

File System

A series of horizontal bars of varying lengths in a light blue color, stacked vertically on the left side of the slide.

File System

1. File System Basics

- Definition of file system and files
- Directory structure

2. File Attributes

- Name, type, size, location, protection, timestamps

3. File Operations

- Create, open, read, write, delete, append, truncate, close

4. File Access Methods

- Sequential, direct, indexed access



File System

5. Directory Management

- Directory structure and common operations

6. File Systems in OS

- Types: FAT, NTFS, ext, HFS/APFS
- Pros and cons

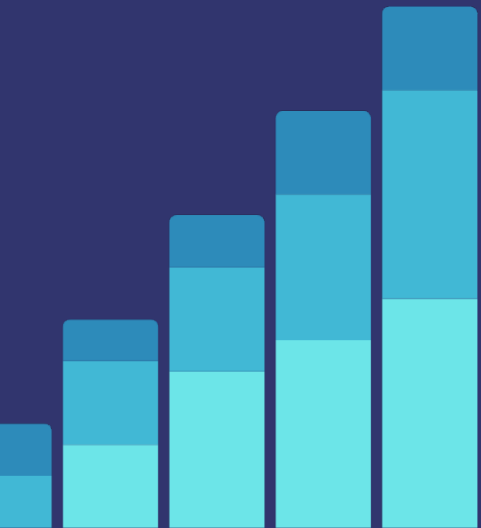
7. File Allocation Methods

- Contiguous, linked, FAT, indexed allocation

8. Free Space Management

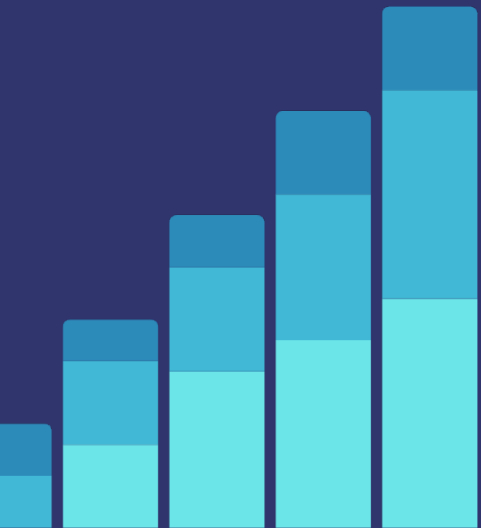
- Bit vector, linked list

WHAT IS File System?



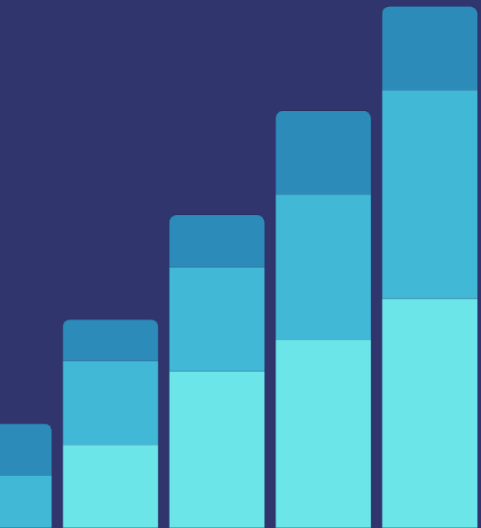
- A file system organizes and manages files stored on disk or in memory.
- A collection of files is called a directory, and a file system comprises multiple directories.

WHAT IS File?



- A file is a data structure storing a sequence of records.
- Files can be simple (e.g., plain text) or complex (formatted data).
- Directories are collections of files, and the file system is made up of directories at different levels.

File Attributes



- File attributes describe properties of a file (e.g., name, type, size, location).
- Attributes can be viewed and modified (e.g., hidden, read-only, system files).
- Attributes may prevent unauthorized file deletion or modification.



Common File Attributes

1. Name: Unique identifier in the file system.
2. Type: File category (e.g., text, video, executable).
3. Location: Storage location in the file system.
4. Size: Amount of memory occupied by the file.
5. Protection: Permissions for users and groups.
6. Time/Date: Last modified timestamp.



File Operations Overview

- Common file operations: create, open, read, write, close, delete, append, truncate, rename.
- File operations are performed using commands provided by the operating system.



