Unit 8

File System

8.1 File Concept
8.2 Access Methods
8.3 File Structure
8.4 Allocation Methods
8.5 Free Space Management

Introduction :

For most users, the file system is the most visible aspect of an operating system. It provides the mechanism for on-line storage and access to both data and programs of the operating system. The file system consists of two distinct parts: a collection of files, each storing related data, and a directory structure, which organizes and provides information about all the files in the system. Since there is only a limited amount of disk space, it is necessary to reuse the space from deleted files for new files. Three major methods of allocating disk space are contiguous, linked, and indexed.

8.1 File Concept :

Computers can store information on various storage media, such as magnetic disks, magnetic tapes, and optical disks. So that the computer system will be convenient to use, the operating system provides a uniform logical view of information storage. The operating system abstracts from the physical properties of its storage devices to define a logical storage unit, the file. Files are mapped by the operating system onto physical devices. These storage devices are usually non-volatile, so the contents are persistent through power failures and system reboots.

8.1.1. File Concept :

A file is a named collection of related information that is recorded on secondary storage. From a user's perspective, a tile is the smallest allotment of logical secondary

Scanned by CamScanner

storage; that is, data cannot be written to secondary storage unless they are within a file. Commonly, files represent programs (both source and object forms) and data. Data files may be numeric, alphabetic, alphanumeric, or binary. Files may be free form, such as text files, or may be formatted rigidly. In general, a file is a sequence of bits, bytes, lines, or records, the meaning of which is defined by the file's creator and user. The concept of a file is thus extremely general.

8.1.2. File Attributes :

A file is named, for the convenience of its human users, and is referred to by its name. A name is usually a string of characters, such as examples. Some systems differentiate between uppercase and lowercase characters in names, whereas other systems do not. A file's attributes vary from one operating system to another but typically consist of these:

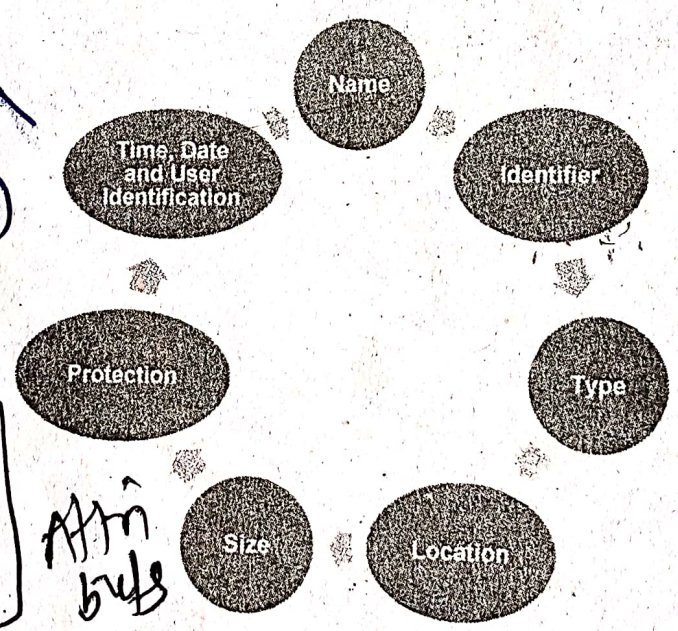


Fig 8.1 : File Attributes

1) **Name :**
The symbolic file name is the only information kept in human readable form.

2) **Identifier :**
This unique tag, usually a number, identifies the file within the file system; it is the non-human-readable name for the file.

3) **Type :**
This information is needed for systems that support different types of files.

Scanned by CamScanner

4) Location :

This information is a pointer to a device and to the location of the file on that device.

5) Size :

The current size of the file (in bytes, words, or blocks) and possibly the maximum allowed size are included in this attribute.

6) Protection :

Access-control information determines who can do reading, writing, executing, and so on.

7) Time, date, and user identification :

This information may be kept for creation, last modification, and last use. These data can be useful for protection, security, and usage monitoring.

8.1.3. File Operations :

A file is an abstract data type. To define a file properly, user need to consider the operations that can be performed on files. The operating system can provide system calls to create, write, read, reposition, delete, and truncate files.

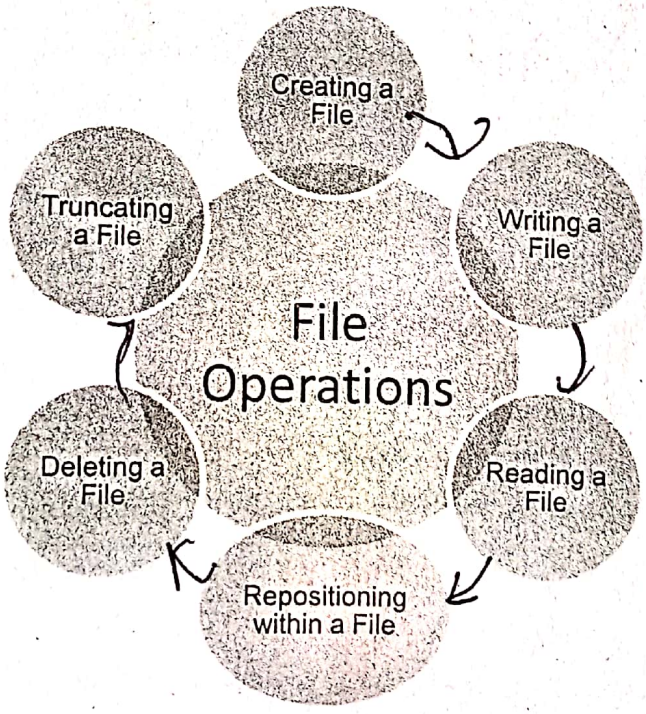


Fig 8.2 : File Operations

Scanned by CamScanner

1) **Creating a File :**

Two steps are necessary to create a file. First, space in the file system must be found for the file. Second, an entry for the new file must be made in the directory.

2) **Writing a File :**

To write a file, a system call must be made for specifying both 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 file's location. The system must keep a write pointer to the location in the file, where the next write is to take place. The write pointer must be updated whenever a write occurs.

3) **Reading a File :**

To read from a file, a system call is used 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 entry, and the system needs to keep a read pointer to the location in the file, where the next read is to take place. Once the read has taken place, the read pointer is updated.

4) **Repositioning within a File :**

The directory is searched for the appropriate entry, and the current file-position pointer is repositioned to a given value. Repositioning within a file need not involve any actual I/O. This file operation is also known as a file seeks.

5) **Deleting a File :**

To delete a file, the directory is searched for the named file. Having found the associated directory entry, all file space is released, so that it can be reused by other files, and erase the directory entry.

6) **Truncating a File :**

The user may want to erase the contents of a file but keep its attributes. Rather than forcing the user to delete the file and then recreate it, this function allows all attributes to remain unchanged except for file length; but lets the file be reset to length zero and its file space released.

8.1.4. File Types :

When a file system is designed-indeed, an entire operating system-user always considers whether the operating system should recognize and support file types. If an

Scanned by CamScanner

operating system recognizes the type of a file, it can then operate on the file in reasonable ways. Following table gives the file type with usual extension and function.

File Type	Usual Extension	Function
Executable	exe, com, bin	Read to run machinelanguage program.
Object	obj, o	Compiled, machinelanguage, not linked
Source Code	c, cc, java, pas asm, a	Source code invarious language
Text	txt, doc	Textual data,documents

Table: Types of Files

8.1.5. Types of Access :

The need to protect files is a direct result of the ability to access files. Systems that

Scanned by CamScanner