved.

For example:

- Data for a telephone mail-order firm must be accessed quickly so customer information can be retrieved quickly. To do so, they can use hashing algorithms to directly access their data.
- The program that retrieves information from the data file uses that key in a hashing algorithm to calculate the logical address where your record is stored.
- Records that collide are stored in an overflow area that was set aside when the file was created. Although the program does all the work of linking the records from the overflow area to their corresponding logical address, the File Manager must handle the physical allocation of space.

Indexed sequential record organization:

- It combines the best of sequential and direct access. It's created and maintained through an Indexed Sequential Access Method (ISAM)

application, which removes the burden of handling overflows and preserves record order from the shoulders of the programmer.

- This type of organization doesn't create collisions because it doesn't use the result of the hashing algorithm to generate a record's address. Instead, it uses this information to generate an index file through which the records are retrieved.
- This organization divides an ordered sequential file into blocks of equal size.
- For most dynamic files, indexed sequential is the organization of choice because it allows both direct access to a few requested records and sequential access to many.

Physical Storage Allocation:

- Records are subdivided into fields. In most cases, their structure is managed by application programs and not the operating system. An exception is made for those systems that are heavily oriented to database applications, where the File Manager handles field structure.

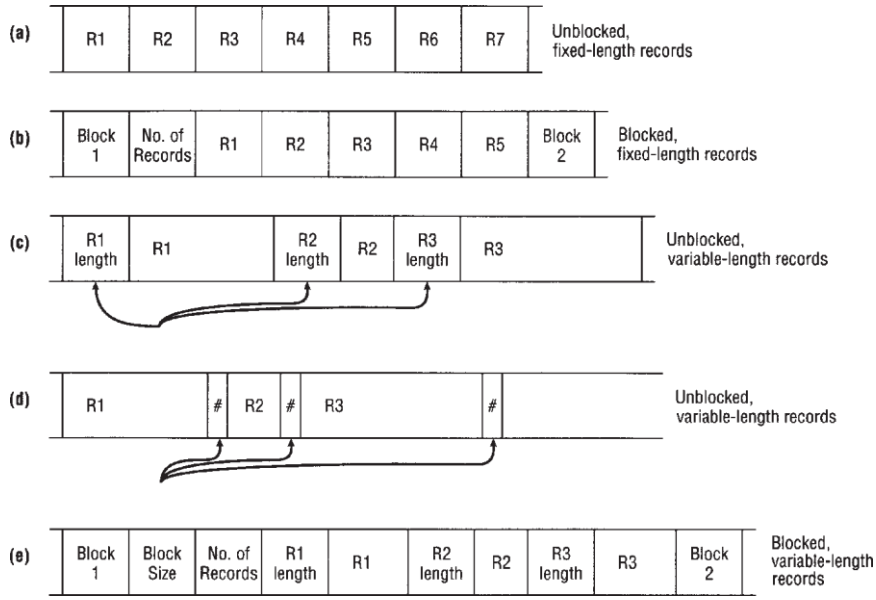


Fig: Five most common Record formats

Contiguous storage:

- It stored one after the other. This was the scheme used in early operating systems. It's very simple to implement and manage. Any record can be found and read, once its starting address and size are known, so the directory is very streamlined.

Free Space	File A Record 1	File A Record 2	File A Record 3	File A Record 4	File A Record 5	File B Record 1	File B Record 2	File B Record 3	File B Record 4	Free Space	File C Record 1

Advantage:

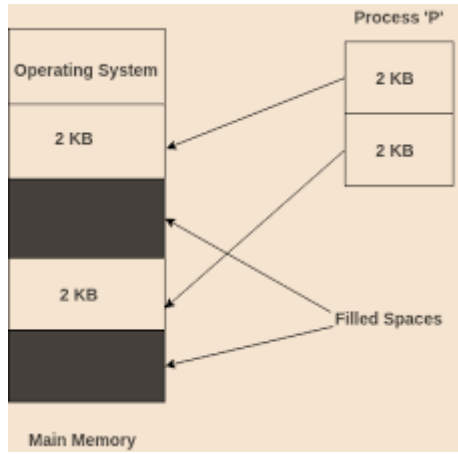
Ease of direct access because every part of the file is stored in the same compact area.

Disadvantage:

A file can't be expanded unless there's empty space available immediately following it, Therefore, room for expansion must be provided when the file is created.

