운영체제 개론(Team Report3)

Lab3 File System

문제:

Analyze FAT file system internal

목차:

- 1. 역할분배
- 2. 개요
- 3. 요구사항 분석
- 4. 해결과정
- 5. 파일 사이즈를 3KB, 20KB로 바꿔보기
- 6. 결과(결론)

과목명	운영체제
담당교수님	최 종 무 교수님
	황 준 일(32131766)
이름(학번)	박 은 영(32151839)
	박 유 림(32161756)
제출일	2018.06.07

1. 역할분배

1) 황준일 : 33.txt 의 Directory Entry 분석

2) 박은영: hex.txt 파일에서 33.txt 파일의 위치 탐색

3) 박유림 : FAT파일 시스템을 생성하고 hex.txt 파일에 정보를 저장하는 과정 분석

2. 개요

- FAT: File Allocation Table의 약어로 파일 시스템 종류 중 하나이다.
- ▶ 구조: BS(Boot Sector), FAT1/2, Root Directory, Data Area
 - ✔ BS(Boot Sector): FAT type, FAT 사이즈, Root Directory 사이즈
 - ✔ Root Directory: 시작 Cluster의 정보, 파일의 사이즈 정보
- ✓ 이 중 FAT은 Bitmap(Sector가 사용 중인지 아닌지)과 **다음 Cluster(disk block)의 위치를 알려주는** inode 의 기능과 같은 역할을 한다.
- ▶ 응용: 소용량 메모리 카드나 플로피 디스크 등의 파일 시스템으로 사용되고 있다.

3. 요구사항 분석

- ramdisk 상에서 FAT 파일 시스템을 생성하고 주어진 파일이 있는 Cluster를 추적하는 과정을 직접 분석하라

a. # is # make

```
root@osLab:/home/oslab/DKU OS LAB/lab3 fs# ls
Makefile create.sh.x mnt ramdisk.c
root@osLab:/home/oslab/DKU OS LAB/lab3 fs# make
make -C /lib/modules/4.13.0-43-generic/build SUBDIRS=/home/oslab/DKU OS LAB/lab3
_fs V= modules
make[1]: 디렉터리 '/usr/src/linux-headers-4.13.0-43-generic' 들어감
  CC [M] /home/oslab/DKU OS LAB/lab3 fs/ramdisk.o
  Building modules, stage 2.
  MODPOST 1 modules
          /home/oslab/DKU OS LAB/lab3 fs/ramdisk.mod.o
LD [M] /home/oslab/DKU_OS_LAB/lab3_fs/ramdisk.ko
make[1]: 디렉터리 '/usr/src/linux-headers-4.13.0-43-generic' 나감
root@osLab:/home/oslab/OKU_OS_LAB/lab3_fs# ls
Makefile
                create.sh.x modules.order ramdisk.ko
                                                              ramdisk.mod.o
Module.symvers ant
                              ramdisk.c
                                              ramdisk.mod.c camdisk.o.
```

b. # insmod ramdisk.ko 모듈이 커널에 동적으로 삽입된다.

root@osLab:/home/oslab/DKU_OS_LAB/lab3_fs# insmod ramdisk.ko

c. # lsmod

모듈이 제대로 올라갔는지 확인한다.

```
root@osLab:/home/oslab/DKU_OS_LAB/lab3_fs# lsmod
Module Size Used by
ramdisk 16384 0
nls_utf8 16384 1
isofs 40960 1
```

d. # dd if=/dev/zero of=/dev/ramdisk dd 명령어로 불필요한 데이터를 초기화한다.

```
root@osLab:/home/oslab/DKU_OS_LAB/lab3_fs# dd if=/dev/zero of=/dev/ramdisk
dd: '/dev/ramdisk' 로 쓰는 중: 장치에 담은 공간이 없음
2097153+0 레코드 들어옴
2097152+0 레코드 나감
1073741824 bytes (1.1 GB, 1.0 GiB) copied, <u>6</u>.32208 s, 170 MB/s
```

