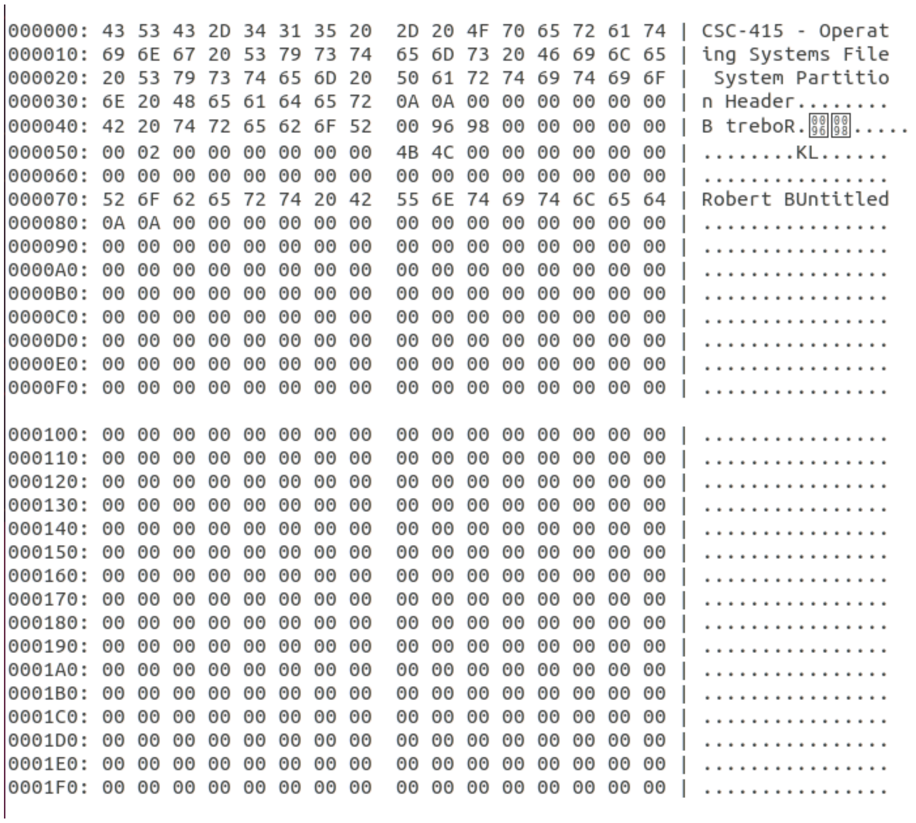
Team EHNA

Evan Giampaoli 922361476, Nathan Rennacker 921348958, Hann Zhao 917565433, Antonio Indindoli 920356733

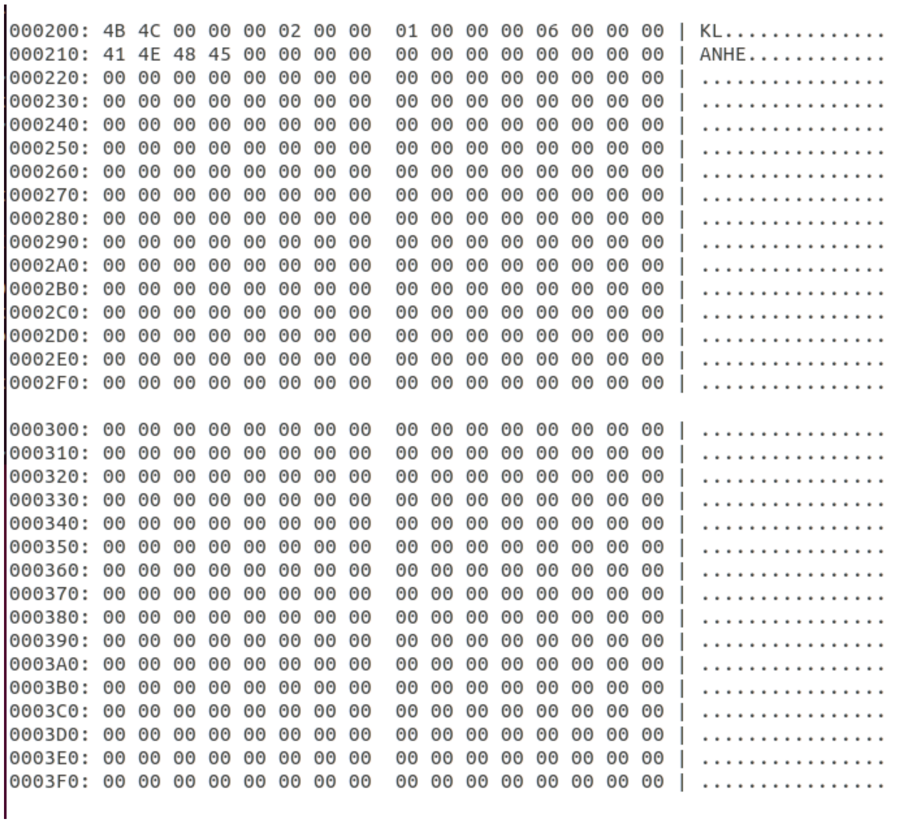
Github Repository: [hazhao33]