Noncontiguous storage allocation

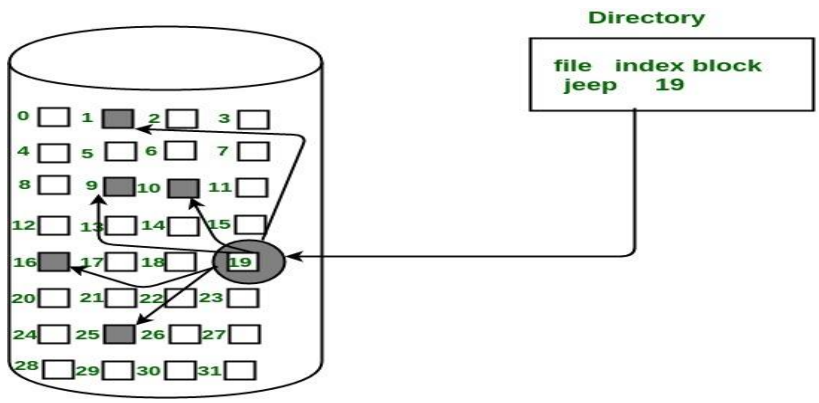
- It allows files to use any storage space available on the disk. A file's records are stored in a contiguous manner, only if there's enough empty space. Any remaining records and all other additions to the file are stored in other sections of the disk.
- In some systems these are called the extents of the file and are linked together with pointers. The physical size of each extent is determined by the operating system and is usually 256—or another power of two—bytes.
- The directory entry consists of the filename, the storage location of the first extent, the location of the last extent, and the total number of extents not counting the first.



- Files are usually declared to be either sequential or direct when they're created, so the File Manager can select the most efficient method of storage allocation: contiguous for direct files and noncontiguous for sequential. Operating systems must have the capability to support both storage allocation schemes.

Indexed storage allocation:

- Indexed storage allocation allows direct record access by bringing together the pointers linking every extent of that file into an index block.

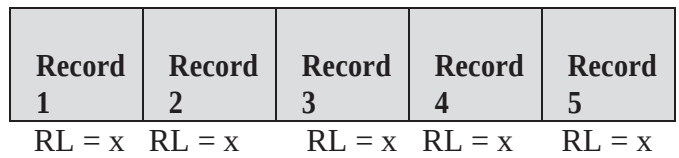


- Every file has its own index block, which consists of the addresses of each disk sector that make up the file. The index lists each entry in the same order in which the sectors are linked.
- To find a desired record, the File Manager accesses the first index (the highest

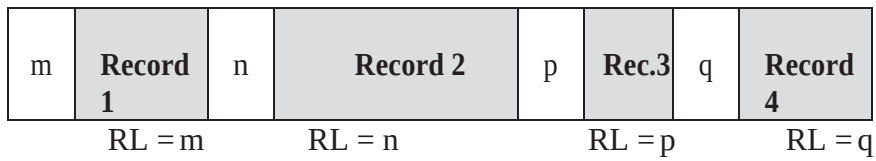
level), which points to a second index (lower level), which points to an even lower-level index and eventually to the data record.

File access methods:

- When a file is used, information is read and accessed into computer memory and there are several ways to access this information of the file. Some systems provide only one access method for files.
- The File Manager uses the address of the last byte read to access the next sequential record. Therefore, the current byte address (CBA) must be updated every time a record is accessed, such as when the READ command is executed.



(a) Fixed- length records



(b) Variable-length records

Sequential Access:

- It is the simplest access method. Information in the file is processed in order, one record after the other. This mode of access is by far the most common. **For example,** Editor and compiler usually access the file.
- Sequential access of fixed-length records, the CBA is updated simply by incrementing it by the record length (RL), which is a constant:

$$CBA = CBA + RL$$

Direct Access:

- Another method is direct access method also known as relative access method. A filed-length logical record that allows the program to read and write record rapidly. in no particular order. The direct access is based on the disk model of a file since disk allows random access to any file block.
- In the case of direct access with fixed-length records, the CBA can be computed directly from the record length and the desired record number RN (information provided through the READ command) minus 1:

$$CBA = (RN - 1) * RL$$

For example, if we're looking for the beginning of the eleventh record and the fixed record length is 25 bytes, the CBA would be:

$$(11 - 1) * 25 = 250$$

Levels in a File Management:

- The highest-level module is called the "basic file system," and it passes information through the access control verification module to the logical file system.
- The higher positioned modules pass information to the lower modules, so that they, in turn, can perform the required service and continue the communication down the chain to the lowest module, which communicates with the physical device and interacts with the Device Manager. Only then is the record made available to the user's program.

