

운영체제(SW) 3분반

Lab3. Ext2 Analysis (digital forensic)

학번 : 32203660

성명 : 이현정

1. 목표

학생의 학번의 마지막 3자리중 1번째 자리의 숫자가 디렉토리 이름이고, 2번째~3번째 숫자와 역순 숫자가 파일 이름으로 학생마다 할당된 10개(각 파일마다 5개 block 할당)의 disk block을 찾는 것이 목표입니다. 저는 32203660번으로 마지막 3자리가 660이기 때문에 디렉토리 6과 파일 6번과 파일 60번을 찾아야합니다. ramdisk를 이용해서 ext2를 구성한 뒤에 파일 시스템을 접근하기 위해서 mount를 진행하고, 해당 프로그램에서 파일들을 생성하는 과정을 거쳐서 위에 언급한 디렉토리 6과 파일 60번을 찾는 과정을 진행할 것입니다.

추가로, 파일 시스템을 mount 하는 경우에 각 학생의 학번과 이름이 출력되도록 하는 것으로 리눅스 커널 레벨에서 ext2라는 커널 소스를 수정하는 과정을 거쳐 보는 것입니다.

2. 분석 결과 & 스냅샷

1) ramdisk 모듈 설치

권리자 권한을 위한 sudo su 명령어를 이용하고, 해당 프로그램을 make 하였습니다.

```
oslab@oslab:~/2022_DKU_OS/lab3_filesystem$ ls
append.c  create.sh  Makefile  ramdisk.c
oslab@oslab:~/2022_DKU_OS/lab3_filesystem$ sudo su
[sudo] password for oslab:
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# make
make -C /lib/modules/5.4.0-84-generic/build M=/home/oslab/2022_DKU_OS/lab3_filesystem modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-84-generic'
  CC [M]  /home/oslab/2022_DKU_OS/lab3_filesystem/ramdisk.o
  Building modules, stage 2.
  MODPOST 1 modules
  CC [M]  /home/oslab/2022_DKU_OS/lab3_filesystem/ramdisk.mod.o
  LD [M]  /home/oslab/2022_DKU_OS/lab3_filesystem/ramdisk.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-84-generic'
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# ls
append.c  modules.order  ramdisk.ko      ramdisk.mod.o
create.sh  Module.symvers  ramdisk.mod     ramdisk.o
Makefile   ramdisk.c       ramdisk.mod.c
```

ramdisk 모듈을 커널에 넣고, mount를 수행하기 위한 mnt 디렉토리를 생성하였습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# insmod ramdisk.ko
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# lsmod | grep ramdisk
ramdisk                16384  0
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# mkdir mnt
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# ls
append.c  Makefile  modules.order  ramdisk.c  ramdisk.mod  ramdisk.mod.o
create.sh  mnt       Module.symvers  ramdisk.ko  ramdisk.mod.c  ramdisk.o
```

파일시스템을 포맷하고 mnt에 mount하였습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# mkfs.ext2 /dev/ramdisk
mke2fs 1.44.1 (24-Mar-2018)
```

Creating filesystem with 131072 4k blocks and 32768 inodes

Filesystem UUID: f8f8faeb-12f1-4390-84e1-59485cb71808

Superblock backups stored on blocks:

