

# Report-File System

---

Author: Shao Jiaqi

StudentID: 119010256

## Introduction

### Environment

I use the environment provided by CSC4005.

```
Operating System: CentOS Linux 7 (Core)
```

```
$ g++ --version
g++ (GCC) 4.8.5 20150623 (Red Hat 4.8.5-28)
Copyright © 2015 Free Software Foundation, Inc.

$ cmake --version
cmake **version** 3.21.2
```

### CUDA & GPU Device

```
Device 0: "NVIDIA GeForce RTX 2080 Ti"
deviceQuery, CUDA Driver = CUDART, CUDA Driver Version = 11.4, CUDA
Runtime Version = 11.4, NumDevs = 1
```

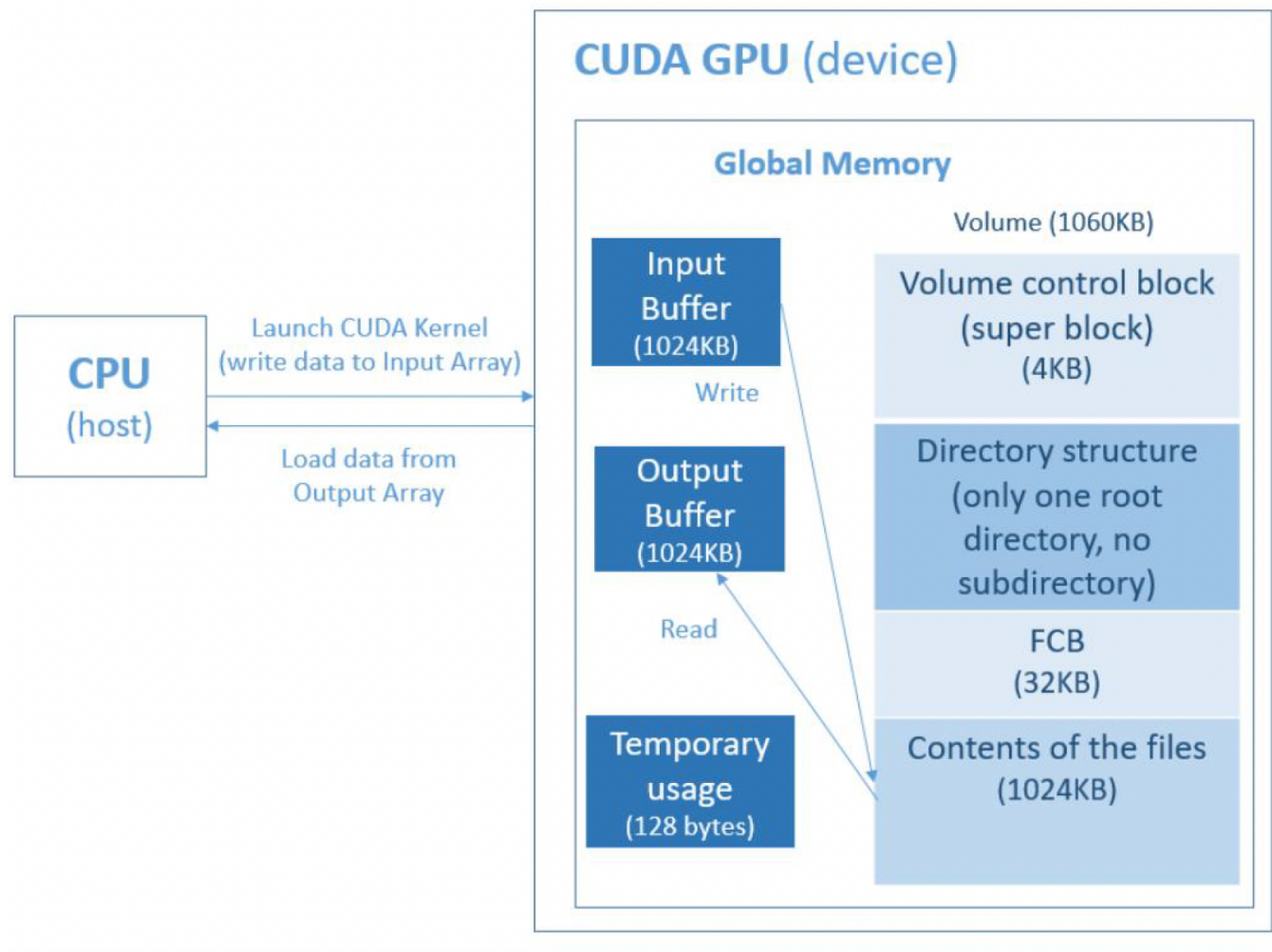
## Background

The File system is a method and data structure that the operating system uses to control how data is stored and retrieved. By separating the data into pieces and giving each piece a name, the data is easily isolated and identified. Taking its name from the way paper-based data management system is named, each group of data is called a "file." The structure and logic rules used to manage the groups of data and their names is called a "file system".

