

# File System

## Milestone One

Team Name	<b>Alibaba</b>	
Team Github	<a href="https://github.com/CSC415-2023-Fall/csc415-filesystem-siid14">https://github.com/CSC415-2023-Fall/csc415-filesystem-siid14</a>	
Team Member	Sidney Thomas	918656419
	Hoang-Anh Tran	922617784
	Ruxue Jin	923092817
	Yee-Tsing Yang	922359864

**compile 1**(When volume signature does not match our signature):

```
student@student-VirtualBox:~/Desktop/FS/csc415-filesystem-testing$ make
gcc -c -o fsshell.o fsshell.c -g -I.
gcc -c -o fsInit.o fsInit.c -g -I.
gcc -c -o fsDir.o fsDir.c -g -I.
gcc -c -o fsFree.o fsFree.c -g -I.
gcc -o fsshell fsshell.o fsInit.o fsDir.o fsFree.o fsLow.o -g -I. -lm -l readline -l pthread
student@student-VirtualBox:~/Desktop/FS/csc415-filesystem-testing$ make run
./fsshell SampleVolume 10000000 512
File SampleVolume does not exist, errno = 2
File SampleVolume not good to go, errno = 2
Block size is : 512
Created a volume with 9999872 bytes, broken into 19531 blocks of 512 bytes.
Opened SampleVolume, Volume Size: 9999872; BlockSize: 512; Return 0
Initializing File System with 19531 blocks with a block size of 512

Signature not found, start formatting

|-----|
|----- Command -----| - Status - |
| ls                  | OFF |
| cd                  | OFF |
| md                  | OFF |
| pwd                 | OFF |
| touch               | OFF |
| cat                 | OFF |
| rm                  | OFF |
| cp                  | OFF |
| mv                  | OFF |
| cp2fs               | OFF |
| cp2l                | OFF |
|-----|

Prompt > exit
System exiting
student@student-VirtualBox:~/Desktop/FS/csc415-filesystem-testing$
```

**compile 2**(When the signature is matches our signature):

```
student@student-VirtualBox:~/CSC415/GroupProject$ make
gcc -c -o fsshell.o fsshell.c -g -I.
gcc -c -o fsInit.o fsInit.c -g -I.
gcc -c -o fsDir.o fsDir.c -g -I.
gcc -c -o fsFree.o fsFree.c -g -I.
gcc -o fsshell fsshell.o fsInit.o fsDir.o fsFree.o fsLow.o -g -I. -lm -l readline -l pthread
student@student-VirtualBox:~/CSC415/GroupProject$ make run
./fsshell SampleVolume 10000000 512
File SampleVolume does exist, errno = 0
File SampleVolume good to go, errno = 0
Opened SampleVolume, Volume Size: 9999872; BlockSize: 512; Return 0
Initializing File System with 19531 blocks with a block size of 512

Signature found, reloading free space

-----|
|----- Command -----| - Status -|
| ls                      |      OFF |
| cd                      |      OFF |
| md                      |      OFF |
| pwd                    |      OFF |
| touch                  |      OFF |
| cat                   |      OFF |
| rm                    |      OFF |
| cp                    |      OFF |
| mv                   |      OFF |
| cp2fs                 |      OFF |
| cp2l                  |      OFF |
|-----|
Prompt > █
```

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

HexDump block 1 (block 0) for VCB:

```
student@student-VirtualBox:~/CSC415/GroupProject$ ./Hexdump/hexdump.linux SampleVolume --start 1 --count 1
Dumping file SampleVolume, starting at block 1 for 1 block:

000200: 4B 4C 00 00 00 02 00 00 D2 04 00 00 00 00 00 00 | KL.....♦.....
000210: 01 00 00 00 06 00 00 00 00 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: 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 | .....

student@student-VirtualBox:~/CSC415/GroupProject$
```

```

000200: 4B 4C 00 00 00 02 00 00 D2 04 00 00 00 00 00 00 | KL.....
000210: 01 00 00 00 06 00 00 00 00 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: 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 | .....

