CSC 415 - 01 OPERATING SYSTEM PRINCIPLES Summer 2024

File System Project

Milestone One

“Team of Three”

Team Member	Student ID
Siarhei Pushkin	
Philip Karnatsevich	
Yahya Obeid	

GitHub Link:

https://github.com/CSC415-2024-Summer/csc415-filesystem-yahyaobeid

Table of Contents

#1. Screenshots of the compilation and HexDump of the volume

Screenshot of compilation:

```
student@student:~/Desktop/HW/csc415-filesystem-yahyaobeid$ make
gcc -c -o fsshell.o fsshell.c -g -I.
gcc -c -o fsInit.o fsInit.c -g -I.
gcc -o fsshell fsshell.o fsInit.o fsLowM1.o -g -I. -lm -l readline -l pthread
```

Screenshot of the execution:

```
student@student:~/Desktop/HW/csc415-filesystem-yahyaobeid$ 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
File system initialized successfully.
|---------------------------------|
|------- 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:~/Desktop/HW/csc415-filesystem-yahyaobeid$ ./Hexdump/hexdump.linuxM1 SampleVolume --start 1 --count 1
```

HexDump(1):

```
student@student:~/Desktop/HW/csc415-filesystem-yahyaobeid$ ./Hexdump/hexdump.linuxM1 SampleVolume --start 1 --count 1
Dumping file SampleVolume, starting at block 1 for 1 block:

000200: 42 20 74 62 52 00 00 00  00 02 00 00 4B 4C 00 00 | B tbR.......KL..
000210: 45 4C 00 00 06 00 00 00  01 00 00 00 E4 B8 00 00 | EL..........◆..
000220: 00 78 54 DB BC 13 CE F6  10 E4 14 C5 FF FF 00 00 | .xT¿.◆◆.◆.◆◆◆..
000230: 48 1B F2 55 E4 B8 00 00  00 02 00 00 00 00 00 00 | H.◆U◆..........
000240: 4B 4C 00 00 00 00 00 00  67 2E 07 CA 01 00 00 00 | KL......g..◆....
000250: B0 B6 EB 7D E4 B8 00 00  70 E4 14 C5 FF FF 00 00 | ◆◆◆}◆..p◆.◆◆◆..
000260: FC 73 46 F2 F9 EE 00 00  E8 E5 14 C5 FF FF 00 00 | ◆sF◆◆◆..◆◆.◆◆◆..
000270: CC 41 72 F2 04 00 00 00  E8 E5 14 C5 00 00 00 00 | ◆Ar◆....◆◆.◆....
000280: 00 96 98 00 00 00 00 00  00 02 00 00 00 00 00 00 | .◆◆.............
000290: DF F3 14 C5 FF FF 00 00  70 E4 14 C5 FF FF 00 00 | ◆◆.◆◆◆..p◆.◆◆◆..
0002A0: C0 73 46 F2 F9 EE 00 00  E8 E5 14 C5 FF FF 00 00 | ◆sF◆◆◆..◆◆.◆◆◆..
0002B0: 00 78 54 DB BC 13 CE F6  80 E5 14 C5 FF FF 00 00 | .xT¿.◆◆◆◆.◆◆◆..
0002C0: CC 74 46 F2 F9 EE 00 00  74 67 72 F2 F9 EE 00 00 | ◆tF◆◆◆..tgr◆◆◆..
0002D0: 54 1A F2 55 E4 B8 00 00  10 E6 14 C5 04 00 00 00 | T.◆U◆...◆.◆....
0002E0: E8 E5 14 C5 FF FF 00 00  E8 E5 14 C5 FF FF 00 00 | ◆◆.◆◆◆..◆◆.◆◆◆..
0002F0: 04 00 00 00 00 00 00 00  78 3C F3 55 E4 B8 00 00 | ........x<◆U◆..

000300: 40 E0 74 F2 F9 EE 00 00  54 1A F2 55 E4 B8 00 00 | @◆t◆◆◆..T.◆U◆..
000310: 00 A0 5D F2 F9 EE 00 00  00 00 00 00 00 00 00 00 | .◆]◆◆◆.........
000320: 10 E6 14 C5 FF FF 00 00  78 3C F3 55 E4 B8 00 00 | .◆.◆◆◆..x<◆U◆..
000330: 00 00 00 00 00 00 00 00  70 E4 14 C5 FF FF 00 00 | ........p◆.◆◆◆..
000340: 1C F9 72 77 09 ED 6A C7  00 00 00 00 00 2E 2F 66 | .◆rw.◆j◆....../f
000350: AC 6E 20 40 0F FC 6A C7  00 00 00 00 00 00 00 00 | ◆n @.◆j◆........
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 C0 00 00 00 | ............◆...
0003A0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................
0003B0: 00 A0 5D F2 F9 EE 00 00  80 B0 5D F2 F9 EE 00 00 | .◆]◆◆◆..◆◆]◆◆◆..
0003C0: 00 78 54 DB BC 13 CE F6  00 00 00 00 00 00 00 00 | .xT¿.◆◆........
0003D0: 70 13 F2 55 E4 B8 00 00  00 00 00 00 00 00 00 00 | p.◆U◆..........
0003E0: 00 00 00 00 00 00 00 00  40 13 F2 55 E4 B8 00 00 | ........@.◆U◆..
0003F0: 00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 | ................

student@student:~/Desktop/HW/csc415-filesystem-yahyaobeid$
```

