```
/**************************************************************
* Class: CSC-415-01 Spring 2020
* Group Name: Team Alpha 1
* Name: Daniel Belmeur
* Student ID: 913525345
* Name: Brad Peraza
* Student ID: 916768260
* Name: Amily Wu
* Student ID:916929226
*
* Project: <Assignment 3-File System
*
* File: <CSC CSC 415 Assignment 3 Report>
*
*Description: A PDF writeup on project
**************************************************************/
```

A description of your file system:

Our file system creates files. It will first create a default virtual hard drive that is stored in secondary memory. When the program exits and then opens again, it will go open and operate on the default virtual hard drive which is like a save point for files created previously.

Issues you had:

The issues we had to face on this project was the communication in our current situation. We weren't able to meet face to face which limited our discussion, concentration, and understanding. One of our team members was unresponsive and never contacted us despite appearing in zoom classes throughout the duration of this project. Another issue we face is the difficulty managing our roles because we were all lost at what to do more or less. It was difficult to understand because we weren't too familiar with the C programming language and overall confused at how to execute our plan and steps. Understanding the material was difficult due to the unaccustomed

terms and references. Our confusions and questions didn't emerge from specific or particular parts of the material but rather the whole thing which made it nebulously harder to solve and detect where the real problems lie.

Detail of how your driver program works:

When you first run it without any values/args it will make default values to make a virtual harddrive to store the files. When you end the program and start it again it will confirm if the default virtual hard drive exists. If you run it with custom values it will make that the hard drive. Then it sets up the volume control block and lists out the general information of the directory. The driver will then set up a rootDirectory as well as a freespace map for open files. It prints the actual directory entries and the total blocks, bytes, and freeBlocksNeeded we have currently. After listing all the general information the program will await for an input from the user. If the user needs help there is an option above that tells the user to type ":h" for help and more information. If ":h" is typed and entered it will list out the options that the user can do. The commands the user can do in the file system is ":o" open a file, ":t" edit a file, ":e" exit a file, and ":q" exit the program. ":h" and ":q" are the only ones working currently while the rest are still a work in progress

Screenshots showing each of the commands listed above:

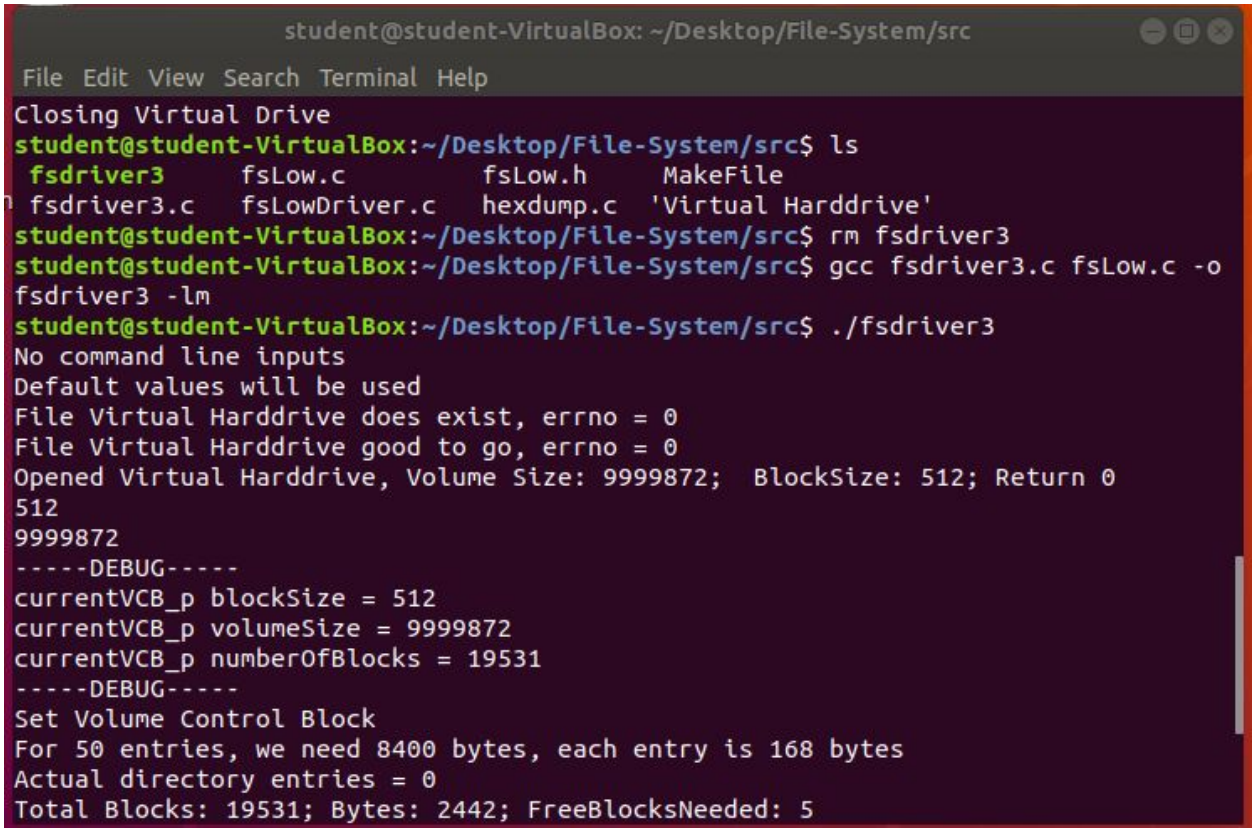
Order :h, :o, :e, :q

```
Please enter in a command: :h
--------USER COMMANDS---------
Template for Commands: 'What you will do' [command]
Exit Program [:q]
Open File [:o]
Edit File [:t]
Exit File [:e]
Seek File [:s]
Please enter in a command:
```

```
Please enter in a command: :o
Please enter in a filename to open: test.txt
We are in myfsOpen
Please enter in a command:
```

```
Please enter in a command: :e
Please enter in a filename to close: test.txt
We are in myfsClose but i dont believe it is setup to work other than free memory
Please enter in a command:
```

```
Please enter in a command: :q
Closing Virtual Drive
student@student-VirtualBox:~/File-System/src$
```