32768, 98304

Allocating group tables: done

Writing inode tables: done

Writing superblocks and filesystem accounting information: done

위 내용을 확인한 내용입니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	2.0G	0	2.0G	0%	/dev
tmpfs	412M	1.4M	411M	1%	/run
/dev/sda1	9.8G	6.7G	2.6G	73%	/
tmpfs	2.1G	0	2.1G	0%	/dev/shm
tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
tmpfs	2.1G	0	2.1G	0%	/sys/fs/cgroup
/dev/loop0	82M	82M	0	100%	/snap/gtk-common-themes/1534
/dev/loop2	2.5M	2.5M	0	100%	/snap/gnome-calculator/884
/dev/loop1	249M	249M	0	100%	/snap/gnome-3-38-2004/99
/dev/loop3	62M	62M	0	100%	/snap/core20/1081
/dev/loop4	62M	62M	0	100%	/snap/core20/1434
/dev/loop5	56M	56M	0	100%	/snap/core18/2409
/dev/loop6	219M	219M	0	100%	/snap/gnome-3-34-1804/77
/dev/loop7	219M	219M	0	100%	/snap/gnome-3-34-1804/72
/dev/loop8	242M	242M	0	100%	/snap/gnome-3-38-2004/70
/dev/loop9	2.7M	2.7M	0	100%	/snap/gnome-system-monitor/174
/dev/loop10	768K	768K	0	100%	/snap/gnome-characters/741
/dev/loop11	2.7M	2.7M	0	100%	/snap/gnome-calculator/920
/dev/loop12	640K	640K	0	100%	/snap/gnome-logs/112
/dev/loop13	768K	768K	0	100%	/snap/gnome-characters/726
/dev/loop14	128K	128K	0	100%	/snap/bare/5
/dev/loop15	640K	640K	0	100%	/snap/gnome-logs/106
/dev/loop16	56M	56M	0	100%	/snap/core18/2128
/dev/loop17	2.5M	2.5M	0	100%	/snap/gnome-system-monitor/163
/dev/loop18	66M	66M	0	100%	/snap/gtk-common-themes/1515
/dev/loop19	45M	45M	0	100%	/snap/snapd/15904
tmpfs	412M	28K	412M	1%	/run/user/1000
tmpfs	412M	0	412M	0%	/run/user/0
/dev/ramdisk	504M	396K	478M	1%	/home/oslab/2022_DKU_OS/lab3_filesystem/mnt

create.sh로 파일 시스템 관련 디렉토리와 파일을 생성한 후에 apd 프로그램으로 생성하였습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# ./create.sh
create files ...
```

```
done
```

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# ls mnt
```

```
0 1 2 3 4 5 6 7 8 9 lost+found
```

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# ls mnt/6
```

```
0 14 2 25 30 36 41 47 52 58 63 69 74 8 85 90 96
1 15 20 26 31 37 42 48 53 59 64 7 75 80 86 91 97
10 16 21 27 32 38 43 49 54 6 65 70 76 81 87 92 98
11 17 22 28 33 39 44 5 55 60 66 71 77 82 88 93 99
12 18 23 29 34 4 45 50 56 61 67 72 78 83 89 94
13 19 24 3 35 40 46 51 57 62 68 73 79 84 9 95
```

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# ls -l mnt/6/6
```

```
-rw-r--r-- 1 root root 12294 5월 25 10:44 mnt/6/6
```

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# ls -l mnt/6/60
```

```
-rw-r--r-- 1 root root 12295 5월 25 10:44 mnt/6/60
```

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# ./apd mnt/6/60 13 6/60-13
```

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# ./apd mnt/6/6 13 6/6-13
```

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# ls -l mnt/6/60
```

```
-rw-r--r-- 1 root root 49160 5월 25 10:52 mnt/6/60
```

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# ls -l mnt/6/6
```

```
-rw-r--r-- 1 root root 49159 5월 25 10:52 mnt/6/6 _
```

EXT2의 정보는 메타 데이터는 리틀 엔디안으로 저장 되어있으므로 바이트별로 반대로 읽어야합니다.

<super block 영역>

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x100 -s 0x400
/dev/ramdisk
```

```
00000400: 00800000 00000200 99190000 8ff70100 .....
00000410: f57f0000 00000000 02000000 02000000 .....
00000420: 00800000 00800000 00200000 39ab8d62 ..... ..9..b
00000430: 39ab8d62 0100ffff 53ef0000 01000000 9..b....S.....
00000440: 30ab8d62 00000000 00000000 01000000 0..b.....
00000450: 00000000 0b000000 00010000 38000000 .....8...
00000460: 02000000 03000000 78848351 f3814bf2 .....x..Q..K.
00000470: 8b1dca66 48c8e43a 00000000 00000000 ...fH...:.....
00000480: 00000000 00000000 2f686f6d 652f6f73 ...../home/os
00000490: 6c61622f 32303232 5f444b55 5f4f532f lab/2022_DKU_OS/
000004a0: 6c616233 5f66696c 65737973 74656d2f lab3_filesystem/
000004b0: 6d6e7400 00000000 00000000 00000000 mnt.....
000004c0: 00000000 00000000 00000000 00001f00 .....
000004d0: 00000000 00000000 00000000 00000000 .....
000004e0: 00000000 00000000 00000000 feab2e93 .....
000004f0: b9e34b1b a6f09a5a 52dcb169 01000000 ..K....ZR..i....
```

```
inode count : 0x8000
```

```
block count : 0x2000
```

```
log block size : 0x2
```

```
blocks per group : 0x8000, inodes per group : 0x2000, block group number : 0x0
```