Key File Operations

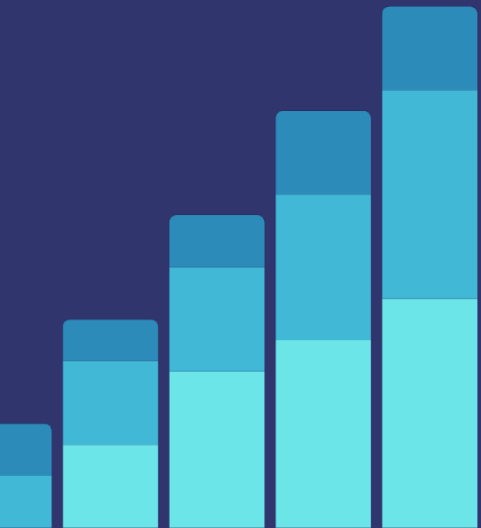
1. Create: Adds a new file to the system.
2. Open: Accesses an existing file for processing.
3. Write: Adds data to a file, increasing its size.
4. Read: Retrieves data from the file.



Additional File Operations

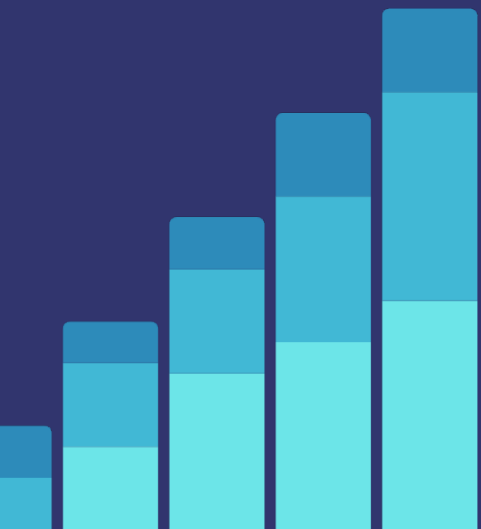
1. Seek/Re-position: Moves the file pointer for direct access.
2. Delete: Removes a file, freeing space.
3. Truncate: Deletes file content, retaining attributes.
4. Append: Adds data to the end of a file.
5. Close: Saves changes and releases resources.

File Access Methods



- Three main methods: sequential, direct, and indexed.
- Each method has specific advantages, depending on the application.

Sequential Access



- Files are accessed one record at a time, starting from the beginning.
- The pointer moves through the file as each record is read or written.



Advantages of Sequential Access

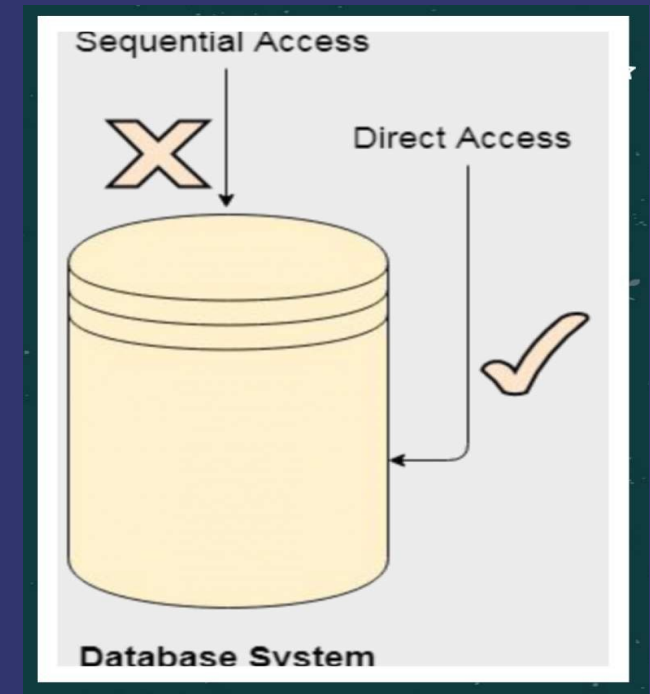
1. Simple to implement.
2. Efficient for reading large files and backup operations.
3. Less prone to data corruption.



