

Operating System Lab 3

운영체제(SW) 2분반, 32170578, 김산

[Lab 3 EXT2 file system 실습]

File System layout analysis

1) 모듈 적재후 확인

```
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# lsmod | grep ramdisk
ramdisk                16384  0
```

2) 파일시스템 포맷후 ./mnt에 마운트

```
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# lsmod | grep ramdisk
ramdisk                16384  0
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# mkfs.ext2 /dev/ramdisk
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 131072 4k blocks and 32768 inodes
Filesystem UUID: 13ac5a52-1d1f-4dda-9f71-1c11d424e0a1
Superblock backups stored on blocks:
    32768, 98304

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# mount /dev/ramdisk ./mnt
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
udev	2.4G	0	2.4G	0%	/dev
tmpfs	494M	1.4M	493M	1%	/run
/dev/sda5	9.3G	7.3G	1.6G	83%	/
tmpfs	2.5G	0	2.5G	0%	/dev/shm
tmpfs	5.0M	4.0K	5.0M	1%	/run/lock
tmpfs	2.5G	0	2.5G	0%	/sys/fs/cgroup
/dev/loop1	56M	56M	0	100%	/snap/core18/1997
/dev/loop2	219M	219M	0	100%	/snap/gnome-3-34-1804/66
/dev/loop3	66M	66M	0	100%	/snap/gtk-common-themes/1515
/dev/loop4	52M	52M	0	100%	/snap/snap-store/518
/dev/loop0	56M	56M	0	100%	/snap/core18/2066
/dev/loop5	32M	32M	0	100%	/snap/snapd/11036
/dev/loop6	65M	65M	0	100%	/snap/gtk-common-themes/1514
/dev/loop7	33M	33M	0	100%	/snap/snapd/11841
/dev/sda1	511M	4.0K	511M	1%	/boot/efi
tmpfs	494M	28K	494M	1%	/run/user/1000
/dev/ramdisk	504M	396K	478M	1%	/home/oslab/2021_DKU_OS/lab3_filesystem/mnt

```
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem#
```

3) ./create.sh 실행

```
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# ls mnt
0 1 2 3 4 5 6 7 8 9 lost+found
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# ls -l mnt/5/78
-rw-r--r-- 1 root root 8199  5월 19 13:18 mnt/5/78
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# ./apd mnt/5/78 13 5/78-13
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# ls -l mnt/5/78
-rw-r--r-- 1 root root 49160  5월 19 13:19 mnt/5/78
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem#
```

create.sh 스크립트를 수행한 후 0~9번 디렉토리가 생성된 것을 확인하였습니다.

제 학번은 32170578이므로 ls -l mnt/5/78에 apd를 통해 불락을 추가하여 파일크기가 49160이 된 것을 확인하였습니다.

1. Root Directory에서 찾을 File이 속한 Directory의 Inode Number 찾기

Super Block 영역 분석

첫번째 Group Descriptor Table은 램디스크의 1블록 이후부터 시작하므로 xxd 명령어를 통해 0x1000(16^3B = 4KB) 부터 256byte를 읽었습니다.

```
oslab@oslab-DKU:~/2021_DKU_OS/lab3_filesystem$ sudo xxd -g 4 -l 0x100 -s 0x400 /dev/ramdisk
00000400: 00800000 00000200 99190000 8ff70100 .....
00000410: f57f0000 00000000 02000000 02000000 .....
00000420: 00800000 00800000 00200000 b06aa060 .....j.
00000430: b06aa060 0100ffff 53ef0000 01000000 .j. ....S.....
00000440: 7f6aa060 00000000 00000000 01000000 .j. ....
00000450: 00000000 0b000000 00010000 38000000 .....8...
00000460: 02000000 03000000 2858a3e5 dea84cb0 .....(X....L.
00000470: ada5e077 f1b82d14 00000000 00000000 ...w...-.....
00000480: 00000000 00000000 2f686f6d 652f6f73 ...../home/os
00000490: 6c61622f 32303231 5f444b55 5f4f532f lab/2021_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 8d4af650 .....J.P
000004f0: 87cb4e2d b6baf755 0eb6023e 01000000 ..N-...U...>....
```

- inode count : 0x8000
- block count : 0x20000
- log block size : 0x2
- blocks per group : 0x8000
- inodes per group : 0x2000
- block group number : 0x0

Group Descriptor Table 영역 분석

