assignment-final.md 2024-12-04

File System Explorer Assignment

Problem Statement

Implement a file system explorer using an n-ary tree with linked list implementation. Each node represents either a file or directory, with directories able to contain multiple files and subdirectories.

Node Structure Requirements

class Node {

string name;

bool isDirectory;

string content; // For files only

Node\* firstChild; // Points to first child (for directories) Node\* nextSibling; // Points to next sibling in the linked list Node\* parent; // Points to parent directory

};

Core Operations

. Directory Operations

Create directory (mkdir)

Change directory (cd)

Print working directory (pwd)

List contents (ls)

. File Operations

Create file (touch)

Write to file (write)

Read file (cat)

Delete file (rm)

. Path Handling

Support absolute paths (/home/user)

Support relative paths (./docs, ../user)

Maximum path length: 255 characters

Implementation Requirements

Use linked list for sibling connections

Implement proper memory management

Handle basic error cases (file exists, invalid path)

Support file content storage

1 / 2

assignment-final.md 2024-12-04

Sample Usage

int main() {

FileSystem fs;

fs.mkdir("/home");

fs.mkdir("/home/user");

fs.touch("/home/user/notes.txt", "Hello World!");

fs.cd("/home/user");

fs.ls(); // Should show: notes.txt fs.cat("notes.txt"); // Should print: Hello World!

return 0;

}

Evaluation

Correct implementation of n-ary tree with linked list

Proper memory management

Working core operations

Error handling

2 / 2