# In-Memory File System Documentation

## Introduction

This is a simple in-memory file system implemented in Python, providing basic functionalities similar to those found in a traditional file system. The file system operates within a console interface, allowing users to interact with it through various commands.

## Class: InMemoryFileSystem

### Attributes

- `root`: A dictionary representing the root directory of the file system.

- `current\_directory`: A reference to the current working directory within the file system.

### Methods

#### `\_\_init\_\_(self)`

Initializes the file system with a root directory.

#### `mkdir(self, dirname: str)`

Creates a new directory with the specified name in the current directory.

#### `cd(self, path: str)`

Changes the current directory based on the provided path (relative or absolute).

#### `ls(self, path: str = '.')`

Lists the contents of the specified directory. Defaults to the current directory if no path is provided.

#### `grep(self, filename: str, pattern: str) -> List[str]`

Searches for a specified pattern in the contents of a file.

#### `cat(self, filename: str)`

Displays the contents of a file.

#### `touch(self, filename: str)`

Creates a new empty file with the specified name.

#### `echo(self, filename: str, text: str)`

Writes text to a file, creating the file if it does not exist.

#### `mv(self, source: str, destination: str)`

Moves a file or directory to another location within the file system.

#### `cp(self, source: str, destination: str)`

Copies a file or directory to another location within the file system.

#### `rm(self, path: str)`

Removes a file or directory from the file system.

## Functions

### `exec\_command(file\_system: InMemoryFileSystem, command: str)`

Executes a user command on the file system.

## Usage

The file system can be interacted with through the console interface. Users can input various commands to perform operations such as creating directories, navigating the file system, listing directory contents, searching for patterns in files, and more.

### Basic Commands

- `mkdir <directory\_name>`: Create a new directory.

- `cd <path>`: Change the current directory.

- `ls [path]`: List the contents of the current or specified directory.

- `grep <filename> <pattern>`: Search for a pattern in a file.

- `cat <filename>`: Display the contents of a file.

- `touch <filename>`: Create a new empty file.

- `echo <filename> <text>`: Write text to a file.

- `mv <source> <destination>`: Move a file or directory.

- `cp <source> <destination>`: Copy a file or directory.

- `rm <path>`: Remove a file or directory.

- `exit`: Exit the file system.

Note: The script supports saving and loading the state using command-line arguments for improved persistence.