

Operating Systems  
Course Code: **71203002004**  
*Logical File System*

*by -  
Minal Rajwar*



# Logical File System View

- Refers to how the user or program **perceives and accesses** the data.
- Provides an **abstraction** that hides the details of physical storage.
- Defines record descriptions and access paths to physical files.

# Logical File System View

## Key Characteristics:

- Does not occupy storage space (contains no actual data).
- Represents one or more physical files.
- Can contain multiple record formats (up to 32).
- Cannot exist without a physical file.
- Can be created/deleted independently of the physical file (though tied to it).
- Uses commands like **CRTL F**.

**Example:** A logical file providing a sorted view of the employee records by department, even though the physical file stores them sequentially by ID.

# File Directories and File System Structures

## 1. File Directories

A **file directory** is a collection of files that contains information about each file, including attributes, location, and type. The operating system manages this information.

### Key Points:

- A directory itself is a file, accessible through file management routines.
- Directories allow efficient file organization, naming, and grouping of related files (e.g., all Java programs, all games).

### Advantages of Maintaining Directories:

1. **Efficiency:** Files can be located quickly.
2. **Naming:** Allows multiple users to have files with the same name without conflict.
3. **Grouping:** Logical grouping of files based on properties or type.

# File Directories and File System Structures

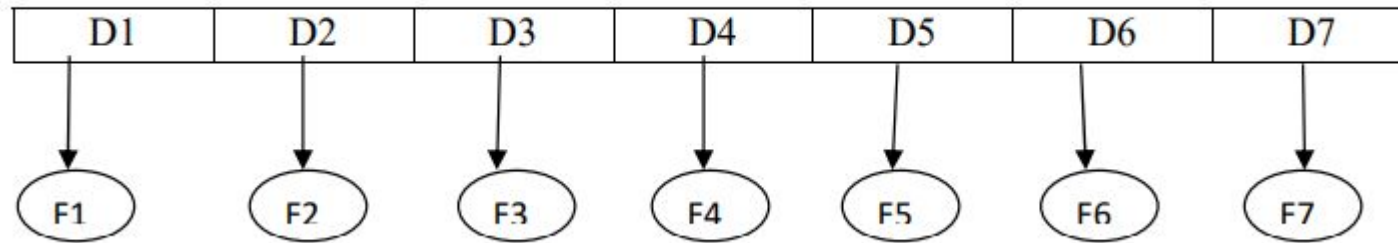
## 2. Directory Structures

Directories can be organized in different ways depending on the complexity and number of users.

### 2.1 Single-Level Directory

- All files are stored in a single directory.
- **Advantages:** Simple, easy to implement, fast for small numbers of files.
- **Disadvantages:** Name conflicts; difficult to manage large numbers of files.

