Unit 13

FILES TYPES

A 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. In general, a file is a sequence of bits, bytes, lines or **records** whose meaning is defined by the files creator and user.

File attributes

A file has certain attributes which vary from one OS to another but typically consist of these:-

Type: This information is needed for those systems that support different types.

Location: this information is a pointer to a device and to the location of the file on that device.

Size (current size): This is the number of bytes in the file.

Max. size: the number of bytes the file may grow to.

Protection: this specifies who can access the file and in what way.

Creator: This identifies the person who created the file.

Owner: this specifies the current owner of the file.

Time/date: this shows the date and time when the file was created, when the file was last accessed and when the file was last modified.

FILE OPERATIONS

a) Creating a file: The purpose of this is to announce that the file is coming and to set some of the attributes. Two stages are necessary to create a file i.e.

- If space in the file system must be found for the file.
- An entry for the new file must be made in directory.

b) To write a file: we make a system call specifying both the name 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 system must keep a write pointer to the location in the file where the next write is to take place.

FILE OPERATIONS

c) Reading a file: to read a file, we use a system call that specifies the name of the file and where in the memory the next block of the file should be put.

d) Deleting a file: To delete a file, we search the directory for the named file. Having found the associated directory entry, we release all the file space so that it can be used by other files and erase the directory entry. This frees up disk space.

e) Opening a file: Before using a file, a process must open it. The purpose of open call is to allow the system to fetch the attributes and list of disk to fetch the attributes and list of disk addresses into main memory for rapid access.

FILE OPERATIONS

f) Closing a file: When all the accesses are finished, the attributes and disk addresses are no longer needed, so the file should be closed, free up internal table space. Many systems encourage disk by imposing the maxim number of open files on processes. A disk is written in blocks, and closing a file forces writing of the file's last block, even though that block may not be entirely full yet.

g) Append: This is used to add data to the end of the file. However, systems provide multiple ways of doing the same thing.

h) Rename a file: This is done when the user needs to change the name of an existing file.

File Structure

A File Structure should be according to a required format that the operating system can understand.

1. A file has a certain defined structure according to its type.
2. A text file is a sequence of characters organized into lines.
3. A source file is a sequence of procedures and functions.
4. An object file is a sequence of bytes organized into blocks that are understandable by the machine.

When operating system defines different file structures, it also contains the code to support these file structure. Unix, MS-DOS support minimum number of file structure.

File Type

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. Many operating systems support many types of files. Operating system like MS-DOS and UNIX have the following types of files −

Ordinary files

These are the files that contain user information.

These may have text, databases or executable program.

The user can apply various operations on such files like add, modify, delete or even remove the entire file.

Directory files

These files contain list of file names and other information related to these files.

Special files

These files are also known as device files.

These files represent physical device like disks, terminals, printers, networks, tape drive etc.

These files are of two types −

Character special files − data is handled character by character as in case of terminals or printers.

Block special files − data is handled in blocks as in the case of disks and tapes.

File Access Mechanisms

File access mechanism refers to the manner in which the records of a file may be accessed. There are several ways to access files –

1. Sequential access
2. Direct/Random access
3. Indexed sequential access

File Access Mechanisms

Sequential access

A sequential access is that in which the records are accessed in some sequence, i.e., the information in the file is processed in order, one record after the other. This access method is the most primitive one. Example: Compilers usually access files in this fashion.

File Access Mechanisms

Direct/Random access

Random access file organization provides, accessing the records directly.

Each record has its own address on the file with by the help of which it can be directly accessed for reading or writing.

The records need not be in any sequence within the file and they need not be in adjacent locations on the storage medium.

File Access Mechanisms

Indexed sequential access

This mechanism is built up on base of sequential access.

An index is created for each file which contains pointers to various blocks.

Index is searched sequentially and its pointer is used to access the file directly.

Space Allocation

Files are allocated disk spaces by operating system. Operating systems deploy following three main ways to allocate disk space to files.

1. Contiguous Allocation
2. Linked Allocation
3. Indexed Allocation

Contiguous Allocation

1. Each file occupies a contiguous address space on disk.
2. Assigned disk address is in linear order.
3. Easy to implement.
4. External fragmentation is a major issue with this type of allocation technique.

Linked Allocation

1. Each file carries a list of links to disk blocks.
2. Directory contains link / pointer to first block of a file.
3. No external fragmentation
4. Effectively used in sequential access file.
5. Inefficient in case of direct access file.

Indexed Allocation

1. Provides solutions to problems of contiguous and linked allocation.
2. A index block is created having all pointers to files.
3. Each file has its own index block which stores the addresses of disk space occupied by the file.
4. Directory contains the addresses of index blocks of files.