Chapter-4

* File systems *

* **Introduction:** All computer applications need to store and retrieve info.
- secondary storage devices such as magnetic disk, magnetic tape, external hard drives & computer's built in hard disk are used for storage.
1) The data on secondary storage devices are persistent (permanently stored). A user can access these data at any time as per the requirement.
2) It is possible to store a very large amount of data.
3) multiple processes can be able to access the info. concurrently.
- Info. or data is nothing but the file itself.
- To store files on the disk and retrieve them when needed, the OS has a mechanism called the **file system.**

→ **File system:** The file system is that part of OS which is responsible for the management and organization of various files in a system.

→ **Files:** Files are logical unit of info. created by processes.
- how they are structured, named, accessed, used, protected, implemented, and managed are major topics in operating system design.
- The part of OS dealing with files is known as **filesystem.**

4.1) → **How files are used and what Properties they have?**

4.1.1) **File Naming:** Files are an abstraction mechanism.
- They provide a way to store info. on the disk and read it back later. Files can store different types

types of data such as the graphics, executable code, sound, videos etc.
- On the basis of the data within it, a file can be categorized as a data file, graphic file, database, executable file, sound file, video file etc.
- The structure of a file is based on the type of file. Ex- (i) graphical file : it is an organized collection of pixels.
(ii) Database file : It is a collection of tables.
(iii) Batch file : It is a collection of commands.
- When a process creates a file, it gives a name. file
When a process terminates, the file continues to exist and can be accessed by other processes using its name.
- The exact rules for file naming vary from system to system.
- But all current OS allows strings of one to eight letters as legal file names.
- Now digits and special characters are also permitted.
- Some file systems support names as long as 255 characters.
- Some filesystems distinguish between upper and lower case where as others don't.
Ex- Unix / Linux - case sensitive
MS-DOS - falls into 2nd category.
- Windows-95 and windows-98 both use the MS-DOS file system, called FAT-16 & FAT-32
- Many OS support two part file name, with two parts seperated by a period, as in prog.c
- The part following the period is called the file-extention and usually indicates something about file. refer fig: 4.1 of Tanenbaum (page-286)

Scanned with CamScanner

4.1.2] File structure :- The file structure refers to the internal structure of the file, that is, how a file is internally stored in the system. Files can be structured in several ways. Three common structures are shown in fig:4.2 (Tanenbaum)

(a) Byte sequence :- In this file structure, each file is made up of sequence of 8-bit bytes, having no fixed structure. This type of structure provides flexibility to the user programs. Unix OS supports this type of file structure.

(b) Record sequence :- In this file structure, a file consists of a sequence of fixed-length records, where an arbitary number of records can be read from or written to a file. The records can't be inserted or deleted in the middle of a file. In this system, the read operation returns one record and the written operation appends or overwrites one record. CP/M OS supports this type of scheme.

(c) Tree Structure :- In this file structure, a file consists of a tree of disk blocks where each block holds a number of records of varied lengths. Each record contains a key field at a fixed position. The records are search on key value and new records can be inserted any where in the file structure. This type of file structure is used by main frame systems, where it is called ISAM (Indexed sequential Access method).

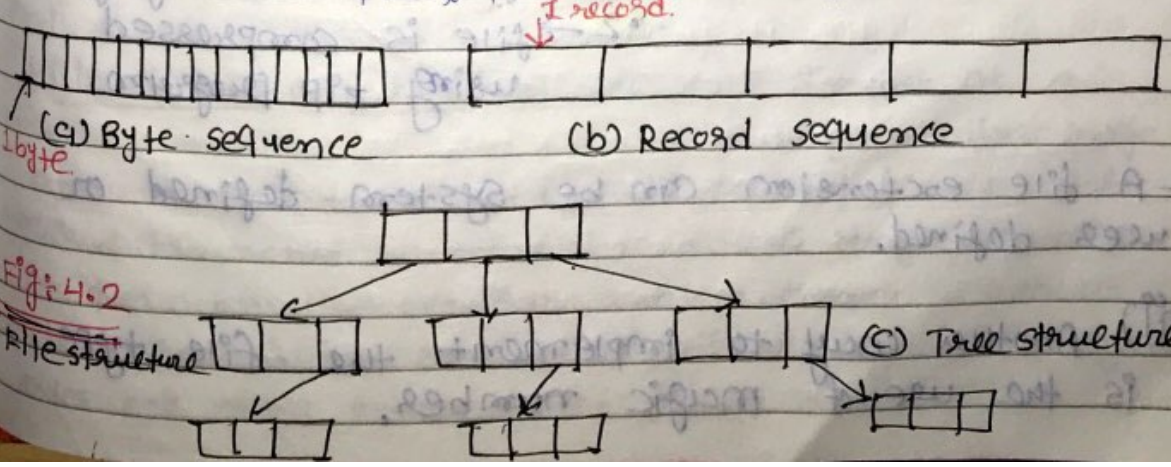

Fig:4.2
File structure

Scanned with CamScanner

4.1.3] File types: Files can be of different type like data file, graphic file, executable file, sound file, video file etc.

- The OS can handle a file in a reasonable way only if it recognizes and supports that file type. EX- A user request to open an executable file with a text editor will produce garbage if the OS has not been told that it is an executable file.

