Team Name: incomplete

team members

Parampal Singh #922759406

Kendall Powell #921279668

Pankuri Khare #921385202

Ashley Ching #921824056

GitHub:

https://github.com/CSC415-2023-Fall/csc415-filesystem-kpowell2/tree/newbranch

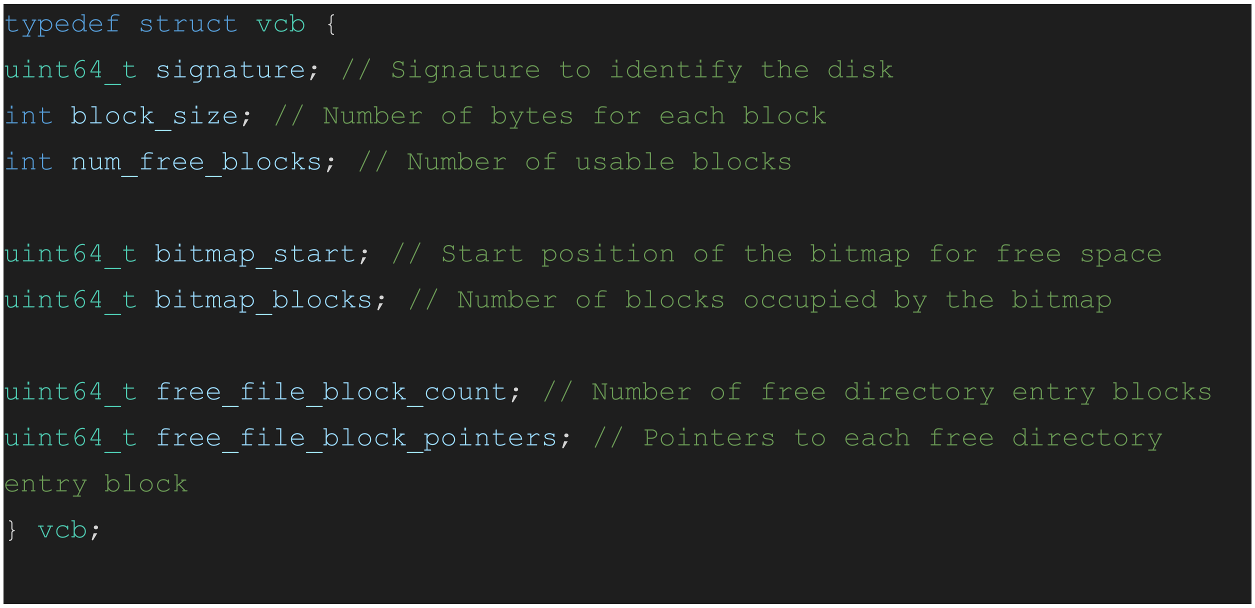
File System Milestone 1

Description: for the first milestone of the file system project we were given the task of formatting the volume. this means that the VCB is written to memory, the memory management system is initialized and finally, the root directory is initialized and written to memory.

Approach / What We Did:

we began with storing the values passed into initFileSystem which were the number of blocks and block size. then we do a check to see if the vcb has already been written to memory by checking it with a function called isVCBSet which checks the signature with the magic number. we then calculate the size needed for the bitmap and then call the bitmapinit function to initialize it. for simplicity, we gave each part of milestone one their read and write functions. we call them when we are ready to write the different parts of milestone 1 into memory. finally, we initialize the root directory with the initRootDirectory function which mallocs enough memory to store the "." and ".." directories. After this, we call initDirectoryEntry which takes a directory entry structure and sets the parameter values. after this, we call LBAwrite to write it to memory and return the value from LBAwrite the fsinit.

VCB structure



```
typedef struct vcb {
uint64_t signature; // Signature to identify the disk
int block_size; // Number of bytes for each block
int num_free_blocks; // Number of usable blocks

uint64_t bitmap_start; // Start position of the bitmap for free space
uint64_t bitmap_blocks; // Number of blocks occupied by the bitmap

uint64_t free_file_block_count; // Number of free directory entry blocks
uint64_t free_file_block_pointers; // Pointers to each free directory
entry block
} vcb;
```

Bit Map structure

In this bitmap, each bit corresponds to the present state of a memory block. when a block of memory is used we set the corresponding bit to the used state and when we free memory we set it to the free state. we wrote a few functions to make using the bitmap easy such as the search function to find the next free bit in the map.

