FAT32

강대명(charsyam@naver.com)

FAT

- File Allocation Table
  - FAT Table 의 크기에 따라서 FAT12, 16, 32 등으로 구분할 수 있다.
- 0번, 1번 클러스터는 존재하지 않는다.

FAT 12, 16, 32 간단한 비교

구분	FAT12	FAT16	FAT32
사용 용도	플로피디스크(거의 안씀)	저용량 하드(거의 안씀) 2GB 이하	일반 하드(거의 안씀) 2TB 이하
클러스터 표현 비트 수	12	16	32
최대 클러스터 개수	4,084	65,524	2^28
최대 볼륨 크기	16MB	2GB	2TB
파일의 최대 크기	볼륨 크기	볼륨 크기	4GB
디렉토리당 최대 파일 수	X	65,534	65,534
루트 디렉토리 파일 개수 제한 _ 루트 디렉토리가 하나의 클러스터로 고정	있음	있음	없음

일반적인 클러스터 사이즈(현재는 보통 4k)

볼륨 크기	FAT32	NTFS
32~64MB	512	512
64~128MB	1024	512
128~256MB	2048	512
256~512MB	4096	512
512MB~1GB	4096	1024
1GB~2GB	4096	2048
4GB~8GB	4096	4096
8~16GB	8192	4096
16~32GB	16084	4096
32GB~2TB	인식가능	4096

클러스터 크기에 따른 장단점

구분	장점	단점
클러스터 크기가 작을 경우	버려지는 용량이 작다. (클러스터 슬랙이 작아진다.)	FAT 사용량이 커진다. 하나의 파일을 표현할 때, 더 많은 FAT Table이 필요하다.
클러스터 크기가 클 경우	FAT 영역의 사용량이 줄어든다.	버려지는 영역(슬랙)이 크다.

FAT32 Volume 추상 Layout

- FAT 볼륨은 크게 VBR, FAT, Data Blocks 로 이루어진다.