e. # mkfs.fat -F 32 /dev/ramdisk

File system FAT32로 초기화 및 마운트 한다.

```
root@osLab:/home/oslab/DKU_OS_LAB/lab3_fs# mkfs.fat -F 32 /dev/ramdisk
mkfs.fat 3.0.28 (2015-05-16)
root@osLab:/home/oslab/DKU_OS_LAB/lab3_fs# mount /dev/ramdisk mnt
```

f. # ./create.sh.x

스크립트 실행을 위해 모든 권한을 허용하도록 설정하였다.

```
root@osLab:/home/oslab/DKU_OS_LAB/lab3_fs# ./create.sh.x
bash: ./create.sh.x: 허가 거부
```

root@ośLab:/home/oślab/DKU_OS_LAB/láb3_fs# chmod a+rwx ./create.sh.x

```
root@osLab:/home/oslab/DKU_OS_LAB/lab3_fs# ./create.sh.x
1+0 레코드 들어옴
1+0 레코드 나감
8192 bytes (8.2 kB, 8.0 KiB) copied, 0.000294851 s, 27.8 MB/s
1+0 레코드 들어옴
:1+0 레코드 나감
8192 bytes (8.2 kB, 8.0 KiB) copied, 0.000376044 s, 21.8 MB/s
1+0 레코드 들어옴
1+0 레코드 나감
8192 bytes (8.2 kB, 8.0 KiB) copied, 0.000414918 s, 19.7 MB/s
```

g. # xxd -a -g 1 -s+0x00 /dev/ramdisk > hex.txt xxd 출력 파일을 작성한다.

root@osLab:/home/oslab/DKU OS LAB/lab3 fs# xxd -a -g 1 -s+0x00 /dev/ramdisk > hex.txt

```
00000c00: eb58906d 6b66732e 66617400 02082000 .X.mkfs.fat....
00000c10: 02000000 00f80000 10000400 00000000
                                          . . . . . . . . . . . . . . . . . . .
00000c20: 00002000 fc070000 00000000 02000000
00000c30: 01000600 00000000 00000000 00000000
00000c40: 80002990 17d8094e 4f204e41 4d452020
                                          ..)...NO NAME
00000c50: 20204641 54333220 20200e1f be777cac
                                           FAT32 ...W .
                                          ".t.V.....^..2
00000c60: 22c0740b 56b40ebb 0700cd10 5eebf032
00000c70: e4cd16cd 19ebfe54 68697320 6973206e ......This is n
00000c80: 6f742061 20626f6f 7461626c 65206469 ot a bootable di
00000c90: 736b2e20 20506c65 61736520 696e7365 sk. Please inse
                                                            Boot
00000ca0: 72742061 20626f6f 7461626c 6520666c rt a bootable fl
00000cb0: 6f707079 20616e64 0d0a7072 65737320 oppy and..press
00000cc0: 616e7920 6b657920 746f2074 72792061
                                          any key to try a
00000cd0: 6761696e 202e2e2e 200d0a00 00000000
                                          gain ... ......
00000ce0: 00000000 00000000 00000000 00000000
                                          00000df0: 00000000 00000000 00000000 000055aa
                                          .....U.
00000e00: 00000000 00000000 00000000 000<u>0</u>00000
00004000: f8ffff0f ffffff0f f8ffff0f 00000000
00004010: 00000000 ffffff0f ffffff0f ffffff0f
00004020: ffffff0f ffffff0f ffffff0f
00004030: ffffff0f ffffff0f ffffff0f
00004040: ffffff0f ffffff0f ffffff0f 00000000
                                          . . . . . . . . . . . . . . . . . . .
......
000040c0: 00000000 32000000 ffffff0f 34000000
                                          ....2......4...
000040d0: ffffff0f 36000000 ffffff0f 38000000
                                          . . . . 6 . . . . . . . 8 . . .
000040e0: ffffff0f 3a000000 ffffff0f 3c000000
                                          ....:...........
000040f0: ffffff0f 3e000000 ffffff0f 40000000
                                          00004100: ffffff0f 42000000 ffffff0f 44000000
00004110: ffffff0f 46000000 ffffff0f 48000000
00004120: ffffff0f 4a000000 ffffff0f 4c000000
                                          ....J.....L...
00004130: ffffff0f 4e000000 ffffff0f 00000000
                                          ....N.......
```

[create.sh.x 실행 후 hex.txt]

빨간색 글씨는 최종에서 제거해야할 메모입니다!

