project

Questions Assigned

Question 1

File System Simulator:

Data Structures: Tree (linked list representation) for directory structure, linked list for file information within directories.
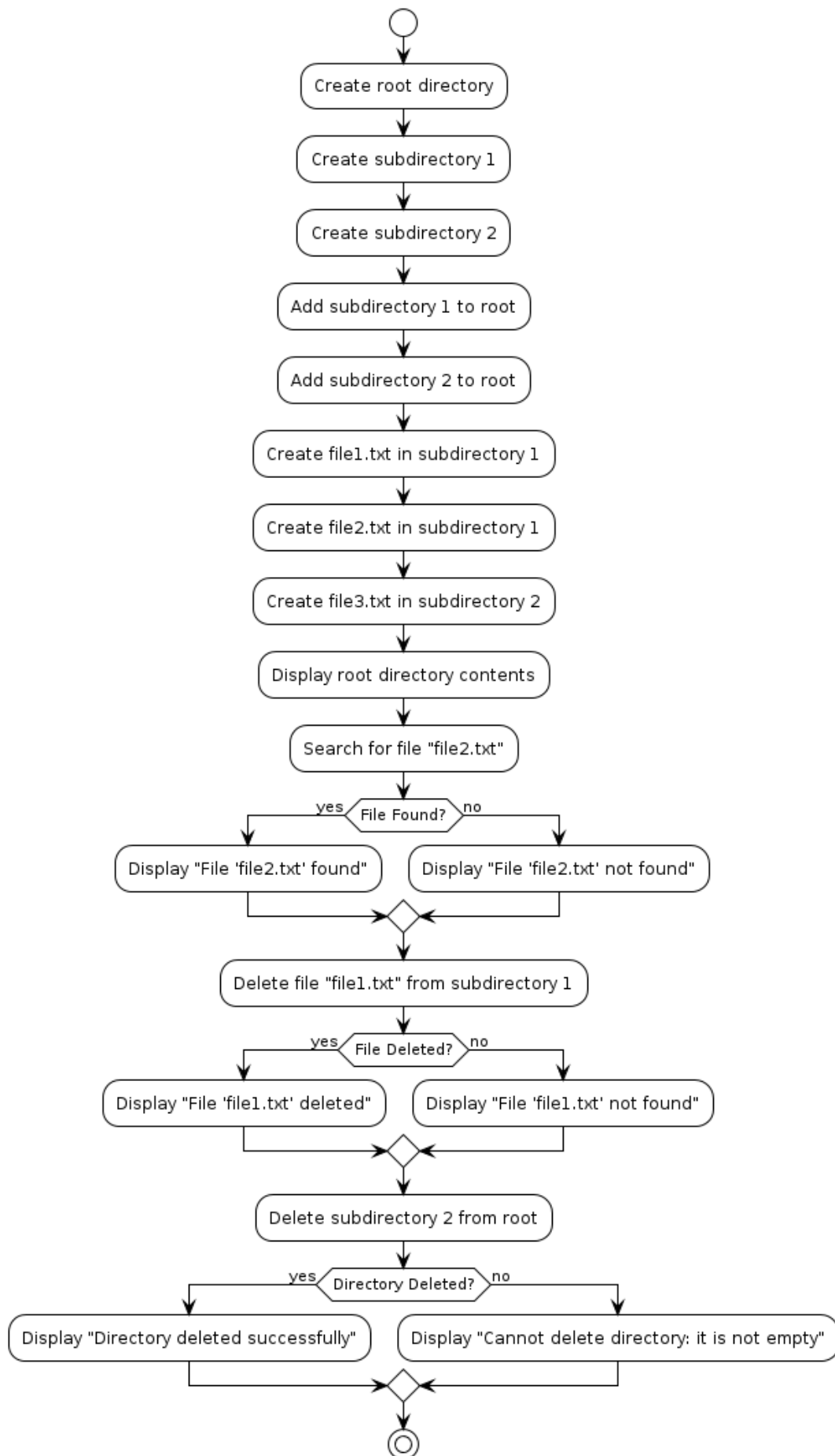
Functionality:

Create directories and files.

View directory contents (list files and subdirectories).

Navigate into subdirectories.

Delete files and directories (handle potential issues like non-empty directories).