Directory System

the directory system is currently comprised of two functions. the first initializes directory entry structures and the second writes them to memory. the directory entry structure was written as

```
typedef struct DirectoryEntry {
char name[MAX_NAME_LENGTH + 1]; // Name of the file or directory
uint64_t start_block; // Starting block of the file/directory data
uint64_t size; // Size of the file or directory
time_t creation_time; // Time of creation
time_t last_access_time; // Last accessed time
time_t last_modified_time; // Last modified time
uint8_t is_directory; // 1 if directory, 0 if file
} DirectoryEntry;
```

who did what

volume control init	bitmap init	root directory init
Parampal Singh	Parampal Singh	Parampal Singh
	Kendall Powell	
Pankuri Khare		
		Ashley Ching

how we work together

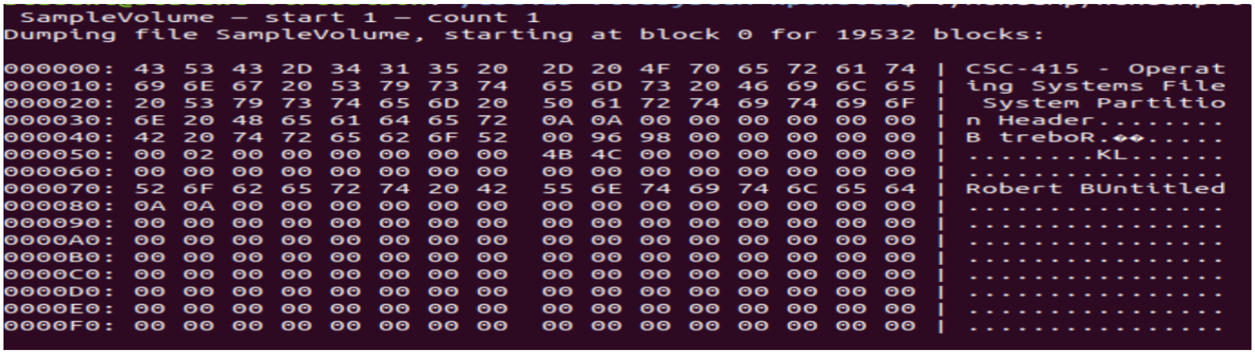
to get this milestone finished we divided the work between the four of us except for Parampal who had the best idea of how to get the work done. we met over Discord several times to discuss who could work on what and find the best times for us to compare ideas and strategies to finish the milestone. we ended up having kendall work on the bitmap, pankuri work on the initialization of the volume control block, and Ashley work on the root directory. Parampal worked on all three with us as he had the most understanding of the project.

issues and resolutions

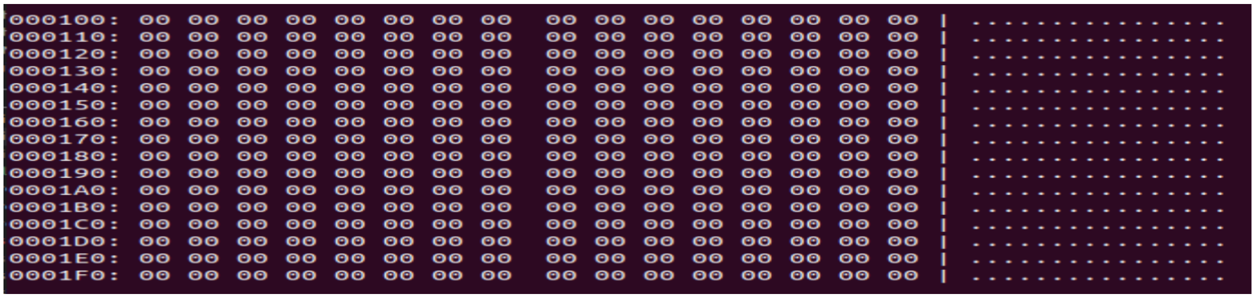
there were two main issues we had. the first issue was time management because many of us are either working or participating in other clubs for the school we found it difficult to find times in which we could all meet together on Discord. we resolved this by creating a schedule to better organize meeting times.