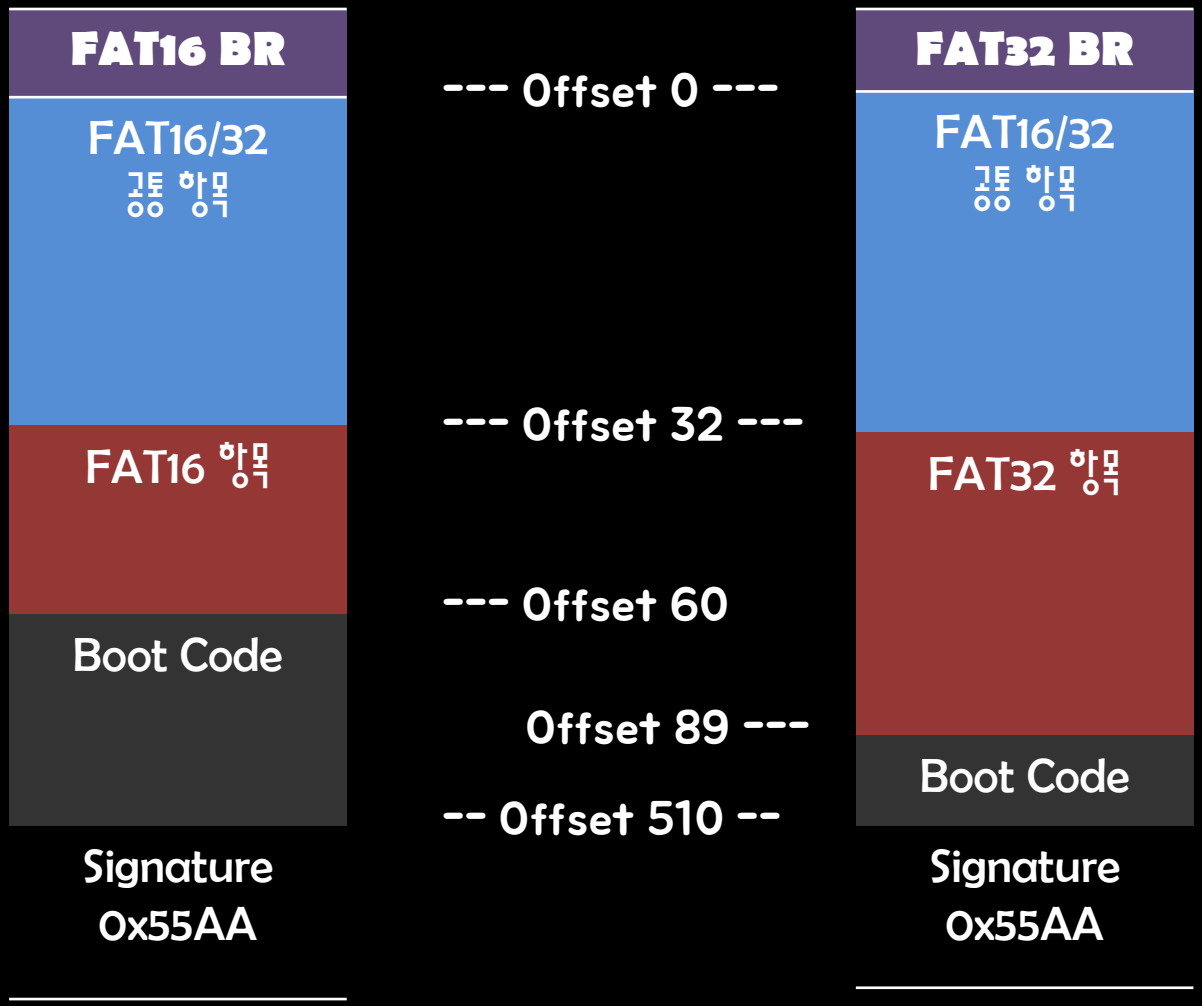
FAT16 Volume

- FAT16 에서 Root Directory 는 FAT2 바로 다음에 존재.

FAT32 Volume

- FAT32 에서 Root Directory 는 일반 디렉토리 영역이므로 데이터 영역의 어디에든 존재 가능.
- 단 보통은 FAT2 뒤에 둔다.(찾기 쉬움)

Boot Record FAT16 and FAT32



FAT 16/32 공통 영역

내용	시작 위치	사이즈	비고
Jump Boot Code(EB 3c 90)	0	3	
OEM Name	3	8	
Bytes Per Sector(512)	11	2	
Sector Per Cluster(8)	13	1	
Reserved Sector Count	14	2	FAT16: 1, FAT32: 32
Number of FATs(2)	16	1	
Root Dir Entry Count	17	2	FAT16: 512, FAT32: 0
Total Sector 16	19	2	FAT32: 0, FAT16만의 값
Media	21	1	0xF8
FAT Size 16	22	2	FAT32: 0
Sector Per Track	24	2	63
Number of Heads	26	2	255
Hidden Sector	28	4	32
Total Sector 32	32	4	FAT32용

FAT16 항목

내용	시작 위치	사이즈	비고
Drive Number	36	1	0x80
Reserved1	37	1	0
Boot Signature	38	1	0x29
Volume ID	39	4	
Volume Label	43	11	
File System Type	54	8	FAT16

