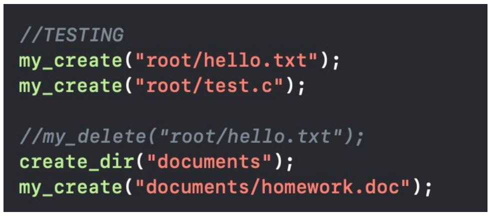
Lab 4 Results
Kateryna Osadchuk

Part 1: Testing

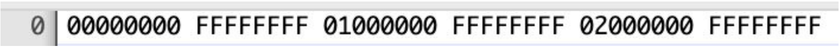
Test One - Creating Files and Directories

command



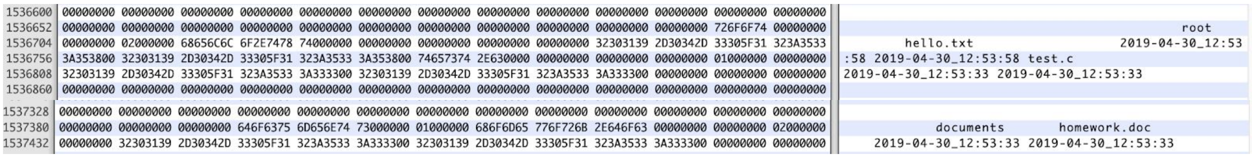
```
//TESTING
my_create("root/hello.txt");
my_create("root/test.c");

//my_delete("root/hello.txt");
create_dir("documents");
my_create("documents/homework.doc");
```

FAT



```
0 | 00000000 FFFFFFFF 01000000 FFFFFFFF 02000000 FFFFFFFF
```

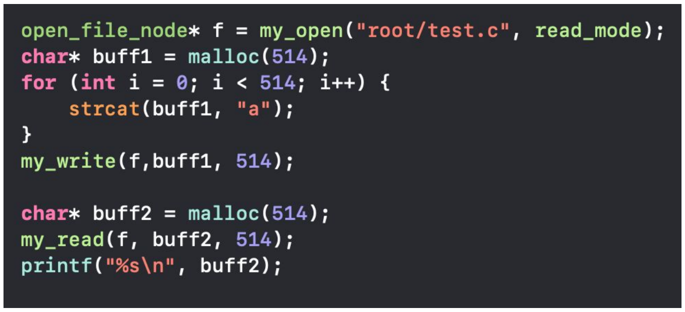
Directory entries



```
1536600 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
1536652 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 726F6F74 00000000                                       root
1536704 00000000 02000000 68656C6C 6F2E7478 74000000 00000000 00000000 00000000 00000000 32303139 2D30342D 33305F31 323A3533        hello.txt                2019-04-30_12:53
1536756 3A353800 32303139 2D30342D 33305F31 323A3533 3A353800 74657374 2E630000 00000000 00000000 00000000 01000000 00000000 :58 2019-04-30_12:53:58 test.c
1536808 32303139 2D30342D 33305F31 323A3533 3A333300 32303139 2D30342D 33305F31 323A3533 3A333300 00000000 00000000 00000000 2019-04-30_12:53:33 2019-04-30_12:53:33
1536860 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

1537328 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
1537380 00000000 00000000 00000000 646F6375 6D656E74 73000000 01000000 686F6D65 776F726B 2E646F63 00000000 00000000 02000000             documents       homework.doc
1537432 00000000 32303139 2D30342D 33305F31 323A3533 3A333300 32303139 2D30342D 33305F31 323A3533 3A333300 00000000 00000000     2019-04-30_12:53:33 2019-04-30_12:53:33
```

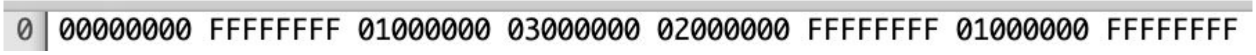
Test Two - Opening Files, Reading, Writing

Command



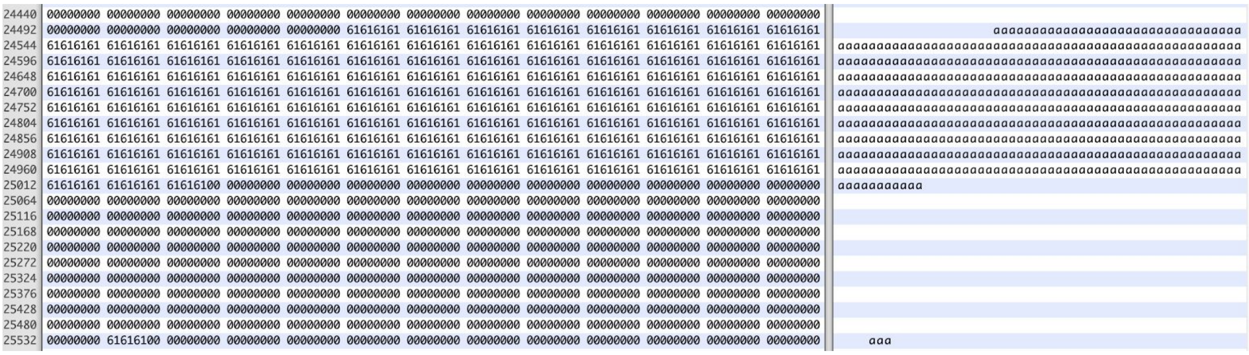
```
open_file_node* f = my_open("root/test.c", read_mode);
char* buff1 = malloc(514);
for (int i = 0; i < 514; i++) {
    strcat(buff1, "a");
}
my_write(f,buff1, 514);

char* buff2 = malloc(514);
my_read(f, buff2, 514);
printf("%s\n", buff2);
```

