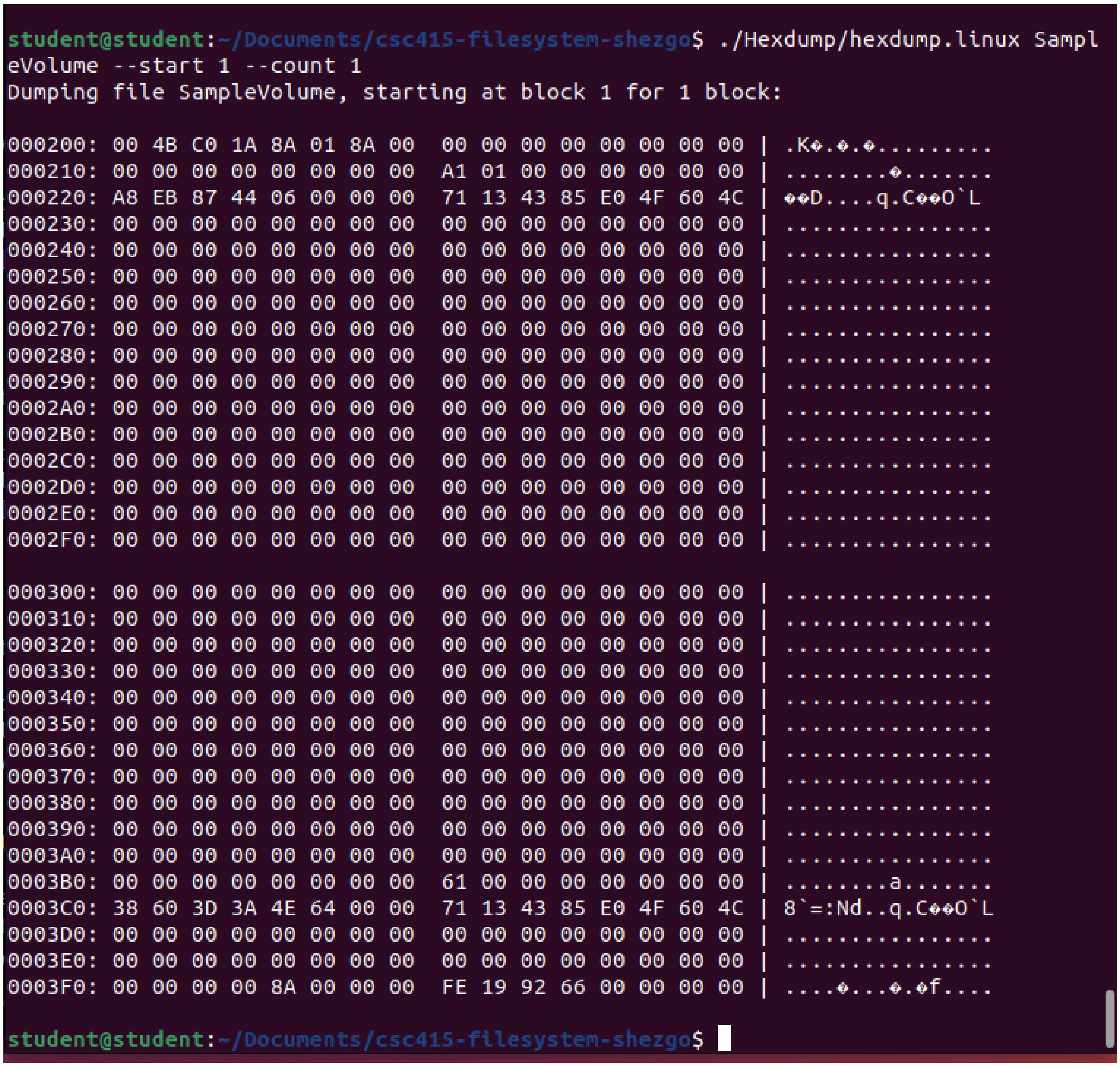
Course: CSC 415 - 01 OPERATING SYSTEM PRINCIPLE
Team name: Spork
Assignment: File System Milestone 1
Team members:
Shez Rahman (ID: 916867424)
Austin Kuykendall (ID: 920222066)
Robel Ayelew (ID: 922419937)
Awet Fikadu (ID: 922130310)
Github: https://github.com/CSC415-2024-Summer/csc415-filesystem-shezgo.git

1. Hex Dumps:

   Volume control block at block 1:



