

CSC415-03
File System Project
Team: Compiler Errors
Github: kylepet

Kyle Petkovic - ID: 923683186
Nicholas Pagcanlungan - ID: 922298361
Matthew Hernandez - ID: 920214682
Neal Olorvida - ID: 915905996

Milestone 1
October 27, 2023

Milestone 1

Hexdump:

The VCB at block 1:

```
kylep@kyles-vm: ~/github/csc415/csc415-filesystem-k... 
kylep@kyles-vm:~/github/csc415/csc415-filesystem-kylepet$ ./Hexdump/hexdump
.linuxM1 SampleVolume --start 1 --count 1
Dumping file SampleVolume, starting at block 1 for 1 block:

000200: 4E 65 61 6C 4B 79 6C 65  4D 61 74 74 4E 69 63 68 | NealKyleMattNich
000210: 6F 6C 61 73 00 00 00 00  4B 4C 00 00 4B 4C 00 00 | olas....KL..KL..
000220: 00 02 00 00 9F 00 00 00  05 00 00 00 99 00 00 00 | ....♦.....♦...
000230: 70 79 11 E6 AA AA 00 00  80 AB 12 E6 AA AA 00 00 | py.様..♦..様..
000240: 9F 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ♦. .....
000250: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | .....
000260: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | .....
000270: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | .....
000280: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | .....
000290: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | .....
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: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | .....
0002D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | .....
0002E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | .....
0002F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | .....

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 00 00 00 00 | .....
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 00 00 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  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 00 00 00 00 | .....
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 | .....

kylep@kyles-vm:~/github/csc415/csc415-filesystem-kylepet$ 
```

The FAT table, starting at block 2:

The FAT table is longer, but the rest of it is unallocated and has no data. The FAT table is from block 2 to block 154.

```
kylep@kyles-vm: ~/github/csc415/csc415-filesystem-k... kylep@kyles-vm: ~/github/csc415/csc415-filesystem-kylepet$ ./Hexdump/hexdump .linuxM1 SampleVolume --start 2 --count 2
Dumping file SampleVolume, starting at block 2 for 2 blocks:

000400: 00 00 00 00 02 00 00 00 03 00 00 00 04 00 00 00 | .....
000410: 05 00 00 00 06 00 00 00 07 00 00 00 08 00 00 00 | .....
000420: 09 00 00 00 0A 00 00 00 0B 00 00 00 0C 00 00 00 | .....
000430: 0D 00 00 00 0E 00 00 00 0F 00 00 00 10 00 00 00 | .....
000440: 11 00 00 00 12 00 00 00 13 00 00 00 14 00 00 00 | .....
000450: 15 00 00 00 16 00 00 00 17 00 00 00 18 00 00 00 | .....
000460: 19 00 00 00 1A 00 00 00 1B 00 00 00 1C 00 00 00 | .....
000470: 1D 00 00 00 1E 00 00 00 1F 00 00 00 20 00 00 00 | .....
000480: 21 00 00 00 22 00 00 00 23 00 00 00 24 00 00 00 | !...#"...$...
000490: 25 00 00 00 26 00 00 00 27 00 00 00 28 00 00 00 | %...&...'...(...).
0004A0: 29 00 00 00 2A 00 00 00 2B 00 00 00 2C 00 00 00 | )...*...+...,...
0004B0: 2D 00 00 00 2E 00 00 00 2F 00 00 00 30 00 00 00 | -...../....0...
0004C0: 31 00 00 00 32 00 00 00 33 00 00 00 34 00 00 00 | 1...2...3...4...
0004D0: 35 00 00 00 36 00 00 00 37 00 00 00 38 00 00 00 | 5...6...7...8...
0004E0: 39 00 00 00 3A 00 00 00 3B 00 00 00 3C 00 00 00 | 9....:...;...<...
0004F0: 3D 00 00 00 3E 00 00 00 3F 00 00 00 40 00 00 00 | =....>...?...@...

000500: 41 00 00 00 42 00 00 00 43 00 00 00 44 00 00 00 | A...B...C...D...
000510: 45 00 00 00 46 00 00 00 47 00 00 00 48 00 00 00 | E...F...G...H...
000520: 49 00 00 00 4A 00 00 00 4B 00 00 00 4C 00 00 00 | I...J...K...L...
000530: 4D 00 00 00 4E 00 00 00 4F 00 00 00 50 00 00 00 | M...N...O...P...
000540: 51 00 00 00 52 00 00 00 53 00 00 00 54 00 00 00 | Q...R...S...T...
000550: 55 00 00 00 56 00 00 00 57 00 00 00 58 00 00 00 | U...V...W...X...
000560: 59 00 00 00 5A 00 00 00 5B 00 00 00 5C 00 00 00 | Y...Z...[...`...
000570: 5D 00 00 00 5E 00 00 00 5F 00 00 00 60 00 00 00 | ]...^..._...`...
000580: 61 00 00 00 62 00 00 00 63 00 00 00 64 00 00 00 | a...b...c...d...
000590: 65 00 00 00 66 00 00 00 67 00 00 00 68 00 00 00 | e...f...g...h...
0005A0: 69 00 00 00 6A 00 00 00 6B 00 00 00 6C 00 00 00 | i...j...k...l...
0005B0: 6D 00 00 00 6E 00 00 00 6F 00 00 00 70 00 00 00 | m...n...o...p...
0005C0: 71 00 00 00 72 00 00 00 73 00 00 00 74 00 00 00 | q...r...s...t...
0005D0: 75 00 00 00 76 00 00 00 77 00 00 00 78 00 00 00 | u...v...w...x...
0005E0: 79 00 00 00 7A 00 00 00 7B 00 00 00 7C 00 00 00 | y...z...{....|...
0005F0: 7D 00 00 00 7E 00 00 00 7F 00 00 00 80 00 00 00 | }...~....@...
```

```
kylep@kyles-vm: ~/github/csc415/csc415-filesystem-k... | Y...Z...[...]\...
000560: 59 00 00 00 5A 00 00 00 5B 00 00 00 5C 00 00 00 | ]...^..._.....
000570: 5D 00 00 00 5E 00 00 00 5F 00 00 00 60 00 00 00 | a...b...c...d...
000580: 61 00 00 00 62 00 00 00 63 00 00 00 64 00 00 00 | e...f...g...h...
000590: 65 00 00 00 66 00 00 00 67 00 00 00 68 00 00 00 | i...j...k...l...
0005A0: 69 00 00 00 6A 00 00 00 6B 00 00 00 6C 00 00 00 | m...n...o...p...
0005B0: 6D 00 00 00 6E 00 00 00 6F 00 00 00 70 00 00 00 | q...r...s...t...
0005C0: 71 00 00 00 72 00 00 00 73 00 00 00 74 00 00 00 | u...v...w...x...
0005D0: 75 00 00 00 76 00 00 00 77 00 00 00 78 00 00 00 | y...z...{...|...
0005E0: 79 00 00 00 7A 00 00 00 7B 00 00 00 7C 00 00 00 | }...~.....
0005F0: 7D 00 00 00 7E 00 00 00 7F 00 00 00 80 00 00 00 | .....*.....
000600: 81 00 00 00 82 00 00 00 83 00 00 00 84 00 00 00 | .....*.....
000610: 85 00 00 00 86 00 00 00 87 00 00 00 88 00 00 00 | .....*.....
000620: 89 00 00 00 8A 00 00 00 8B 00 00 00 8C 00 00 00 | .....*.....
000630: 8D 00 00 00 8E 00 00 00 8F 00 00 00 90 00 00 00 | .....*.....
000640: 91 00 00 00 92 00 00 00 93 00 00 00 94 00 00 00 | .....*.....
000650: 95 00 00 00 96 00 00 00 97 00 00 00 98 00 00 00 | .....*.....
000660: 99 00 00 00 00 00 00 00 9B 00 00 00 9C 00 00 00 | .....*.....
000670: 9D 00 00 00 9E 00 00 00 00 00 00 00 A0 00 00 00 | .....*.....
000680: A1 00 00 00 A2 00 00 00 A3 00 00 00 A4 00 00 00 | .....*.....
000690: A5 00 00 00 A6 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
0006A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
0006B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
0006C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
0006D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
0006E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
0006F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
000700: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
000710: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
000720: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
000730: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
000740: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
000750: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
000760: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
000770: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
000780: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
000790: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
0007A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
0007B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
0007C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
0007D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
0007E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
0007F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....*.....
kylep@kyles-vm: ~/github/csc415/csc415-filesystem-kylepet$
```

The free space bitmap: It is 4 more blocks long, but since the other blocks are unallocated they are all 00's

```
kylep@kyles-vm: ~/github/csc415/csc415-filesystem-k... Q = x

kylep@kyles-vm:~/github/csc415/csc415-filesystem-kylepet$ ./Hexdump/hexdump
.linuxM1 SampleVolume --start 155 --count 1
Dumping file SampleVolume, starting at block 155 for 1 block:

013600: FF | ++++++.....
013610: FF FF FF FF 7F 00 00 00 00 00 00 00 00 00 00 00 | +.....
013620: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013630: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013640: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013650: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013660: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013670: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013680: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013690: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0136A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0136B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0136C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0136D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0136E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0136F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

013700: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013710: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013720: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013730: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013740: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013750: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013760: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013770: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013780: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
013790: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0137A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0137B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0137C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0137D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0137E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0137F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

kylep@kyles-vm:~/github/csc415/csc415-filesystem-kylepet$
```