The File System is organized into layers: application programs->logical file system->file-organization module->base file system->I/O control->device. The application programs provide users with APIs to use system calls. The file organization module translates logical block to physical block, and manages free space and disk allocation.

In this project, I have implemented the functions of some APIs with structures: Volume Control Block--contains total # of blocks, # of free blocks... File Control Block--file size, time, filename...The structures are organized on

GPU:



The file system is simulated with only one directory (no subdirectory), and with multiple subdirectories (the max depth is 3) by a tree structure on GPU.

## Design Method

### File System Structure

#### 1. Super Block

```
struct SuperBlock
{
    int free_block_count; // how many free block
    u16 free_block_start; // the first start free block number;
    int total_file = 0;   // how many files in the storge
};
```

#### 2. File Control Block

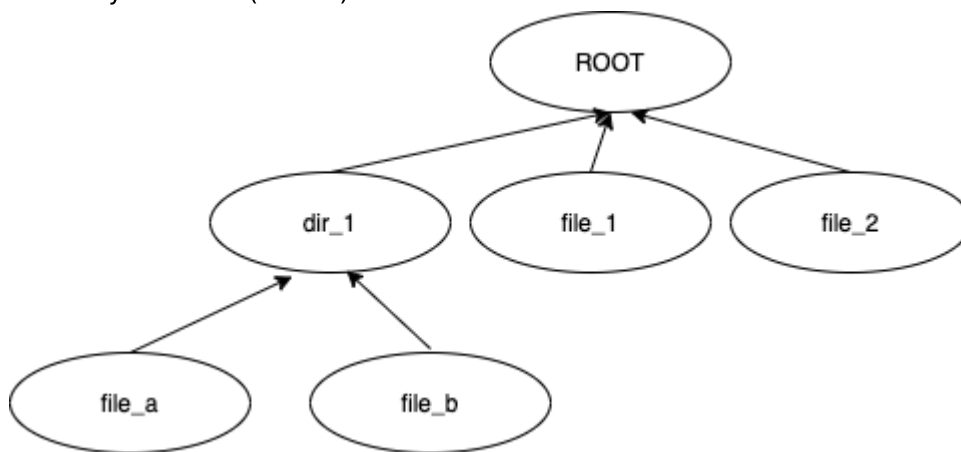
```
#pragma pack(1)
struct FCB
{
```

```
    u32 modified_time; // 4 bytes
    u32 create_time;   // 4 bytes
    u16 file_size;      // 2 bytes
    u16 start_block;
    char filename[20];
};
#pragma pack()
```

Declaration in the File System:

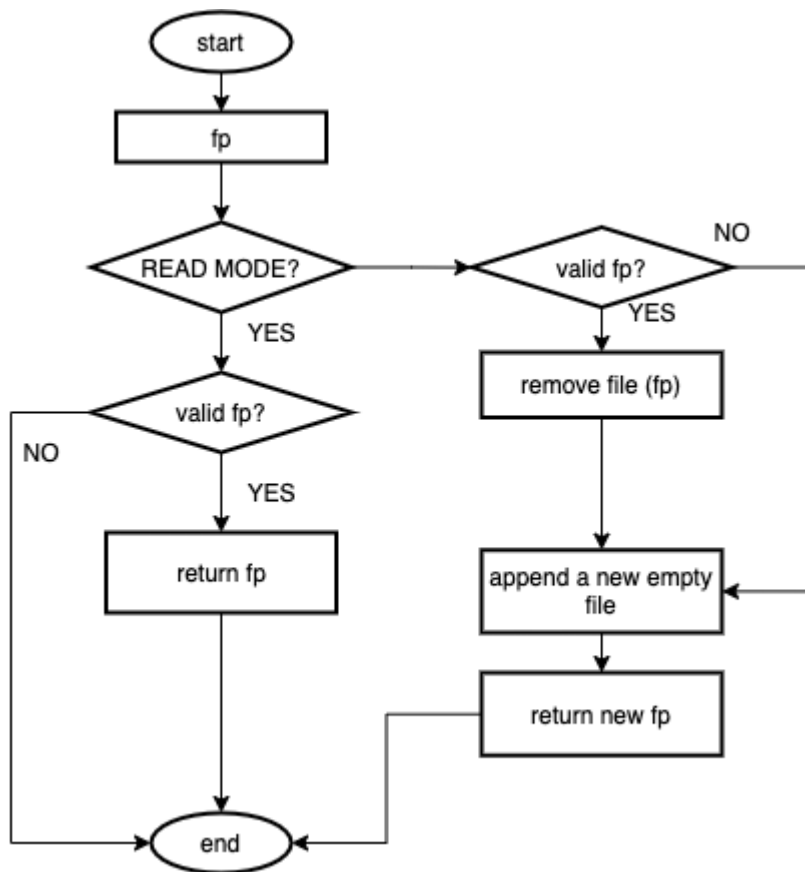
```
SuperBlock *superBlock_ptr;
struct FCB *FCB_arr;
uchar *fileContent_ptr;
```

Directory structure (**Bonus**)



API

**fs\_open(fs, name, G\_READ/G\_WRITE)**



### fs\_write(fs, input, size, fp)

```

for (u32 i = start_addr; i < start_addr + size; i++)
{
    fs->fileContent_ptr[i] = input[i - start_addr];
} // write to File Content from input buffer.

// update FCB_arr
fs->FCB_arr[fp].file_size = size;
fs->FCB_arr[fp].modified_time = gtime++;

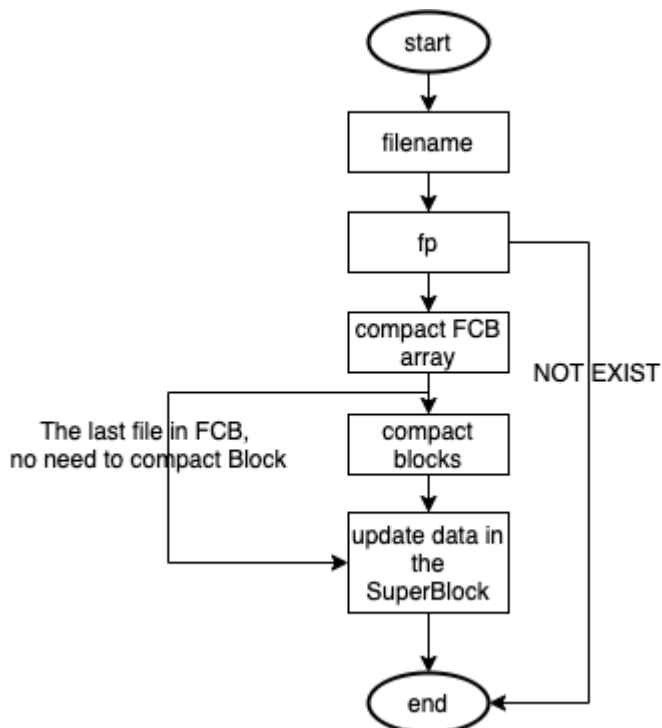
// update superBlock_ptr
int delta_block = block_needs - origin_blocks;
fs->superBlock_ptr->free_block_count -= delta_block;
fs->superBlock_ptr->free_block_start += delta_block;