Directory

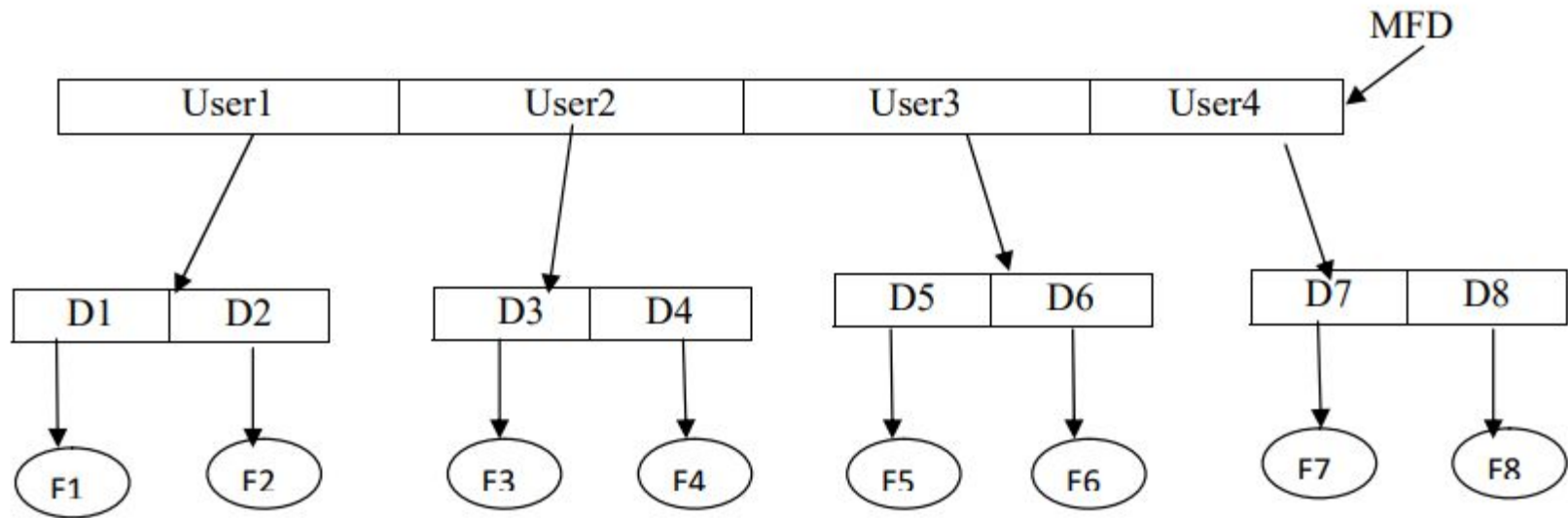


# File Directories and File System Structures

## 2.2 Two-Level Directory

- Each user has their own user file directory (UFD).
- The system maintains a master file directory (MFD) pointing to each UFD.
- **Advantages:** Users can have files with the same name; searching is simplified with pathnames.