```
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x100 -s 0x1000 /dev/ramdisk
00001000: 21000000 22000000 23000000 d77d2b1f !..."...#....}+.
00001010: 04000400 00000000 00000000 00000000 .....
00001020: 21800000 22800000 23800000 dd7dd11e !..."...#....}..
00001030: 03000400 00000000 00000000 00000000 .....
00001040: 00000100 01000100 02000100 3a72361f .....:r6.
00001050: 02000400 00000000 00000000 00000000 .....
00001060: 21800100 22800100 23800100 dd7dd11e !..."...#....}..
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 : 0x23 블록부터 시작

Ext2에서 Root inode number는 2번이고, inodes per group의 값은 0x2000이므로

- root's block group : (2-1) / 0x2000 = 0
- root's index : (2-1) % 0x2000 = 1
- root는 0번 Block Group의 Inode Table의 1번째(index=1)에 위치한다

Inode Table 영역 분석

```
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x1000 -s 0x23000 /dev/ramdisk
00023000: 00000000 00000000 4891a460 4891a460 .....H..`H..`
00023010: 4891a460 00000000 00000000 00000000 H..`.....
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 a391a460 9991a460 .A.....
00023110: 9991a460 00000000 00000d00 08000000 ...`.....
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 bc7e14dc bc7e14dc 44827d1a .....~...~..D.}.
00023190: 4891a460 00000000 00000000 00000000 H..`.....
000231a0: 00000000 00000000 00000000 00000000 .....
000231b0: 00000000 00000000 00000000 00000000 .....
000231c0: 00000000 00000000 00000000 00000000 .....
000231d0: 00000000 00000000 00000000 00000000 .....
000231e0: 00000000 00000000 00000000 00000000 .....
000231f0: 00000000 00000000 00000000 00000000 .....
```

Inode의 크기는 0x100 byte 이고, Root가 속한 Block Group은 0, Index는 1이므로 0번 Block Group의 Inode Table 0x100 부터 Root Inode이다.

```
00023100: ed410000 00100000 a391a460 9991a460 .A.....
00023110: 9991a460 00000000 00000d00 08000000 ...`.....
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 bc7e14dc bc7e14dc 44827d1a .....~...~..D.}.
00023190: 4891a460 00000000 00000000 00000000 H..`.....
000231a0: 00000000 00000000 00000000 00000000 .....
000231b0: 00000000 00000000 00000000 00000000 .....
000231c0: 00000000 00000000 00000000 00000000 .....
000231d0: 00000000 00000000 00000000 00000000 .....
000231e0: 00000000 00000000 00000000 00000000 .....
000231f0: 00000000 00000000 00000000 00000000 .....
```

- mode : 0x41ed = 0100000 111 101 101 = d rwx r-x r-x
- block pointer : 0x223

Data 영역 분석

```
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x1000 -s 0x223000 /dev/ramdisk
00223000: 02000000 0c000102 2e000000 02000000 .....
00223010: 0c000202 2e2e0000 0b000000 14000a02 .....
00223020: 6c6f7374 2b666f75 6e640000 01400000 lost+found...@..
00223030: 0c000102 30000000 01200000 0c000102 ....0....
00223040: 31000000 01600000 0c000102 32000000 1.....2...
00223050: 0c000000 0c000102 33000000 71000000 .....3...q...
00223060: 0c000102 34000000 66600000 0c000102 ....4...f`.....
00223070: 35000000 66200000 0c000102 36000000 5...f .....6...
00223080: 66400000 0c000102 37000000 cb600000 f@.....7.....
00223090: 0c000102 38000000 cb200000 680f0102 ....8....h...
002230a0: 39000000 00000000 00000000 00000000 9.....
```


- 5번 Directory
 - inode number : 0x6066
 - file type : 0x2 = directory

```
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem/mnt# stat 5
File: 5
Size: 4096          Blocks: 8          IO Block: 4096   directory
Device: fc00h/64512d Inode: 24678        Links: 2
Access: (0755/drwxr-xr-x)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2021-05-19 16:37:55.866645715 +0900
Modify: 2021-05-19 13:18:33.911082638 +0900
Change: 2021-05-19 13:18:33.911082638 +0900
Birth: -
```

stat명령어를 통해 분석을 통해 얻은 5번디렉토리의 inode number와 file type이 일치하는지 확인하였습니다.