```

### fs\_read(fs, output, size, fp)

- fp => get **start\_block** and **file\_size**
- Get physical start address and end address
- Read from File Content, and put result to **output** buffer.

### fs\_gsys(fs, RM, name)



### fs\_gsys (LS\_D / LS\_S)

- Use Bubble Sort
- LS\_D: LS\_D list all files name in the directory and order by modified time of files, least modified, first print.
- LS\_S: list all files name and size in the directory and order by descending size. If there are several files with the same size, then first create first print.

-----Bonus-----

### fs\_open(fs, name, G\_READ/G\_WRITE)

- update: when create a new file, modify the current directory's `file_size` and `modified_time`;
- update: when remove a file, modify the current directory's `file_size`;

### fs\_gsys(fs, MKDIR, name)

- Similar with create an empty file in `fs_open`

### fs\_gsys(fs, CD, name)

```
fp = linear search to get FCD index by name;
fs->curr_dir = fp; // update the current directory;
```

### fs\_gsys(fs, CD\_P)

```
// change to the parent directory of the current directory;
fs->curr_dir = fs->FCB_arr[fs->curr_dir].parent;
```

### fs\_gsys(fs, RM\_RF, name)

```
// recursive_remove_dir(fs, fp);
if (directory fp is empty) remove the empty directory fp;
for (i in range of total files number) {
    if file i is a file {
        remove file i;
    } else if (file i is a directory) {
        recursive_remove_dir(fs, i);
    }
}
remove empty directory fp;
```

### **fs\_gsys(fs, PWD)**

- Back track the parent directory until reach to the root directory. Print out the current path.

### **fs\_gsy(fs, LS\_D\LS\_S)**

- update: list the files as well as directories. For a file, list it name (with size) only. For a directory, add an symbol 'd' at the end.

## Execution

```
.
├── bonus
├── source
└── Report_119010256.pdf
```

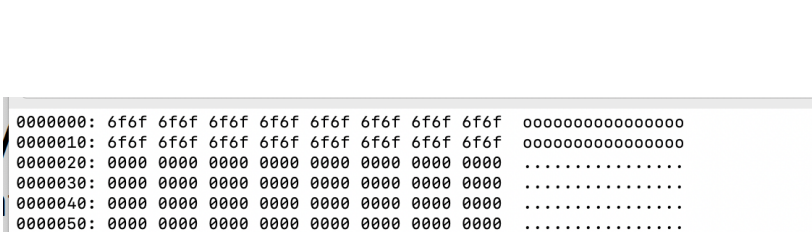
```
# under the source folder or the bonus folder ...
mkdir build && cd build
cmake ..
cp ../data.bin ./
make -j4

# under the build folder
./cuda (or ./as4_bonus)
```