준비하여 보고서에 추가하면 좋을 것 같은 것!

- -Boot Sector, FAT, Root Directory가 어디인지 간단하게 캡쳐 => 0
- create.sh.x 하기 전 hex.txt와 한 후의 hex.txt 살짝 비교

2) 33.txt Tracking 과정

- ▶ Tracking 방법: Root Directory에서의 시작 클러스터 번호와 같은 정보를 이용해 FAT에서 해당 파일 또는 디렉토리의 클러스터를 얼마만큼 사용중인지에 대한 정보를 추적한다. 또한 이를 통해 User Data에서 파일의데이터 정보를 알아낸다.
- ▶시작 클러스터는 Root Directory에서 20-21(High-order address of first cluster), 26-27(Low-order address of first cluster)위치의 값을 합친 것이다.
- ▶ Root Directory에서 11번째 바이트 값이 0x10일 경우 해당 파일은 디렉토리이다.
- ▶ Root Directory는 두 줄씩 묶어 하나의 Entry로 생각한다.
- ▶이 파일 시스템은 클러스터가 8개의 sectors를 가지고 있으며 한 sector당 크기는 512B이다.
- ▶ create.sh.x 실행 후 hex.txt에서

(FAT 이전: Boot Sector)

(0x004000주소~: FAT)

(0x201000주소~: User Data)

(유의: 특이하게도 이번 과제에서 파일 시스템 환경은 root directory가 user data 안에 들어가 있는 구조임)

① LAB3

00203060: 4c414233 20202020 202020**10** 00003337 LAB3 ...37 00203070: c44cc44c **0000**3337 c44c**0600** 00000000 .L.L..37.L.....

- ▶시작 클러스터: 0000(First Cluster High 2Byte) 0006(First Cluster Low 2Byte) => 00000006
- ▶ FAT의 00000006 인덱스 추적: 0fffffff(=> 더 이상 연결된 클러스터 없음! 여기서 끝) 00004010: 00000000 ffffff0f ffffff0f
- ▶ Attribute(속성): 10, Directory
- ▶ File Size: 00000000, Directory
- ▶ 0x201000(User data 시작 주소) + 0x6 * 0x1000(클러스터 한 개 크기) = 0x207000

계산의 결과로 나온 주소 0x207000으로 가서 어떤 파일들이 있는지 알아낸다.

00207000: 2e202020 20202020 20202010 0000333737

00207010: c44cc44c 00003337 c44c0600 00000000 .L.L..37.L.....

00207030: c44cc44c 00003337 c44c0000 00000000 .L.L..37.L.....

00207040: 41640072 00690076 0065000f 00137200 Ad.r.i.v.e...r.

00207050: 0000ffff fffffff ffff0000 ffffffff

00207060: 44524956 45522020 20202010 00003337 DRIVER ...37

00207070: c44cc44c 00003337 c44c0a00 00000000 .L.L..37.L.....

00207080: 41650073 00630061 0070000f 00656500 Ae.s.c.a.p...ee.

00207090: 0000ffff ffffffff ffff0000 ffffffff

002070a0: 45534341 50452020 20202010 00003337 ESCAPE ...37

002070b0: c44cc44c 00003337 c44c0b00 00000000 .L.L..37.L.....

002070c0: 41660069 006c0065 0073000f 006b7900 Af.i.l.e.s...ky.

002070d0: 73007400 65006d00 00000000 ffffffff s.t.e.m......

002070e0: 46494c45 53597e31 20202010 00003337 FILESY~1 ...37

002070f0: c44cc44c 00003337 c44c0c00 00000000 .L.L..37.L.....

