CSE 222 HOMEWORK 4: File & Folder Directory Tree System

Gebze Technical University Computer Engineering Department

Submission Deadline: 22th April 2024 23:59

1 Introduction

This assignment involves the implementation of a File and Folder Directory Tree system using recursion and object-oriented programming. The objective is to simulate directory structures and file management operations found in real-world file systems.

2 Assignment Requirements

2.1 Directory and File Structure

Construct a directory system where each node is an instance of a class representing a file or directory. Nodes have the following attributes:

- **name** - The file or directory name.
- **date created** - The creation timestamp using Java's `Timestamp` class.
- **parent** - A reference to the parent directory.

3 OOP Design Specifications

3.1 Class Hierarchy

Develop the following classes for a robust directory system:

- **FileSystemElement**: An abstract class with common attributes and methods for both files and directories. Attributes include:
  - **name** (String): The name of the element.
  - **dateCreated** (Timestamp): The creation timestamp.
  - **parent** (FileSystemElement): Reference to the parent directory. Null if this is the root.
- **File**: Inherits from FileSystemElement and represents a file.

- **Directory**: Inherits from FileSystemElement and represents a directory that can contain other files or directories. It must manage a list of **FileSystemElement** objects representing its children. It includes:
  - **children** (Linked List of FileSystemElement): A **Linked List** containing child elements.
  - Methods to add, remove, and manage children.
- **FileSystem**: Manages the root directory and contains methods for user interactions such as navigating the directory tree, and managing files and directories.

4 Core Features

Implement the following functionalities:

- Create, delete, and move files and directories. **When moving a directory, do not make any copying operation. You should manipulate directory's reference and put it to the children list of its parent.**
- Recursively delete directories and their contents.
- Display the current path.
- Search for files or folders recursively.
- Print the directory tree showing its structure.
- List contents of the current directory, with directories marked.

5 Advanced Features

Include features such as:

- Sorting files within a directory by creation date.
- An interactive command-line interface for system operations.

6 User Interface

The user interface for this project will be a command-line interface (CLI) designed to facilitate interaction with the file system. Below is a template of the main menu, which users will see upon starting the application:

```
===== File System Management Menu =====
1. Change directory
2. List directory contents
3. Create file/directory
4. Delete file/directory
5. Move file/directory
6. Search file/directory
```

```
7. Print directory tree
8. Sort contents by date created
9. Exit
Please select an option:
```

Users will interact with the system by typing the number corresponding to their choice and pressing Enter. The system will then prompt the user for any additional information required to perform the selected action.

7 Example Outputs

This section provides detailed examples of expected outputs and command-line interactions for various operations within the file system.

7.1 Changing Directory

```
Current directory: /home/user/Documents
Enter new directory path: /home/user/Documents/Projects
Directory changed to: /home/user/Documents/Projects
```

7.2 Listing Directory Contents

```
Listing contents of /home/user/Documents/Projects:
* Project1/
* Project2/
Report.docx
Notes.txt
```

7.3 Creating Files and Directories

```
Current directory: /home/user/Documents/Projects
Create file or directory (f/d): d
Enter name for new directory: Project3
Directory created: Project3/
```

```
Current directory: /home/user/Documents/Projects
Create file or directory (f/d): f
Enter name for new file: MeetingNotes.txt
File created: MeetingNotes.txt
```

7.4 Deleting Files and Directories

```
Current directory: /home/user/Documents/Projects
Enter name of file/directory to delete: Project3
Directory deleted: Project3/
```

```
Current directory: /home/user/Documents/Projects
Enter name of file to delete: MeetingNotes.txt
File deleted: MeetingNotes.txt
```

7.5 Moving Files and Directories

```
Current directory: /home/user/Documents/Projects
Enter the name of file/directory to move: Report.docx
Enter new directory path: /home/user/Documents/Reports
File moved: Report.docx to /home/user/Documents/Reports
```

7.6 Searching for Files and Directories

```
Search query: Report.docx
Searching from root...
Found: /home/user/Documents/Reports/Report.docx
```

7.7 Printing the Directory Tree

```
Path to current directory from root:
* root/
  * home/
    * user/
      * Documents/
        * Projects/
          * Project2/ (Current Directory)
            project-outline.txt
            research-notes.txt
```

7.8 Sorting Directory Contents

```
Sorted contents of /home/user/Documents by date created:
* Projects/ (2024-01-15 12:34:56)
* Reports/ (2024-02-20 08:15:00)
Resume.pdf (2024-03-01 09:00:30)
```

8 Submission Guidelines

Submissions should include complete JavaDoc documentation, PDF report, and a Makefile for compilation. Code should be well-commented, detailing purpose, inputs, outputs, and side effects.

9 Evaluation Criteria

Projects will be evaluated based on functionality, adherence to OOP principles, code quality, interface usability, and documentation completeness.