Disadvantages of Sequential Access

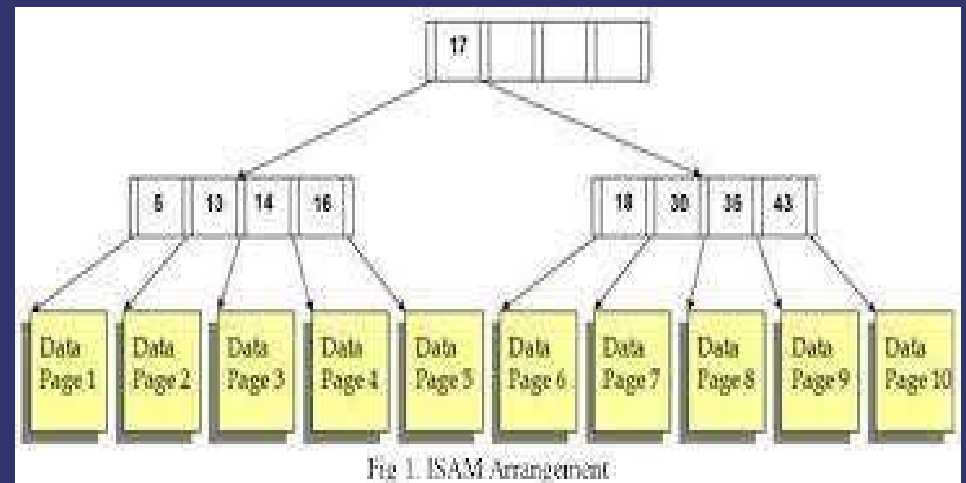
1. Slow if the desired record is not adjacent.
2. Not suitable for quick access to specific records.
3. Inefficient for frequent updates or modifications.

Direct Access



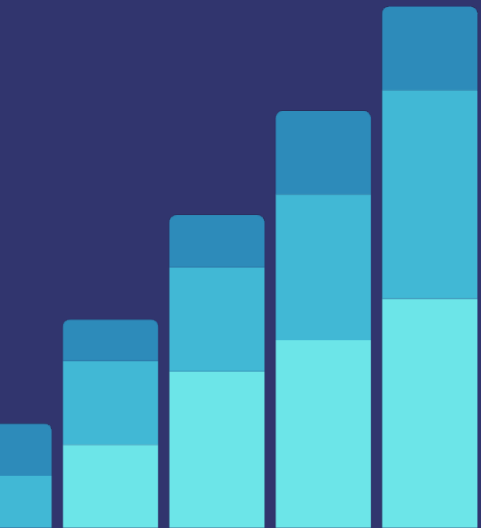
- Direct access retrieves data from any block in a file without sequential traversal.
- Efficient for databases and systems that require specific data quickly.

Indexed Access



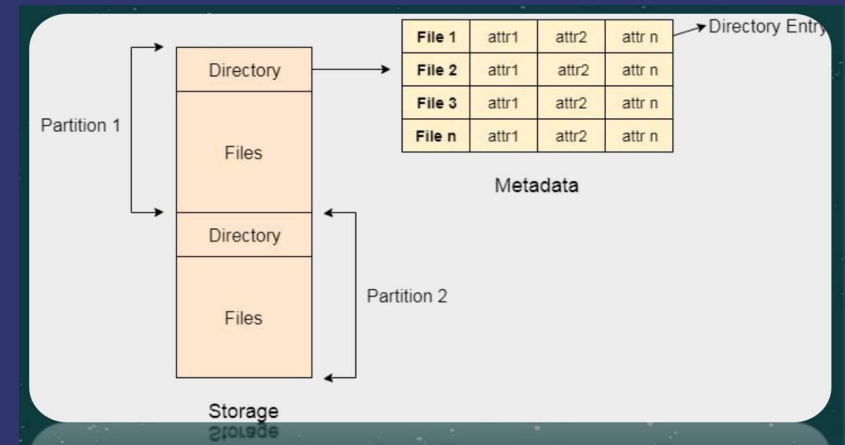
- An index is used to quickly locate specific records in a file.
- Efficient for large databases but requires extra memory for the index.
- Combines benefits of direct access with fast searching.

Directory Structure



- A directory lists related files, storing their attributes.
- Hard disks can be partitioned into volumes, with each partition containing directories.
- A directory entry holds information about each file.