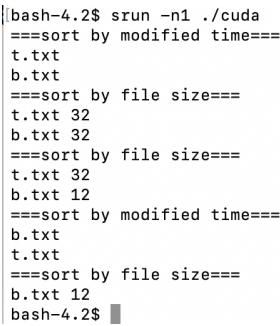
## Result

### 1. TEST1

---

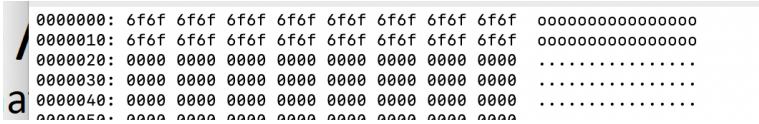


<Fig.3.1 TEST1's snapshot>

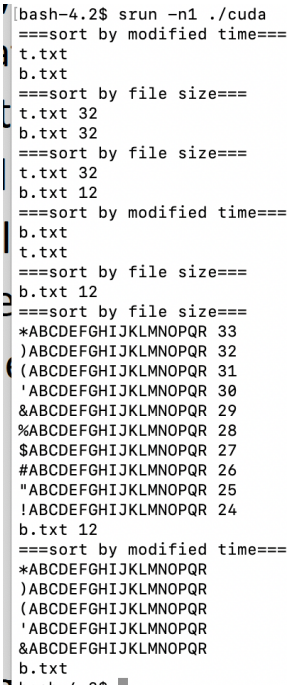


<Fig.3.2 TEST1's output>

2. TEST2



<Fig.3.3 TEST2's snapshot>



<Fig.3.4 TEST2's output>

3. TEST3

```

00000000: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000003e0: 0000 0000 0000 0000 fae5 17f8 a2dc 72af .....r.
0000003f0: 4ba0 28c0 c318 25ee e69f f444 5a7e 8ee9 K.(...%...DZ~..
000000400: 95c5 e424 e374 29de d9bf 57fc 9dca ace8 ...$.t)...W....
000000410: 6b54 2aae ebce 1dd3 ee12 9749 90a5 b2a6 kT*.....I....
000000420: 6b97 ca50 8c73 ae66 3407 63d1 d110 3abc k..P.s.f4.c....
000000430: 64e3 6b51 b388 a422 1abb 6baa 619d 51cc d.kQ...".k.a.Q.
000000440: b59c 9c42 104c a844 530d 95a4 9c50 e081 ...B.L.DS....P..
000000450: b3bc d267 54f6 89ed b274 9814 926a 6048 ...gT....t...j`H
000000460: 07fd 8a96 4a33 db1d bff0 415d c022 de75 ....J3....A].".u
000000470: ed31 5cc1 2866 af5a dac7 edec b14e 3538 .1\.(f.Z.....N58
000000480: cb3f cf95 f22b 32b2 1c73 10dd 956e 5303 ..?..+2...s..nS.
000000490: 1faf 44c6 16f3 2171 3c8e dded 5d14 2729 ..D...!q<...].')
0000004a0: 53f6 3fc5 2271 79bd e509 1bfa f7ed 7e17 S.?. "qy.....~.
0000004b0: 1ec2 deb3 b700 a4f3 8f83 61ec 1788 95e9 .....a.....
0000004c0: fed4 b021 472a 5fac 337a a7aa e826 c207 ...!G*_3z...&..
0000004d0: e9a1 ba21 21df 9430 63f5 1d7a fe33 64fd ...!!...0c.z.3d.
0000004e0: 8715 9fce befe fa72 f8a3 1d61 49df 6833 .....r...aI.h3
0000004f0: 81a3 d323 83e7 53e6 5ef0 e05d 24c4 5bab ....#.S.^...$.[.
000000500: da7a fa19 f8f5 8b72 19a9 d363 093d 160b .z.....r...c.=..
000000510: 60ea 2ee3 5281 4ab0 722b 0e16 efe9 424a `...R.J.r+...BJ
000000520: 64bc 64dd 326f 504c 9824 afa2 6145 2d41 d..d.2oPL$.aE-A
000000530: af5b 2503 5cee b34f 9942 650a 2c27 5410 .[%.\..0.Be., 'T.
000000540: e338 ed17 283e 63c0 e292 6344 d790 0688 .8..(>c...cD....
000000550: 6b2b 0bc8 9abe 1834 80fc 3e2c 2513 3d09 k+....4..>,%.=.
000000560: caaa 20f2 6903 b44c 9597 10ed a816 f514 .. .i..L.....
000000570: 4201 5cdc 3ff3 9040 f1cf 6c17 e2a9 9fad B.\.?..@..l.....
000000580: 55c0 a1be 43d5 8ad9 6d9a 4795 b13d 2af3 U...C...m.G...=*.
000000590: bd86 507c 7be0 bc6d 302a 0492 53a3 c028 ..P|{.m0*..S..(
0000005a0: e3e1 6628 b7f0 81a4 8cc8 3b3e 8565 b1c2 ..f(f.....);>.e.
0000005b0: 6b81 bee6 e17c d313 2657 2579 7be5 a2c2 k...|...&W%y{._