<group descriptor table 영역 분석>

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x100 -s 0x1000 /dev/ramdisk
```

```
00001000: 21000000 22000000 23000000 d47dc61e !..."...#....}..
00001010: 05000400 00000000 00000000 00000000 .....
00001020: 21800000 22800000 23800000 cb7a361f !..."...#....z6.
00001030: 02000400 00000000 00000000 00000000 .....
00001040: 00000100 01000100 02000100 6871361f .....hq6.
00001050: 02000400 00000000 00000000 00000000 .....
00001060: 21800100 22800100 23800100 da7dd11e !..."...#....}..
00001070: 03000400 00000000 00000000 00000000 .....
00001080: 00000000 00000000 00000000 00000000 .....
00001090: 00000000 00000000 00000000 00000000 .....
000010a0: 00000000 00000000 00000000 00000000 .....
000010b0: 00000000 00000000 00000000 00000000 .....
000010c0: 00000000 00000000 00000000 00000000 .....
000010d0: 00000000 00000000 00000000 00000000 .....
000010e0: 00000000 00000000 00000000 00000000 .....
000010f0: 00000000 00000000 00000000 00000000 .....
```

block bitmap : 0x21 블록부터 시작

inode bitmap : 0x22 블록부터 시작

inode table : 0x23블록부터 시작

해당 과정에서 블록은 4KB입니다.

ext2에서 root inode number은 2번이고, inode per group은 0x2000이므로

root의 block group : $(2-1) / (0x2000) = 0$

root의 index : $(2-1) \% 0x2000 = 1$

0번 block group의 inode table은 인덱스1에 위치합니다.

<inode table 영역 분석>

inode table이 0x23부터 시작하므로 0x23000을 이용해서 출력하면, index table의 인덱스 1을 보기 위해서는 0x23100을 보시면, ed41으로 root inode가 존재하는 것을 확인할 수 있습니다.

root 디렉토리는 0x223000에 있다는 것이 보여집니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x200 -s 0x2300
```

```
0 /dev/ramdisk
00023000: 00000000 00000000 30ab8d62 30ab8d62 .....0..b0..b
00023010: 30ab8d62 00000000 00000000 00000000 0..b.....
00023020: 00000000 00000000 00000000 00000000 .....
00023030: 00000000 00000000 00000000 00000000 .....
00023040: 00000000 00000000 00000000 00000000 .....
00023050: 00000000 00000000 00000000 00000000 .....
00023060: 00000000 00000000 00000000 00000000 .....
00023070: 00000000 00000000 00000000 00000000 .....
00023080: 00000000 00000000 00000000 00000000 .....
00023090: 00000000 00000000 00000000 00000000 .....
000230a0: 00000000 00000000 00000000 00000000 .....
000230b0: 00000000 00000000 00000000 00000000 .....
000230c0: 00000000 00000000 00000000 00000000 .....
000230d0: 00000000 00000000 00000000 00000000 .....
000230e0: 00000000 00000000 00000000 00000000 .....
000230f0: 00000000 00000000 00000000 00000000 .....
00023100: ed410000 00100000 e8ab8d62 42ab8d62 .A.....bB..b
00023110: 42ab8d62 00000000 00000d00 08000000 B..b.....
00023120: 00000000 0a000000 23020000 00000000 .....#.....
00023130: 00000000 00000000 00000000 00000000 .....
00023140: 00000000 00000000 00000000 00000000 .....
00023150: 00000000 00000000 00000000 00000000 .....
00023160: 00000000 00000000 00000000 00000000 .....
00023170: 00000000 00000000 00000000 00000000 .....
00023180: 20000000 d4cd0462 d4cd0462 00b4eb17 .....b...b...
```