=> Driver, Escape, filesystem 이라는 디렉토리가 LAB3 디렉토리 하위에 존재함을 알 수 있다.

```
② LAB3 > Driver
  00207060: 44524956 45522020 20202010 00003337 DRIVER
  00207070: c44cc44c 00003337 c44c0a00 00000000 .L.L..37.L.....
▶시작 클러스터: 0000(First Cluster High 2Byte) 000a(First Cluster Low 2Byte) => 0000000a
▶ FAT의 0000000a 인덱스 추적: 0fffffff(=> 더 이상 연결된 클러스터 없음! 여기서 끝)
  00004020: ffffff0f ffffff0f ffffff0f ......
▶ Attribute(속성): 10, Directory
► File Size: 00000000, Directory
▶ 0x201000(User data 시작 주소) + 0xa * 0x1000(클러스터 한 개 크기) = 0x20b000
계산의 결과로 나온 주소 0x20b000으로 가서 어떤 파일들이 있는지 알아낸다.
0020b000: 2e202020 20202020 20202010 00003337 .
                                                   ...37
0020b010: c44cc44c 00003337 c44c0a00 00000000 .L.L..37.L.....
0020b020: 2e2e2020 20202020 20202010 00003337 ..
                                                   ...37
0020b030: c44cc44c 00003337 c44c0600 00000000 .L.L..37.L.....
0020b040: 41440072 00610067 006f000f 00b56e00 AD.r.a.g.o...n.
0020b050: 0000ffff fffffff ffff0000 ffffffff ......
0020b060: 44524147 4f4e2020 20202010 00003337 DRAGON ...37
0020b070: c44cc44c 00003337 c44c1000 00000000 .L.L..37.L.....
=> DRAGON 이라는 디렉토리가 Driver 디렉토리 하위에 존재함을 알 수 있다.
③ LAB3 > Driver > DRAGON
  0020b060: 44524147 4f4e2020 20202010 00003337 DRAGON
  0020b070: c44cc44c 00003337 c44c1000 00000000 .L.L..37.L.....
▶시작 클러스터: 0000(First Cluster High 2Byte) 0010(First Cluster Low 2Byte) => 00000010
▶ FAT의 00000010 인덱스 추적: 0fffffff(=> 더 이상 연결된 클러스터 없음! 여기서 끝)
  00004040: ffffff0f ffffff0f ffffff0f 00000000 ......
▶ Attribute(속성): 10, Directory
▶ File Size: 00000000, Directory
▶ 0x201000(User data 시작 주소) + 0x10 * 0x1000(클러스터 한 개 크기) = 0x211000
계산의 결과로 나온 주소 0x211000으로 가서 어떤 파일들이 있는지 알아낸다.
00211000: 2e202020 20202020 20202010 00003337 .
00211010: c44cc44c 00003337 c44c1000 00000000 .L.L..37.L.....
00211020: 2e2e2020 20202020 20202010 00003337 ..
                                                    ...37
00211030: c44cc44c 00003337 c44c0a00 00000000 .L.L..37.L.....
00211040: 41330033 002e0074 0078000f 005b7400 A3.3...t.x...[t.
00211050: 0000ffff fffffff ffff0000 ffffffff ......
00211060: 33332020 20202020 54585420 00003337 33 TXT ..37
00211070: c44cc44c 00003337 c44c3d00 00200000 .L.L..37.L=.. ..
00211080: 41390039 002e0074 0078000f 00e67400 A9.9...t.x....t.
00211090: 0000ffff fffffff ffff0000 ffffffff ......
002110a0: 39392020 20202020 54585420 00003337 99
002110b0: c44cc44c 00003337 c44c3f00 00200000 .L.L..37.L?.. ..
002110c0: 41310035 00310035 002e000f 009e7400 A1.5.1.5.....t.
002110d0: 78007400 0000ffff ffff0000 ffffffff x.t.....
002110e0: 31353135 20202020 54585420 00003337 1515
002110f0: c44cc44c 00003337 c44c4100 00200000 .L.L..37.LA.. ..
=> 33.txt / 99.txt / 1515.txt 파일이 DRAGON 디렉토리 하위에 존재함을 알 수 있다.
```

4 LAB3 > Driver > DRAGON > 33.txt

00211060: 33332020 20202020 545854**20** 00003337 33 TXT ..37 00211070: c44cc44c **0000**3337 c44c**3d00** 00200000 .L.L..37.L=.. ..