```

**Data from the VCB structure is stored in little Endian format (so byte swapping is needed to read values)**

From **000200 to 000203**: 4 bytes is the unsigned integer for the numberOfBlocks.

Value: 0x0000 4C4B. Converting the hexadecimal value to decimal, we get a decimal value of 19531 that matches the desired number of blocks in our 10MB volume.

From **000204 to 000207**: 4 bytes is the unsigned integer for the blockSize.

Value: 0x0000 0200. Converting the hexadecimal value to decimal, we get a decimal value of 512 that matches the desired size of each block in our volume.

From **000208 to 00020F**: 8 bytes is the unsigned long for the signature.

Value: 0x0000 0000 0000 04D2. Converting the hexadecimal value to decimal, we get a decimal value of 1234 that matches our volume's signature.

From **000210 to 000213**: 4 bytes is the unsigned integer for the bitMapLocation.

Value: 0x0000 0001. Converting the hexadecimal value to decimal we get a decimal value of 1 that matches the starting block number of the bitmap.

From **000214 to 000217**: 4 bytes is the unsigned integer for the rootDirLocation.

Value: 0x0000 0006. Converting the hexadecimal value to decimal we get a decimal value of 6 that matches the starting block number of the root directory.

HexDump block 2 to block 6 (block 1 - 5) for freespace:

```
student@student-VirtualBox:~/CSC415/GroupProject$ ./Hexdump/hexdump.linux SampleVolume --start 2 --count 1
Dumping file SampleVolume, starting at block 2 for 1 block:
```

```
000400: FF FF FF FF E0 00 00 00 00 00 00 00 00 00 00 00 | ****.....
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 | .....
```

```
student@student-VirtualBox:~/CSC415/GroupProject$
```

```

student@student-VirtualBox:~/CSC415/GroupProject$ ./Hexdump/hexdump.linux SampleVolume --start 3 --count 1
Dumping file SampleVolume, starting at block 3 for 1 block:

000600: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000610: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000620: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000630: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000640: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000650: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000660: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000670: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000680: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000690: 00 00 00 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 | .....
0006B0: 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 | .....
0006D0: 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 | .....
0006F0: 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 | .....
000710: 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 | .....
000730: 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 | .....
000750: 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 | .....
000770: 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 | .....
000790: 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 | .....
0007B0: 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 | .....
0007D0: 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 | .....
0007F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

student@student-VirtualBox:~/CSC415/GroupProject$

```



```

student@student-VirtualBox:~/CSC415/GroupProject$ ./Hexdump/hexdump.linux SampleVolume --start 4 --count 1
Dumping file SampleVolume, starting at block 4 for 1 block:

000800: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000810: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000820: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000830: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000840: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000850: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000860: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000870: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000880: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000890: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0008A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0008B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0008C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0008D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0008E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0008F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

000900: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000910: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000920: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000930: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000940: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000950: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000960: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000970: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000980: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000990: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0009A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0009B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0009C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0009D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0009E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0009F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

student@student-VirtualBox:~/CSC415/GroupProject$ ./Hexdump/hexdump.linux SampleVolume --start 5 --count 1

```



```

student@student-VirtualBox:~/CSC415/GroupProject$ ./Hexdump/hexdump.linux SampleVolume --start 5 --count 1
Dumping file SampleVolume, starting at block 5 for 1 block:

000A00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000A10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000A20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000A30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000A40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000A50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000A60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000A70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000A80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000A90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000AA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000AB0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000AC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000AD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000AE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000AF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

000B00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000B10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000B20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000B30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000B40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000B50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000B60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000B70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000B80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000B90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000BA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000BB0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000BC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000BD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000BE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000BF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

student@student-VirtualBox:~/CSC415/GroupProject$

```

```

student@student-VirtualBox:~/CSC415/GroupProject$ ./Hexdump/hexdump.linux SampleVolume --start 6 --count 1
Dumping file SampleVolume, starting at block 6 for 1 block:

000C00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000C10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000C20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000C30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000C40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000C50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000C60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000C70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000C80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000C90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000CA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000CB0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000CC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000CD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000CE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000CF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

000D00: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000D10: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000D20: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000D30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000D40: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000D50: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000D60: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000D70: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000D80: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000D90: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000DA0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000DB0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000DC0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000DD0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000DE0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
000DF0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

3 student@student-VirtualBox:~/CSC415/GroupProject$

```

000400: FF FF FF FF E0 00 00 00 00 00 00 00 00 00 00 00 00 | ◆◆◆◆◆.....