0x6066은 10진수로 24678이므로 일치합니다.

- 속한 Block Group : $(0x6066 - 1) / 0x2000 = 3$ 번 Block Group
- Inode Table Index : $(0x6066 - 1) \% 0x2000 = 65$
- 5번 디렉터리의 Inode는 3번 Block Group의 Inode Table에서 65번째(index=65)에 위치함을 알 수 있습니다.

2. Diretory가 속한 Block Group의 Inode Table에서 디렉토리의 Inode찾기

Inode의 크기는 0x100 byte 이고, 5번 디렉터리가 속한 Block Group은 3, Index는 65이므로 3번 Block Group의 Inode Table 0x6500 부터 5번 디렉터리의 inode임을 알 수 있습니다.

```
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x100 -s 0x1000 /dev/ramdisk
00001000: 21000000 22000000 23000000 d77d2b1f  !...".#....}+.
00001010: 04000400 00000000 00000000 00000000  .....
00001020: 21800000 22800000 23800000 dd7dd11e  !...".#....}..
00001030: 03000400 00000000 00000000 00000000  .....
00001040: 00000100 01000100 02000100 3a72361f  .....:r6.
00001050: 02000400 00000000 00000000 00000000  .....
00001060: 21800100 22800100 23800100 dd7dd11e  !...".#....}..
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  .....
```

Group Descriptor Table 영역을 분석하여 3번 Block Group의 Inode table의 시작부분이 0x18023임을 알 수 있습니다.

inode table의 65번째 인덱스의 inode번호는 $0x18023000 + 0x6500 = 0x18029500$ 입니다.

따라서 0x18029500에 해당하는 위치를 xxd를 통해 조회하면 다음과 같습니다.

```

root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x100 -s 0x18029500 /dev/ramdisk
18029500: ed410000 00100000 53c0a460 9991a460 .A.....S...`
18029510: 9991a460 00000000 00000200 08000000 ...`.....
18029520: 00000000 65000000 07020100 00000000 ....e.....
18029530: 00000000 00000000 00000000 00000000 .....
18029540: 00000000 00000000 00000000 00000000 .....
18029550: 00000000 00000000 00000000 00000000 .....
18029560: 00000000 2be4bbeb 00000000 00000000 ....+.....
18029570: 00000000 00000000 00000000 00000000 .....
18029580: 20000000 381238d9 381238d9 4cdb9fce ...8.8.8.L...
18029590: 9991a460 381238d9 00000000 00000000 ...`8.....
180295a0: 00000000 00000000 00000000 00000000 .....
180295b0: 00000000 00000000 00000000 00000000 .....
180295c0: 00000000 00000000 00000000 00000000 .....
180295d0: 00000000 00000000 00000000 00000000 .....
180295e0: 00000000 00000000 00000000 00000000 .....
180295f0: 00000000 00000000 00000000 00000000 .....

```

- mode : 0x41ed = 0100000 111 101 101 = d rwx r-x r-x
- block pointer : 0x10207
5번 디렉토리의 inode가 가리키고있는 data block은 0x10207번 입니다.

3. 찾은 Directory Entry에서 File의 Inode Number찾기

각 블록의 크기는 0x1000(4Kbyte)이므로 0x10207에 0x1000을 곱한 값을 통해 5번 디렉토리의 데이터 블록을 조회합니다.

```

root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x400 -s 0x10207000 /dev/ramdisk
10207350: 3b380000 acb00000 0c000201 3b390000 b8... ..b9..
10207360: ad600000 0c000201 37300000 ae600000 .`.....70...`
10207370: 0c000201 37310000 af600000 0c000201 ....71...`.....
10207380: 37320000 b0600000 0c000201 37330000 72...`.....73..
10207390: b1600000 0c000201 37340000 b2600000 .`.....74...`..
102073a0: 0c000201 37350000 b3600000 0c000201 ....75...`.....
102073b0: 37360000 b4600000 0c000201 37370000 76...`.....77..
102073c0: b5600000 0c000201 37380000 b6600000 .`.....78...`..
102073d0: 0c000201 37390000 b7600000 0c000201 ....79...`.....
102073e0: 38300000 b8600000 0c000201 38310000 80...`.....81

```

data영역을 조회하면 78번 파일의 inode number가 60b5임을 확인할 수 있습니다.