<data 영역 분석> 디렉토리 6번

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x200 -s 0x223000 /dev/ramdisk
```

```
00223000: 02000000 0c000102 2e000000 02000000 .....
00223010: 0c000202 2e2e0000 0b000000 14000a02 .....
00223020: 6c6f7374 2b666f75 6e640000 01400000 lost+found...@..
00223030: 0c000102 30000000 01600000 0c000102 ....0....`.....
00223040: 31000000 01200000 0c000102 32000000 1.... .....2...
00223050: 0c000000 0c000102 33000000 71000000 .....3...q...
00223060: 0c000102 34000000 66200000 0c000102 ....4...f .....
00223070: 35000000 66600000 0c000102 36000000 5...f`.....6...
00223080: 66400000 0c000102 37000000 cb600000 f@.....7....`..
00223090: 0c000102 38000000 d6000000 680f0102 ....8.....h...
002230a0: 39000000 00000000 00000000 00000000 9.....
```

1번째 노란색 부분 -> 디렉토리 6번 inode : 0x6066

2번째 노란색 부분 -> 파일 타입 : 0x2

3번째 노란색 부분 -> 디렉토리 6번의 파일

디렉토리 6번은 (0x6066 -1) / 2000로 3번 block group에 존재하고, node table 인덱스는 65입니다.

그래서 6번 디렉토리의 inode는 block group의 inode table에서 65번째에 위치합니다.

1000의 위치로 이동해서 block group을 확인합니다.

2줄씩 인덱스 0부터 해당하므로 3번 block group은 인덱스 7번째 줄에 위치가 나타납니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x200 -s 0x1000 /dev/ramdisk
```

```
00001000: 21000000 22000000 23000000 d47dc61e !..."...#....}..
00001010: 05000400 00000000 00000000 00000000 .....
00001020: 21800000 22800000 23800000 3776361f !..."...#...7v6.
00001030: 02000400 00000000 00000000 00000000 .....
00001040: 00000100 01000100 02000100 fc75361f .....u6.
00001050: 02000400 00000000 00000000 00000000 .....
00001060: 21800100 22800100 23800100 da7dd11e !..."...#....}..
```

node table 인덱스가 65이므로 $18023000 \rightarrow 18023000 + 6500 = 18029500$ 으로 이동합니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x200 -s 0x18029500 /dev/ramdisk
```

```
18029500: ed410000 00100000 59b78d62 59b78d62 .A.....Y..bY..b
18029510: 59b78d62 00000000 00000200 08000000 Y..b.....
18029520: 00000000 65000000 24820100 00000000 ....e...$.
18029530: 00000000 00000000 00000000 00000000 .....
18029540: 00000000 00000000 00000000 00000000 .....
18029550: 00000000 00000000 00000000 00000000 .....
18029560: 00000000 6b054aba 00000000 00000000 ....k.J.....
18029570: 00000000 00000000 00000000 00000000 .....
18029580: 20000000 b05cbcc8 b05cbcc8 a439c8c7 ....\...\...9..
18029590: 59b78d62 a439c8c7 00000000 00000000 Y..b.9.....
```

6번 파일의 inode는 0x6c00만큼 인덱스로 가야한다는 것을 확인하였습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x500 -s 0x18224000 /dev/ramdisk
18224000: 66600000 0c000102 2e000000 02000000  f`.....
18224010: 0c000202 2e2e0000 67600000 0c000101  ....g`.....
18224020: 30000000 68600000 0c000101 31000000  0...h`.....1...
18224030: 69600000 0c000101 32000000 6a600000  i`.....2...j`..
18224040: 0c000101 33000000 6b600000 0c000101  ....3...k`.....
18224050: 34000000 6c600000 0c000101 35000000  4...l`.....5...
18224060: 6d600000 0c000101 36000000 6e600000  m`.....6...n`..
18224070: 0c000101 37000000 6f600000 0c000101  ....7...o`.....
18224080: 38000000 70600000 0c000101 39000000  8...p`.....9...
18224090: 71600000 0c000201 31300000 72600000  q`.....10..r`..
182240a0: 0c000201 31310000 73600000 0c000201  ....11..s`.....
```