(i) The most common technique to implement a file a file type is by providing an extension to a file. The file name is divided into two parts, with two parts separated by a period (.) symbol where, the first part is the name and the second part after the period is file-extension. A file extension is generally one to three characters long, it indicates the type of the file and the ops (read, write, execute) that can be performed on that file.

EX- Prog.c, the file with .txt extension will be opened with text editor

- And the file with .mp3 extension will be opened with a music player supporting .mp3 file.

- Unix supports use of double extension to a file name. EX- file1.c.z where

.c → file is a c language file.

.z → file is compressed using zip program.

- A file extension can be system defined or user defined.

(ii) Another way to implement the file type is the use of magic numbers.

Scanned with CamScanner

magic number is a sequence of bits, placed at the starting point of the file to indicate the type of of file.
The unix system makes use of magic number to recognize the file type.
However, not all its files have magic number, so, to help its users to determine the type of contents of the file, it allows file-name-extension hints.

4.1.4] File Access Methods : The info^n stored in the file can be accessed in one of the two ways: either sequential access or direct access (random access).

(i) sequential access : when the info^n in the file is accessed in order, one record after the other, it is called sequential access. It is the easiest file access method. Compilers, multimedia applications, sound files and editors are the most common examples of the programs using sequential access.
The most common and frequent operations performed on a file are read and write operations. In case of read operation, the record at the location pointed by the file pointer is read and the file pointer is then advanced to the next record. similarly, in case of write operation, the record is written to the end of the file and pointer is advanced to the end of new record.

(ii) Direct access (random access) : For disk as a storage media (large amount of data can be stored), sequential access of this data could be come very slow & lengthy process. To over come this problem random access is there. Files whose bytes or records can be read in any order are called random access files.

Scanned with CamScanner

- They are required by many applications
  Ex- Railway reservation system
- In railway reservation system if a customer makes a request to check the status of his reservation on a train, the system must be able to access the record of that customer directly, without having to access all other customer's records.
- Most applications with large databases require direct access method, enabling immediate access to large amount of info.
- In this access where to start reading?
  (i) Every read operation gives the position in the file from where start reading.
  (ii) A special operation seek is provided to set that position. [used in Unix & windows]

4.1.5] File Attributes: Every file has a name and its data.
- All OS associates other info with each file. for example - the date and time the file was last modified and the file size.
- we call these extra items the file's attributes or metadata.
- The list of attributes vary system to system.
- Some of them are listed in fig: 4.4 (Tanenbaum)
  Ex- Name - helps to identify and locate a file in a system.
  type: it indicates file type
  Size: stores info about current size of the file.
  Identifier: A unique tag, helps filesystem to recognize file
  Location: location of the file on the device.
  date and time: date & time of last modified, creation time &
  Protection: stores info about the access permissions.
  Password: needed to access the file.
  Creator etc.

Scanned with CamScanner

4.1.6] File operations :-

1) Create file :- To bring a file into existence, the create system call is used.

2) Open file :- To open a file, the open system call is used which accepts the file name and the access mode as a parameters and returns a pointer to the entry in the open file table.

3) Write file :- To write into a file, the write system call is used which accepts the file name and the data to be written to the file as parameters.

4) Read file :- The read system call is used to retrieve data from a file. It accepts the file name, amount of data to be read and a read pointer to point to the position from where the data is to be read as parameters.

5) Seek file :- To position the pointer at a specific position in a file, the seek system call is used. once the pointer is positioned, data can be read from and written to that position.

6) Close file :- When all the operations on a file are completed, it must be closed using the close systemcall.

7) Delete file :- When a file is no longer needed, the delete system call is used. The OS searches the file name in the directory listing, having found the associated entry, it releases all space allocated to the file.

8) Append file :- To add data at the end of an existing file, append system call is used.

9) Rename file :- To change the name of an existing file, rename system call is used. This system call changes the existing entry for the file name in the directory to a new file name.

Scanned with CamScanner

Imp

*Directories :- To keep track of files, file systems normally have directories or folders, which in many systems are themselves files.

(i) single level directories :- The simplest form of directory system is having one directory containing all the files. Some times it is called the root directory, but since it is the only one, the name does not matter much.

- On early personal computers, this system was common, in part because there was only one user.
- The world's first super computer, the CDC 6600, also had only a single directory for all files, even though it was used by many users at once.
- An example of a system with one directory is given in fig :- 4.3. Here the directory contains four files. The advantages of this scheme are its simplicity and the ability to locate files quickly - there is only one place to look, after all. It is often used on simple embedded devices such as telephones, digital cameras, and some portable music players.

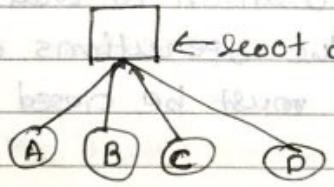

Fig:- 4.3 A single level directory system containing four files.

(ii) Hierarchical (multilevel) directory systems :-

Scanned with CamScanner

Fig:3.12 A TLB to speed up paging

valid	Virtual page	Modied	Protection	page frame
1	140	1	RW	31
1	20	0	RX	38
1	130	1	RW	29
1	129	1	RW	62
1	19	0	RX	50
1	21	0	RX	45
1	860	1	RW	14
1	861	1	RW	75

Scanned with CamScanner