the second and biggest issue was that our hex dump remained mostly empty after we wrote everything to memory we were unable to find a solution to this problem but will continue to work on it to ensure our file system works appropriately. also, we were unable to decipher the hex dump and do not have a clear idea if our code worked or not.

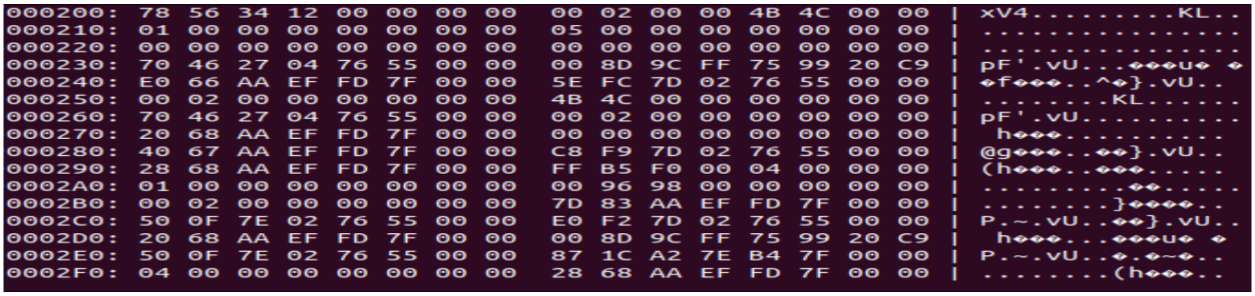
hex dump



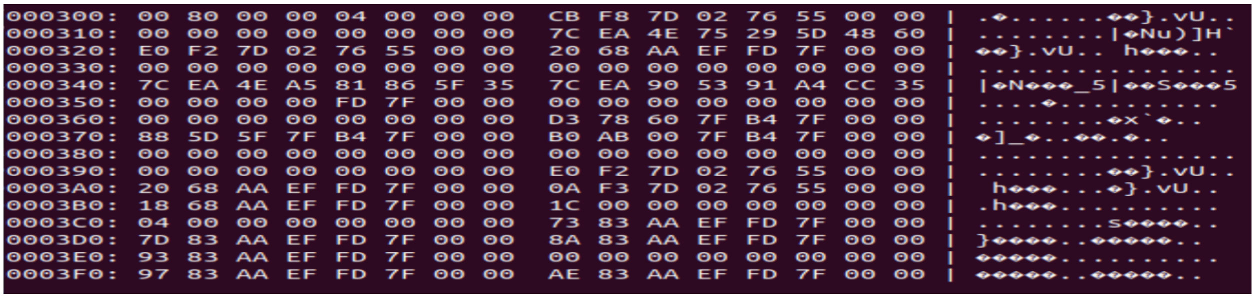
```
 SampleVolume — start 1 — count 1
Dumping file SampleVolume, starting at block 0 for 19532 blocks:

000000: 43 53 43 2D 34 31 35 20  2D 20 4F 70 65 72 61 74 | CSC-415 - Operat
000010: 69 6E 67 20 53 79 73 74  65 6D 73 20 46 69 6C 65 | ing Systems File
000020: 20 53 79 73 74 65 6D 20  50 61 72 74 69 74 69 6F |  System Partitio
000030: 6E 20 48 65 61 64 65 72  0A 0A 00 00 00 00 00 00 | n Header........
000040: 42 20 74 72 65 62 6F 52  00 96 98 00 00 00 00 00 | B treboR.◆◆.....
000050: 00 02 00 00 00 00 00 00  4B 4C 00 00 00 00 00 00 | ........KL......
000060: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000070: 52 6F 62 65 72 74 20 42  55 6E 74 69 74 6C 65 64 | Robert BUntitled
000080: 0A 0A 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000090: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0000A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0000B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0000C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0000D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0000E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0000F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
```



```
000100: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000110: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000120: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000130: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000140: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000150: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000160: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000170: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000180: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000190: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0001A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0001B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0001C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0001D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0001E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0001F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
