File System Project

CSC415 - Spring 2021

The A-Team

Duy Nguyen
Vi Dao
Alex Shirazi
Manuel Hernandez

NAME:	SFSU STUDENT ID	GITHUB
Alex Shirazi	918794583	Alex-Shirazi1
Duy Nguyen	917131389	duybuin
Vi Dao	920353977	vdao182
Manuel Hernandez		mhernandez29

Github link for group submission:
https://github.com/CSC415-Spring2021/filesystemproject-mhernandez29

Purpose: The main objective of this project was to work as a team to implement a file system that functions on the LINUX OS. To start we were given a set of instructions to follow, an api header file to implement,and a shell that utilizes this api.

Scope of Work: The api header file we were to implement was the mfs.h file. The fsshell.c file was the main driver behind our family system and interacts with the file system through the mfs.h api. Our main job was to trace and interpret how the api was meant to work by looking at how the api function calls were used within fsshell along with the structures and function parameters passed in.

Problems: The main issue we had was designing how our file system was to be implemented. We were given an open ended set of instructions on what features the file system was to have and were left on our own on how to implement it. The way we implemented it was up to us and we based this on the mfs.h file api and how the fsshell would call the api. We needed to make sure that our implemented api functioned in the expected way consistently.

Another issue was working as a group remotely, as this meant less facetime and added difficulty in conveying logic and intent. To try and combat this we set up a general chat room using discord where we mainly utilised a text chat room and a voice chat room. The text chat room was very useful as a sort of white board where we could post links to informational resources. Despite the schedule conflicts since everyone had a busy schedule, we managed to communicate with each other using discord and meet up whenever we were available to discuss and help each other with debugging issues.

Description: Our file system uses a linked list like system for our allocation method and a bitmap for freespace management. The main files for our file system are b_io, FSAPI, VCB, bitmap, fsBoot, fsshell,fsLow and mfs. B_io files deal with input/output for the system taking data from linux into the file system and vice versa. FSAPI is mainly a collection of single purpose functions we found we needed in several files so instead of rewriting the logic multiple times we just made a single function call. VCB sets up the

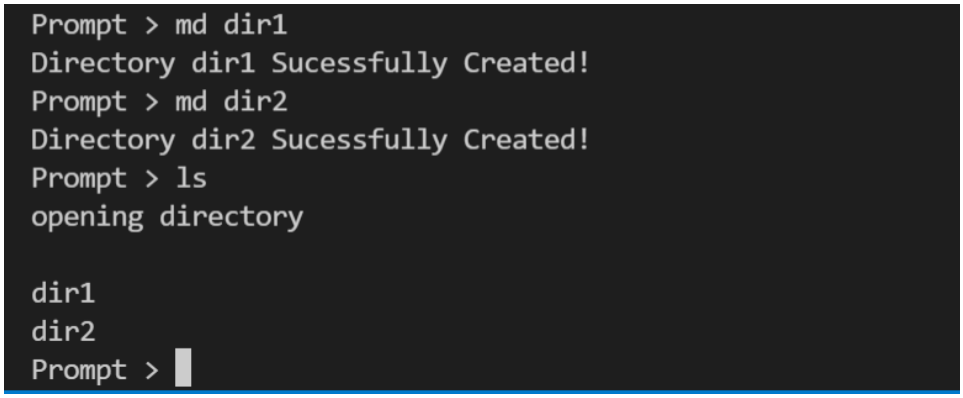
volume control block using arguments passed in by the user. Bitmap initializes the bitmap and also allocates and deallocates memory blocks used by other functions such as fs_mkdir. Mfs contains all the functions called by fsshell and acts as the api between fsshell and the rest of the file system. Bootfs calls all the inits from bitmap, VCB and fsLow. fsLow holds the operating system level functions that execute the low level data transfers and finally fsshell takes user input in the form of commands and passes them to the mfs api to be executed.

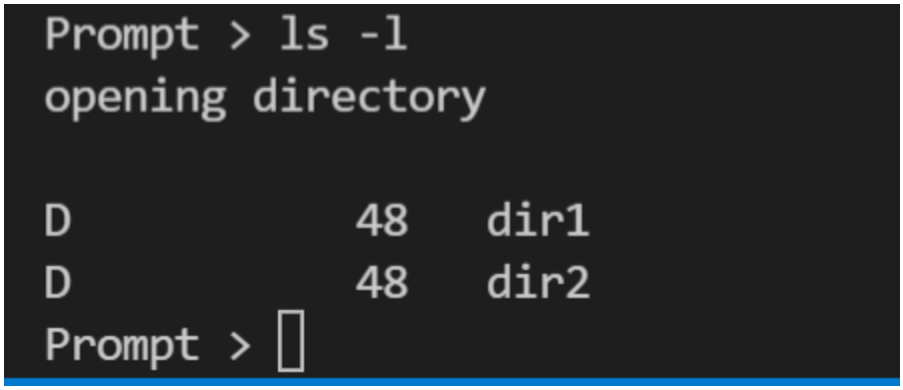
Mfs held two important structs that were the basis of our directory system and by extension our file system. fs_diriteminfo was an in memory only structure where we read in data for directory entries to be used by the rest of the mfs api. fdDir was the second struct and actually held all the data relevant to directory entries and was the structure actually saved to disk. When ever the system read information from disk for directory entries they were in the form of fdDir structs.