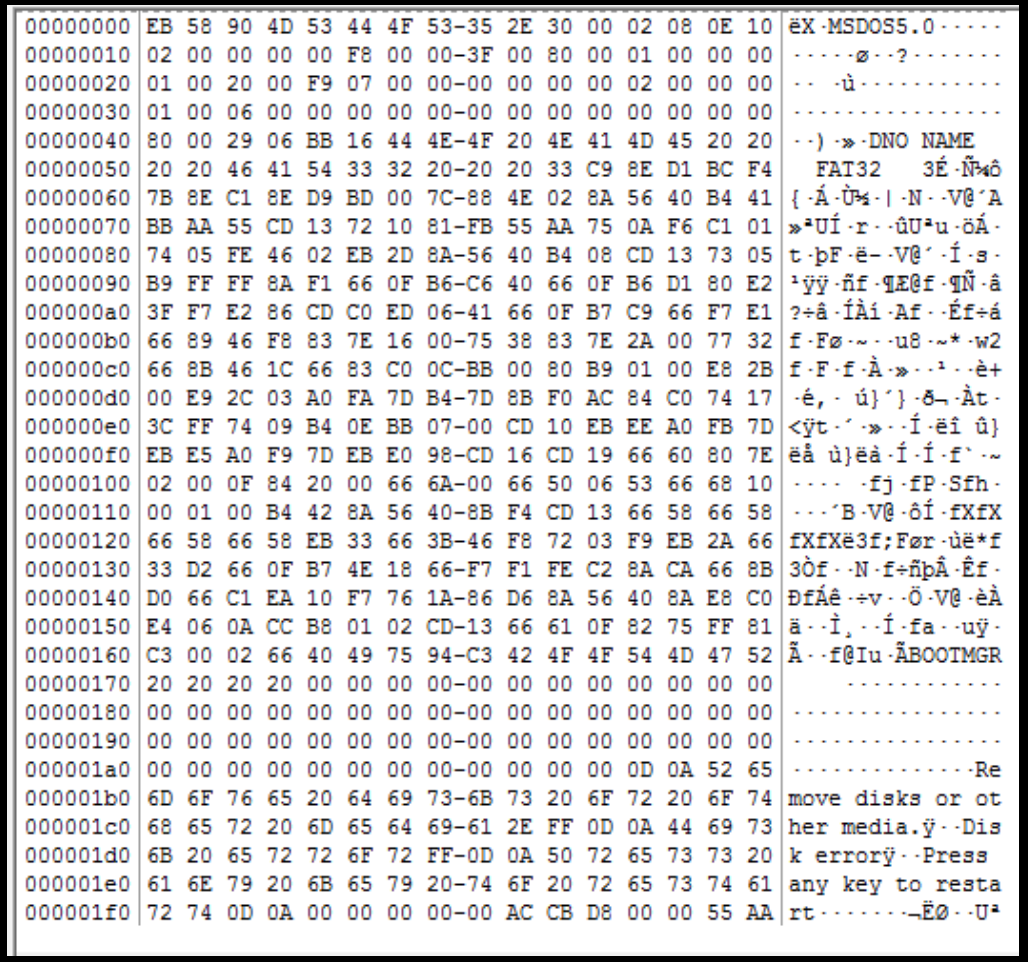
FAT32 항목

내용	시작 위치	사이즈	비고
FAT Size 32	36	4	
Ext Flags	40	2	0x00
File System Version	42	2	0x00
Root Dir Cluster	44	4	2
File System Info	48	2	1
Boot Record Backup Sec	50	2	6
Reserved	52	12	0
Drive Number	64	1	1
Reserved1	65	1	0
Boot Signature	66	1	0x29
Volume ID	67	4	
Volume Label	71	11	
File System Type	82	8	*FAT32*

FAT32 Volume

- FAT32 에서 Root Directory 는 일반 디렉토리 영역이므로 데이터 영역의 어디에든 존재 가능.
- 단 보통은 FAT2 뒤에 둔다.(찾기 쉬움)