The entire root directory:

It has 50 entries and is 8 blocks long.

```
kylep@kyles-vm: ~/github/csc415/csc415-filesystem-kylepet$ ./Hexdump/hexdump.  
linuxM1 SampleVolume --start 160 --count 8  
Dumping file SampleVolume, starting at block 160 for 8 blocks:  
  
014000: 2E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
014010: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
014020: 00 00 00 00 00 00 00 00 A0 0F 00 00 9F 00 00 00 | .....?e...  
014030: 01 00 00 00 00 00 00 00 DE OD 3F 65 00 00 00 00 | .....?e...  
014040: DE OD 3F 65 00 00 00 00 DE OD 3F 65 00 00 00 00 | ..?e...?e...  
014050: 2E 2E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
014060: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
014070: 00 00 00 00 00 00 00 00 A0 0F 00 00 9F 00 00 00 | .....?e...  
014080: 01 00 00 00 00 00 00 00 DE OD 3F 65 00 00 00 00 | .....?e...  
014090: DE OD 3F 65 00 00 00 00 DE OD 3F 65 00 00 00 00 | ..?e...?e...  
0140A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0140B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0140C0: 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF | .....?e...  
0140D0: 00 00 00 00 00 00 00 00 DE OD 3F 65 00 00 00 00 | .....?e...  
0140E0: DE OD 3F 65 00 00 00 00 DE OD 3F 65 00 00 00 00 | ..?e...?e...  
0140F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
  
014100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
014110: 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF | .....?e...  
014120: 00 00 00 00 00 00 00 00 DE OD 3F 65 00 00 00 00 | .....?e...  
014130: DE OD 3F 65 00 00 00 00 DE OD 3F 65 00 00 00 00 | ..?e...?e...  
014140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
014150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
014160: 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF | .....?e...  
014170: 00 00 00 00 00 00 00 00 DE OD 3F 65 00 00 00 00 | .....?e...  
014180: DE OD 3F 65 00 00 00 00 DE OD 3F 65 00 00 00 00 | ..?e...?e...  
014190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0141A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0141B0: 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF | .....?e...  
0141C0: 00 00 00 00 00 00 00 00 DE OD 3F 65 00 00 00 00 | .....?e...  
0141D0: DE OD 3F 65 00 00 00 00 DE OD 3F 65 00 00 00 00 | ..?e...?e...  
0141E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0141F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
  
014200: 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF | .....?e...  
014210: 00 00 00 00 00 00 00 00 DE OD 3F 65 00 00 00 00 | ..?e...?e...  
014220: DE OD 3F 65 00 00 00 00 DE OD 3F 65 00 00 00 00 | ..?e...?e...  
014230: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
014240: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
014250: 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF | .....?e...  
014260: 00 00 00 00 00 00 00 00 DE OD 3F 65 00 00 00 00 | .....?e...  
014270: DE OD 3F 65 00 00 00 00 DE OD 3F 65 00 00 00 00 | ..?e...?e...  
014280: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
014290: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0142A0: 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF | .....?e...  
0142B0: 00 00 00 00 00 00 00 00 DE OD 3F 65 00 00 00 00 | .....?e...  
0142C0: DE OD 3F 65 00 00 00 00 DE OD 3F 65 00 00 00 00 | ..?e...?e...  
0142D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0142E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0142F0: 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF | .....?e...
```



```

014F00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
014F10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
014F20: 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF | .....$.....
014F30: 00 00 00 00 00 00 00 00 00 DE 0D 3F 65 00 00 00 00 | .....?e....
014F40: DE 0D 3F 65 00 00 00 00 DE 0D 3F 65 00 00 00 00 | ..?e....?e....
014F50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
014F60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
014F70: 00 00 00 00 00 00 00 00 00 00 00 FF FF FF FF | .....$.....
014F80: 00 00 00 00 00 00 00 00 00 DE 0D 3F 65 00 00 00 00 | .....?e....
014F90: DE 0D 3F 65 00 00 00 00 DE 0D 3F 65 00 00 00 00 | ..?e....?e....
014FA0: 00 00 00 00 00 00 00 00 31 00 00 00 00 00 00 00 | .....1.....
014FB0: 9F 00 00 00 A0 00 00 00 A1 00 00 00 A2 00 00 00 | ..$....$....$...
014FC0: A3 00 00 00 A4 00 00 00 A5 00 00 00 A6 00 00 00 | ..$....$....$...
014FD0: 00 00 00 00 00 00 00 00 81 C6 00 00 00 00 00 00 | .....$.....
014FE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
014FF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

```

Description of VCB:

Our Volume control block contains the signature, number of blocks in the control block, the size of the block, an integer that represents the bitmap length in free space, the length of the FAT, pointers to freespace and the FAT table that are updated at runtime, an integer that represents the next free block, and the integer that points to the root directory.