Address: 000400: 1 byte, 8 bits. Value: 0xFF. Converting the hexadecimal value to binary we get a 1111 1111, matching the block 0 to block 7 have been used.

Address: 000401: 1 byte, 8 bits. Value: 0xFF. Converting the hexadecimal value to binary we get a 1111 1111, matching the block 8 to block 15 have been used.

Address: 000402: 1 byte, 8 bits. Value: 0xFF. Converting the hexadecimal value to binary we get a 1111 1111, matching the block 16 to block 23 have been used.

Address: 000403: 1 byte, 8 bits. Value: 0xFF. Converting the hexadecimal value to binary we get a 1111 1111, matching the block 24 to block 31 have been used.

Address: 000404: 1 byte, 8 bits. Value: 0xE0. Converting the hexadecimal value to binary we get a 1110 0000, matching the block 32 to block 34 have been used.

In Total, 35 blocks (from block 0 to block 34) have been used. This matches our design: 1 (volume control block) + 5 (bitmap) + 29 (root directory) = 35.

HexDump block 7 to block 35 (6 - 34) for root directory:

## Block 7

```
student@student-VirtualBox:~/CSC415/FileSystem$ ./Hexdump/hexdump.linux SampleVolume --start 7 --count 1
Dumping file SampleVolume, starting at block 7 for 1 block:

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: D0 39 00 00 00 00 00 00 06 00 00 00 01 00 00 00 | ♦9.....
000F10: E9 36 3C 65 00 00 00 00 E9 36 3C 65 00 00 00 00 | ♦6<e....♦6<e....
000F20: E9 36 3C 65 00 00 00 00 2E 2E 00 00 00 00 00 00 | ♦6<e.....
000F30: 00 00 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 | .....

student@student-VirtualBox:~/CSC415/FileSystem$
```

```

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: D0 39 00 00 00 00 00 00 06 00 00 00 01 00 00 00 | ♦9.....
000F10: E9 36 3C 65 00 00 00 00 E9 36 3C 65 00 00 00 00 | ♦6<e....♦6<e....
000F20: E9 36 3C 65 00 00 00 00 | 2E 2E 00 00 00 00 00 00 | ♦6<e.....

```

**From 000E00 to 000F27: 296 bytes of the first directory entry “.” information. “.” represents the current directory, which is root directory after initialization for milestone 1**

**The data is in little Endian format, byte swapping is needed to read value**

From 000E00 to 000EFF: 256 bytes is the char array reserved for the fileName(max 256 bytes, including NULL).

Value at the first byte: 0x2E. Converting the hexadecimal value to decimal we get a decimal value of 46. 46 represents the character ‘.’ in the ASCII table. So it matches the name of the first directory entry ‘.’

```
| 46 2E 056 &#46; .
```

From **000F00 to 000F07**: 8 bytes long is the actual size of the root directory.

Value: 0x00000000 000039D0. Converting the hexadecimal value to decimal we get a decimal value of 14800 matching the size of the root directory in bytes.

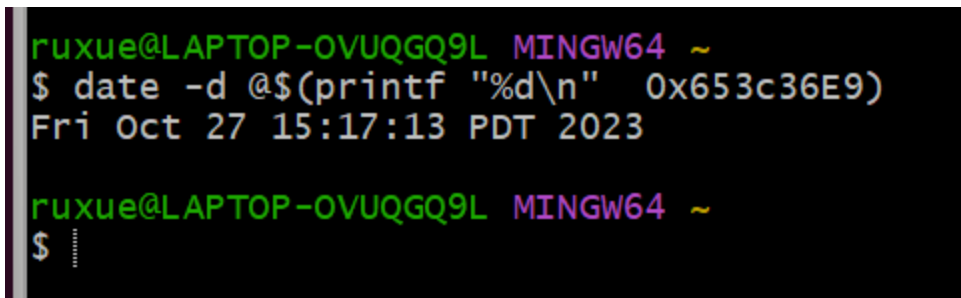
From **000F08 to 000F0B**: 4 bytes integer is the location of the root directory.

