Team: Zombies
Salaar Karimzadeh SFSU ID: 920765141
Jasmine Thind SFSU ID: 920767299
Brandon Cruz-Youll SFSU ID: 902899041
Brian Adams SFSU ID: 921039987
CSC415-02

File System Milestone One

https://github.com/CSC415-2022-Spring/csc415-filesystem-BeeSeeWhy

Description:

This assignment required us to partition a volume, initialize a volume control block, initialize a freespace structure and the root directory. After the volume has been initialized, we need to initialize and write the volume control block, free space, and root directory. Then run a hexdump to show your SampleVolume has been initialized correctly, and that the volume control block, free space structure and root directory have successfully been written to the SampleVolume and are located in the intended blocks. Finally, we needed to analyze the hex dump and determine what was written.

Approach/What We Did:

A description of the VCB structure:

1. **VCB Structure Definition:**
    a. VolumeControlBlock structure that holds the variables that will be written into volume. It includes the total number of blocks, the block size, the start of the free space, the start of the root directory, and the magic number. We also included a prototype of a volume control block initializing function called vcbInit() that contains 5 parameters
2. **Implementation:**
    a. VCB structure that will handle the initializations of variables such as; number of total blocks, size of the blocks, the free block count, all block pointers, and the root directory pointers. Through the usage of pointers it will allow for dereferencing and access to different members in the structure.
3. **Initialization fsInit.c**

a. Followed the steps of milestone 1 pdf and malloc a block of memory which we set as BLOCK_SIZE as a volume control block pointer.

Then called the LBAread() function which takes a buffer count of the block and the starting block number 0. It returns the number of blocks read. Next was to check if our pointer to the structure matched the magic number, to do this we used an if statement and if the magic number does not match it will initialize the values of the volume control block, the free space, and the root directory.

A description of the Free Space structure:

4. **FreeSpace Structure Definition:**
    a. In a free space structure we need to be able to keep track of all the free blocks for space allocation to files when they are created. We will need to be able to reuse the space when the files have been deleted. This makes free space management crucial. The free space system maintains a free space list that will keep track of the disk blocks that have not yet been allocated to the directory.
5. **Type of bitmap used:**
    a. In this project the bitmap that is being used is a bit array that will keep track of 0's or 1's, and that will tell the state of the block. If a block is not being used then it will be marked with a 0, if a block is being used then it will be marked with a 1.
6. **Implementation:**
    a. In the Freespace structure we create an array of ints that are defaulted to 0. This array is then written to the Sample Volume, once the VCB has been written. In the freespace structure we keep track of the block size, volume size, the size of the array, the location of that array(which will be initialized at 1) and the free space location
    b. We initialized the bits using malloc by multiplying the array size * the block size

A description of the Directory system:

1. **Directory Entry Structure**

    a. The directory entry structure holds everything that is needed to identify the file that will be described in the entry–including, but not limited to, the file id, name, size of the file and the volume block the file is located.
2. **Directory Structure**
    a. The directory is made up of an array of directory entry objects and the structure includes some file metadata such as the parent block location, the block location of itself , the name of the directory and the magic number.
3. **Initialization of Root Directory**
    a. Memory is allocated for a directory structure object with the name “root”. The “.” and “..” are added to the root directory. All other entries are given a file id of -1 to show they are open entries.

A table of who worked on which components:

Volume Control Block	Jasmine Thind
FreeSpace	Salaar Karimzadeh
Directory system	Brian Adams/Brandon Cruz-Youll
Write Up	The Zombies Team
fsInit	The Zombies Team
Debugging	Brian Adams

How did your team work together, how often you met, how did you meet, how did you divide up the tasks

1. Our team works together well. We have been using teamwork and collaboration skills to ensure each team member’s needs are met.
2. Meetings:
    a. Our team presented a schedule of availability. The meetings were on a Discord server that was created by the team, allowing for voice and text communication.

i. Meeting dates:
    1. 3/28/2022 - from 11:00am to 1:00pm
    2. 3/30/2022 - from 3:00pm to 4:00pm
    3. 3/31/2022 - from 11:00am to 1:00pm
    4. 4/4/2022 - from 11:00am to 1:00pm & 7:00pm to 11:00pm
    5. 4/5/2022 - from 11:00am-12:15pm and 3:00pm-11:00pm
3. We talked amongst ourselves to figure out which of us would like to take certain tasks. We allowed people to pick what they wanted to work on, but also kept close communication between each other, because there were a lot of pieces that worked together.