해당 inode를 분석하면

- 속한 Block Group : $(0x60b5 - 1) / 0x2000 = 3$ 번 Block Group
- Inode Table Index : $(0x60b5 - 1) \% 0x2000 = b4$

inode table의 b4번 인덱스의 inode번호는 $0x18023000 + 0xb400 = 0x1802E400$ 입니다.

따라서 0x18029500에 해당하는 위치를 xxd를 통해 조회하면 다음과 같습니다.

```

root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x400 -s 0x1802e400 /dev/ramdisk
1802e400: a4810000 08c00000 df19a560 c291a460 .....`...
1802e410: c291a460 00000000 00000100 28000000 ...`.....(...
1802e420: 00000000 01000000 42060100 42080100 .....B...B...
1802e430: 600e0100 00000000 00000000 00000000 .....
1802e440: 00000000 00000000 00000000 00000000 .....
1802e450: 00000000 00000000 00000000 0c020100 00000000 .....
1802e460: 00000000 276537f6 00000000 00000000 ....'e7.....
1802e470: 00000000 00000000 00000000 00000000 .....

```

78번 파일의 inode가 가리키고있는 데이터블럭이 0x10642, 0x10842, 0x10E60 임을 알 수 있습니다.
이를 통해 각 데이터 블록을 조회하면 다음과 같이 파일의 블록을 확인할 수 있습니다.

```

root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x100 -s 0x10642000 /dev/ramdisk
10642000: 352f3738 2d310a00 00000000 00000000 5/78-1.....
10642010: 00000000 00000000 00000000 00000000 .....
10642020: 00000000 00000000 00000000 00000000 .....

```

```

root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x100 -s 0x10842000 /dev/ramdisk
10842000: 352f3738 2d320a00 00000000 00000000 5/78-2.....
10842010: 00000000 00000000 00000000 00000000 .....
10842020: 00000000 00000000 00000000 00000000 .....
10842030: 00000000 00000000 00000000 00000000 .....
10842040: 00000000 00000000 00000000 00000000 .....

root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# xxd -g 4 -l 0x100 -s 0x10e60000 /dev/ramdisk
10e60000: 352f3738 2d330a00 00000000 00000000 5/78-3.....
10e60010: 00000000 00000000 00000000 00000000 .....

```

[EXT2 file system 실습 Bouns]

```

static struct dentry *ext2_mount(struct file_system_type *fs_type,
int flags, const char *dev_name, void *data)
{
    printk(KERN_ERR "os_ext2 : San Kim OS Lab3\n"); //print "os_ext2 : Your Name OS Lab3"
    return mount_bdev(fs_type, flags, dev_name, data, ext2_fill_super);
}

```

vim /mount

```

root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem/os_ext2# insmod os_ext2.ko
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem/os_ext2# lsmod |grep os_ext2
os_ext2              73728  0
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem/os_ext2# cd ..
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# mkfs.ext2 /dev/ramdisk
mke2fs 1.45.5 (07-Jan-2020)
Creating filesystem with 131072 4k blocks and 32768 inodes
Filesystem UUID: 372746ba-6f83-4a3e-bb0b-336c5e9b7bf5
Superblock backups stored on blocks:
    32768, 98304

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting information: done

root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# mount -t os_ext2 /dev/ramdisk ./mnt
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem# dmesg | grep os_ext2
[ 4332.272765] os_ext2 : San Kim OS Lab3
[ 4332.272998] os_ext2 filesystem being mounted at /home/oslab/2021_DKU_OS/lab3_filesystem/mnt supports timestamps until 2038 (0x7fffffff)
root@oslab-DKU:/home/oslab/2021_DKU_OS/lab3_filesystem#

```

Discussion

PPT를 따라하면서 막혔던 곳이 3번 Block Group을 찾을때였습니다.