Value: 0x0000 0006. Converting the hexadecimal value to decimal we get a decimal value of 6 matching the number of the first block of the root directory.

From **000F0C to 000F0F**: 4 bytes integer indicates whether it is a directory or not. Value: 0x0000 0001. Converting the hexadecimal value to decimal we get a decimal value of 1 indicating it is a directory.

From **000F10 to 000F17**: 8 bytes time\_t shows the timeCreated.

Value: 0x0000 0000 653C 36E9, matching the creation time of this root directory.



```

ruxue@LAPTOP-OVUQGQ9L MINGW64 ~
$ date -d @$(printf "%d\n" 0x653c36E9)
Fri Oct 27 15:17:13 PDT 2023

ruxue@LAPTOP-OVUQGQ9L MINGW64 ~
$

```

From **000F18 to 000F1F**: 8 bytes time\_t shows the timeLastModified.

Value: 0x0000 0000 653C 36E9, is the same as timeCreated, since it is modified during initialization.

From **000F20 to 000F27**: 8 bytes time\_t shows the timeLastAccessed.

Value: 0x0000 0000 653C 36E9, is the same as timeCreated, since it is accessed during initialization



## Block 8

```

student@student-VirtualBox:~/CSC415/FileSystem$ ./Hexdump/hexdump.linux SampleVolume --start 8 --count 1
Dumping file SampleVolume, starting at block 8 for 1 block:

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 D0 39 00 00 00 00 00 00 | .....9.....
001030: 06 00 00 00 01 00 00 00 E9 36 3C 65 00 00 00 00 | .....6<e....
001040: E9 36 3C 65 00 00 00 00 E9 36 3C 65 00 00 00 00 | 6<e....6<e....
001050: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
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 | .....

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: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
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 | .....

student@student-VirtualBox:~/CSC415/FileSystem$

```



```

000F20:  E9 36 3C 65 00 00 00 00 | 2E 2E 00 00 00 00 00 00 | | 6<e.....
000F30:  00 00 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 | D0 39 00 00 00 00 00 00 | | .....9.....
001030:  06 00 00 00 | 01 00 00 00 | E9 36 3C 65 00 00 00 00 | | .....6<e....
001040:  E9 36 3C 65 00 00 00 00 | E9 36 3C 65 00 00 00 00 | | 6<e....6<e....

```

**From 000F28 to 00104F: 296 bytes of the second directory entry “..” information. “..” represents the parent directory. The parent of the root directory is itself. The data is in little Endian format, and byte swapping is needed to read values.**

From 000F28 to 001027: 256 bytes are the char array reserved for the fileName(max 256 bytes, including NULL).

From 000F28 to 000F29: 2 bytes char is the name for the second directory entry.

Value: 0x2E2E. Single hexadecimal value 2E represents the character ‘.’ in the ASCII table.

Combination of two 2E is “..”, so it matches the name of the second directory entry “..”

From 001028 to 00102F: 8 bytes long is the actual size of the root directory.

Value: 0x00000000 000039D0. Converting the hexadecimal value to decimal we get a decimal value of 14800 matching the size of the root directory in bytes.

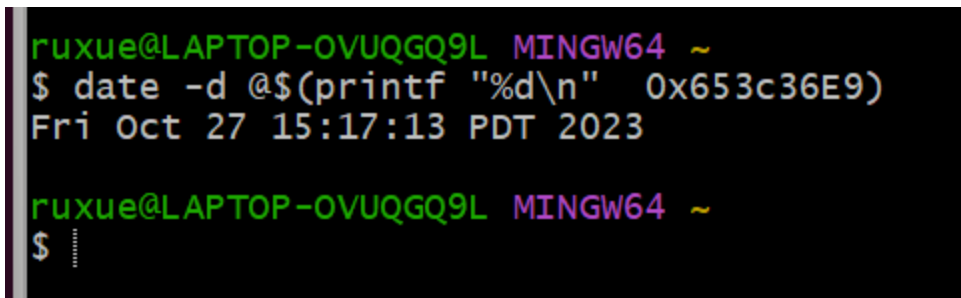
From 001030 to 001033: 4 bytes integer is the location of the root directory.

