CS232 Operating Systems
Assignment 1: A simple file system.

CS Program
Habib University

Fall 2023
Due Date: 18 September 2023 @ 11:59PM

1 Introduction

In this assignment we will simulate a simple file system.

1. The whole disk is 128 KB in size.
2. The top most directory is the root directory (/).
3. The system can have a maximum of 16 files/directories.
4. A file can have a maximum of 8 blocks (no indirect pointers). Each block is 1 KB in size.
5. A file/directory name can be of 8 chars max (including NULL char). There can be only one file of a given name in a directory.

2 Disk Layout

The disk layout[1] is as follows: The 128 blocks are divided into 1 super block and 127 data blocks. The superblock contains the 128 byte free block list where each byte contains a boolean value indicating whether that particular block is free or not. Just after the free block list, in the super block, we have the inode table containing the 16 inodes themselves. Each inode is 56 bytes in size and contains metadata about the stored files/directories as indicated by the data structure in the accompanying `filesystem.c`.

Inodes can contain metadata about a file or a directory. The contents of a directory are the a series of directory entries comprising of dirent structures (see `filesystem.c`).

The file or directory names can be a maximum of 8 characters including the NULL character.

3 Required Tasks

Your program should be able to process the following commands:

3.1 Create a file

`syntax: CR filename size`

This command should create a file titled filename of the given size. The filename will be an absolute path.

If there's not enough space in the disk, it should output an error saying ”not enough space”, otherwise it should create a file of the required size filling the file content with small alphabets [a-z] repeated.

[1] NB: this layout is slightly different from the one in the book.

If a directory in the given path does not exist, it should output an error message saying ”the directory XXX in the given path does not exist” where XXX is the name of the missing directory.

If a file with a given pathname already exist, it should give an error ”the file already exists”.

3.2 Delete a file

syntax: `DL filename`

This command should delete a file titled filename. The filename will be an absolute path.

If a directory in the given path does not exist, it should output an error message saying ”the directory XXX in the given path does not exist” where XXX is the name of the missing directory.

If a file with a given pathname does not exist, it should give an error ”the file does not exist”.

3.3 Copy a file

syntax: `CP srcname dstname`

This command should copy a file titled srcname to a file titled dstname. The srcname and dstname will be an absolute paths.

If there’s not enough space in the disk, it should output an error saying ”not enough space”, otherwise it should create a copy of the source file at the destination.

If a directory in the given paths does not exist, it should output an error message saying ”the directory XXX in the given path does not exist” where XXX is the name of the missing directory.

If a file with a given pathname already exist, it should overwrite it.

If either srcname or dstname is a directory, it should give an error saying ”can’t handle directories”.

3.4 Move a file

syntax: `MV srcname dstname`

This command should move a file titled srcname to a file titled dstname. The srcname and dstname will be an absolute paths.

This command should not fail due to space limitations are the source and destination files are of the same size. You can assume that the source file is temporarily stored in RAM, while it is deleted from hard disk and the destination file created.

If a directory in the given paths does not exist, it shoudl output an error message saying ”the directory XXX in the given path does not exist” where XXX is the name of the missing directory.

If a file with a given pathname already exists, it should overwrite it.

If either srcname or dstname is a directory, it should give an error saying ”can’t handle directories”.

3.5 Create a directory

syntax: `CD dirname`

This command should create an empty directory at the path indicated by dirname. The dirname will be an absolute path.

If a directory in the given path does not exist, it should output an error message saying ”the directory XXX in the given path does not exist” where XXX is the name of the missing directory.

If a directory with the given name with the given path exists, it should give an error message saying ”the directory already exists”.

3.6 Remove a directory

`syntax: DD dirname`

This command should remove the directory at the path indicated by dirname. The dirname will be an absolute path. This is a recursive operation, it should remove everything inside the directory from the file system.

If a directory in the given path does not exist, it should output an error message saying ”the directory XXX in the given path does not exist” where XXX is the name of the missing directory.

If a directory with the given name at the given path does not exist, it should give an error message saying ”the directory does not exist”.

3.7 List all files

`syntax: LL`

This command should list all the files/directories on the hard disk along with their sizes. Each file/directory should be listed on a separate line with a space between the name and the size (in bytes).

4 Input

Your program should take a command line argument which will be a file containing the commands to be executed as given in the sampleinput.txt. It should read the commands and execute them one by one. You can assume that the input will always be in the correct format.

After executing every command the program should update the state of the hard disk in a file called ”myfs” in the current directory. When your program terminates, this fill will contain 3 the snapshot of the hard disk at the end of the program.

At the start of your program, it should look for a file titled ”myfs” in the current directory and be able to read the hard disk state (it was stored by your program). If it does not find the file titled ”myfs” in the current directory it should create an empty hard disk by formatting it according to the specified layout and creating a the first root directory (/).

5 Submission and Rubric

5.1 Submission

You will submit the following:

1. `filesystem.c` containing your complete C code
2. `makefile` for compiling and running of your C code. It should contain `compile`, `build` and `clean` targets.
3. PDF report on the data structures and algorithms you have used in your implementation.

Compress all of these files into a zip file and rename it with your CS registration number (CSxxyyyy.zip). Submit the zip file on the Assignment 1 submission module on LMS.

5.2 Rubric

The details are given in the following. Note that you might also be called for a viva at the instructor’s discretion.

5.2.1 Marks

- file commands work as specified: - 35 marks
- directory commands and LL work as specified: - 35 marks
- makefile: - 10 marks
- initialization and saving state works correctly and general - 10 marks
- submission (code legible, commented, PDF correclty formatted): - 10 marks

5.2.2 Penalties

- code doesn't compile: -100 marks
- code has warnings (compile with -Wall): -20 marks
- code has memory leaks: -30 marks WIll be listed in Wall?
- makefile without required targets: -10*number of missing targets
- program crashes: -30 marks How?
- late submission: -20 marks for missing deadline + -10*num days

Mark obtained = max (marks+penalties, 0) Explain this pls

6 Using chatGPT or other AI software

You are not allowed to use any AI software to obtain the code for this assignment. Appropriate tool will be used to evaluate your submission for AI tool usage. If you are found using such a tool, you will be given a straight 0 and an Academic Conduct will be filed against you for academic dishonesty.

7 Plagiarism Policy

We have zero tolerance for plagiarism. Every submission will be screened using a plagiarism detection software. If there is any evidence of plagiarism, the case will be reported to the Office of Academic Conduct and all offenders will get a 0. This is applicable even for cases when the code is copied or a significant amount has been obtained from an online repository on open source platforms like bitbucket or github without proper attribution. In case you are taking any material from online sources, we expect that a proper credit/reference is given to the source.

A short overview of the task in hand?

work in root? or can work on any directory

can be anywhere

code crashes in Wall?
unnecessary memory usage?

Memory leaks where?

Mark obtained function

Can include c files in one main file

we have to make structure for the commands

PDF -> listings and make a report with minor explanations

By Wall, memory leaks good
Code should compile and not get stuck

root pointer -> point to other pointers