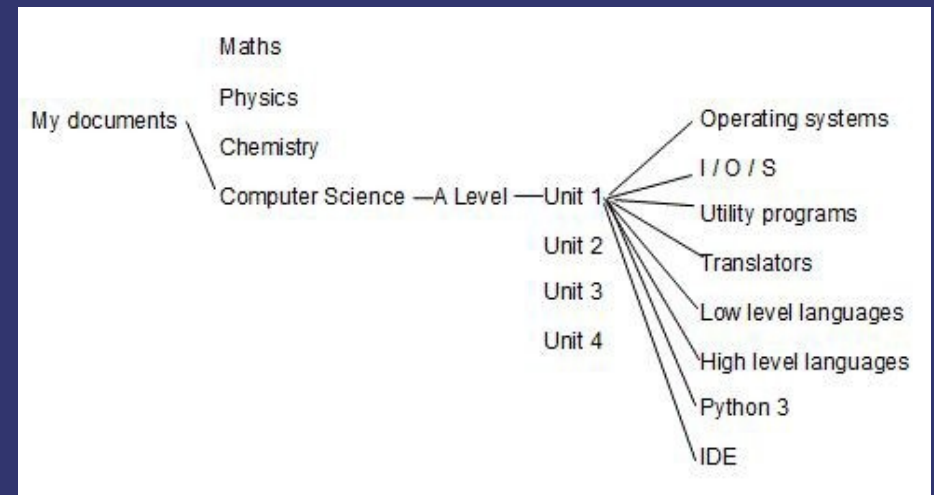
What is a Directory?



- A directory is a container that organizes files and folders.
- Directories are often called "folders" and can contain subdirectories.
- Hierarchical directories are structured like a tree, with the root as the base (e.g., C: drive).
- Directories help logically organize files for easier access.

Directory Structure

Example



- Example: A student's "My Documents" directory may contain subdirectories for each subject.
- Subdirectories further organize files into focused categories, making them easy to locate later.
- Files related to each topic or subject are stored in relevant subdirectories.



Common Directory Operations

File operations in a directory include:

1. File Creation
2. File Search
3. File Deletion
4. File Renaming
5. Traversing Files
6. Listing Files



File Systems in Operating Systems

- A file system organizes and manages files on storage devices.
- Common file systems:
 - FAT (File Allocation Table) – used by older Windows versions.
 - NTFS (New Technology File System) – modern Windows systems.
 - ext – used by Linux and Unix-based systems.
 - HFS/APFS – used by macOS.



Advantages of File Systems

1. Organization: Helps categorize and locate files easily.
2. Data Protection: Supports features like permissions, backup, and error correction.
3. Performance: Efficient file organization improves reading and writing speeds.



Disadvantages of File Systems

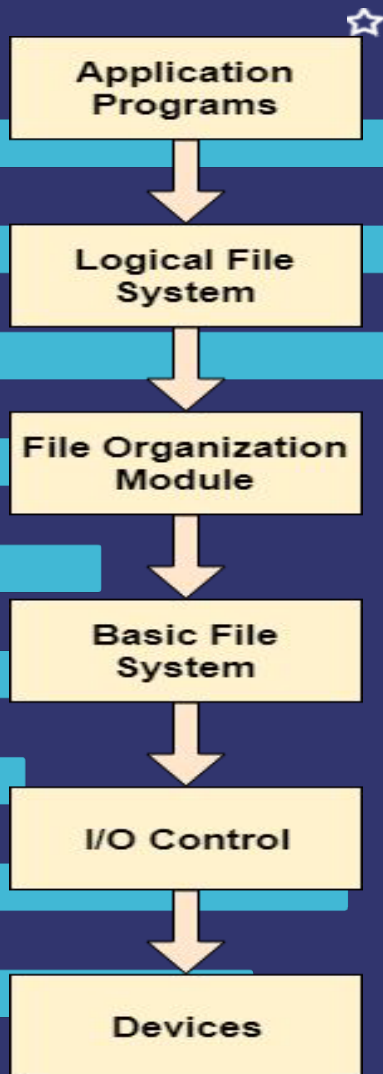
1. Compatibility Issues: Different systems may not be compatible across platforms.
2. Disk Space Overhead: Metadata storage reduces available user space.
3. Vulnerability: File systems are prone to corruption and malware attacks.



File System Responsibilities

1. **File Structure:** Maintains optimal file storage.
2. **Recovering Free Space:** Reallocates space after file deletion.
3. **Disk Space Assignment:** Determines file placement on disk.
4. **Tracking Data Location:** Keeps track of non-contiguous data blocks.