Index

Hexdump



```
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: 4B 4C 00 00 00 02 00 00  01 00 00 00 06 00 00 00 | KL..............
000210: 41 4E 48 45 00 00 00 00  00 00 00 00 00 00 00 00 | ANHE............
000220: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
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  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 | ................
```

Our volume control block starts at block 0, which is byte : 00200 to 003FF in the hex dump pictured above.

000200 - 000203 : That is the struct member *numBlocks,* which is 1951 blocks.
If you look at that location then you will see, 4B 4C 00 00 which is read as 00 00 4C 4B. This is the hexadecimal equivalent of 19531.

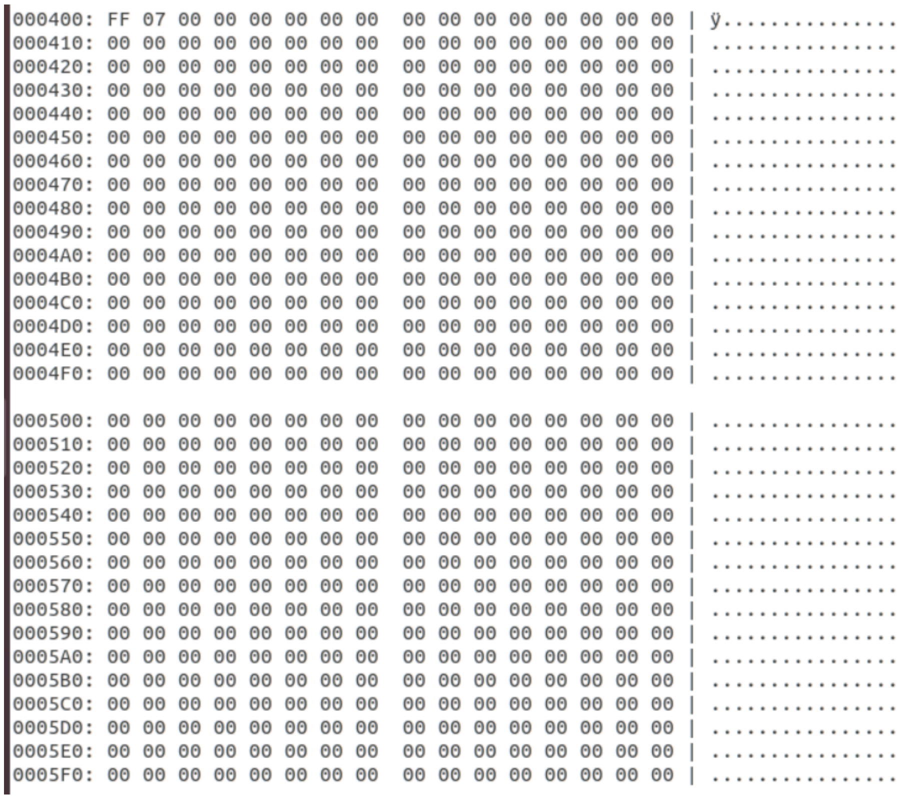
000204 - 000207 : This location shows the VCB struct member *sizeBlocks.*
Which is 512 bytes.
If you look at that location then you will see, 00 02 00 00 which is read as 00 00 02 00. This is the hexadecimal equivalent of 512, which is matches with how big each block size is a matches the info stores in our volume control block.

000208 - 00020B : This location shows the VCB struct member *bitmapLocation*
Our bitmap is located at block position 1.
If you look at that location then you will see, 01 00 00 00 which is read as 00 00 00 01. This is the hexadecimal equivalent of 1, which correctly matches the location of our bitmap.