Value: 0x0000 0006. Converting the hexadecimal value to decimal we get a decimal value of 6 matching the number of the first block of the root directory.

From 001034 to 001037: 4 bytes integer indicates whether it is a directory or not. Value: 0x0000 0001. Converting the hexadecimal value to decimal we get a decimal value of 1 indicating it is a directory.

From 001038 to 00103F: 8 bytes time\_t shows the timeCreated.

Value: 0x0000 0000 653C 36E9, matching the creation time of this root directory.



```
ruxue@LAPTOP-OVUQGQ9L MINGW64 ~
$ date -d @$(printf "%d\n" 0x653c36E9)
Fri Oct 27 15:17:13 PDT 2023

ruxue@LAPTOP-OVUQGQ9L MINGW64 ~
$
```

From 001040 to 001047: 8 bytes time\_t shows the timeLastModified.

Value: 0x0000 0000 653C 36E9, is the same as timeCreated, since it is modified during initialization.

From 001048 to 00104F: 8 bytes time\_t shows the timeLastAccessed.

Value: 0x0000 0000 653C 36E9, is the same as timeCreated, since it is accessed during initialization.

## Block 9 - 34

The rest of the hexdump of the root directory from block 9 to block 34 are all 0s, since the rest of 48 entries are initialized with name “\0” and 0 for other values.

```
student@student-VirtualBox:~/Desktop/FS/csc415-filesystem-testing$ ./Hexdump/hexdump.linux SampleVolume --start 9 --count 26
Dumping file SampleVolume, starting at block 9 for 26 blocks:
```

```
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: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
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: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
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 | .....
```





[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

```
004500: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
004510: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
004520: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
004530: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
004540: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
004550: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
004560: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
004570: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
004580: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
004590: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0045A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0045B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0045C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0045D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0045E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
0045F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....  
  
student@student-VirtualBox:~/Desktop/FS/csc415-filesystem-testing$
```

>> End of Block 9 - 34



## Block 35

The hexdump of block 35, the last block of root directory, has a strange value.

```
student@student-VirtualBox:~/Desktop/FS/csc415-filesystem-testing$ ./Hexdump/hexdump.linux SampleVolume --start 35 --count 1
Dumping file SampleVolume, starting at block 35 for 1 block:

004600: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004610: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004620: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004630: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004640: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004650: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004660: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004670: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004680: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004690: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0046A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0046B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0046C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0046D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0046E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0046F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

004700: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004710: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004720: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004730: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004740: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004750: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004760: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004770: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004780: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
004790: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0047A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0047B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0047C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0047D0: 00 00 00 00 00 00 00 00 00 F1 C2 01 00 00 00 00 | .....
0047E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0047F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....

student@student-VirtualBox:~/Desktop/FS/csc415-filesystem-testing$
```

```
0047D0: 00 00 00 00 00 00 00 00 00 00 F1 C2 01 00 00 00 00 00 | .....
0047E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
0047F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | .....
```

From 0047D0 to 0047FF: 48 bytes are the bytes wasted

From 0047D8 to 0047DA: Value: 0x0001 C2F1; We assume it is a signature to mark the start of wasted bytes, however, it should start at 0047D0.

Our root directory has 50 directory entries. So, it needs  $50 \times 296 = 14800$  bytes. We malloc 29 blocks, which result 14848 bytes. 48 bytes would be wasted.