60번의 파일의 inode는 0xa200 인덱스만큼 가야한다는 것을 확인하였습니다.

```
182242b0: 0c000201 35350000 9f600000 0c000201  ....55...`.....
182242c0: 35360000 a0600000 0c000201 35370000  56...`.....57..
182242d0: a1600000 0c000201 35380000 a2600000  .`.....58...`..
182242e0: 0c000201 35390000 a3600000 0c000201  ....59...`.....
182242f0: 36300000 a4600000 0c000201 36310000  60...`.....61..
18224300: a5600000 0c000201 36320000 a6600000  .`.....62...`..
18224310: 0c000201 36330000 a7600000 0c000201  ....63...`.....
18224320: 36340000 a8600000 0c000201 36350000  64...`.....65..
```

파일 6번과 60번을 찾아야 합니다.

1) 6번 파일입니다.

0x18023000 + 0x6c00 -> 0x18029c00로 이동하였습니다.

(1) direct pointer로 첫 블록은 '845e'를 가리키고 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x200 -s 0x18029c00 /dev/ramdisk
18029c00: a4810000 07c00000 59b78d62 8ab78d62  ....Y..b...b
18029c10: 8ab78d62 00000000 00000100 30000000  ...b.....0...
18029c20: 00000000 01000000 5e840000 f6860000  ....^.....
18029c30: 5a870000 be870000 00000000 00000000  Z.....
18029c40: 00000000 00000000 00000000 00000000  .....
18029c50: 00000000 00000000 26820000 00000000  ....&.....
18029c60: 00000000 1d666a21 00000000 00000000  ....fj!.....
18029c70: 00000000 00000000 00000000 00000000  .....
18029c80: 20000000 6cbe507c 6cbe507c b05cbcc8  ...l.P|l.P|.\..
18029c90: 59b78d62 b05cbcc8 00000000 00000000  Y..b.\.....
18029ca0: 00000000 00000000 00000000 00000000  .....
```

0x545e000으로 이동하면, 파일 6의 1번째 블록을 확인할 수 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x50 -s 0x845e000 /dev/ramdisk
0845e000: 362f362d 310a0000 00000000 00000000  6/6-1.....
0845e010: 00000000 00000000 00000000 00000000  .....
0845e020: 00000000 00000000 00000000 00000000  .....
0845e030: 00000000 00000000 00000000 00000000  .....
0845e040: 00000000 00000000 00000000 00000000  .....
```

2번째 블록으로는 'f686'으로 0x86f6000으로 이동하면 확인할 수 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x50 -s 0x86f6000 /dev/ramdisk
086f6000: 362f362d 320a0000 00000000 00000000  6/6-2.....
086f6010: 00000000 00000000 00000000 00000000  .....
086f6020: 00000000 00000000 00000000 00000000  .....
086f6030: 00000000 00000000 00000000 00000000  .....
086f6040: 00000000 00000000 00000000 00000000  .....
```

3번째 블록으로는 '5a87'로 0x875a000으로 이동하면 확인할 수 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x50 -s 0x875a000 /dev/ramdisk
0875a000: 362f362d 330a0000 00000000 00000000  6/6-3.....
0875a010: 00000000 00000000 00000000 00000000  .....
0875a020: 00000000 00000000 00000000 00000000  .....
0875a030: 00000000 00000000 00000000 00000000  .....
0875a040: 00000000 00000000 00000000 00000000  .....
```