Layers of the File System



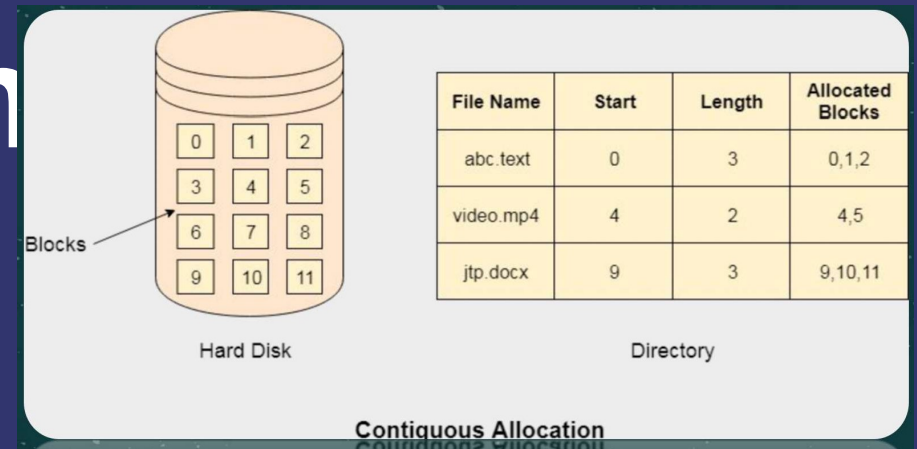
1. Logical File System: Handles file metadata and directory structure.
2. File Organization Module: Maps logical blocks to physical blocks.
3. Basic File System: Issues commands to the I/O control.
4. I/O Control: Manages device drivers and accesses the hard disk.



Allocation Methods

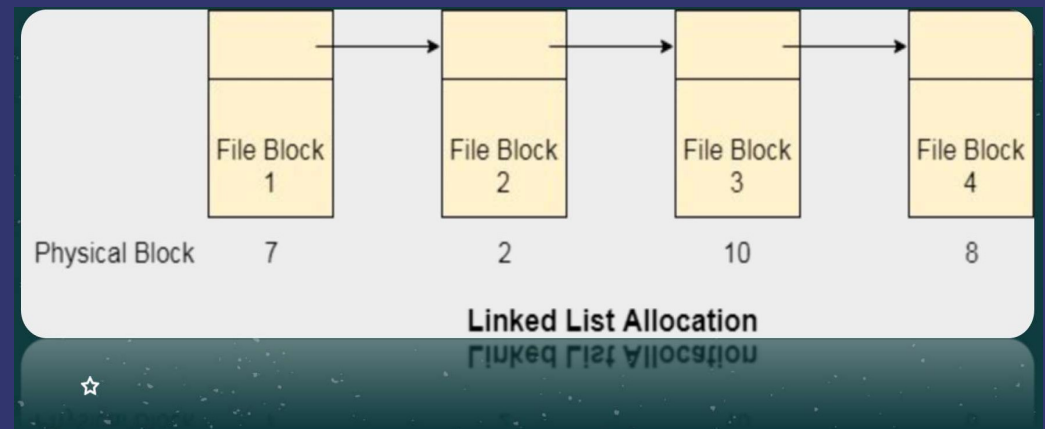
- Disk space allocation methods impact performance:
 - **Contiguous Allocation:** Stores files in sequential blocks.
 - **Linked Allocation:** Files stored in non-contiguous blocks using pointers.
 - **FAT:** Stores pointers in a table.
 - **Indexed Allocation:** Uses an indexed block containing pointers to file blocks.

Contiguous Allocation



- Advantages: Simple to implement, supports random access, good performance.
- Disadvantages: Causes fragmentation, hard for files to grow in size.

Linked List Allocation



- **Advantages:** No fragmentation, files can grow as needed.
- **Disadvantages:** No random access, pointers consume space, risk of broken links.



File Allocation Table (FAT)

- **Advantages:** Provides random access, uses entire disk block for data.
- **Disadvantages:** FAT size grows with the number of blocks, causing inefficiency.



Indexed Allocation

- **Advantages:** Supports direct access, isolates bad blocks to specific data.
- **Disadvantages:** Bad index block can cause file loss, overhead with small files.

**THANK
YOU**