  We follow this I/O instruction through the file management system:

  READ RECORD NUMBER 7 FROM FILE CLASSES INTO STUDENT

- CLASSES is the name of a direct access file previously opened for input, and STUDENT is a data record previously defined within the program and occupying specific memory locations.

- The logical file system transforms the record number to its byte address using the familiar formula:

$$CBA = (RN - 1) * RL$$

- If there's more than one record in that block, it computes the record's offset within that block using these formulas:

$$\text{block number} = \text{integers}\left(\frac{\text{byte address}}{\text{physical record}}\right) + \text{address of the first}$$

$$\text{offset} = \text{remainder}\left(\frac{\text{byte address}}{\text{physical block size}}\right)$$

Access Control	No. of Links	Group	Owner	No. of Bytes	Date	Time	Filename
drwxrwxr-x	2	Journal	Comp	12820	Jan 10	19:32	ArtWarehouse
drwxrwxr-x	2	Journal	Comp	12844	Dec 15	09:59	bus_transport
-rwxr-xr-x	1	Journal	Comp	2705221	Mar 6	11:38	CLASSES
-rwxr--r--	1	Journal	Comp	12556	Feb 20	18:08	PAYroll
-rwx------	1	Journal	Comp	8721	Jan 17	07:32	supplier

(TABLE) A typical list of files stored in the directory called journal.

ACCESS CONTROL VERIFICATION MODULE:

- The first operating systems couldn't support file sharing among users. early systems needed 10 copies of a compiler to serve 10 users. Today's systems require only a single copy to serve everyone, regardless the number of active programs in the system. In fact, any file can be shared from data files and user-owned program files to system files.

- There are five possible actions that can be performed on a file—the ability to READ only, WRITE only, EXECUTE only, DELETE only, or some combination of the four. Each file management system has its own method to control file access.

Advantages:

- File sharing are numerous. In addition to saving space, it allows for synchronization of data updates, as when two applications are updating the same data file.

- It improves the efficiency of the system's resources because if files are shared in main memory, then there's a reduction of I/O operations.

Disadvantage:

- The file sharing is that the integrity of each file must be safeguarded; that calls for control over who is allowed to access the file and what type of access is permitted.

Access Control Matrix:

- An access control matrix is a table that states a subject's access rights on an object. A subject's access rights can be of the type read, write, and execute. Each column of the access control matrix is called an Access Control List (ACL).

- Each row is called a capability list. An ACL is connected to the object and outlines actions each subject can perform on that object. A capability list is connected to the subject and outlines the actions that a specific subject is allowed to perform on each object.

	User 1	User 2	User 3	User 4	User 5
File 1	RWED	R-E-	----	RWE-	--E-
File 2	----	R-E-	R-E-	--E-	----
File 3	----	RWED	----	--E-	----
File 4	R-E-	----	----	----	RWED
File 5	----	----	----	----	RWED

In the actual implementation, the letters RWEDare represented by bits 1 and 0: 1 indicates that access is allowed, and a 0 indicates access is denied.

Access control List:

- The access control list is a modification of the access control matrix. Each file is entered in the list and contains the names of the users who are allowed to access it and the type of access each is permitted.

File	Access
File 1	USER1 (RWED), USER2 (R-E-), USER4 (RWE-), USER5 (--E-), WORLD ( )
File 2	USER2 (R-E-), USER3 (R-E-), USER4 (--E-), WORLD ( )
File 3	USER2 (RWED), USER4 (--E-),WORLD ( )
File 4	USER1 (R-E-), USER5 (RWED),WORLD ( )
File 5	USER5 (RWED),WORLD ( )

- Some systems shorten the access control list even more by putting every user into a category: system, owner, group, and world. SYSTEM or ADMIN is designated for system personnel who have unlimited access to all files in the system.
- The OWNER has absolute control over all files created in the owner's account. An owner may create a GROUP file so that all users belonging to the appropriate group have access to it. WORLD is composed of all other users in the system.

Capability List:

- A capability list shows the access control information from a different perspective. It lists every user and the files to which each has access.

User	Access
User 1	File 1 (RWED), File 4 (R-E-)
User 2	File 1 (R-E-), File 2 (R-E-), File 3 (RWED)
User 3	File 2 (R-E-)
User 4	File 1 (RWE-), File 2 (--E-), File 3 (--E-)
User 5	File 1 (--E-), File 4 (RWED), File 5 (RWED)

- Capability lists are gaining in popularity because in operating systems such as Linux or UNIX, they control access to devices as well as to files.
- Access control list can be equated to the reservation list in a restaurant that has limited seating, with each seat assigned to a certain individual