```



```
000200: 78 56 34 12 00 00 00 00  00 02 00 00 4B 4C 00 00 | xV4.........KL..
000210: 01 00 00 00 00 00 00 00  05 00 00 00 00 00 00 00 | ................
000220: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000230: 70 46 27 04 76 55 00 00  00 8D 9C FF 75 99 20 C9 | pF'.vU...���u� �
000240: E0 66 AA EF FD 7F 00 00  5E FC 7D 02 76 55 00 00 | �f���..^�}.vU..
000250: 00 02 00 00 00 00 00 00  4B 4C 00 00 00 00 00 00 | ..........KL......
000260: 70 46 27 04 76 55 00 00  00 02 00 00 00 00 00 00 | pF'.vU..........
000270: 20 68 AA EF FD 7F 00 00  00 00 00 00 00 00 00 00 |  h���..........
000280: 40 67 AA EF FD 7F 00 00  C8 F9 7D 02 76 55 00 00 | @g���..��}.vU..
000290: 28 68 AA EF FD 7F 00 00  FF B5 F0 00 04 00 00 00 | (h���..���.....
0002A0: 01 00 00 00 00 00 00 00  00 96 98 00 00 00 00 00 | .........��.....
0002B0: 00 02 00 00 00 00 00 00  7D 83 AA EF FD 7F 00 00 | ........}����..
0002C0: 50 0F 7E 02 76 55 00 00  E0 F2 7D 02 76 55 00 00 | P.~.vU..��}.vU..
0002D0: 20 68 AA EF FD 7F 00 00  00 8D 9C FF 75 99 20 C9 |  h���...���u� �
0002E0: 50 0F 7E 02 76 55 00 00  87 1C A2 7E B4 7F 00 00 | P.~.vU..�.�~�..
0002F0: 04 00 00 00 00 00 00 00  28 68 AA EF FD 7F 00 00 | ........(h���..
```



```
000300: 00 80 00 00 04 00 00 00  CB F8 7D 02 76 55 00 00 | .�......��}.vU..
000310: 00 00 00 00 00 00 00 00  7C EA 4E 75 29 5D 48 60 | ........|�Nu)]H`
000320: E0 F2 7D 02 76 55 00 00  20 68 AA EF FD 7F 00 00 | ��}.vU.. h���..
000330: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000340: 7C EA 4E A5 81 86 5F 35  7C EA 90 53 91 A4 CC 35 | |�N���_5|��S���5
000350: 00 00 00 00 FD 7F 00 00  00 00 00 00 00 00 00 00 | ....�..........
000360: 00 00 00 00 00 00 00 00  D3 78 60 7F B4 7F 00 00 | ........�x`�..
000370: 88 5D 5F 7F B4 7F 00 00  B0 AB 00 7F B4 7F 00 00 | �]_�..��.�..
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  E0 F2 7D 02 76 55 00 00 | ........��}.vU..
0003A0: 20 68 AA EF FD 7F 00 00  0A F3 7D 02 76 55 00 00 |  h���...�}.vU..
0003B0: 18 68 AA EF FD 7F 00 00  1C 00 00 00 00 00 00 00 | .h���..........
0003C0: 04 00 00 00 00 00 00 00  73 83 AA EF FD 7F 00 00 | ........s����..
0003D0: 7D 83 AA EF FD 7F 00 00  8A 83 AA EF FD 7F 00 00 | }����..�����..
0003E0: 93 83 AA EF FD 7F 00 00  00 00 00 00 00 00 00 00 | �����..........
0003F0: 97 83 AA EF FD 7F 00 00  AE 83 AA EF FD 7F 00 00 | �����..�����..
```