FAT32 Volume



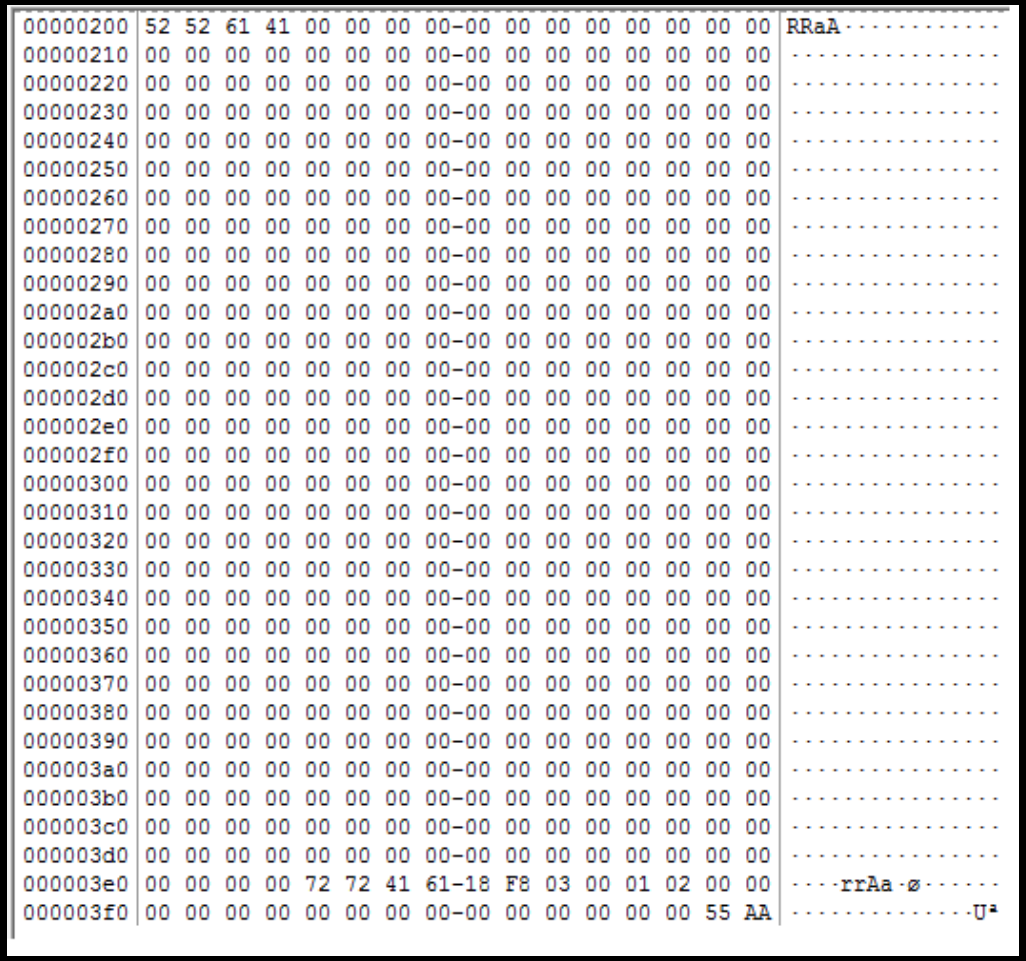
```
00000000 EB 58 90 4D 53 44 4F 53-35 2E 30 00 02 08 0E 10  ëX·MSDOS5.0·····
00000010 02 00 00 00 00 F8 00 00-3F 00 80 00 01 00 00 00  ·····ø··?·······
00000020 01 00 20 00 F9 07 00 00-00 00 00 00 02 00 00 00  ··  ·ù··········
00000030 01 00 06 00 00 00 00 00-00 00 00 00 00 00 00 00  ················
00000040 80 00 29 06 BB 16 44 4E-4F 20 4E 41 4D 45 20 20  ··)·»·DNO NAME
00000050 20 20 46 41 54 33 32 20-20 20 33 C9 8E D1 BC F4    FAT32   3É·Ñ¼ô
00000060 7B 8E C1 8E D9 BD 00 7C-88 4E 02 8A 56 40 B4 41  {·Á·Ù½·|·N··V@´A
00000070 BB AA 55 CD 13 72 10 81-FB 55 AA 75 0A F6 C1 01  »ªUÍ·r··ûUªu·öÁ·
00000080 74 05 FE 46 02 EB 2D 8A-56 40 B4 08 CD 13 73 05  t·þF·ë-·V@´·Í·s·
00000090 B9 FF FF 8A F1 66 0F B6-C6 40 66 0F B6 D1 80 E2  ¹ÿÿ·ñf·¶Æ@f·¶Ñ·â
000000a0 3F F7 E2 86 CD C0 ED 06-41 66 0F B7 C9 66 F7 E1  ?÷â·ÍÀí·Af··Éf÷á
000000b0 66 89 46 F8 83 7E 16 00-75 38 83 7E 2A 00 77 32  f·Fø·~··u8·~*·w2
000000c0 66 8B 46 1C 66 83 C0 0C-BB 00 80 B9 01 00 E8 2B  f·F·f·À·»··¹··è+
000000d0 00 E9 2C 03 A0 FA 7D B4-7D 8B F0 AC 84 C0 74 17  ·é,· ú}´}·ð¬·Àt·
000000e0 3C FF 74 09 B4 0E BB 07-00 CD 10 EB EE A0 FB 7D  <ÿt·´·»··Í·ëî û}
000000f0 EB E5 A0 F9 7D EB E0 98-CD 16 CD 19 66 60 80 7E  ëå ù}ëà·Í·Í·f`·~