A discussion of what issues you faced and how your team resolved them:

Issues:

1. We had a hard time conceptualizing how to run the code and check if everything is writing to the hexdump. We knew how to get the output for the hexdump, as it was shown in class, but at first we did not know how to initialize the hexdump with the vcb, directory, and freespace.
2. It was difficult for us to wrap our brains around this project as it was the first time for all of us creating a file system, but I can assume it was the same for every group.
    a. Rewatching the lectures and lecture recordings helped us with understanding what needed to be done. Reading the Milestone requirements multiple times and making a
3. We had issues with scheduling and working issues as everyone was busy with other classes. A lot of the time we intended to work together but people needed to study for midterms or had other projects
    a. We tried our best to work around these projects and study time, trying our best to at least get a couple of hours in to work together as a lot of the functions needed different parts from different files.

4. We expended an extreme amount of time to actually create and implement a fsLow.c for LBAWrite and LBARead, but soon realized that an fsLow.o with every functionality was given to us.
    a. On our part we were really confused at first, but went through the github again and read through each instruction. We then realized that a fsLow.o has each function that needed to be called. But we took an extra amount of time to try and create a partition.
5. It took us a while to understand the changes to be made to the MakeFile. Once we figured out the files were not linking together during the compiling, we remembered the additional objects to add to the MakeFile.

Analysis:

000200-000210: VCB

76,63, and 62 represent the vcb, which is a string that is being initialized to the dump

000230-000250: Root Directory

72 6f 6f 74 is the root dir name

01 represents the directory entry size

54 represents the character T, this indicates whether the root directory is a directory.

F4 53 is the OS which represents the author

000260-0002B0: Root Entry[0] Parent .. (self)

72 6F 6F 74 2- 45 6E 74 72 79 translates to root Entry, which is the name of the Directory entry

01 represents the directory entry size

54 represents T, this indicates that this directory entry is an entry

F4 53 translates to OS, like the directory itself represents the author which in this case is the os

0002C0-0002F0: Root Entry[1] Self

(This is identical to Root Entry[0] as it is the directory entry for self which is the same as the parent)

72 6F 6F 74 2- 45 6E 74 72 79 translates to root Entry, which is the name of the Directory entry

01 represents the directory entry size

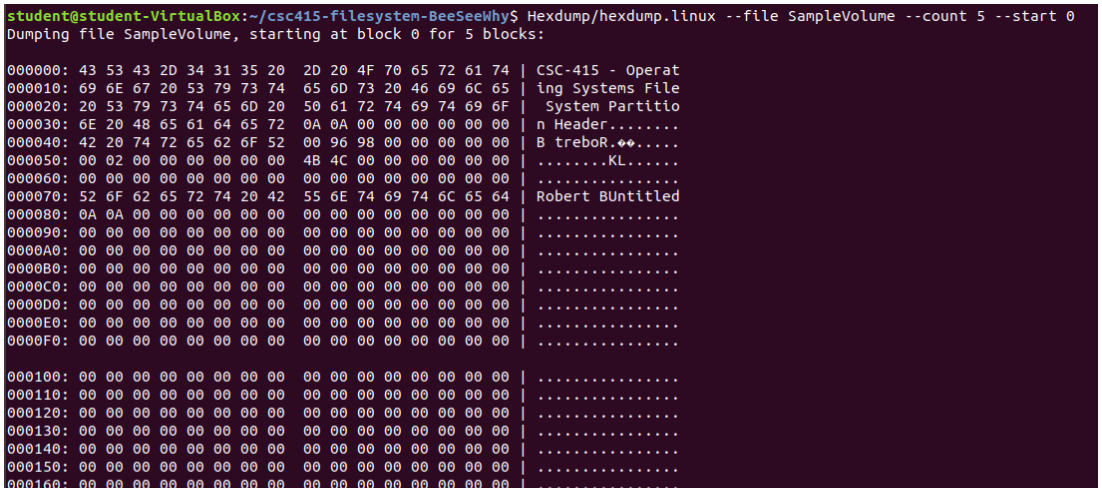
54 represents T, this indicates that this directory entry is an entry

F4 53 translates to OS, like the directory itself represents the author which in this case is the os

000600-0009F8: FreeSpace Array,

values 58,58,58 represent x which are occupied blocks, 4f represent o which means open blocks.

