

Operating Systems

Project: -

Implementation File System using Data Structure

Description: -

Most file systems in use today store the files in a **tree (or hierarchical) structure**.

At the top of the tree is **one (or more) root nodes (partitions in Microsoft Windows)**.

Under the root node, there are **files and directories (folders in Microsoft Windows)**.

Each directory can contain **files and subdirectories**, which in turn can contain files and subdirectories, and so on, potentially to an **almost limitless depth**.

A file system is an application that **stores and organizes** files and their data (content).

Essentially, it organizes these files for storage, organization, manipulation, and retrieval.

The way a computer or an application:

organizes, names, stores, secures, searches and manipulates files.

depends and is implemented through what we refer as a file system.

File systems vary greatly. **(Choose one of them to implement)**

- In some cases, the file system is a **single hierarchy of files with one top-level root directory (Linux)**.
- In other cases, it may have **several distinct file hierarchies, each with its own top-level root directory. (Windows C: or D:)**.

Types of Files: -

1) File: -

a) Attributes: -

- Name:** It is the only information stored in a human-readable form. **E.g.(project.docx)**
- Identifier:** Every file is identified by a unique number within a file system known as an identifier. **E.g. (91011)**
- Location:** logically by its path. **E.g. (C:\Users\YourUsername\Documents)**
- Type:** This attribute is required for systems that support various types of files. **E.g. (Document File)**
- Size:** Attribute used to display the current file size. **E.g. (40.5 KB)**
- Blocks:** number of blocks in which the file is stored **E.g. (If size is 2 bytes and block size is 4 bytes then is stored in 1 block)**
- Protection:** This attribute assigns and controls the access rights of reading (r), writing (w), and executing (x) the file. **E.g. (rw-) which mean have access to read and write.**
- Creation Time (Birth Time):** The timestamp indicating when a file was created
- Modification Time (Last Modified Time):** The timestamp indicating when the content of a file was last modified.

- x) **Access Time (Last Accessed Time):** The timestamp indicating when a file was last accessed or read. E.g. (2015-09-15 13:50:26)
- xi) **Content:** let's consider that the content is string of characters and each character has size 1 byte.

b) Actions and Functions

- i) **Create File:** This action involves the creation of a new file.
- ii) **Read from File:** This action involves reading the content of a file.
- iii) **Write to a File:** This action involves adding or modifying content in a file.
- iv) **Copy a File:** This action involves creating a duplicate of a file in specific path.
- v) **Move/Rename a File:** This action involves changing the location or name of a file.
- vi) **Delete a File:** This action involves removing a file from the file system.
- vii) **Get File Information:** This function retrieves all information about a file.
- viii) **Change File Permissions:** This function allows the modification of file access permissions.

2) Directory

a) Attributes: -

- i) A directory is just a file with the directory type
- ii) without content
- iii) that may have children (**files and subdirectories**)

b) Actions and Functions

- i) **Create a Directory (Folder):** This action involves the creation of a new directory.
- ii) **Delete a Directory:** This action involves removing a directory from the file system.
- iii) **List Directory Contents:** This function retrieves a list of files and directories within a specified directory.
- iv) **Get Directory Information:** This function retrieves all information about a directory.
- v) **Change Directory Permissions:** This function allows the modification of directory access permissions.
- vi) **Search in Directory:** This function take name and retrieve all files and subdirectories that have a similar name.

3) Root Node (Partition)

a) Attributes: -

- i) **Partition Label/Name:** A human-readable name or label assigned to the partition.
- ii) **UUID (Universally Unique Identifier):** A unique identifier assigned to the partition, used for referencing it without relying on names or device paths.
- iii) **Size:** Represents the total size of the partition.
- iv) **Used Space:** Represents the amount of space already utilized on the partition.
- v) **Free Space:** Indicates the amount of available or unallocated space on the partition.
- vi) **Block Size:** determines how many bytes are allocated to each block on file system.