4번째 블록으로는 'be87'로 0x87be000으로 이동하면 확인할 수 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x50 -s 0x87be000 /dev/ramdisk
087be000: 362f362d 340a0000 00000000 00000000  6/6-4.....
087be010: 00000000 00000000 00000000 00000000  .....
087be020: 00000000 00000000 00000000 00000000  .....
087be030: 00000000 00000000 00000000 00000000  .....
087be040: 00000000 00000000 00000000 00000000  ....._.....
```

(2) indirect pointer는 '2682'를 가리키고 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x200 -s 0x18029c00 /dev/ramdisk
18029c00: a4810000 07c00000 59b78d62 8ab78d62  ....Y..b...b
18029c10: 8ab78d62 00000000 00000100 30000000  ...b.....0...
18029c20: 00000000 01000000 5e840000 f6860000  .....^.....
18029c30: 5a870000 be870000 00000000 00000000  Z.....
18029c40: 00000000 00000000 00000000 00000000  .....
18029c50: 00000000 00000000 26820000 00000000  .....&.....
18029c60: 00000000 1d666a21 00000000 00000000  ....fj!.....
18029c70: 00000000 00000000 00000000 00000000  .....
18029c80: 20000000 6cbe507c 6cbe507c b05cbcc8  ...l.P|l.P|.\..
18029c90: 59b78d62 b05cbcc8 00000000 00000000  Y..b.\.....
```

direct pointer로는 ./apd로 만든 파일의 위치를 나타내고 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x200 -s 0x8226000 /dev/ramdisk
08226000: a18b0000 00000000 00000000 00000000  .....
08226010: 00000000 00000000 00000000 00000000  .....
08226020: 00000000 00000000 00000000 00000000  .....
08226030: 00000000 00000000 00000000 00000000  .....
```


0x8ba1000으로 이동하면, 우리가 만든 파일 부분이 보여지게 됩니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x200 -s 0x8ba1000 /dev/ramdisk
08ba1000: 362f362d 31330a00 00000000 00000000 6/6-13.....
08ba1010: 00000000 00000000 00000000 00000000 .....
08ba1020: 00000000 00000000 00000000 00000000 .....
```

2) 60번 파일입니다.

0x18023000 + 0xa200 -> 0x1802d200로 이동하였습니다.

(1) direct pointer로 첫 블록은 '9484'를 가리키고 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x100 -s 0x1802d200 /dev/ramdisk
1802d200: a4810000 08c00000 59b78d62 7ab78d62 .....Y..bz..b
1802d210: 7ab78d62 00000000 00000100 30000000 z..b.....0...
1802d220: 00000000 01000000 94840000 2c870000 .....
1802d230: 90870000 f4870000 00000000 00000000 .....
1802d240: 00000000 00000000 00000000 00000000 .....
1802d250: 00000000 00000000 25820000 00000000 .....%.
1802d260: 00000000 7093fcee 00000000 00000000 ....p.....
1802d270: 00000000 00000000 00000000 00000000 .....
1802d280: 20000000 2cf52ad2 2cf52ad2 b05cbcc8 ...,*.,*..\..
1802d290: 59b78d62 b05cbcc8 00000000 00000000 Y..b.\.....
```

0x872c000으로 이동하면, 파일 60의 1번째 블록을 확인할 수 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x100 -s 0x8494000 /dev/ramdisk
08494000: 362f3630 2d310a00 00000000 00000000 6/60-1.....
08494010: 00000000 00000000 00000000 00000000 .....
08494020: 00000000 00000000 00000000 00000000 .....
08494030: 00000000 00000000 00000000 00000000 .....
08494040: 00000000 00000000 00000000 00000000 .....
```

0x8494000으로 이동하면, 파일 60의 2번째 블록을 확인할 수 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x50 -s 0x872c000 /dev/ramdisk
0872c000: 362f3630 2d320a00 00000000 00000000 6/60-2.....
0872c010: 00000000 00000000 00000000 00000000 .....
0872c020: 00000000 00000000 00000000 00000000 .....
0872c030: 00000000 00000000 00000000 00000000 .....
0872c040: 00000000 00000000 00000000 00000000 .....
```

0x8790000으로 이동하면, 파일 60의 3번째 블록을 확인할 수 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x50 -s 0x8790000 /dev/ramdisk
08790000: 362f3630 2d330a00 00000000 00000000 6/60-3.....
08790010: 00000000 00000000 00000000 00000000 .....
08790020: 00000000 00000000 00000000 00000000 .....
08790030: 00000000 00000000 00000000 00000000 .....
08790040: 00000000 00000000 00000000 00000000 .....
```