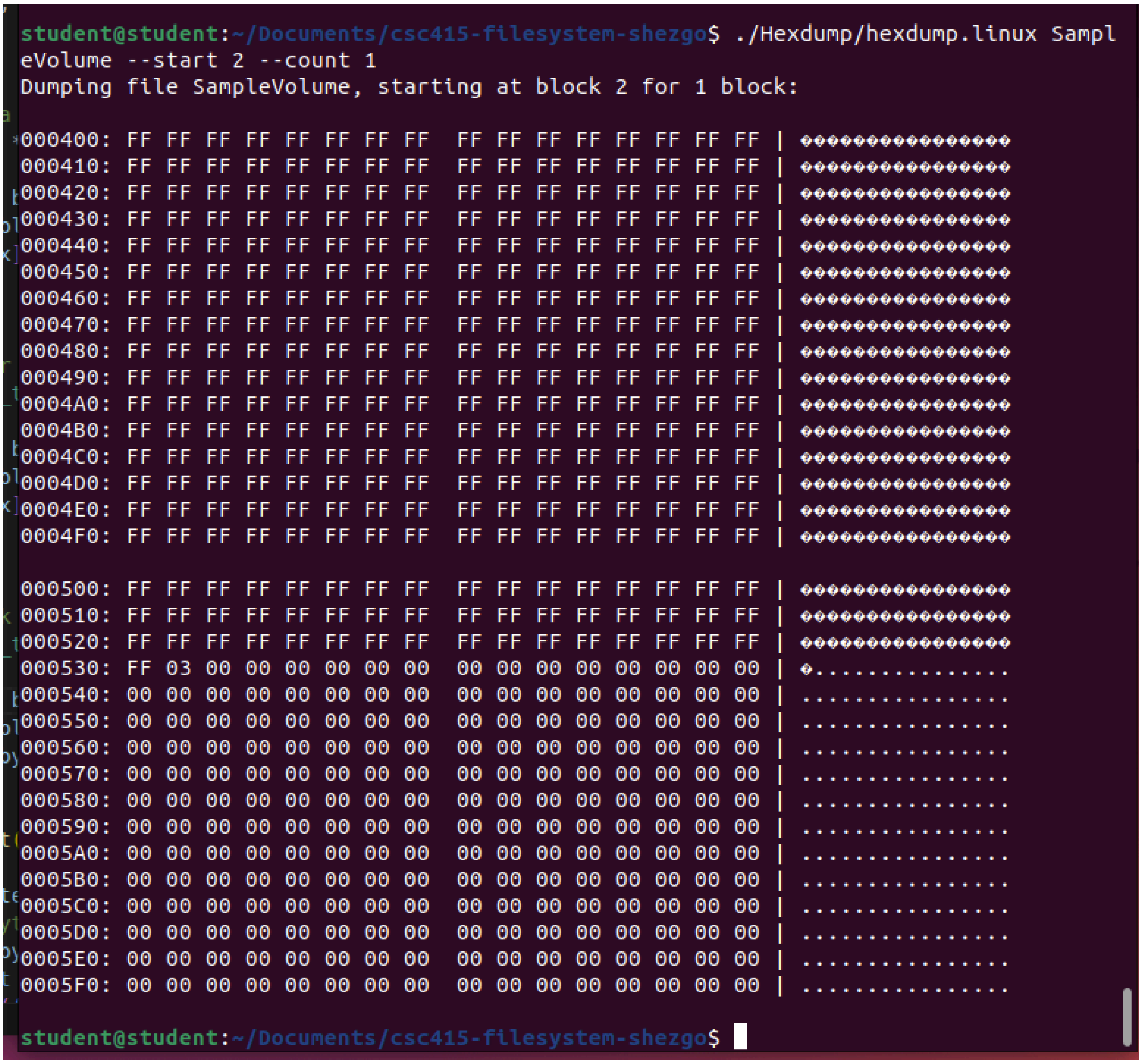
```
student@student:~/Documents/csc415-filesystem-shezgo$ ./Hexdump/hexdump.linux Sampl
eVolume --start 1 --count 1
Dumping file SampleVolume, starting at block 1 for 1 block:

000200: 00 4B C0 1A 8A 01 8A 00  00 00 00 00 00 00 00 00 | .K�.�.�.........
000210: 00 00 00 00 00 00 00 00  A1 01 00 00 00 00 00 00 | ........�.......
000220: A8 EB 87 44 06 00 00 00  71 13 43 85 E0 4F 60 4C | ��D....q.C��O`L
000230: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000240: 00 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  61 00 00 00 00 00 00 00 | ........a.......
0003C0: 38 60 3D 3A 4E 64 00 00  71 13 43 85 E0 4F 60 4C | 8`=:Nd..q.C��O`L
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 8A 00 00 00  FE 19 92 66 00 00 00 00 | ....�...�.�f....

student@student:~/Documents/csc415-filesystem-shezgo$ 
```