Search for files by name.

```
○
↓
Create root directory
↓
Create subdirectory 1
↓
Create subdirectory 2
↓
Add subdirectory 1 to root
↓
Add subdirectory 2 to root
↓
Create file1.txt in subdirectory 1
↓
Create file2.txt in subdirectory 1
↓
Create file3.txt in subdirectory 2
↓
Display root directory contents
↓
Search for file "file2.txt"
↓
```

File Found? — yes → Display "File 'file2.txt' found" | no → Display "File 'file2.txt' not found"

```
↓
Delete file "file1.txt" from subdirectory 1
↓
```

File Deleted? — yes → Display "File 'file1.txt' deleted" | no → Display "File 'file1.txt' not found"

```
↓
Delete subdirectory 2 from root
↓
```

Directory Deleted? — yes → Display "Directory deleted successfully" | no → Display "Cannot delete directory: it is not empty"

```
↓
◎
```

Simulator.c

```c
#include "lltree.h"

#include <stdio.h>

#include <stdlib.h>

#include <string.h>


Directory* createDirectory(char* name) {

    Directory* newDirectory = (Directory*)malloc(sizeof(Directory));

    if (newDirectory == NULL) {

        printf("Memory allocation failed.\n");

        exit(EXIT_FAILURE);

    }

    strcpy(newDirectory->name, name);

    newDirectory->subdirectories = NULL;

    newDirectory->files = NULL;

    newDirectory->next = NULL;

    return newDirectory;

}


File* createFile(char* name) {

    File* newFile = (File*)malloc(sizeof(File));

    if (newFile == NULL) {

        printf("Memory allocation failed.\n");

        exit(EXIT_FAILURE);

    }

    strcpy(newFile->name, name);

    newFile->next = NULL;

    return newFile;

}


void addDirectory(Directory** root, char* name) {

    Directory* newDirectory = createDirectory(name);

    newDirectory->next = *root;

    *root = newDirectory;
```

```c
}

void addFile(Directory* directory, char* name) {
    File* newFile = createFile(name);
    newFile->next = directory->files;
    directory->files = newFile;
}

void viewDirectoryContents(Directory* directory) {
    printf("Directory: %s\n", directory->name);
    printf("Files:\n");
    File* filePtr = directory->files;
    while (filePtr != NULL) {
        printf("- %s\n", filePtr->name);
        filePtr = filePtr->next;
    }
    printf("Subdirectories:\n");
    Directory* subdirectoryPtr = directory->subdirectories;
    while (subdirectoryPtr != NULL) {
        printf("- %s\n", subdirectoryPtr->name);
        subdirectoryPtr = subdirectoryPtr->next;
    }
}

Directory* navigateToSubdirectory(Directory* root, char* name) {
    Directory* current = root->subdirectories;
    while (current != NULL) {
        if (strcmp(current->name, name) == 0)
            return current;
        current = current->next;
    }
    printf("Subdirectory '%s' not found.\n", name);
    return NULL;
}
```

```c
void deleteFile(Directory* directory, char* name) {
    File* current = directory->files;
    File* prev = NULL;
    while (current != NULL && strcmp(current->name, name) != 0) {
        prev = current;
        current = current->next;
    }
    if (current == NULL) {
        printf("File '%s' not found.\n", name);
        return;
    }
    if (prev == NULL)
        directory->files = current->next;
    else
        prev->next = current->next;
    free(current);
}


void deleteDirectory(Directory** root, char* name) {
    Directory* current = *root;
    Directory* prev = NULL;
    while (current != NULL && strcmp(current->name, name) != 0) {
        prev = current;
        current = current->next;
    }
    if (current == NULL) {
        printf("Directory '%s' not found.\n", name);
        return;
    }
    if (prev == NULL)
        *root = current->next;
    else
        prev->next = current->next;
    free(current);
}
```

```c
File* searchFile(Directory* directory, char* name) {

    File* current = directory->files;

    while (current != NULL) {

        if (strcmp(current->name, name) == 0)

            return current;

        current = current->next;

    }

    return NULL;

}
```

Simulator .h

```c
#ifndef LLTREE_H

#define LLTREE_H


// Structure for a file

struct File {

    char name[100];

    struct File* next;

};


typedef struct File File;


// Structure for a directory

struct Directory {

    char name[100];

    struct Directory* subdirectories;

    struct File* files;

    struct Directory* next;

};


typedef struct Directory Directory;


// Function declarations
```

```c
Directory* createDirectory(char* name);

File* createFile(char* name);

void addDirectory(Directory** root, char* name);

void addFile(Directory* directory, char* name);

void viewDirectoryContents(Directory* directory);

Directory* navigateToSubdirectory(Directory* root, char* name);

void deleteFile(Directory* directory, char* name);

void deleteDirectory(Directory** root, char* name);

File* searchFile(Directory* directory, char* name);


#endif /* LLTREE_H */
```