0x87f4000으로 이동하면, 파일 60의 4번째 블록을 확인할 수 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x50 -s 0x87f4000 /dev/ramdisk
087f4000: 362f3630 2d340a00 00000000 00000000 6/60-4.....
087f4010: 00000000 00000000 00000000 00000000 .....
087f4020: 00000000 00000000 00000000 00000000 .....
087f4030: 00000000 00000000 00000000 00000000 .....
087f4040: 00000000 00000000 00000000 00000000 .....
```

(2) indirect pointer는 '2582'를 가리키고 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x100 -s 0x1802d200 /dev/ramdisk
1802d200: a4810000 08c00000 59b78d62 7ab78d62 .....Y..bz..b
1802d210: 7ab78d62 00000000 00000100 30000000 z..b.....0...
1802d220: 00000000 01000000 94840000 2c870000 .....
1802d230: 90870000 f4870000 00000000 00000000 .....
1802d240: 00000000 00000000 00000000 00000000 .....
1802d250: 00000000 00000000 25820000 00000000 .....%.
1802d260: 00000000 7093fcee 00000000 00000000 ....p.....
1802d270: 00000000 00000000 00000000 00000000 .....
1802d280: 20000000 2cf52ad2 2cf52ad2 b05cbcc8 ...,*.,*..\..
1802d290: 59b78d62 b05cbcc8 00000000 00000000 Y..b.\.....
```

direct pointer로는 ./apd로 만든 파일의 위치를 나타내고 있습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x100 -s 0x8225000 /dev/ramdisk
08225000: a08b0000 00000000 00000000 00000000 .....
08225010: 00000000 00000000 00000000 00000000 .....
08225020: 00000000 00000000 00000000 00000000 .....
08225030: 00000000 00000000 00000000 00000000 .....
08225040: 00000000 00000000 00000000 00000000 .....
```

0x8ba0000으로 이동하면, 우리가 만든 파일 부분이 보여지게 됩니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x50 -s 0x8ba0000 /dev/ramdisk
08ba0000: 362f3630 2d31330a 00000000 00000000 6/60-13.....
08ba0010: 00000000 00000000 00000000 00000000 .....
08ba0020: 00000000 00000000 00000000 00000000 .....
08ba0030: 00000000 00000000 00000000 00000000 .....
08ba0040: 00000000 00000000 00000000 00000000 ....._.....
```

* 보너스

umount, rmmod로 ramdisk를 정리 -> ext2의 super.c 소스 수정 -> make로 컴파일 -> insmod로
집어넣기 -> mount로 자신의 학번 출력의 순서로 진행하였습니다.

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# umount /dev/ramdisk
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# rmmod ramdisk
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# lsmod | grep ramdisk
```

< 수정된 super.c >

```
if (!(bh = sb_bread(sb, logic_sb_block))) {
    ext2_msg(sb, KERN_ERR, "error: unable to read superblock");
    goto failed_sbi;
}

printf("os_ext2 : Lee Hyeonjeong OS Lab3");
/*
 * Note: s_es must be initialized as soon as possible because
 * some ext2 macro-instructions depend on its value
 */
es = (struct ext2_super_block *) (((char *)bh->b_data) + offset);
sbi->s_es = es;
sb->s_magic = le16_to_cpu(es->s_magic);

