Team details:

Member1:

Name : S Shreyas

SRN : PES1UG22CS494

Member2:

Name : Saathvick R Shankar

SRN : PES1UG22CS497

Member3:

Name : Shashank P

SRN : PES1UG22CS548

Problem statement:

You are tasked with creating a file and directory management system that simulates the functionality of a file system on a computer. The system should allow users to efficiently organise, navigate files and directories.

Core Features:

1. Metadata Management
   * Manage metadata for files and directories, including attributes like file size, creation date, and modification date. Use data structures to store and retrieve metadata efficiently.
2. Path Resolution Using path resolution access the metadata of specific files
   * Here's how the file path resolution process works:

i. *Relative Path:* The relative path to the "report.docx" file is "..\Work\report.docx." It indicates that we need to move one directory up from the current working directory and then access the "Work" directory to find the file.

ii. *Resolving the Absolute Path:* - Starting with the current working directory, which is "C:\Users\John\Documents," we move one directory up to "C:\Users\John." This represents the ".." part of the path. - Next, we append "Work" to the path, resulting in "C:\Users\John\Work." - Finally, we add "report.docx" to the path, giving us the complete absolute path: "C:\Users\John\Work\report.docx."

iii. *File Access:* With the absolute path "C:\Users\John\Work\report.docx," the system can access the "report.docx" file directly, and you can perform operations on it, such as opening, editing, or copying.

**Solution:**

These are the functions used in the program.

Here the metadata of the code is displayed in the display hierarchical function.

1. **initializeFileSystem:**
   * Initializes the file system with a root directory at "C:".
2. **createNode:**
   * Creates a new tree node (file or directory) with the given name and type.
3. **addNodeToDirectory:**
   * Adds a child node to a directory, maintaining the linked list of children.
4. **listDirectories:**
   * Lists the directories within a given directory.
5. **findDirectoryByName:**
   * Searches for a directory with a specified name within a starting directory.
6. **createDirectory:**
   * Creates a new directory with the given name and adds it to the current directory.
7. **createFile:**
   * Creates a new file with the given attributes and adds it to the current directory.
8. **printAbsolutePath:**
   * Prints the absolute path of a given node in the file system.
9. **printRelativePath:**
   * Prints the relative path of a given node in the file system.
10. **changeDirectory:**
    * Allows the user to change the current directory based on user input.
11. **displayFileSystem:**
    * Displays the hierarchy of the file system, including file details for each file node.
12. **printOptions:**
    * Prints the available options for the user to interact with the file system.

This is basically everything that’s present in our code.