UNIT III – FILE SYSTEMS, I/O AND DEVICE MANAGEMENT

File systems: FAT, EXT4, inodes, journaling, Disk I/O operations, buffering, caching, DMA, Device drivers: concept and real use (kernel modules overview), Real-world: Android file system, SD card permissions, Linux device trees

3.1 FILE SYSTEMS

An OS file system is a method used by an operating system to organize, manage, and store files and directories on a storage device. It acts as a bridge between the OS and the physical storage, allowing users and applications to access and manage data efficiently. Essentially, it's how your computer keeps track of where everything is stored.

FILE CONCEPT

File: A file is a collection of related information that shares a common name that is recorded on secondary storage. From a user's perspective, a file is the smallest allotment of logical secondary storage and data cannot be written to secondary storage unless they are within a file. The file is treated as a single entity by users and applications and may be referenced by name.

File Attributes:

Identifier – A unique number (e.g., inode) used by the OS to recognize the file.

Type – Specifies whether the file is text, executable, image, or another format.

Location – The exact position of the file on the storage device.

Size – The total amount of data the file contains, usually in bytes.

Protection – Defines who can read, write, or execute the file.

Time & Date – Records when the file was created, modified, or last accessed.

User ID – Identifies the owner of the file.

File Operations:

Reading a file – Allocating space and initializing metadata for a new file.

Writing a file – Storing data into a file at a specified location.

Reading a file – Retrieving data from a file into memory.

Repositioning within a file – Moving the file pointer to a specific position for read/write.

Deleting a file – Removing the file's directory entry and freeing its storage space.

Truncating a file – Erasing all file content without deleting the file itself.

File Types: The files are classified into different categories as follows:

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.

ACCESS METHODS

1. **Sequential Access** – Data is read or written in a linear order from start to end.
2. **Direct (Random) Access** – Data can be accessed directly at any position using its address.
3. **Indexed Access** – Uses an index table to quickly locate and access data blocks in a file.

DIRECTORY STRUCTURE

1. **Single-level Directory** – All files are stored in a single directory for all users.
2. **Two-level Directory** – Each user has their own separate directory under a master directory.
3. **Tree-structured Directory** – Directories are arranged in a hierarchy with subdirectories.
4. **Acyclic Graph Directory** – Allows shared subdirectories and files without cycles.
5. **General Graph Directory** – Similar to acyclic graph but allows cycles (needs extra control to avoid loops).