Description of Freespace Structures:

The free space system was made in freeSpace.c and freeSpace.h files. There is a bitmap that has one bit for each block that exists in the filesystem. It's made up of ints whose bits are manually changed with functions. There is also a FAT table where each index represents its respective block. Each entry in the table points to the next block with a special EOF of 0 meaning it's the end of the data. 0 is chosen since I always know that will be the VCB. A FAT table was used because the size of files can be expanded. These low level functions and saving the new FAT entries and bitmap values have been abstracted into a function that returns n blocks that are free in an array. These blocks aren't necessarily contiguous, so another function exists to write those specific blocks to the volume.

Description of Directory:

Similar to how we structured the freespace functions, we decided to put all the functions related to directories in separate files. initDir can initialize the root directory and other directories. It calls the free space system for free blocks and then the write function in order to save the data to the disk. Directories store the name of the file, the size in bytes, the location in blocks, if it's a dir, and some timestamps for metadata.

Who worked on what components:

	VCB	Freespace	Directory
Matthew	Worked on		Worked on
Kyle	Worked on	Worked on	Worked on
Nicholas	Worked on		
Neal	Worked on		

Teamwork:

For the most part, our team communicated in our team Discord, since it was hard to align our schedules together. For how we divided the work, we essentially looked at the steps, and each assigned a step to each other. Although we did not set scheduled meetings for each other, we frequently checked in with each other in discord, and updated one another when we made changes, or needed help. Overall, despite our troubles scheduling a meeting, we adapted and were able to communicate efficiently so that we could complete our tasks. In addition, to not step over anyone's work, we made separate files for each of us to work on so that there would be no merging errors.

Issues and Resolutions:

1. There were issues with compiling the initDir function. This was due to taking the pseudo code too literally, and not understanding what needed to be properly initialized and malloc'd. This was resolved by reviewing what was already written to fully understand what needed to be initialized.
2. There was some confusion with variable types while writing in fsinit.c. This was cleared up by looking at the variables of the VCB struct, and properly organizing and managing them in the function.
3. There were some issues passing variables into a function that was designed to init the file system if it was not initialized yet(step 2). We resolved this by removing the file and just directly writing it into the fsinit.c file.

Kyle:

1. When using structs in other header files the make run would fail because there would be an infinite loop of importing files. I had to fix this by using #ifndef and #define to ensure that data structures would only be declared one time. I also had to forward declare structs.
2. I was overcomplicating the code by reading in the VCB from the filesystem every time. This added a lot of extra complexity. I resolved this by creating a global variable for the VCB so it can just stay in memory and easily be accessed.

3. I was receiving malloc assertion errors when running the code. This was because I wasn't malloc'ing enough memory and the read was going past the allocated memory. The reason was because I was getting memory based on the exact size of the struct and not taking into account that it would be a bit larger since I was reading it based on blocks.
4. Data wasn't being saved across runs. This was because I forgot to re-write data back to the volume after changing it. I also had issues with my FAT entries not being written correctly since my variable counts were slightly off. I was able to confirm this was working properly by directly reading the volume.
5. Initially the filesystem design didn't have any way to increase the size of a file past its last block. In order to resolve this I implemented a FAT table that keeps track of what blocks data was written at and a pointer to the next block or EOF if it was at the end.
6. The free space system uses a bitmap to quickly see what blocks have been allocated and which blocks haven't; however, the data wasn't changing in the volume. I realized I actually forgot to update it with the allocated blocks after writing the VCB, FAT, and the free space bitmap and committing it to the volume.
7. When running the program after the volume was already initialized the pointers for the FAT table and the free space bitmap weren't working. This was because every time you run the program the pointers change, so the written portions in the VCB are essentially garbage data after the program finishes running. I solved this by reading the two data structures from the volume and then saving them to the VCB in memory if the volume was already initialized.
8. Because a FAT table is being used, that means there is a chance that the data is not contiguous. This isn't really an issue now since nothing can be unallocated, but when I was re-writing the function that writes data to the disk, I kept on getting duplicate data for every block. The issue was that I needed to increment the buffer pointer by the block size in order for the rest of the data to properly be written.