

# In-Memory Management System

## System Overview:

The in-memory file system is implemented in Java as a console-based application. The system supports basic file operations such as creating directories, navigating directories, listing directory contents, searching for patterns in files, displaying file contents, creating new files, writing to files, moving files or directories, copying files or directories, and removing files or directories.

## Data Structures Used:

Directory Class:

- Represents a directory and contains references to child directories and files.
- Stores child directories in a `Map<String,Directory>` for efficient lookup.
- Stores files in a `Map<String,File>` for efficient lookup.

File Class:

- Represents a file and contains the file content.

## Design Decisions:

Command-Line Interface (CLI):

- The system is designed as a command-line interface to interact with the user.

In-Memory Storage:

- The file system's state is stored in memory using Java data structures (`Map`, `String`, etc.).

Object Serialization for State Saving/Loading:

- The system supports saving and loading the file system state using Java's object serialization. The state can be saved to a file and loaded from that file.

Command Processing:

- The `processCommand` method interprets user input commands and calls the corresponding methods.

## Features

- `mkdir`: Create a new directory.
- `cd`: Change the current directory, supporting navigation using "." and absolute paths.
- `ls`: List the contents of the current or specified directory.
- `grep`: Search for a specified pattern in a file (Bonus feature).
- `cat`: Display the contents of a file.
- `touch`: Create a new empty file.
- `echo`: Write text to a file.
- `mv`: Move a file or directory to another location.
- `cp`: Copy a file or directory to another location.
- `rm`: Remove a file or directory.

### Commands:

- `mkdir <directory_path>`: Create a new directory.
- `cd <directory_path>`: Change the current directory.
- `ls [<directory_path>]`: List the contents of the current or specified directory.
- `grep <pattern> <file_path>`: Search for a pattern in a file.
- `cat <file_path>`: Display the contents of a file.
- `touch <file_path>`: Create a new empty file.
- `echo <content> <file_path>`: Write text to a file.
- `mv <source_path> <destination_path>`: Move a file or directory.
- `cp <source_path> <destination_path>`: Copy a file or directory.
- `rm <file_path>`: Remove a file or directory.

## Setup Script or Dockerfile:

To run the Java application, you need to have Java Development Kit (JDK) installed on your system.

For running the file either run the file `Main.java` or do the input the following commands in the terminal

Compile the Main file

```
javac Main.java
```

## Run the Main file

*java Main*

Also incase for saving or loading state take an input that tell if the state of the file system needs to be saved in persistent memory when the process is terminated and one that tells if it needs to be reloaded when the new process starts.

For Saving: *java Main --save-state true --path /path/to/save/state.dat*

For Loading :*java Main --load-state true --path /path/to/load/state.dat*