0000005c0: c788 877f 7a88 a407 d05f c4d5 c476 98af ....z....._..v..
0000005d0: f758 9759 54ea 6c7a c212 733e f716 9dc0 .X.YT.lz..s>....
0000005e0: 9fa5 bf99 2ee3 207f 43e4 5587 5bee 38d2 ..... .C.U.[.8.
0000005f0: c64f 2c1b 3a19 95fc 2b09 baa2 9f59 63be .0.,:..+....Yc.
000000600: 7ea3 d72c 87f7 ab4b dc02 d2b7 700b 8a37 ~.,.,.,.K....p..7
000000610: 5a36 d115 ce67 9179 ef4d 1d0f a600 4d25 Z6...g.y.M....M%
000000620: 2325 d0ab 1d7d 76f9 7f49 31ef 55bb a62f #%...}v...I1.U../
000000630: 7278 44c0 5fd6 3bce 24d7 5e4a d7ab eefb rxD_.;$.^J....
000000640: 51bf 27ed 3d9d 683c 6699 abbb d552 6b48 Q.'..h<f....RkH
000000650: 4aaf 88a9 86c3 f82a 9b57 74f3 8263 efd3 J.....*.Wt..C..
000000660: a396 c2e0 342b 9d9a 4449 d61a 1c42 e166 ....4+...DI...B.f
000000670: 716b 9078 2f89 a24b e097 3fe2 faae b71e qk.x/..K..?.....
000000680: 45f9 00f8 a49d 93e8 66e9 0482 abe5 e91e E.....f.....
000000690: 517a 9601 83b8 cbe3 500b c6cb b9fd e97e Qz.....P.....~
0000006a0: f769 779c 078a 06ed 750a 7021 6fd9 be41 .iw.....u.p!o..A
0000006b0: d355 c157 0f8d 3bde 1882 aad1 8095 cff8 .U.W.;.....
0000006c0: fe47 1586 d11b 74c6 a4e4 6815 bf27 5693 .G....t...h... 'V.
0000006d0: fc97 6b8b 2526 6a3d a816 8e2a ab5e 232a ..k.%&j=...*.^#*
0000006e0: 2538 b0f6 d325 be78 8a27 8dc9 cd63 dcca %8...%.x.'...c..
0000006f0: fa48 d59f ee41 dc97 d66b 4102 49e3 2c6e .H...A...kA.I.,n
000000700: 1d5d e570 0224 e88c caf6 5698 d9b3 e355 .].p.$....V....U
000000710: fbb9 f4ea 7a52 03d0 3d44 d287 297f 75c5 ....zR...=D...)..u.
000000720: dc5b 36de 7f9e 6c4a 9542 63ef f547 4572 .[6...lJ.Bc...GEr
000000730: 803a dcfb 8cdf cc4a a420 d14d 9f47 927c .:.....J..M.G.|
000000740: 23c8 5ba2 e647 6d7d 0ad0 6d00 1832 f118 #. [.Gm}..m..2..
000000750: eb4f 9379 aee0 c353 0115 20a0 dbb2 9cfe .0.y...S.. ....
000000760: 7b77 22e2 3f8f 6049 df4d c8f7 7fbb 8f6b {w".?.`I.M....k
000000770: 0b24 6439 05a7 8d06 3cad 2619 e042 97db .sd9....<.&..B..
000000780: b939 be78 c89e c128 eb8b 9fea c630 d651 .9.x...(. ....0.Q
000000790: 54ba 8a59 e297 de9e c584 b7a6 46cf 827f T..Y.....F...
0000007a0: 0942 f852 e0ba 7a4d c51b b78c 4b8e 5d1f .B.R...zM...K.).
0000007b0: c9e8 f7ac 8056 4bc5 5a83 6ca0 d2ef 205b .....VK.Z.l... [
0000007c0: b119 ad92 54a8 df1a 4318 270e a604 2def ....T...C.'...-.
0000007d0: eca4 1ced fae7 b355 eaa0 f5bd 9095 1942 .....U.....B
0000007e0: 2f47 5483 6fb4 1d32 0000 0000 0000 0000 /GT.o..2.....
0000007f0: 0000 0000 0000 0000 0000 0000 0000 0000 .....