00000100 02 00 0F 84 20 00 66 6A-00 66 50 06 53 66 68 10  ···· ·fj·fP·Sfh·
00000110 00 01 00 B4 42 8A 56 40-8B F4 CD 13 66 58 66 58  ···´B·V@·ôÍ·fXfX
00000120 66 58 66 58 EB 33 66 3B-46 F8 72 03 F9 EB 2A 66  fXfXë3f;Før·ùë*f
00000130 33 D2 66 0F B7 4E 18 66-F7 F1 FE C2 8A CA 66 8B  3Òf··N·f÷ñþÂ·Êf·
00000140 D0 66 C1 EA 10 F7 76 1A-86 D6 8A 56 40 8A E8 C0  ÐfÁê·÷v··Ö·V@·èÀ
00000150 E4 06 0A CC B8 01 02 CD-13 66 61 0F 82 75 FF 81  ä··Ì¸··Í·fa··uÿ·
00000160 C3 00 02 66 40 49 75 94-C3 42 4F 4F 54 4D 47 52  Ã··f@Iu·ÃBOOTMGR
00000170 20 20 20 20 00 00 00 00-00 00 00 00 00 00 00 00      ············
00000180 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ················
00000190 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00  ················
000001a0 00 00 00 00 00 00 00 00-00 00 00 00 0D 0A 52 65  ··············Re
000001b0 6D 6F 76 65 20 64 69 73-6B 73 20 6F 72 20 6F 74  move disks or ot
000001c0 68 65 72 20 6D 65 64 69-61 2E FF 0D 0A 44 69 73  her media.ÿ··Dis
000001d0 6B 20 65 72 72 6F 72 FF-0D 0A 50 72 65 73 73 20  k errorÿ··Press
000001e0 61 6E 79 20 6B 65 79 20-74 6F 20 72 65 73 74 61  any key to resta
000001f0 72 74 0D 0A 00 00 00 00-00 AC CB D8 00 00 55 AA  rt·····¬ËØ··Uª
```