- ▶시작 클러스터: 0000(First Cluster High 2Byte) 003d(First Cluster Low 2Byte) => 0000003d
- ▶FAT의 0000003d 인덱스 추적: 3e000000(3e번지로 가라) -> 0fffffff(=> 더 이상 연결된 클러스터 없음! 여기서 끝)

000040f0: ffffff0f **3e000000 ffffff0f** 40000000>.....@...

- ▶ Attribute(속성): 20, Archive file
- ► File Size: 00002000, Archive file
- ▶ 0x201000(User data 시작 주소) + 0x3d * 0x1000(클러스터 한 개 크기) = **0x23e000**

계산의 결과로 나온 주소 0x23e000으로 가서 어떤 정보가 있는지 알아낸다.

0023e000: 0572e6fb 83a08e9b 4c685975 1795cb23 .r....LhYu...#

=> 0x23e000부터 0x240000주소까지 33.txt의 데이터가 있다고 볼 수 있다.(클러스터 3d, 3e번째 총 2개 사용 중)

<	hex.txt>	33.t	xt S	Size	: 4	KB										
L	00203060: 00203070: 00004010:	₫ 4	c ċ	4c 6	0 00	∂ dc∙	· 22 · c	3 40	96	00 O	0 00	99 99	.L.		d.¶ L	<us><user data=""></user></us>
→	00207060: 00207070: 00004020:	₫ 4	c ċB	4c 6	0 00	ð"dc∙	· 22 · c	3 4c	0a	00 O	0 00	99 99	.L.	VER L	d.¶ L	파일의 생성, 수정 날짜가 서로 다름을 알 수 있다.
	00211060: 00211070: 000040f0:	₫ 4	c cB	4€ €	0 00	∂ dc∙	· 22 · c	3 4c	3d	00 0	0 20	00 00	.L.	L".	XT .d." L=	파일내용의 시작위치 0x0023e000
į	0023e000:	87 8	a 36	a5 2	4 80	: f5	90 6	1 ac	l d2	d8 e	c fd	3d 0 a	a6	.\$a	=.	
 	0023fff0:	94 0	F Ь0	71 d	9 79) 5e	f8 9	a es	ed	1b 8	8 91	. 06-86	5	q.у^		
•	00004000: 00004010: 00004020: 00004030: 00004040: 00004050:	00 00 ff ff ff ff ff ff	9 00 F ff F ff F ff	00 f 0f f 0f f	f f1 f f1 f f1 f f1	f ff f ff f ff f ff	0f f 0f f 0f f 0f f	f ff f ff f ff f ff	ff ff ff ff	0f f 0f f 0f f 0f 0	f ff f ff f ff 0 00	ff 01 ff 01 ff 01 00 00	f f f			<fat> 0x3e000000 ~ 0xffffff0f 는 기존 데이터 4KB 인 33.txt 내용의 클러스터의 위치이다.</fat>
→	000040c0: 000040d0: 000040e0: 000040f0: 00004100: 00004110: 00004120: 00004130: 00004140:	ff	f ff f ff f ff f ff f ff f ff	0f 3 0f 3 0f 4 0f 4 0f 4	6 00 a 00 2 00 6 00 a 00 e 00	9 00 9 00 9 00 9 00 9 00 9 00	00 f 00 f 00 f 00 f 00 f 00 f	f ff f ff f ff f ff f ff f ff	ff ff ff ff ff ff ff	0f 3 0f 4 0f 4 0f 4 0f 4 0f 4	8 06 c 06 0 06 4 06 8 06 c 06	99 99 99 99 99 99 99 99 99 99 99 99 99	9 9 9 9 9	.6 .: . 3 .B	@ D H	
<	hex.txt>	33.t	xt S	Size	: 3	KB										
	- 00203060: 00203070: 00004010:	₫ 40	с4	4c 0	9 00	e6 ∙ 7	75 · c4	4c	96 Q	9 00	99 E	99 99	.L.L.	L		<user data=""></user>