00020C - 00020F : This location shows the VCB struct member *rootLocation*.
Root location is block 6.
If you look at that location then you will see, 06 00 00 00 which is read as 00 00 00 06. This is the hexadecimal equivalent of 6
To the block number the root is located.

000210 - 000213 : This location shows the VCB struct member *magicNumber*.
Our magic number is 0x45484E41.
If you look at this location then you will see, 41 4E 48 45 which is read as 45 48 4E 41. This correctly matches our file system's magic number.

00214 - 0003FF:
This remaining location is the remaining part of the first block for our volume. Because the first block only contains the volume control block struct which is less than 512 bytes, the remaining part of the block is all 0s.



```
000400: FF 07 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ÿ...............
000410: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000420: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000430: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000440: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000450: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000460: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
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: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0004A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0004B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0004C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
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 00 00 00 00 | ................
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 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 | ................
```

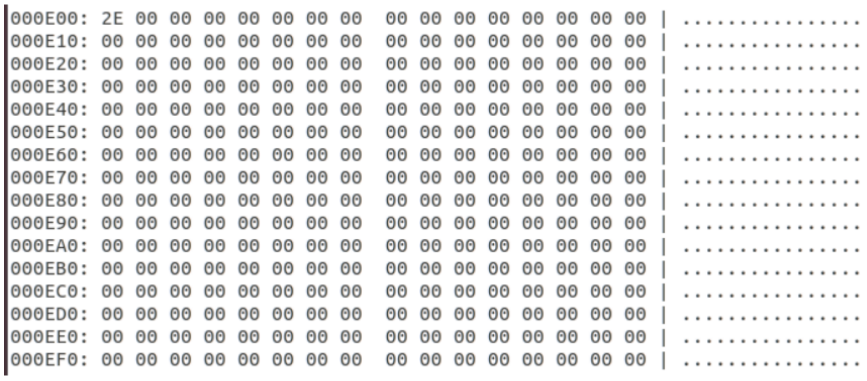
000400-000DF0 : 5 bitMap associated blocks

400 : the first 11 bits are filled for the VCB, the bitMap itself, and the directory System – FF 07 → 0000 0111 1111 1111

↑

Empty* blocks between these posted blocks ***empty for free space managing**

↓



```
000E00: 2E 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E10: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E20: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E30: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E40: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E50: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E60: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E70: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E80: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000E90: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000EA0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000EB0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000EC0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000ED0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000EE0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000EF0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
```

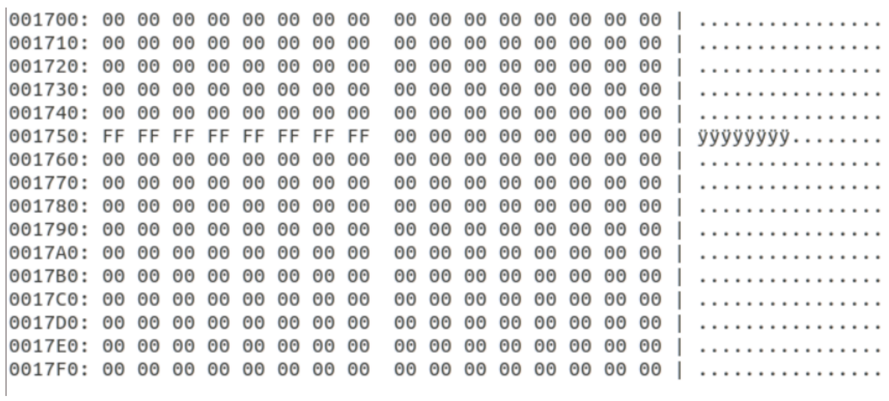
```
000F00: 06 00 00 00 80 09 00 00  01 00 00 00 31 30 2F 32 | ...........10/2
000F10: 37 2F 32 30 32 32 00 00  31 30 2F 32 37 2F 32 30 | 7/2022..10/27/20
000F20: 32 32 00 00 31 30 2F 32  37 2F 32 30 32 32 00 00 | 22..10/27/2022..
000F30: 2E 2E 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000F40: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000F50: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000F60: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000F70: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000F80: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000F90: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000FA0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000FB0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000FC0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000FD0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000FE0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
000FF0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

001000: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001010: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001020: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001030: 06 00 00 00 80 09 00 00  01 00 00 00 31 30 2F 32 | ...........10/2
001040: 37 2F 32 30 32 32 00 00  31 30 2F 32 37 2F 32 30 | 7/2022..10/27/20
001050: 32 32 00 00 31 30 2F 32  37 2F 32 30 32 32 00 00 | 22..10/27/2022..
001060: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001070: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001080: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001090: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0010A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0010B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0010C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0010D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0010E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0010F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
```