**Disadvantages:** Users cannot easily share files; grouping similar file types is limited.

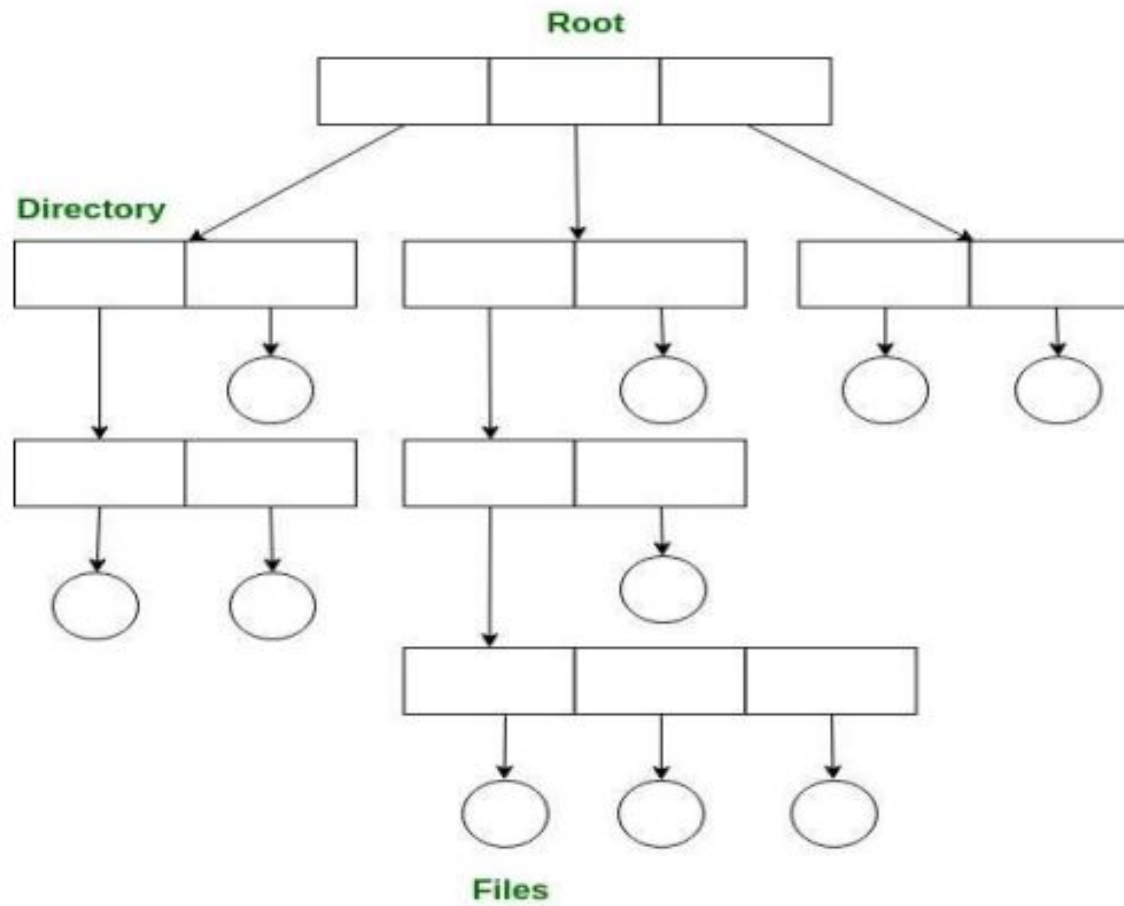




# File Directories and File System Structures

## 2.3 Tree-Structured Directory

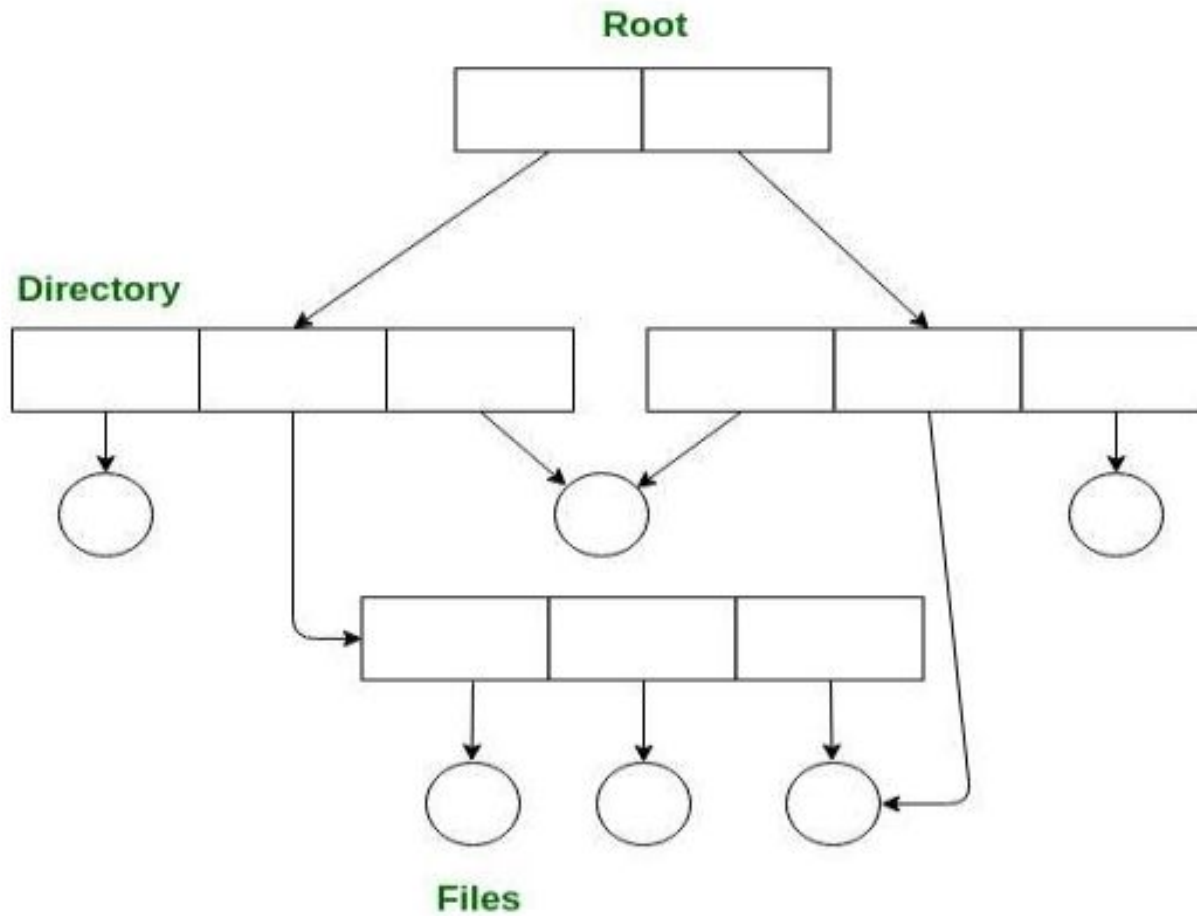
- Directories are organized in a hierarchical tree, starting from a root directory.
- Users can create subdirectories to organize their files.
- **Advantages:** Efficient searching, supports grouping and path-based access.
- **Disadvantages:** Files that don't fit the hierarchy are hard to manage; sharing is limited.



# File Directories and File System Structures

## 2.4 Acyclic Graph Directory

- Directories can share subdirectories and files; a single file can appear in multiple directories.
- **Advantages:** More flexible, allows shared files.
- **Disadvantages:** Requires careful handling of links; dangling pointers may occur if files are deleted.