   Free space bit map at block 2:



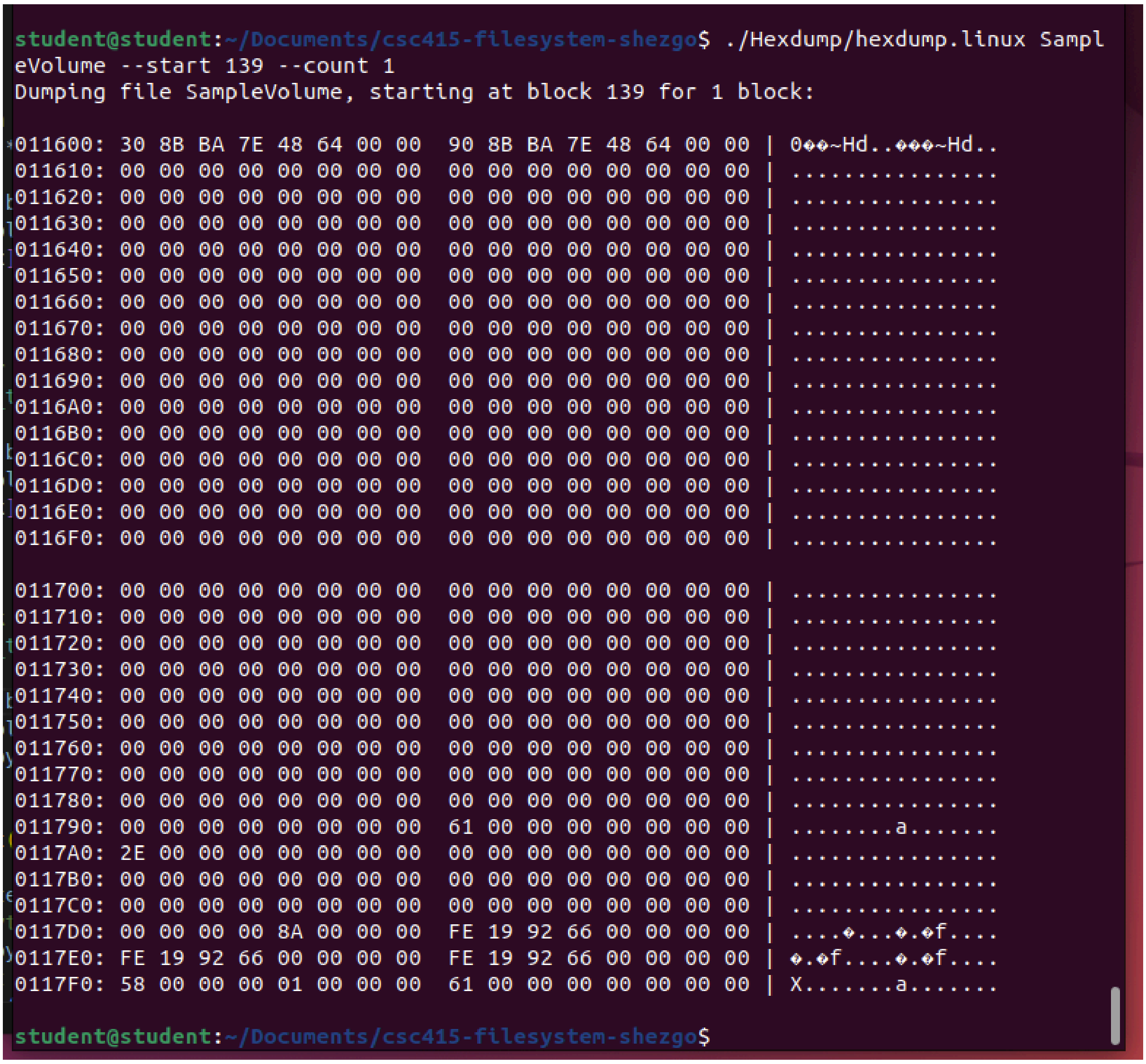
```
student@student:~/Documents/csc415-filesystem-shezgo$ ./Hexdump/hexdump.linux Sampl
eVolume --start 2 --count 1
Dumping file SampleVolume, starting at block 2 for 1 block:

000400: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
000410: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
000420: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
000430: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
000440: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
000450: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
000460: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
000470: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
000480: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
000490: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
0004A0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
0004B0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
0004C0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
0004D0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
0004E0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
0004F0: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������

000500: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
000510: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
000520: FF FF FF FF FF FF FF FF  FF FF FF FF FF FF FF FF | ����������������
000530: FF 03 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 00 00 00 00 | ................
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 00 00 00 00 | ................
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 | ................

student@student:~/Documents/csc415-filesystem-shezgo$ 
```