•	• 00207060: [00207070: 00004020:	₫ 40	с 4	4c 0	9 00	e6 · 7	75 · c4	4c	0a 0	9 00	99 8	99 99	DRIVER	0 .L	.d.[]	파일의 생성, 수정 날짜가 서로 다름을 알 수 있다.
•	 00211060: 00211070:									9 00	64 6					
							70.04	4c	75 0				33 .L.L.	TXT v.Lu.		파일내용의 시작위치가 0x00276000으로 바뀌었다.
•	00276000: 	03 12	2 72	41 e	≥ 8f				_	9 66	ଡିପ ହି	99 99	.L.L.	v.Lu.		
•	00276000: 00004000: 00004010: 00004020:	f8 ff	F ff 9 9 0	0f f 00 f	f ff f ff	eb 9	9d 5d 0f f8 0f ff	26 ff ff	d2 9	9 00 c 8c f 00 f ff	98 c	9 28 9 00 9 00 Ff 0f	.L.L.	v.Lu.	(0x00276000으로 바뀌었다. 파일을 수정하였지만 원래 위치한 내용은 없어지고 UserData의 마지막
•	- 00004000: 00004010:	f8 ff 00 00 ff ff ff ff	F ff 9 00 F ff F ff	0f f 00 f 0f f 0f f	f ff f ff f ff f ff	eb 9	9d 5d 0f f8 0f ff 0f ff 0f ff	26 ff ff ff ff ff	ff 0. ff 0. ff 0. ff 0. ff 0.	9 00 5 8c 6 6f 6 ff 6 ff 6 00	98 c	99 28 99 28 99 00 99 00 96 06 96 06 96 06	.L.L.	y.Lu.	(0x00276000으로 바뀌었다. 파일을 수정하였지만 원래 위치한 내용은 없어지고 UserData의 마지막 부분부터 파일이 다시 쓰여졌다.
•	90004000: 90004010: 90004010: 90004030: 90004040: 900044050: * 900044060: 900044060: 900044100: 900044100:	f8 ff 60 00 ff ff ff ff 60 00 ff ff ff ff ff ff ff ff	F ff 9 00 F ff F ff F ff F ff F ff F ff F ff	0f f 00 f 0f f 0f f 00 0 00 3 0f 3 0f 3	f ff f ff f ff f ff f ff 2 00 2 00 3 00 2 00 5 00 5 00	eb 9 ff 6 f	9d 5d 0f f8 0f ff 0f ff 0f ff 00 00 00 ff 00 ff 00 ff 00 ff	26 ff f	ff 0:	6 00 00 00 00 00 00 00 00 00 00 00 00 00	98 c	29 28 29 28 29 29 29 29 29 29 29 29 29 29 29 29 29	.L.LrA.]&	4 8 @ H	0x00276000으로 바뀌었다. 파일을 수정하였지만 원래 위치한 내용은 없어지고 UserData의 마지막 부분부터 파일이 다시 쓰여졌다. <fat> 33.txt가 3KB로 수정되면서 기존에 위치한 클러스터의 내용이</fat>
•	90904090: 90904010: 90904030: 90904030: 90904040: * 909044060: 909040406: 90904060: 90904060:	f8 ff 60 00 ff ff ff ff 60 00 ff	F ff 3 00 F ff F ff	0f f f	f ff f ff f ff f ff f ff 2 00 2 00 3 00 2 00 3 00 2 00 3 00 3 00	eb 9 ff 6 ff 6 ff 6 6 6 6 6 6 6 6 6 6 6 6	9d 5d 0f f8 0f ff 0f ff 0f ff 00 ff	26 ff f	d2 9 ff 0 f	60 00 00 00 00 00 00 00 00 00 00 00 00 0	98 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	29 28 29 28 29 29 29 29 29 29 29 29 29 29 29 29 29	rA.]&	4 8 9 H	Ox00276000으로 바뀌었다. 파일을 수정하였지만 원래 위치한 내용은 없어지고 UserData의 마지막 부분부터 파일이 다시 쓰여졌다. <fat> 33.txt가 3KB로 수정되면서 기존에 위치한 클러스터의 내용이 모두 00으로 변하였고 User Data에서 시작 클러스터가 3e에서 75로 바뀌었다. FAT에서 0x000041d0 에서 시작 클러스트의</fat>