처음에는 5번 디렉토리의 inode를 확인하고 0x6066이므로 3번 Blcok Group을 찾기위해서 0번 그룹의 inode table에 Block Group의 size만큼 더해주면 될거라 생각을 했습니다. 0x23000에 group당 block의 개수가 0x8000이므로 blcok의 size(0x1000)을 곱하여 0번그룹의 inode table위치를 더하면 1번 그룹의 inode table을 찾을 수 있을 것이라 생각을 했는데 이상한 값들이 나와서 당황했습니다.

그래서 Ext2 File system내용을 다시 찾아보면서 Group Descriptor Table 영역에 다른 그룹들의 정보가 들어있을 것이라는 생각이 들었고 Group Descriptor Table 그림을 참고하면서 다음 그룹들의 inode table위치를 확인할 수 있었습니다.

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f		
00	block bitmap				inode bitmap				inode table				free blk cnt		free ino cnt			
10	used dir cnt		padding		reserved (padding)													

3번 그룹의 inode table의 위치를 확인한 후 파일까지 찾아가는 과정은 수월했습니다.

평소에 자주 사용하던 탐색기를 통해 기존의 파일을 열람하는것은 이미 상당히 추상화 되어 있기 때문에 파일시스템에 대한 이해가 필요 없었습니다. 하지만 이번 Lab3과제를 진행하면서 파일시스템(특히 Ext2 파일시스템)의 구조가 어떻게 이루어져 있는지 조금더 구체적으로 확인할 수 있었고, 힘들여 샅샅이 시간만 있었던 만큼 Ext2파일시스템의 구조와 원리에 대해 조금더 오래 기억할 수 있을것 같습니다.

Bonus과제에서는 단순히 모듈을 수정하는 과정이었지만 어떤 함수를 수정해야할지 찾는 과정에서 파일 시스템 생성 모듈이 어떻게 코드로 구현되어있는지 접할 수 있었습니다. 함수들이 사용용도와 목적에 맞는 이름들을 가지고 있었고 인자의 순서도 어떤 정해진 위치에 따라 정갈하게 구현되어있는것이 제가 앞으로 어떻게 코딩을 해 나가야 할 지 방향점을 시사하고 있는것 같았습니다.

이번 과제를 하면서 코딩을 따로 하는 부분은 없었지만 리눅스 파일 시스템의 내부구조를 직접 눈으로 확인하고 삽질하면서 좀더 오래 기억할 수 있을 것 같고, 특히나 제가보았던 모듈의 코드부분은 자주 참고하게 될 것 같습니다.

추가적으로 커널 모듈을 다루는 과정에 흥미를 느껴서 간단한 모듈을 생성하고 적재해 보았습니다.

```
#include <linux/init.h>
#include <linux/module.h>
MODULE_LICENSE("Dual BSD/GPL"); /* optional */

static int __init hello_init(void)
{
    printk(KERN_NOTICE "Hello, world #2\n");
    return 0; /* return zero on successful loading */
}

static void __exit hello_exit(void)
{
    printk(KERN_NOTICE "Goodbye\n");
}

module_init(hello_init);
module_exit(hello_exit);
```

위와 같은 코드를 통해 모듈을 작성하고, dmesg명령어를 통해 모듈 동작을 확인하였습니다.

```
root@oslab-DKU:/home/oslab/workspace# sudo rmmod mymod
root@oslab-DKU:/home/oslab/workspace# dmesg | tail
[16690.747416] raid6: using algorithm sse2x2 gen() 16137 MB/s
[16690.747416] raid6: .... xor() 11164 MB/s, rmw enabled
[16690.747417] raid6: using ssse3x2 recovery algorithm
[16690.751738] xor: measuring software checksum speed
[16691.145735]    prefetch64-sse: 53911.000 MB/sec
[16691.253541]    generic_sse: 49870.000 MB/sec
[16691.253542] xor: using function: prefetch64-sse (53911.000 MB/sec)
[16691.284269] Btrfs loaded, crc32c=crc32c-intel
[17384.379917] Hello, World #2
[17404.584994] Goodbye
```