FSInfo Sector : FAT32 Only

- FAT 관련한 추가적인 정보를 가지고 있다.

내용	시작 위치	사이즈	비고
Lead Signature	0	4	0x41615252
Reserved1	4	479	0
Struct Signature	484	4	0x61417272
Free Cluster Count	488	4	현재 비어있는 클러스터 수 0xFFFFFFFF라면 Free Cluster를 직접 계산해야함. 항상 옳지 않음
Next Free Cluster	492	4	현재 비어있는 클러스터 번호. 할당이 필요할 때 따로 찾지않고 이 값을 이용할 수 있다. 항상 옳지 않음
Reserved2	496	12	0
Trail Signature	508	4	0xAA550000

FSInfo Sector



00000200	52	52	61	41	00	00	00	00	00	00	00	00	00	00	RRaA
00000210	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000220	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000230	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000240	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000250	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000260	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000270	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000280	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000290	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000002a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000002b0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000002c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000002d0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000002e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000002f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000300	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000310	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000320	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000330	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000340	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000350	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000360	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000370	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000380	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000390	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000003a0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000003b0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000003c0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000003d0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000003e0	00	00	00	00	72	72	41	61	F8	03	00	01	02	00	00	····rrAaø·····
000003f0	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA	··············Uª

클러스터 위치 계산

- 클러스터는 2번 부터 시작한다. 다만 FAT 내부에 0, 1번 값은 존재한다.
- Bps
  - Bytes_Per_Sector
- Root_dir_sector
  - (Root_dir_entry_number * 32 + (bps - 1))/bps
- First_data_sector
  - Reserved_sector_count + fat size * number_of_fats + root_dir_sector
- ClusterToSector
  - (Cluster-2) * sectors_per_cluster + first_data_sector

FAT Table

Media Type	Partition State	Cluster 2	Cluster 3
Cluster 4	Cluster 5	Cluster 6	Cluster 7
Cluster 8	Cluster 9	Cluster 10	Cluster 11
…	…	…	…
…	…	…	…

- FAT32에서 각 항목은 4 bytes, FAT16 에서는 2 bytes

FAT 의 Cluster 연결 방식

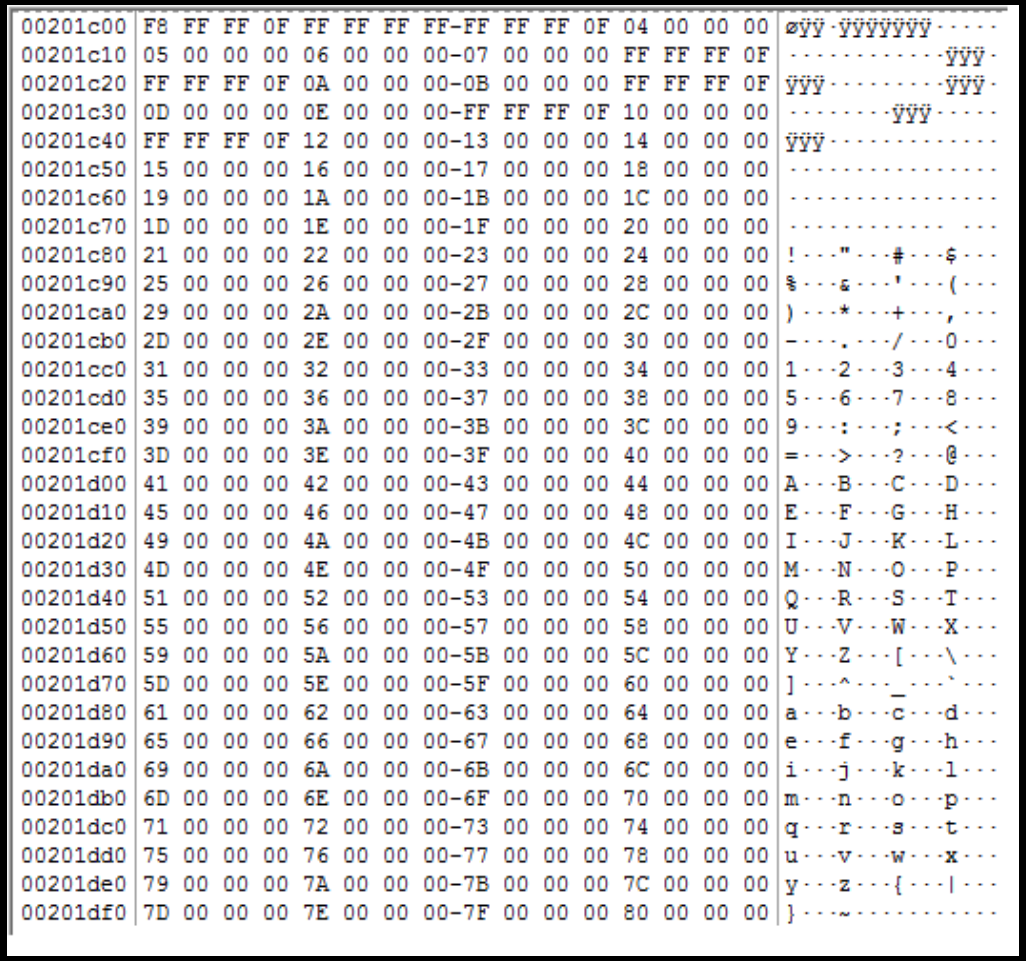
Media Type	Partition State	3	4
0F FF FF FF	Cluster 5	Cluster 6	Cluster 7
Cluster 8	Cluster 9	Cluster 10	Cluster 11
…	…	…	…
…	…	…	…