000E00 -001050: Root Directory Entry
The first 256 byte is allocate to file Name
The next 12 byte is for the location of the file, the file size , and the flag for tracking if that is a directory.
001030-001050: Store the root directory metadata
Which include the create date, modified date, and access date.

```
001100: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001110: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001120: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001130: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001140: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001150: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001160: FF FF FF FF FF FF FF FF  00 00 00 00 00 00 00 00 | ÿÿÿÿÿÿÿÿ........
001170: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001180: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001190: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0011A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0011B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0011C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0011D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0011E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0011F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

001200: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001210: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001220: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001230: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001240: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001250: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001260: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001270: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001280: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001290: FF FF FF FF FF FF FF FF  00 00 00 00 00 00 00 00 | ÿÿÿÿÿÿÿÿ........
0012A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0012B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0012C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0012D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0012E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0012F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
```

```
001300: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001310: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001320: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001330: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001340: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001350: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001360: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001370: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001380: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001390: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0013A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0013B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0013C0: FF FF FF FF FF FF FF FF  00 00 00 00 00 00 00 00 | ÿÿÿÿÿÿÿÿ........
0013D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0013E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0013F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

001400: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001410: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001420: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001430: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001440: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001450: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001460: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001470: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001480: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001490: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0014A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0014B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0014C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0014D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0014E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0014F0: FF FF FF FF FF FF FF FF  00 00 00 00 00 00 00 00 | ÿÿÿÿÿÿÿÿ........
```

```
001500: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001510: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001520: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001530: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001540: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001550: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001560: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001570: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001580: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001590: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0015A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0015B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0015C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0015D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0015E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0015F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

001600: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001610: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001620: FF FF FF FF FF FF FF FF  00 00 00 00 00 00 00 00 | ÿÿÿÿÿÿÿÿ........
001630: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001640: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001650: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001660: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001670: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001680: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001690: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0016A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0016B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0016C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0016D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0016E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0016F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
```


```
001700: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001710: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001720: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001730: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001740: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001750: FF FF FF FF FF FF FF FF  00 00 00 00 00 00 00 00 | ÿÿÿÿÿÿÿÿ........
001760: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001770: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001780: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
001790: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0017A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0017B0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0017C0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0017D0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0017E0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0017F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
```

001750: empty directory entry
The location and the file size has been set to -1 to indicate it is an empty directory entry

VCB Structure

```
struct VCB {
        int numBlocks;         // number of blocks in our volume
        int sizeBlocks;        // size of each block
        int bitmapLocation;  // block index location of our bitmap
        int rootLocation;      // block index location of our root directory
        long magicNumber;  // unique signature of our filesystem
};
```

Our Volume Control Block Structure is located at block 0 of our LBA. It contains all of the necessary information about our volume. This includes the number of blocks in our volume, the size of each block, the location of our free space map in the LBA, the location of our Root Directory in the LBA and the magic number for our file system.

For our volume size, we use the limit of 10MB. It contains 19531 blocks with each block being 512 bytes. Our file system's unique magic number is 0x45484E41, which is the hexadecimal representation of our team name. The location of our free space map starts at block index 1 while our root directory starts at block index 6. Therefore,

numBlocks = 19531

sizeBlocks = 512

bitmapLocation = 1

rootLocation = 6

magicNumber = 0x45484E41

Free Space Structure

We decided on a bitmap for overseeing and controlling our available free space. This map is sized according to the maximum block size for the project which is 19,531 blocks, which corresponds to 19,531 bits or 2,442 (2,441.375) bytes. We decided to use an unsigned integer array for mapping. Each unsigned integer is 4 bytes so that means we need a total array of size of 611 to map each of the bits.

Total Size Needed	19,531 blocks	19,531 bits
Size in Bytes	2,442 bytes	
Unsigned Integer Size	4 bytes	32 bits
Array Size	611 Integers	19,552 bits

Notice that there are some extra bits that aren't being used which could be problematic if those were accessible. To fix this, a maximum block count of 19,531 was added in bitmap.c that ensures that the total block count isn't being exceeded–and we aren't trying to write to space that doesn't exist.