Root directory at block 139:



```
student@student:~/Documents/csc415-filesystem-shezgo$ ./Hexdump/hexdump.linux Sampl
eVolume --start 139 --count 1
Dumping file SampleVolume, starting at block 139 for 1 block:

011600: 30 8B BA 7E 48 64 00 00  90 8B BA 7E 48 64 00 00 | 0��~Hd..���~Hd..
011610: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011620: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011630: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011640: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011650: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011660: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011670: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011680: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011690: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0116A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0116B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0116C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0116D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0116E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0116F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

011700: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011710: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011720: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011730: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011740: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011750: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011760: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011770: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011780: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
011790: 00 00 00 00 00 00 00 00  61 00 00 00 00 00 00 00 | ........a.......
0117A0: 2E 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0117B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0117C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0117D0: 00 00 00 00 8A 00 00 00  FE 19 92 66 00 00 00 00 | ....�...�.�f....
0117E0: FE 19 92 66 00 00 00 00  FE 19 92 66 00 00 00 00 | �.�f....�.�f....
0117F0: 58 00 00 00 01 00 00 00  61 00 00 00 00 00 00 00 | X.......a.......

student@student:~/Documents/csc415-filesystem-shezgo$
```

2. Volume Control Block

The Volume control block is the backbone of our file system, meaning it manages the file system by tracking the size of each block, the total number of blocks, current free blocks, and so on. The VCB is initialized in fsInit.c which writes the VCB to the disk at physical block 0, which we know is actually logical block 1 because Professor's partition table uses block 0 when displayed with the hex dump.

3. Free Space Structure

As per the file system design, our free space structure to track the free blocks and occupied blocks will be Bitmap. The way that the Bitmap structure works is that each block is represented by a bit, meaning each used block will be set to 1 and other blocks that are free will be set to 0, therefore this approach is more efficient when it comes to space efficiency, simple implementation, and ease of use compared to other structures such as LinkedList, Grouping, Counting, and so on. This bitmap is assisted with helper functions to set and clear bits, check if a bit is being used or free, and return the first free bit because our memory is contiguous.

4. Directory System

We have a Directory struct which contains an array of 50 DirectoryEntry objects, as well as helper functions to initialize the directory, add a directory entry, and find a directory entry using its name as an identifier. We will add more members and functions as needed for the next milestone.

5. Table of Contributions

Team Names	Component
Shez Rahman	Implementation, writeup, research, collaborative coding
Austin Kuykendall	DirectoryEntry Amount, Address Population, research, collaborative coding
Robel Ayelew	Milestone 1 writeup [VCB, Free Space Structure, Issues], research, collaborative coding
Awet Fikadu	Research, collaborative coding

5. Team organization

Project breakdown is assessed whenever the group meets. Every other day we at least host a zoom meeting to talk about shared project resources. Within our team. Making it online with zoom is good for the commuters of our group avoiding traffic.

6. Issues

The issue that we have in this assignment is making or integrating our VCB and Free space management structure in the file system initialization function, meaning we have been able to implement the above components in the fsinit. It was difficult to decipher the difference between a directory and directory entry, and how the two interact with each other. The confusion stemmed from the fact that a directory entry contains a flag saying if it is a directory or not. As a team, we reviewed the lectures and found that a directory contains an array of directory entries, and resolved the issue by creating a separate directory struct with basic operations to interact with its array of directory entries.

Another issue was learning how to do bitwise operations, and making sure the free space map operated as intended. By iteratively doing research and testing calculations with simple output statements, we were able to debug and make sure we were getting expected values (such as the number of bytes to be occupied with the bitmap) and setting dependent values (such as the block containing the root directory) accordingly. Helper functions were set along the way to interact with the free space bitmap.