A dump (use the provided HexDump utility) of the volume file that shows the VCB, FreeSpace, and root directory:



```
000200: 00 00 00 00 00 02 00 00  10 00 00 00 00 00 00 00 | ................
000210: 00 00 00 00 76 63 62 00  00 00 00 00 00 00 00 00 | ....vcb.........
000220: 00 00 00 00 00 00 00 00  01 04 00 00 00 00 00 00 | ................
000230: 72 6F 6F 74 00 00 00 00  00 00 00 00 00 00 00 00 | root............
000240: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000250: 01 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00 | ................
000260: 72 6F 6F 74 20 45 6E 74  72 79 00 00 00 00 00 00 | root Entry......
000270: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000280: 01 00 00 00 01 00 00 00  54 00 00 00 00 00 00 00 | ........T.......
000290: 00 00 00 4F 53 00 00 00  00 00 00 00 00 00 00 00 | ...OS...........
0002A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0002B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0002C0: 72 6F 6F 74 20 45 6E 74  72 79 00 00 00 00 00 00 | root Entry......
0002D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0002E0: 01 00 00 00 01 00 00 00  54 00 00 00 00 00 00 00 | ........T.......
0002F0: 00 00 00 4F 53 00 00 00  00 00 00 00 00 00 00 00 | ...OS...........

000300: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000310: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............▯▯▯▯
000320: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000330: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000340: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000350: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000360: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000370: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............▯▯▯▯
000380: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000390: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0003A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0003B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0003C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0003D0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............▯▯▯▯
0003E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0003F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
```

```
000400: 72 6F 6F 74 00 00 00 00  00 00 00 00 00 00 00 00 | root............
000410: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000420: 01 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00 | ................
000430: 72 6F 6F 74 20 45 6E 74  72 79 00 00 00 00 00 00 | root Entry......
000440: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000450: 01 00 00 00 01 00 00 00  54 00 00 00 00 00 00 00 | ........T.......
000460: 00 00 00 4F 53 00 00 00  00 00 00 00 00 00 00 00 | ...OS...........
000470: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000480: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000490: 72 6F 6F 74 20 45 6E 74  72 79 00 00 00 00 00 00 | root Entry......
0004A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0004B0: 01 00 00 00 01 00 00 00  54 00 00 00 00 00 00 00 | ........T.......
0004C0: 00 00 00 4F 53 00 00 00  00 00 00 00 00 00 00 00 | ...OS...........
0004D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0004E0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............▯▯▯▯
0004F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

000500: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000510: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000520: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000530: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000540: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............▯▯▯▯
000550: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000560: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000570: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000580: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000590: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0005A0: 00 00 00 00 00 00 00 00  00 00 00 00 FF FF FF FF | ............▯▯▯▯
0005B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0005C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0005D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0005E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0005F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
```

```
000600: 58 58 58 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | XXXO0000000000000
000610: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000620: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000630: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000640: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000650: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000660: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000670: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000680: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000690: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0006A0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0006B0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0006C0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0006D0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0006E0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0006F0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000

000700: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000710: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000720: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000730: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000740: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000750: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000760: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000770: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000780: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000790: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0007A0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0007B0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0007C0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0007D0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0007E0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0007F0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
```

```
000800: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000810: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000820: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000830: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000840: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000850: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000860: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000870: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000880: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000890: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0008A0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0008B0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0008C0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0008D0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0008E0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0008F0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000

000900: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000910: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000920: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000930: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000940: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000950: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000960: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000970: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000980: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
000990: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0009A0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0009B0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0009C0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0009D0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0009E0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
0009F0: 4F 4F 4F 4F 4F 4F 4F 4F  4F 4F 4F 4F 4F 4F 4F 4F | 0000000000000000
```