## 2. A description of the VCB structure

```

52  ✓  typedef struct VCB
53      {
54          unsigned int numberOfBlocks; // Number of blocks in the volume
55          unsigned int blockSize;      // Size of each block in bytes
56
57          unsigned long signature; // Signature for VCB struct
58
59          unsigned int bitMapLocation; // starting block num of bitMap
60          unsigned int rootDirLocation; // starting block num of the root directory
61      } VCB;

```

Our Volume Control Block structure includes the following attributes:

a. numberOfBlocks (unsigned int)

The total number of blocks available within the storage volume to indicate the size or capacity of the volume in terms of these blocks.

b. blockSize (unsigned int)

The size of each individual block within the volume (in bytes) to define the granularity at which data is stored, read, and written.

c. signature (unsigned long)

A unique identifier or code that helps verify the compatibility of the VCB structure to ensure that the VCB is correctly recognized and used by the associated file system.

d. bitMapLocation (unsigned int)

The starting block number within the volume where the bitmap is stored to keep a record of which blocks are in use and which are available for data storage.

e. rootDirLocation (unsigned int)

The starting block number of the root directory within the volume like the entry point for navigating the file system's hierarchical structure.

### 3. A description of the Free Space structure

We use the bitmap char array to track the free space.

Each block has a unique number represented by a single bit in the bit map.

Let  $n$  = the block number

- value of bit  $n == 0$ ; block  $n$  is free
- value of bit  $n == 1$ ; block  $n$  is used

Example: Blocks 3 and 4 are free

Block 0	Block 1	Block 2	Block 3	Block 4	Block 5	Block 6	Block 7
1	1	1	0	0	1	1	1

We malloc bitmap 5 blocks, which are 2560 bytes, but only 2442 bytes are needed to track 19531 bits (blocks) and 118 bytes are wasted. To avoid allocating invalid blocks, we use the number of the last block (block 19530) to check the block num is valid or not.

Our functions for managing free space:

- `initFreeSpace(int blockCount, int bytesPerBlock)`
  - Initializes the Free Space structure.
  - Allocates memory for the bitMap, setting all blocks as initially free.
  - Marks blocks used by the VCB and bitmap as "used."
  - Writes the bitmap to the disk for persistence.
- `loadFreeSpace(int blockCount, int bytesPerBlock)`
  - Loads the existing Free Space structure from the disk.
  - Reads the bitmap into memory, allowing the file system to use the previously initialized Free Space structure.
- `setBitUsed(unsigned int blockNum)`
  - To set a specific block's bit to "used," indicating allocation.
- `setBitFree(unsigned int blockNum)`
  - To mark a block's bits as "free" for allocation.
- `isBitUsed(unsigned int blockNum)`
  - To check if a block is marked as "used" or "free" in the bitmap.

- f. `getFreeBlockNum()`
  - Finds the first free block for allocation.
  - Returns the next available free block number according to the bitmap.
  - Returns -1 if all blocks are in use.
- g. `allocBlocksCont(int blocksNeeded)`
  - Allocates a sequence of contiguous blocks for storage.
  - Marks and updates the bitmap if sufficient contiguous free blocks are available.
  - Returns -1 on failure or insufficient free blocks.

#### 4. A description of the Directory system

```
#define MAX_FILENAME_LEN 255 // maximum filename length

// Directory Entry structure
typedef struct DE
{
    char fileName[MAX_FILENAME_LEN + 1]; // file name cstring, +1 for the NULL
    unsigned long size; // size of the directory/file in bytes
    unsigned int location; // starting block number of the directory/file
    unsigned int isDir; // flag indicating if this entry is a directory (1) or
a file (0)

    time_t timeCreated; // time when the file created
    time_t timeLastModified; // time when the file last modified
    time_t timeLastAccessed; // time when the file last accessed
} DE;
```

Each directory entry structure is 296 bytes. Specifically, max fileName is 256 bytes, size is 8 bytes, location and isDir are 4 bytes, timeCreated, timeLastModification, and timeLastAccessed are 8 bytes respectively.

Our Directory Entry structure includes:

- a. fileName:

A character array (C-string) holding the name of the file or directory. It can be up to MAX\_FILENAME\_LEN characters in length, plus one extra character for the NULL terminator.

- b. size: An unsigned long integer representing the size of the file or directory (in bytes).
- c. location: An unsigned integer specifying the starting block number where the file or directory data is stored within the volume.
- d. isDir: An unsigned integer flag that indicates whether this entry represents a directory (1) or a file (0).
- e. timeCreated: A timestamp representing when the file or directory was initially created.
- f. timeLastModified: A timestamp indicating when the file or directory was last modified.
- g. timeLastAccessed: A timestamp recording the most recent access time to the file or directory.