```

```

GA 45
FA 44
DA 42
CA 41
BA 40
AA 39
@A 38
?A 37
>A 36
=A 35
<A 34
*ABCDEFGHIJKLMNOPQRSTUVWXYZ 33
;A 33
)ABCDEFGHIJKLMNOPQRSTUVWXYZ 32
:A 32
(ABCDEFGHIJKLMNOPQRSTUVWXYZ 31
9A 31
'ABCDEFGHIJKLMNOPQRSTUVWXYZ 30
8A 30
&ABCDEFGHIJKLMNOPQRSTUVWXYZ 29
7A 29
6Atxt 28
5Atxt 27
4Atxt 26
3Atxt 25
2Atxt 24
b.txt 12

```

&lt;Fig.3.5 TEST3's snapshot&gt;

&lt;Fig.3.6 TEST3's output&gt;

#### 4. BONUS TEST



```

0000000: 6f6f 6f6f 6f6f 6f6f 6f6f 6f6f 6f6f 6f6f 0000000000000000
0000010: 6f6f 6f6f 6f6f 6f6f 6f6f 6f6f 6f6f 6f6f 0000000000000000
0000020: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000030: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000040: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000050: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000060: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000070: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000080: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000090: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000a0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000b0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000c0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000d0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000e0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
00000f0: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000100: 0000 0000 0000 0000 0000 0000 0000 0000 .....
0000110: 0000 0000 0000 0000 0000 0000 0000 0000 .....

```

```

===sort by modified time===
t.txt
b.txt
===sort by file size===
t.txt 32
b.txt 32
===sort by modified time===
app d
t.txt
b.txt
===sort by file size===
t.txt 32
b.txt 32
app 0 d
===sort by file size===
===sort by file size===
a.txt 64
b.txt 32
soft 0 d
===sort by modified time===
soft d
b.txt
a.txt
/app/soft
===sort by file size===
B.txt 1024
C.txt 1024
D.txt 1024
A.txt 64
>> move to parent director
===sort by file size===
a.txt 64
b.txt 32
soft 20 d
/app
>> move to parent director
===sort by file size===
t.txt 32
b.txt 32
app 14 d
===sort by file size===
a.txt 64
b.txt 32
>> move to parent director
===sort by file size===
t.txt 32
b.txt 32
app 10 d
bash-4.2$ srunk -n1 ./a4_bonus > █

```

&lt;Fig.3.7 BONUS TEST's snapshot&gt;

&lt;Fig.3.8 BONUS TEST3's output&gt;

## Problem Encountered

Q: How to handle the external fragment?

A: Compaction. When remove a file, compact the BLOCK, and compact FCB.

## Conclusion

In this project, I have implemented the file system under only one directory and multiple directories. By implementing the functions, I have a better understanding of how the file system works, and how to design the data structure and logical methods.