- Cluster 2 가 시작 주소인 경우 Cluster 연결 방식
  - 연결된 다음 클러스터 번호를 값으로 가짐, 마지막일 경우 0F FF FF FF

FAT 의 Cluster 연결 방식

- Root Dir의 클러스터 번호가 2라면... Root Dir의 크기는 2, 3, 4 해서 3개의 클러스터를 차지한다. 즉 2번, 3번, 4번 클러스터에 걸쳐서 루트 디렉토리가 존재한다.

FAT



```
00201c00 F8 FF FF 0F FF FF FF FF-FF FF FF 0F 04 00 00 00 øÿÿ·ÿÿÿÿÿÿÿ·····
00201c10 05 00 00 00 06 00 00 00-07 00 00 00 FF FF FF 0F ·····------·ÿÿÿ·
00201c20 FF FF FF 0F 0A 00 00 00-0B 00 00 00 FF FF FF 0F ÿÿÿ·------·ÿÿÿ·
00201c30 0D 00 00 00 0E 00 00 00-FF FF FF 0F 10 00 00 00 ·------·ÿÿÿ·····
00201c40 FF FF FF 0F 12 00 00 00-13 00 00 00 14 00 00 00 ÿÿÿ·------------
00201c50 15 00 00 00 16 00 00 00-17 00 00 00 18 00 00 00 ----------------
00201c60 19 00 00 00 1A 00 00 00-1B 00 00 00 1C 00 00 00 ----------------
00201c70 1D 00 00 00 1E 00 00 00-1F 00 00 00 20 00 00 00 ------------ ---
00201c80 21 00 00 00 22 00 00 00-23 00 00 00 24 00 00 00 !---"---#---$---
00201c90 25 00 00 00 26 00 00 00-27 00 00 00 28 00 00 00 %---&---'---(---
00201ca0 29 00 00 00 2A 00 00 00-2B 00 00 00 2C 00 00 00 )---*---+---,---
00201cb0 2D 00 00 00 2E 00 00 00-2F 00 00 00 30 00 00 00 -----.---/---0---
00201cc0 31 00 00 00 32 00 00 00-33 00 00 00 34 00 00 00 1---2---3---4---
00201cd0 35 00 00 00 36 00 00 00-37 00 00 00 38 00 00 00 5---6---7---8---
00201ce0 39 00 00 00 3A 00 00 00-3B 00 00 00 3C 00 00 00 9---:---;---<---
00201cf0 3D 00 00 00 3E 00 00 00-3F 00 00 00 40 00 00 00 =---&gt;---?---@---
00201d00 41 00 00 00 42 00 00 00-43 00 00 00 44 00 00 00 A---B---C---D---
00201d10 45 00 00 00 46 00 00 00-47 00 00 00 48 00 00 00 E---F---G---H---
00201d20 49 00 00 00 4A 00 00 00-4B 00 00 00 4C 00 00 00 I---J---K---L---
00201d30 4D 00 00 00 4E 00 00 00-4F 00 00 00 50 00 00 00 M---N---O---P---
00201d40 51 00 00 00 52 00 00 00-53 00 00 00 54 00 00 00 Q---R---S---T---
00201d50 55 00 00 00 56 00 00 00-57 00 00 00 58 00 00 00 U---V---W---X---
00201d60 59 00 00 00 5A 00 00 00-5B 00 00 00 5C 00 00 00 Y---Z---[---\---
00201d70 5D 00 00 00 5E 00 00 00-5F 00 00 00 60 00 00 00 ]---^---_---`---
00201d80 61 00 00 00 62 00 00 00-63 00 00 00 64 00 00 00 a---b---c---d---
00201d90 65 00 00 00 66 00 00 00-67 00 00 00 68 00 00 00 e---f---g---h---
00201da0 69 00 00 00 6A 00 00 00-6B 00 00 00 6C 00 00 00 i---j---k---l---
00201db0 6D 00 00 00 6E 00 00 00-6F 00 00 00 70 00 00 00 m---n---o---p---
00201dc0 71 00 00 00 72 00 00 00-73 00 00 00 74 00 00 00 q---r---s---t---
00201dd0 75 00 00 00 76 00 00 00-77 00 00 00 78 00 00 00 u---v---w---x---
00201de0 79 00 00 00 7A 00 00 00-7B 00 00 00 7C 00 00 00 y---z---{---|---
00201df0 7D 00 00 00 7E 00 00 00-7F 00 00 00 80 00 00 00 }---~-----------
```

