

FILE SYSTEM ON OPERATING SYSTEM

What is File System?

- A file is a collection of correlated information which is recorded on secondary or non-volatile storage like magnetic disks, optical disks, and tapes.
- It is a method of data collection that is used as a medium for giving input and receiving output from that program.
- In general, a file is a sequence of bits, bytes, or records whose meaning is defined by the file creator and user.
- Every File has a logical location where they are located for storage and retrieval.

The important file concepts include:

1. **File attributes:** A file has certain attributes which vary from one operating system to another.
 - **Name:** Every file has a name by which it is referred.
 - **Identifier:** It is unique number that identifies the file within the file system.
 - **Type:** This information is needed for those systems that support different types of files.

- **Location:** It is a pointer to a device & to the location of the file on that device
- **Size:** It is the current size of a file in bytes, words or blocks.
- **Protection:** It is the access control information that determines who can read, write & execute a file.
- **Time, date & user identification:** It gives information about time of creation or last modification & last use.

2. File operations: The operating system can provide system calls to create, read, write, reposition, delete and truncate files.

- **Creating files:** Two steps are necessary to create a file. First, space must be found for the file in the file system. Secondly, an entry must be made in the directory for the new file.
- **Reading a file:** Data read from the file at the current position. The system must keep a read pointer to know the location in the file from where the next read is to take place. Once the read has been taken place, the read pointer is updated.

- **Writing a file:** Data are written to the file at the current position. The system must keep a write pointer to know the location in the file where the next write is to take place. The write pointer must be updated whenever a write occurs.
- **Repositioning within a file (seek):** The directory is searched for the appropriate entry & the current file position is set to a given value. After repositioning data can be read from or written into that position.
- **Deleting a file:** To delete a file, we search the directory for the required file. After deletion, the space is released so that it can be reused by other files.
- **Truncating a file:** The user may erase the contents of a file but allows all attributes to remain unchanged except the file length which is reset to 0 & the space is released.

3. File types: The file name is split into 2 parts, **Name** & **extension**. Usually these two parts are separated by a period. The user & the OS can know the type of the file from the extension itself.

Listed below are some file types along with their extension:

file type	usual extension	function
executable	exe, com, bin or none	read to run machine- language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rrf, doc	various word-processor formats
library	lib, a, so, dll, mpeg, mov, rm	libraries of routines for programmers
print or view	arc, zip, tar	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes com- pressed, for archiving or storage
multimedia	mpeg, mov, rm	binary file containing audio or A/V information

- The name is split into two parts- a name and an extension, The system uses the extension to indicate the type of the file and the type of operations that can be done on that file.

4. ACCESS METHODS

When a file is used, this information must be accessed and read into computer memory. The information in the file can be accessed in several ways. There are two major access methods as follows:

Sequential Access: Information in the file is processed in order, one record after the other.

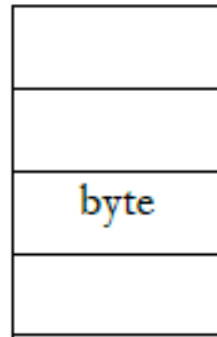
- A read operation reads the next portion of the file and automatically advances a file pointer, which tracks the I/O location.
- Similarly, a write appends to the end of the file and advances to the end of the newly written material (the new end of file).
- Sequential access is based on a tape model of a file, and works as well on sequential-access devices as it does on random-access ones.

Direct Access: A file is made up of fixed length logical records that allow programs to read and write records rapidly in no particular order.

- The direct-access method is based on a disk model of a file, since disks allow random access to any file block.
- For direct access, the file is viewed as a numbered sequence of blocks or records. A direct-access file allows arbitrary blocks to be read or written.
- There are no restrictions on the order of reading or writing for a direct-access file. For the direct-access method, the file operations must be modified to include the block number as a parameter.
- Thus, we have *read n*, where *n* is the block number, rather than *read next*, and *write n* rather than *write next*.

5. File structure: Files can be structured in several ways. Three common possible are:

- **Byte sequence:** The figure shows an unstructured sequence of bytes. The OS doesn't care about the content of file. It only sees the bytes. This structure provides maximum flexibility. Users can write anything into their files & name them according to their convenience. Both UNIX & windows use this approach.



- **Record sequence:** In this structure, a file is a sequence of fixed length records. Here the read operation returns one records & the write operation overwrites or append or record.



- **Tree:** In this organization, a file consists of a tree of records of varying lengths. Each record consists of a key field. The tree is stored on the key field to allow first searching for a particular key.