Simulatormain.c

```c
#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include "lltree.h"


int main() {
    clock_t start, end;

    double cpu_time_used;


    Directory* root = createDirectory("root");


    // Creating directories and files
    addDirectory(&root, "folder1");

    addFile(root, "file1.txt");


    // Viewing directory contents
    viewDirectoryContents(root);


    // Navigating to subdirectory
    Directory* subdirectory = navigateToSubdirectory(root, "folder1");

    if (subdirectory != NULL)
```

```
    printf("Navigated to subdirectory: %s\n", subdirectory->name);


  // Deleting file

  deleteFile(root, "file1.txt");


  // Deleting directory

  deleteDirectory(&root, "folder1");


  // Measure time taken

  start = clock();

  // Perform operations here...

  end = clock();

  cpu_time_used = ((double) (end - start)) / CLOCKS_PER_SEC;

  printf("Time taken: %f seconds\n", cpu_time_used);


  return 0;
```

Explanation:

Directory Structure:

The directory structure can be represented using a tree data structure, where each node represents a directory and its children represent subdirectories and files.

Each node in the tree will contain information such as the directory or file name, and a reference to its parent directory.

You can implement the tree using a linked list representation, where each node has a pointer to its parent directory and a list of child directories/files.

Creating Directories and Files:

To create a directory, you would add a new node to the tree structure under the desired parent directory.

To create a file, you would add information about the file to the list of files within a directory.

Viewing Directory Contents:

To view the contents of a directory, you would traverse the tree starting from the specified directory node and list the files and subdirectories it contains.

Navigating into Subdirectories:

Navigating into a subdirectory involves moving to the corresponding node in the tree structure representing that subdirectory.

Deleting Files and Directories:

Deleting a file involves removing its information from the list of files within its parent directory.

Deleting a directory requires recursively deleting all its contents (files and subdirectories), ensuring to handle non-empty directories properly.

Searching for Files by Name:

Searching for a file involves traversing the tree and looking for files with matching names within the specified directory and its subdirectories.

By implementing these functionalities, you can create a file system simulator that mimics the basic operations of a real file system. Additionally, you can extend the simulator with features like file permissions, file metadata, file copying/moving, and error handling to make it more robust and realistic.

output


```
lltree.c:61:13: note: include '<string.h>' or provide a declaration of 'strcmp'
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ ./lltree
Directory: folder1
Files:
- file1.txt
Subdirectories:
Subdirectory 'folder1' not found.
Time taken: 0.000001 seconds
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ gcc -Wall -pg lltree.c lltreemain.c -o lltree
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ ./lltree
Directory: folder1
Files:
- file1.txt
Subdirectories:
Subdirectory 'folder1' not found.
Time taken: 0.000000 seconds
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ ls gmon.out
gmon.out
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ gprof lltree gmon.out > analysis.txt
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ vim analysis.txt
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ gcc -fprofile-arcs -ftest-coverage lltree.c lltreemain.c -o lltree
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ ./lltree
Directory: folder1
Files:
- file1.txt
Subdirectories:
Subdirectory 'folder1' not found.
Time taken: 0.000003 seconds
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ gcov lltree-lltree.c lltree-lltreemain.c
File 'lltree.c'
Lines executed:68.29% of 82
Creating 'lltree.c.gcov'

File 'lltreemain.c'
Lines executed:93.33% of 15
Creating 'lltreemain.c.gcov'

Lines executed:72.16% of 97
```

Index by function name

rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ gcc -O1 lltreemain.c lltree.c -o lltree
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ ./lltree
Directory: folder1
Files:
- file1.txt
Subdirectories:
Subdirectory 'folder1' not found.
Time taken: 0.000003 seconds
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ gcc -O2 lltreemain.c lltree.c -o lltree
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ ./lltree
Directory: folder1
Files:
- file1.txt
Subdirectories:
Subdirectory 'folder1' not found.
Time taken: 0.000001 seconds
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ gcc -O3 lltreemain.c lltree.c -o lltree
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ ./lltree
Directory: folder1
Files:
- file1.txt
Subdirectories:
Subdirectory 'folder1' not found.
Time taken: 0.000002 seconds
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ gcc -O4 lltreemain.c lltree.c -o lltree
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$ ./lltree
Directory: folder1
Files:
- file1.txt
Subdirectories:
Subdirectory 'folder1' not found.
Time taken: 0.000003 seconds
rps@rps-virtual-machine:~/Documents/demo_repo_ldd/LDD_Batch/manasa$