### 5. A table of who worked on which components

NO	Topic	Status	Task	Member
1	HexDump	<input checked="" type="checkbox"/>	HexDump analysis	Ruxue, Yee-Tsing
2	VCB	<input checked="" type="checkbox"/>	DE and VCB struct	Ruxue
		<input checked="" type="checkbox"/>	mfs.h	Hoang-Anh
3	Free Space	<input checked="" type="checkbox"/>	Implement some functions (loadFreeSpace()) etc...	Sidney
		<input checked="" type="checkbox"/>	Bitmap, block allocation	Yee-Tsing
4	Root Directory	<input checked="" type="checkbox"/>	Help debug the initDir function	Hoang-Anh
		<input checked="" type="checkbox"/>	Implement initDir function	Ruxue
5	Document	<input checked="" type="checkbox"/>	The writeup	Whole team
6	GitHub	<input checked="" type="checkbox"/>	GitHub repo management	Sidney

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

We have a Zoom meeting at least twice a week to keep track of everyone's processes and updates. Besides, we communicate regularly via Discord to help debug and support each other on each component.

Concerning our GITHUB workflow, we decided to work on a testing repository in order to keep the submission repository clean with only working files on it. The testing repo will have work that can

eventually - break the code - test the code - implement any code. Once we reach a satisfactory implementation on the testing repo, only one person will be in charge to merge these files to the working repository.

NO	Time & Date	Details
1	Thu 10/12 at 8:00 PM on Discord	Discussing the directory entry struct
2	Sat 10/14 at 8:00 PM on Discord	Discussing the file system design
3	Fri 10/20 at 2:10 PM on Zoom	<ul style="list-style-type: none"> <li>- Go through the milestone 1 steps together</li> <li>- Divide up the task</li> <li>- Build up the repo and set up git usage rules</li> </ul>
4	Mon 10/23 at 3:30 PM on Zoom	<ul style="list-style-type: none"> <li>- Report the process of each tasks</li> <li>- Debug</li> </ul>
5	Wed 10/25 at 2:00 PM on Zoom	<ul style="list-style-type: none"> <li>- Debug initDir and check the output</li> </ul>
6	Thu 10/26 on Discord	Finished the program and fixed the bug
7	Fri 10/27 at 2:00 PM on Zoom (canceled)	Try to write and finish the document

### **7. A discussion of what issues you faced and how your team resolved them.**

After implementing most of what the M1 required us to implement, we tried to compile the project but we ran into some compilation issues that had to do with the headers and affecting different files as a consequence. To fix it, we had to fix the header file at first and the following files were affected.

Here are some small snippet of the fix we did to have the code to compile as intended

Fix for M1

Headers issue

followed Hoang's approach

```
#include "mfs.h" in      .h  header file
#include "fsLow.h" in    .c  source file
```



All created header files (fsDir.h, fsFree.h) include header guards

Example: fsDir.h

```
#ifndef _FS_DIR_H
#define _FS_DIR_H
// code here
#endif
```

**\*\*fsFree.h\*\***

share bitmap as **global** by using **extern**

**\*\*fsFree.c\*\***

BUG: store bitmap's info (blocks, bytes, bits) in VCB

// VCB is freed in initFileSystem()

FIX: store bitmap's info in fsFree.c

// Alternative: share VCB as global, free in exitFileSystem()

**\*\*fsDir.c\*\***

BUG: `int actualDirEntries = blocksNeeded / sizeof(DE);` // result 0

FIX: `int actualDirEntries = bytesMalloc / sizeof(DE);` // get actual entries

**\*\*fsInit.c\*\***

BUG: `if(strcmp(vcb->signature, "1234") != 0)` // signature is not a string in our VCB

FIX: `if (vcb->signature != SIGNATURE)`

BUG: `strcpy(vcb->signature, SIGNATURE);` // signature is not a string in our VCB

FIX: `vcb->signature = SIGNATURE;`