HexDump(2):

#2. A description of the VCB structure

The VolumeControlBlock (VCB) is an important part of our file system that keeps track of information about the file system's setup and status. It contains various fields such as:

uint64_t signature: A unique identifier is used to verify the validity of the VolumeControlBlock, checking if the block is formatted and contains a valid VolumeControlBlock.

uint32_t blockSizeBytes: Each block represents a certain amount of storage space in the file system, measured in bytes.

uint32_t totalBlockCount: The total number of blocks in the file system shows how much data it can hold.

uint32_t availableBlocks: The number of free blocks available for use helps keep track of how much free space is left in the file system.

uint32_t rootDirectoryBlock: The block number informs where to find the file system's main directory, which is important for getting to the beginning of the directory structure.

uint32_t freeSpaceMapBlock: The block number of the free space map. This map helps us keep track of which storage blocks are being used and which are available, which is important for managing storage space effectively.

#3. A description of the Free Space structure

The file system uses a bitmap in the free space map to manage its free space structure, which is spread across five blocks. Initially, all blocks are set as free by setting their bits to zero. The first six blocks, reserved for system use, including the Volume Control Block and the free space map, have their respective bits set to mark them as

used. This bitmap is written to disk starting at block one, allowing for efficient tracking and managing free and allocated blocks within the file system.

#4. A description of the Directory system

The directory system in our file system helps organize files and directories using a structure called directoryEntry. This structure stores information like file names, timestamps, and attributes. The system includes functions for creating(fs_mkdir) and removing directories (fs_rmdir), opening(fs_opendir), reading(fs_readdir), and closing(fs_closedir), and managing the current working directory (fs_getcwd) (fs_setcwd). It also has tools for checking the types of files (fs_isFile) (fs_isDir) and deleting files(fs_delete). Additionally, the fs_stat structure provides detailed file information, making it easier to manage directories and files within the file system.

#5. A table of who worked on which components

Table of tasks:

Task Description	Assigned To
Volume Control Block	Yahya Obeid
Free Space management	Siarhei Pushkin
The Root Directory	Philip Karnatsevich
Documentation preparation	Philip Karnatsevich, Siarhei Pushkin, Yahya Obeid

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

We utilized a combination of asynchronous chats and live meetings on Discord to address essential aspects of our project in real time. In the beginning, we had meetings to plan our tasks and decide on important features. As we made progress, we had group meetings at the end of each phase to check our work and make sure everyone was on the same page. By using a mix of asynchronous communication and live sessions, we solved problems, shared ideas, and finished our project on time.

Our meetings usually lasted about 30 minutes so we could discuss things in detail without overwhelming our team. We could also organize more meetings if we needed to fix urgent problems or come up with new ideas. By following this plan, we created a team where everyone felt helped and able to work together to reach our goals.

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

Our team faced difficulties implementing complex file system functions in C. We worked through these challenges by having team meetings and continuously learning, and we also used Discord for support. We managed scheduling issues by staying flexible and rescheduling meetings when needed. We split tasks based on our individual strengths and communicated openly to address any teamwork issues. We successfully

handled the project's challenges by creating a supportive environment and solving problems as they occurred.