Resulting FAT



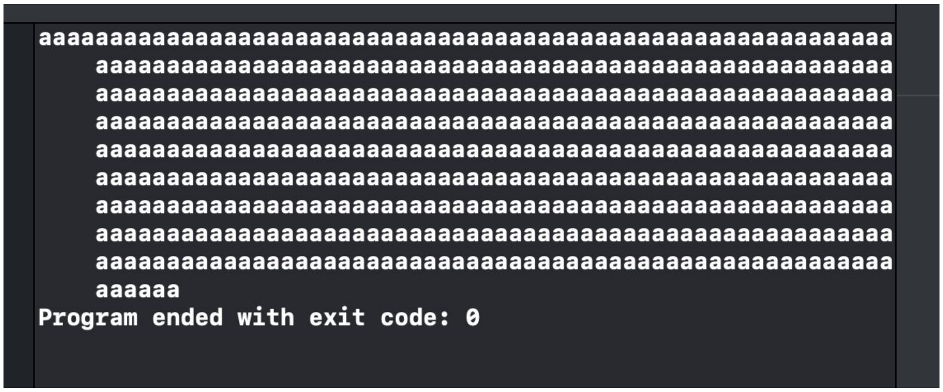
```
0 | 00000000 FFFFFFFF 01000000 03000000 02000000 FFFFFFFF 01000000 FFFFFFFF
```

Entry in Data - Result of Writing - note data blocks aren’t adjacent due to file number 2.



```
24440 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
24492 00000000 00000000 00000000 00000000 00000000 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161                     aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
24544 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
24596 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
24648 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
24700 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
24752 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
24804 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
24856 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
24908 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
24960 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 61616161 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
25012 61616161 61616161 61616100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 aaaaaaaaaaa
25064 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
25116 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
25168 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
25220 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
25272 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
25324 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
25376 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
25428 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
25480 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
25532 00000000 61616100 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000     aaa
```

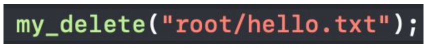
Result of Reading to buff2 and Printing to Screen



```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
    aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
    aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
    aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
    aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
    aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
    aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
    aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
    aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
    aaaaaa
Program ended with exit code: 0
```

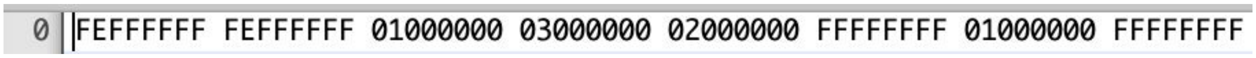
Test Three - Deleting a File

Command



```
my_delete("root/hello.txt");
```

Result in FAT



```
0 FEFFFFFF FEFFFFFF 01000000 03000000 02000000 FFFFFFFF 01000000 FFFFFFFF
```

Part 2: Explanation of Architecture Changes

I made a number of changes in my implementation compared to how I planned it originally. First, I don’t store the open file nodes on the virtual disk because there is no need to make them persistent. The open file nodes are only used to read from files when running the program, so storing them in heap is sufficient.

The second major change is how I handle directories. Due to a time constraint, I had to take a more primitive approach than anticipated. Directories are structs that are stored on the virtual disk and each directory has a name, number of entries, and an array of directory entries. Each directory entry corresponds to a file and stores the file name, file number, and all other metadata. The directories are a seperate section of the virtual disk and take up 6960 bytes. While this approach makes it easy to access the directories, find the file numbers given a file name and edit the metadata, it’s primitive in that it doesn’t allow for directories to store other directories. Though this is a shortcoming, it allows me to finish the assignment and meet the other requirements. I should mention that I know that directories are actually just files that store the information about the files (and other directories) that are found within them, but I wasn’t able to implement this. However, the user can still create and delete directories and create files in directories as required.

Besides these two changes, everything else is as in the previous documentation.

Part 3: Known Issues/Downfalls

While my lab works quite well (especially compared to my original expectations for it), there are a few things I would fix if I had more time:

1. Implement directories as files and allow for nested directories
2. Function to list contents of directory
3. Add more error checking (ex: trying to delete a directory that doesn’t exist)

NOTE*** It might say “abort trap 6” when you run in terminal and honestly I don’t know why that’s happening. I got the code to work without error in Xcode.