<hex.txt> 33.txt Size : 20KB

90213966: 4c 41 42 33 29 29 29 29 29 28 28 10 80 64 66 75 LAB3	4	► hex2i	OS.txt			×															
002070606: 44 52 49 56 45 52 20 20 20 20 20 10 00 64 e6 75 DRIVERd.u DRI																				7	
802070606: 44 52 49 56 45 52 20 20 20 20 20 10 00 64 e6 75 DRIVER .d.u. ### ### ### ### ### ### ### ### ###	_ E	90203070:	c4	4∈	с4	4c	99	99	eb	/5	€4	4c	96	99	99	99	99	00	.L.Lu.L	- 1	
802070700: c4 4c c4 4c 00 00 e6 75 c4 4c 00 00 00 00 00 00 00 00 00 00 00 00 00	Ĭ																				<user data=""></user>
0022170708: c4 4c c4 4c 08 08 e6 75 c4 4c 08 08 08 08 08 08 08 08 08 08 08 08 08	e	90207060:	44	52	49	56	45	52	20	20	20	20	20	10	99	64	efi	75	DRTVERd.u	1	파일의 생성. 수정
알수있다. ### 1808041060: 33 33 20 20 20 20 20 20 20 20 20 54 58 54 20 00 64 55 77 c4 4c 7월 00 00 50 77 c4 4c 7월 00 00 00 00 00 00 00 00 00 00 00 00 00																					
# 000044000: f8 ff ff ef ff	→																			- 1	
00211070: c4 4c c4 4c 00 00 05 77 c4 4c 70 00 00 05 00 00 00 1. L.L. fw.lv.p. 002170000: f8 ff ff ef ff	-																			- 1	
00211070: c4 4c c4 4c 00 00 66.77 c4 4c 70 00 00 00 1	Ιe	90211060:	33	33	20	20	20	20	20	20	54	58	54	20	00	64	66	. 77	33 TXT .dfw		
0027																					0x00277000으로 바뀌었다.
0022	!																			1	피이오 스러워 어지만 이게 이러워
# 000040000: f8 ff		aa27 5 aaa.	93	12	72	41		Я£	٥h	94	54	26	42	96	8-	98	دم.	28	nA 18 (-	
→ 800044000: f8 ff ff 0f ff ff ff 0f f8 ff ff 0f 00 00 00 00 00 00 00 00 00 00			0,5	12	,,	71		01	CD	<i>-</i>	<i>-</i>	20	u.	,,	O.C.	70	-	20			내용은 없어지고 UserData의 마시 티티티디 파이이 다니 ㅆ어져다
80804010: 80 00 00 00 ff																					무군무더 파일이 다시 쓰여졌다.
80804010: 00 00 00 00 ff ff ff ff ff ff ff ff ff																					
808040318: 96 80 80 80 11 11 11 11 11 11 11 11 11 11 11 11 11	→ 8	90004000:	f8	ff	ff	0f	ff	ff	ff	0f	f8	ff	ff	0f	00	99	99	00			~EAT>
00004030: ff	8	90004010:	99	00	90	99	ff	ff	ff	0f	ff	ff	$f \boldsymbol{f}$	0f	ff	ff	ff	0f			
000040404c																					
800004050: 00 00 00 00 00 00 00 00 00 00 00 00 0																			• • • • • • • • • • • • • • • • • • • •		
* 9000440c0: 00 00 00 32 00 00 00 ff ff ff 0f 34 00 00 00																			• • • • • • • • • • • • • • • • • • • •		모두 00으로 변하였고