student@student-VirtualBox: ~/Desktop/File-System/src
File Edit View Search Terminal Help
Closing Virtual Drive
student@student-VirtualBox:~/Desktop/File-System/src$ ls
fsdriver3 fsLow.c fsLow.h MakeFile
fsdriver3.c fsLowDriver.c hexdump.c 'Virtual Harddrive'
student@student-VirtualBox:~/Desktop/File-System/src$ rm fsdriver3
student@student-VirtualBox:~/Desktop/File-System/src$ gcc fsdriver3.c fsLow.c -o
fsdriver3 -lm
student@student-VirtualBox:~/Desktop/File-System/src$ ./fsdriver3
No command line inputs
Default values will be used
File Virtual Harddrive does exist, errno = 0
File Virtual Harddrive good to go, errno = 0
Opened Virtual Harddrive, Volume Size: 9999872; BlockSize: 512; Return 0
512
9999872
-----DEBUG-----
currentVCB_p blockSize = 512
currentVCB_p volumeSize = 9999872
currentVCB_p numberOfBlocks = 19531
-----DEBUG-----
Set Volume Control Block
For 50 entries, we need 8400 bytes, each entry is 168 bytes
Actual directory entries = 0
Total Blocks: 19531; Bytes: 2442; FreeBlocksNeeded: 5


For 50 entries, we need 8400 bytes, each entry is 168 bytes
Actual directory entries = 0
Total Blocks: 19531; Bytes: 2442; FreeBlocksNeeded: 5
Changing bit #0 of the 0 byte in free block with flipper 80
Changing bit #1 of the 0 byte in free block with flipper 40
Changing bit #2 of the 0 byte in free block with flipper 20
Changing bit #3 of the 0 byte in free block with flipper 10
Changing bit #4 of the 0 byte in free block with flipper 8
Changing bit #5 of the 0 byte in free block with flipper 4
-----DEBUG-----
-----DEBUG-----
-----DEBUG-----
Read/Write worked
Enter in ':h' for help
Please enter in a command: :h
--------USER COMMANDS---------
Template for Commands: 'What you will do' [command]
Exit Program [:q]
Open File [:o]
Edit File [:t]
Exit File [:e]
Please enter in a command: :q
Closing Virtual Drive
student@student-VirtualBox:~/Desktop/File-System/src$

```
File Virtual Harddrive not good to go, errno = 2
Block size is : 512
Created a volume with 9999872 bytes, broken into 19531 blocks of 512 bytes.
Opened Virtual Harddrive, Volume Size: 9999872;  BlockSize: 512; Return 0
512
9999872
currentVCB_p blockSize = 512
currentVCB_p volumeSize = 9999872
currentVCB_p numberOfBlocks = 19531
Se Terminal e Control Block
For 50 entries, we need 8400 bytes, each entry is 168 bytes
Actual directory entries = 0
Total Blocks: 19531; Bytes: 2442; FreeBlocksNeeded: 5
Changing bit #0 of the 0 byte in free block with flipper 80
Changing bit #1 of the 0 byte in free block with flipper 40
Changing bit #2 of the 0 byte in free block with flipper 20
Changing bit #3 of the 0 byte in free block with flipper 10
Changing bit #4 of the 0 byte in free block with flipper 8
Changing bit #5 of the 0 byte in free block with flipper 4
Read/Write worked
Enter in ':h' for help
Please enter in a command:
```