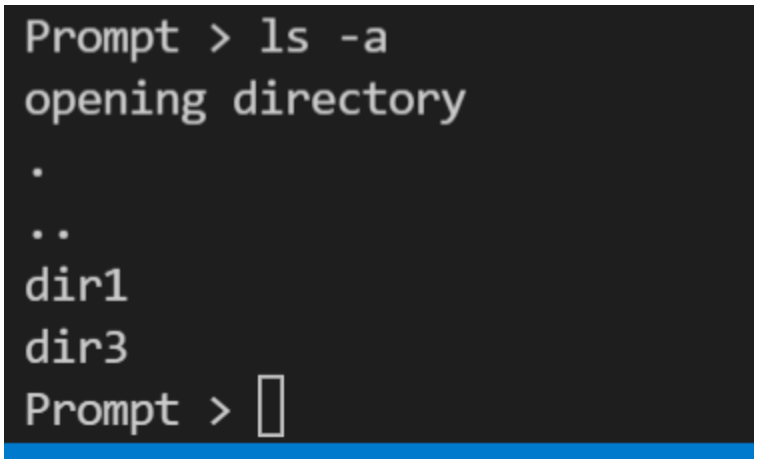
Screenshots of our commands

Md - makes a new directory

Ls - lists out all files/directories in the current directory
Ls - l lists out all files/directories int the current directory with its size and whether the file is a directory or not
Ls -a lists out all directory entries in the current directory including . and ..







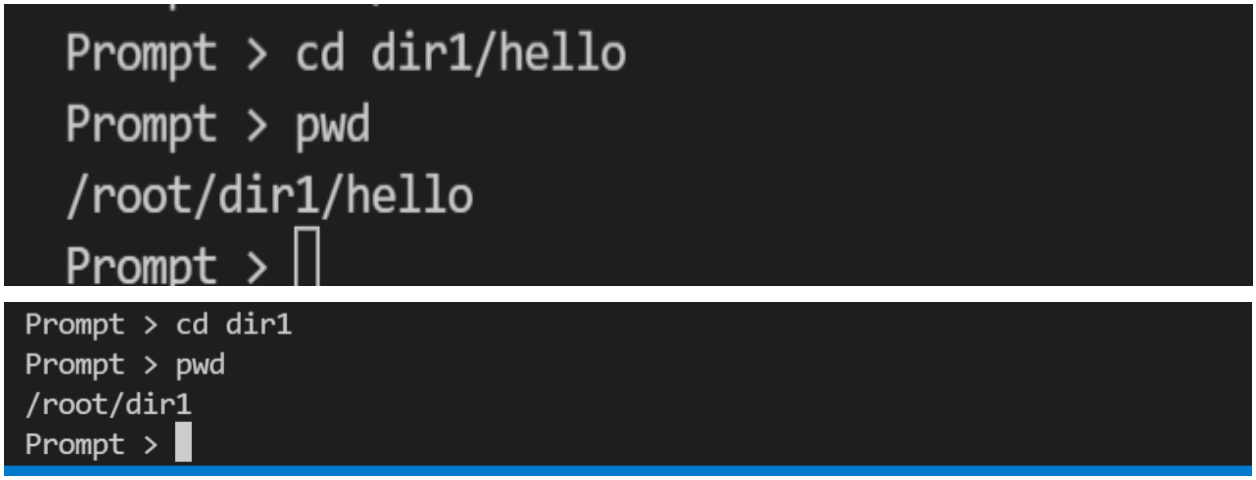
cp2fs - copies file from linux fs to our fs project

```
Prompt > cp2fs test.txt
Prompt > ls
opening directory
dir1
test.txt
Prompt >
```

mv - moves a file from one directory to another

```
Prompt > ls
opening directory
dir1
test.txt
Prompt > mv test.txt dir1
move sucessful!
Prompt > cd dir1
Prompt > ls
opening directory
test.txt
Prompt >
2  tabnine
```

pwd and cd - gets current working directory and changes directory




```
Prompt > cd dir1
Prompt > pwd
/root/dir1
Prompt >
```

rm -removes a directory

```
Prompt > cd /root
Prompt > ls
opening directory
dir1
Prompt > rm dir1

 directory successfully removed
Prompt > ls
opening directory
Prompt >
2  tabnine
```

```
Prompt > md dir1
Directory dir1 Sucessfully Created!
Prompt > ls
opening directory
dir1
Prompt > md dir2
Directory dir2 Sucessfully Created!
Prompt > ls
opening directory
dir1
dir2
Prompt > md dir3
Directory dir3 Sucessfully Created!
Prompt > ls
opening directory
dir1
dir2
dir3
Prompt > rm dir2

 directory successfully removed
Prompt > ls
opening directory
dir1
dir3
Prompt >
```

How to run:

1. Clone the repo from github https://github.com/CSC415-Spring2021/filesystemproject-mhernandez29.git
2. Then type make run to start the fs and type the commands when “prompt>” appears
3. When finished type “exit: to exit the filesystem