# File Directories and File System Structures

## 3. File Allocation Methods

Files may not always be stored contiguously on disk. The OS uses different allocation methods to track file locations:

### 1. Contiguous Allocation

- Files occupy consecutive disk blocks.
- **Pros:** Fast sequential access.
- **Cons:** Difficult to expand; may lead to wasted space.

# File Directories and File System Structures

## Linked Allocation

- Each block points to the next block.
- **Pros:** Flexible size; no fragmentation.
- **Cons:** Slower access; difficult to access random blocks.

## Indexed Allocation

- Each file has an index block containing pointers to all its blocks.
- **Pros:** Supports direct access.
- **Cons:** Index block can become large for very large files.

# File Directories and File System Structures

## 4. Operations on Directories

Common operations performed on directories include:

- **Search:** Locate a file using symbolic names or patterns.
- **Create/Delete:** Add or remove files.
- **Rename:** Change file names.
- **List:** Display all files in a directory.
- **Traverse:** Access every file and subdirectory.

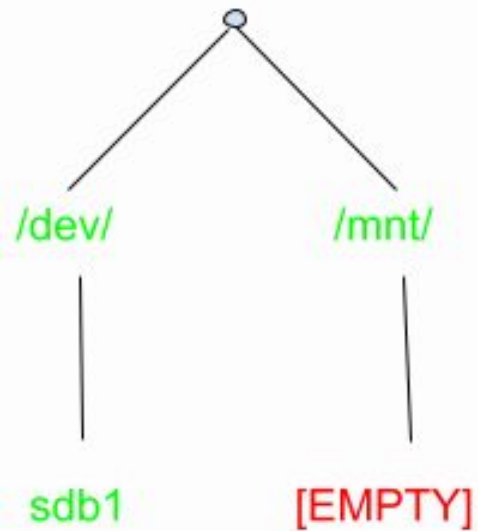
# File System Mounting

- **Mounting** = attaching a file system from a storage device to the computer's main directory tree.
- Files can then be accessed as part of the system.
- Mounted on an **empty directory** called the **mount point**.

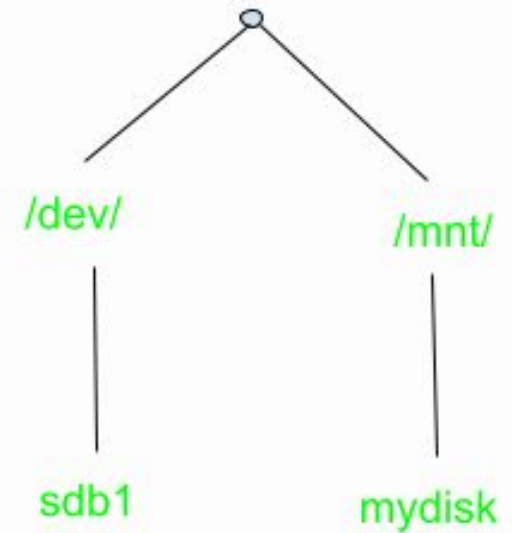
## Key Terms:

- **File System:** Method to organize data (e.g., FAT32, NTFS, ext4).
- **Device Name:** Identifier of the device (e.g., **D:**, **/dev/sdb1**).
- **Mount Point:** Directory where the file system is attached.





(Before Mounting)



(After Mounting)

# File System Mounting

Mounting lets users access files from storage devices by linking them to the system's directory structure.

## In Different OS:

- **Linux:** Use `mount` and `umount` commands.
- **Windows:** Auto-assigns a drive letter (e.g., D:, E:).
- **MacOS:** Auto-mounts in Finder, or use `diskutil`.

## DISCUSSION & REVISION

1. In a directory structure, what is the very first (topmost) directory called?
2. In a two-level directory structure, the system maintains a master directory that points to each user's directory. What is it called?
3. Which directory structure allows files or subdirectories to be shared among multiple users without duplicating them?
4. In file allocation methods, which method uses an index block to hold pointers to all file blocks?
5. Which file allocation method stores file blocks consecutively on disk?

## REFERENCES

1. <https://www.vbspu.ac.in/e-content/File-System-Management.pdf>
2. <https://www.scaler.com/topics/file-systems-in-os/>
3. <https://www.geeksforgeeks.org/operating-systems/physical-and-logical-file-systems/>
4. <https://www.geeksforgeeks.org/operating-systems/file-system-mounting-in-os/>