if (sb->s_magic != EXT2_SUPER_MAGIC)
    goto cantfind_ext2;
```

```
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem/os_ext2# make
make -C /lib/modules/5.4.0-84-generic/build M=/home/oslab/2022_DKU_OS/lab3_filesystem/os_ext2 modules
make[1]: Entering directory '/usr/src/linux-headers-5.4.0-84-generic'
CC [M] /home/oslab/2022_DKU_OS/lab3_filesystem/os_ext2/super.o
CC [M] /home/oslab/2022_DKU_OS/lab3_filesystem/os_ext2/symlink.o
LD [M] /home/oslab/2022_DKU_OS/lab3_filesystem/os_ext2/os_ext2.o
Building modules, stage 2.
MODPOST 1 modules
CC [M] /home/oslab/2022_DKU_OS/lab3_filesystem/os_ext2/os_ext2.mod.o
LD [M] /home/oslab/2022_DKU_OS/lab3_filesystem/os_ext2/os_ext2.ko
make[1]: Leaving directory '/usr/src/linux-headers-5.4.0-84-generic'
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem/os_ext2# insmod os_ext2.ko
insmod: ERROR: could not insert module os_ext2.ko: File exists
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem/os_ext2# ls
acl.c      file.o      Makefile    os_ext2.mod.o  xattr.h
acl.h      ialloc.c    modules.order  os_ext2.o      xattr_security.c
ballocc.c  ialloc.o    Module.symvers  super.c        xattr_trusted.c
ballocc.o  inode.c     namei.c        super.o        xattr_user.c
dir.c      inode.o     namei.o        symlink.c
dir.o      ioctl.c     os_ext2.ko     symlink.o
ext2.h     ioctl.o     os_ext2.mod    tags
file.c     Kconfig    os_ext2.mod.c  xattr.c
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem/os_ext2# lsmod | grep os_ext2
os_ext2                73728  0
root@oslab:/home/oslab/2022_DKU_OS/lab3_filesystem# dmesg | grep os_ext2
[11736.150124] os_ext2 : Lee Hyeonjeong OS Lab3
```

super.c 파일에서 superblock 읽기와 ext2_sb_info 초기화 부분 사이에 해당 문장을 출력하는 문장을 넣어서 ext2 파일시스템에서 mount와 lookup의 경우에 제 이름이 출력이 될 수 있게 구성하였습니다.

3. 논의

저는 이번 과제를 통해서 파일 시스템의 구조와 어떠한 과정을 거쳐서 파일에 접근할 수 있는지에 대해서 배울 수 있었습니다.

기본적으로 EXT2 파일 시스템의 super block에서는 위치에 따라서 inode와 block의 개수, 그룹별로 block과 inode의 개수, block group의 개수 등 원하는 정보를 확인할 수 있다는 점을 알게 되었습니다. super block에서 시작해서 6번 디렉토리, 6번과 60번 파일을 위해서 각 파일의 inode를 통해서 해당 파일의 정보에 접근하고, 파일이 디렉토리인 경우에는 해당 파일 내부에 작성된 파일의 inode 위치를 확인해서 해당 inode를 통하고, direct 혹은 indirect 포인터를 통해 실제로 원하던 6번과 60번 파일의 내용에 접근하기까지의 과정을 사용자는 신경쓰지 않고 메모리의 공간을 이용할 수 있다는 점에서 가상 메모리로 메모리가 추상화되는 것이 얼마나 사용자에게 편의를 제공해주었는지 직접 확인할 수 있었습니다. 이렇게 여러 파일을 통해서 접근하기 때문에 메모리에 접근하는데 시간이 오래 걸려서 메모리에 직접 수정하지 않고, 변경이 발생했다는 내용을 적어두었다가 메모리에 직접 접근해야하는 경우에 해당 내용의 변경 사항을 실제 메모리에 작성하는 것이 얼마나 큰 성능 향상의 효과를 가져오는지 또한 느낄 수 있었습니다.

과제에서는 원하는 파일의 내부 내용을 확인하기 위해 내부 구조에 대해서 파악하고 있으며, 현재 자신의 위치와 해당 숫자가 나타내는 의미, 그리고 원하는 위치로 이동하기 위한 계산 과정에서 어려움을 느꼈습니다. 다행히 수업 시간에 예시를 통해서 설명해주신 부분들로 제가 이동하는 순서와 방향에 대해 알고서 작업을 할 수 있었으며, 파일 시스템의 구조를 이해하는데 많은 도움을 받을 수 있었습니다.

해당 구조를 보면서 최대한 효율적으로 체계적인 접근 경로를 위한 노력을 확인할 수 있었으며, 이러한 구조를 이해한 상태로 프로그램을 하는 개발자가 되어서 이러한 구조가 의미 있게 사용하 러수 있는 사람이 되기 위해 노력하겠습니다!