00000400c0: 00 00 00 00 32 00 00 00 ff ff ff ff 0f 34 00 00 00 00 00 00 00 00 00 00 00 00 00	•		99	99	99	99	99	99	99	00	99	99	99	99	99	99	99	00			
8000440d0: ff ff ff ef 36 00 00 00 ff ff ff ef 38 00 00 0068 900040e0: ff ff ff ef 3a 00 00 00 ff ff ff ef 3c 00 00 006	ļ. "		99	99	99	99	32	99	99	99	ff	ff	ff	øғ	34	99	99	99	24		User Data에서 시작 클러스터가
0000440f0: ff																					3e에서 76로 바뀌었다. FAT에서
000040f0: ff ff ff 0f 0f 00 00 00 00 00 00 00 00	į e	00040e0:	ff	ff	ff	0 f	За	00	99	00	ff	ff	ff	0f	3с	99	99	00			0x000041d0~ 0x000041e0 에서
00004100: ff ff ff 0f 42 00 00 00 ff ff ff 0f 44 00 00 00BD 00004110: ff ff ff 0f 46 00 00 00 ff ff ff 0f 48 00 00 00FH 00004120: ff ff ff 0f 4a 00 00 00 ff ff ff 0f 4c 00 00 00BD 00004130: ff ff ff 0f 4e 00 00 0f ff ff 0f 00 00 00 00N 00004140: 00 00 00 00 00 00 00 00 00 00 00 00 0	İε	900040f0:	ff	ff	ff	0 f	99	00	99	00	90	99	99	00	40	99	99	00			
00004120: ff ff ff 0f 4a 00 00 00 ff ff ff 0f 4c 00 00 00] 00004130: ff ff ff 0f 4e 00 00 00 ff ff ff 0f 00 00 00 00N 00004140: 00 00 00 00 00 00 00 00 00 00 00 00 0	8	90004100:	ff	ff	ff	0f	42	00	99	00	ff	ff	ff	0f	44	99	99	00	BD		1164 111646
800804130: ff ff ff 0f 4e 00 00 06 ff ff ff 0f 00 00 00	0	90004110:	ff	ff	ff	0 f	46	00	99	00	ff	ff	$f \boldsymbol{f}$	0f	48	99	99	00	FH		비크 파이이 스저디머니 크리스티
90004140: 00 00 00 00 00 00 00 00 00 00 00 00 0	0	90004120:	ff	ff	ff	0 f	4a	00	99	00	ff	ff	ff	0f	4 c	99	99	00	JL		
* 33.txt 의 공대스터 999941de: 90 90 90 90 90 90 90 90 90 77.00.00.00.78.00.00.00. 999941ee: 79 90 90 90 7a 90 90 90 ff.ff.ff.of 90 90 90 90 90 90 90 90 90 90 90 90 90																					마시막 무더 파일이 쓰여졌음을
000041d0: 00 00 00 00 00 00 00 00 00 77.00.00.00.00.00.00.00.00.00.00. 000041e0: 79 00 00 00 7a 00 00 00 ff ff ff of 00 00 00 00 00 yz	•		99	99	90	99	99	99	99							99	99	00			왁인 알 수 있나.
000041e0: 79 00 00 00 7a 00 00 00 ff.ff.ff.0f 00 00 00 00 yz	- I "		l aa	aa	aa	aa	ga	99	aa							. 00	. ae	. aa			
[0000-12101 00 00 00 00 00 00 00 00 00 00 00 00																					
	L	. 0000411V	ਦਦ	90	99	99	ਦਦ	90	90	66	66	00	00	99	00	00	90	66			

_한 지막

수 있고

터의