For finding free space a combination of an iterative and recursive solution were used. First the bitMap is iterated through until a free space is found, then a recursive function is called to check if that free space is of the required length. If it is, the bit position is returned, otherwise the next free space is iterated to.

(There are a multitude of ways to make finding free space more efficient, and that is something we will probably revisit later)

Directory System

```
Typedef struct directoryEntry {
Char name [256];        //file name
Int location;           //starting block location
Int fileSize;           //file size in byte
Int isDirectory;                //directory flag if directory = 1 file = 0
Char creationDate [12];         //meta data for date creation date tracking
Char modificationDate [12];             //modification date tracking
Char accessDate[12];                    //access date tracking
}
```

Our root directory starting block is at block 6. And each directory will have a total of 8 entries. Currently the root directory contains the root directory Entry and 6 initiated empty directory entry.Those empty directory contain location = -1 and fileSize = -1 to indicate that it is an empty directory entry. The whole root directory will take up 2432 bytes, which is about 5 blocks. Total wasted byte is 128.

Work Allocation Table

Evan Giampaoli	Steps 1 and 2 of the Milestone provided by Professor Bierman. Determining if the volume needs to be formatted then initializing the volume if it needs to be File : fsInit.c complete formatVolume() function and initFileSystem() function

Nathan Rennacker	Steps 3 of the Milestone provided by Professor Bierman. Freespace Management and initialization Files : bitMap.c, bitMap.h
Hann Zhao	Steps 4 of the Milestone provided by Professor Bierman. Initialize the root directory Files : directory.c, directory.h
Antonio Indindoli	Complete dump of our volume file. Research extents and milestone 2

Milestone One Reflections

This project seemed pretty daunting at first, and probably for good reason. Within our first couple meetings we split up the work simply by going through the potential workload of each step in the "*Steps for milestone 1.pdf"* and assigning those accordingly (this is discussed a little more in **Problem #5** of the next section). And we all got to work!

The time allocated for this portion of the project flew by and thankfully we had planned our time well enough that we weren't scrambling near the deadline attempting tasks that potentially become impossible under a time crunch. The hardest challenge for our team so far was ensuring that each persons' piece of the project meshed well with one another/as a whole *and* worked without implementing bad programming practices. We believe we mostly accomplished this, and beyond that worked cohesively as a team in general.

We try to meet **twice a week** through a discord call–and thankfully we all seem to have pretty flexible schedules so changing a meeting time or quickly planning one hasn't been a massive issue. The project is going well so far, and we've all learned a lot throughout the course of milestone one, from planning to actual coding.

Problems and Solutions

Problem #1

The issue faced with initializing the volume was deciding on how to best organize the code.

Solution

The solution was a combination of two steps. The first being multiple additional read throughs of the steps provided by Professor Bierman, the milestone instructions as a whole as well as the provided code. The second step in the solution was just meeting as a team to have further discussions about how we wanted to organize the code for this section.

Problem #2

We experienced a few issues understanding how an extents system would be implemented.

Solution

We eventually decided that since a working extents system is not needed to meet the requirements of this milestone, that we would work on implementing it later on. We will need to make small adjustments to our code to accommodate extents but ultimately trying to implement an extents system at this point in the project would overcomplicate things and create confusion.

Problem #3

There was an issue with initializing the root directory and how we can combine it with normal directory initialization.

Solution

The solution is to go over the class recording and using the parameter to help decide what kind of directory needed to create.

Problem #4

There was an issue near the end of completing the milestone where the hexdump wasn’t updating correctly with the bitMap information.

Solution

The SampleVolume file had to be remade with the correct bitMap information written to each block. That then had to be copied over to the hexdump file for it to display each block properly.

Problem #5

Equally dividing up the tasks so everyone was contributing equally turned out to be quite the conundrum. We felt that the majority of the work for this milestone was perfectly sized for three people rather than four.

Solution

Admittedly, we are not too sure if we can say that we cleanly resolved the issue but definitely tried our best to do so and hopefully can learn from our mistakes working through this milestone and allocate tasks better in the next. Our working solution involved having three people working on the main coding portion of the milestone while the fourth person managed the hexdump and planned ahead studying more in depth about extents and milestone 2.