문제1: 다음 FAT를 보고 연결된 블럭들을 분리하시오.

Media Type	Partition State	3	6
5	0F FF FF FF	7	8
0F FF FF FF	0	0	0
13	16	15	17
14	0F FF FF FF	...	...

문제1: 다음 FAT를 보고 연결된 블럭들을 분리하시오.

Media Type	Partition State	3	6
5	0F FF FF FF	7	8
0F FF FF FF	0	0	0
13	16	15	17
14	0F FF FF FF	...	...

- 목록
  - 2 -> 3 -> 6 -> 7 -> 8 (5 clusters)
  - 4 -> 5 (2 clusters)
  - 12 -> 13 -> 16 -> 14 -> 15 -> 17(6 clusters)

FAT Entry의 상태 값

FAT16	FAT32	비고
0x0000	0x?0000000	비어 있는 클러스터
0x0001	0x?0000001	예약된 클러스터(미래에 사용할까 봐서 정의된 값)
0x0002~ 0xEEEF	0x?0000002 ~ 0x?FFFFFEF	사용하고 있는 클러스터, 자신에게 연결된 다음 클러스터의 번호
0xFFF0~ 0xFFF6	0x?FFFFFF0~ 0x?FFFFFF6	예약된 클러스터
0xFFF7	0x?FFFFFF7	불량 클러스터
0xFFF8~ 0xFFFF	0x?FFFFFF8~ 0x?FFFFFFF	해당 클러스터 체인의 마지막 클러스터

Data 영역

2	3	4	5	6	7	8	9	...	N

Root Dir	a.txt	bob.zip	DIR1(Dir)	...	N

- 데이터 영역에는 디렉토리와 파일이 존재한다.
- 디렉토리는 디렉토리 엔트리라는 형태로 구성되어있다.

Data 영역 #1

2	3	4	5	6	7	8	9	…	N

Root Dir	a.txt	bob.zip	DIR1(Dir)	…	N

Name	Type	Start Cluster
Bob.zip	File	6
DIR1	Directory	8

Data 영역 #2

2	3	4	5	6	7	8	9	…	N

Root Dir	a.txt	bob.zip	DIR1(Dir)	…	N

Name	Type	Start Cluster
a.txt	File	4

Directory Entry

- Directory Entry는 하나당 32 bytes
  - 한 섹터당 16개

Directory Entry #1

내용	시작 위치	사이즈	비고
Name	0	8	파일명, 기본적으로 파일이나 디렉토리 명은 최대 8자리만 가능, 대문자만 가능하고 빈 공간은 0x20(스페이스)로 채운다.
Ext	8	3	확장자 최대 3자리, Name 과 동일한 속성을 가진다.
Attribute	11	1	
NT Resource	12	1	예약값 : 0
Create Time Tenth	13	1	파일이 생성된 시각을 1/10초 단위로 기록한 항목
Create Time	14	2	
Create Date	16	2	
Last Access Date	18	2	최근 접근 날짜만 기록
First Cluster High 2 bytes	20	2	
Write Time	22	2	

Directory Entry #2

내용	시작 위치	사이즈	비고
Write Time	22	2	
Write Date	24	2	
First Cluster Low 2 Bytes	26	2	
Filesize	28	4	디렉토리면 0

- 첫 문자로 0x20은 올 수 없다.
- 첫 문자로 0x05를 제외한 0x20보다 적은 값도 올 수 없다.

Filename + Ext

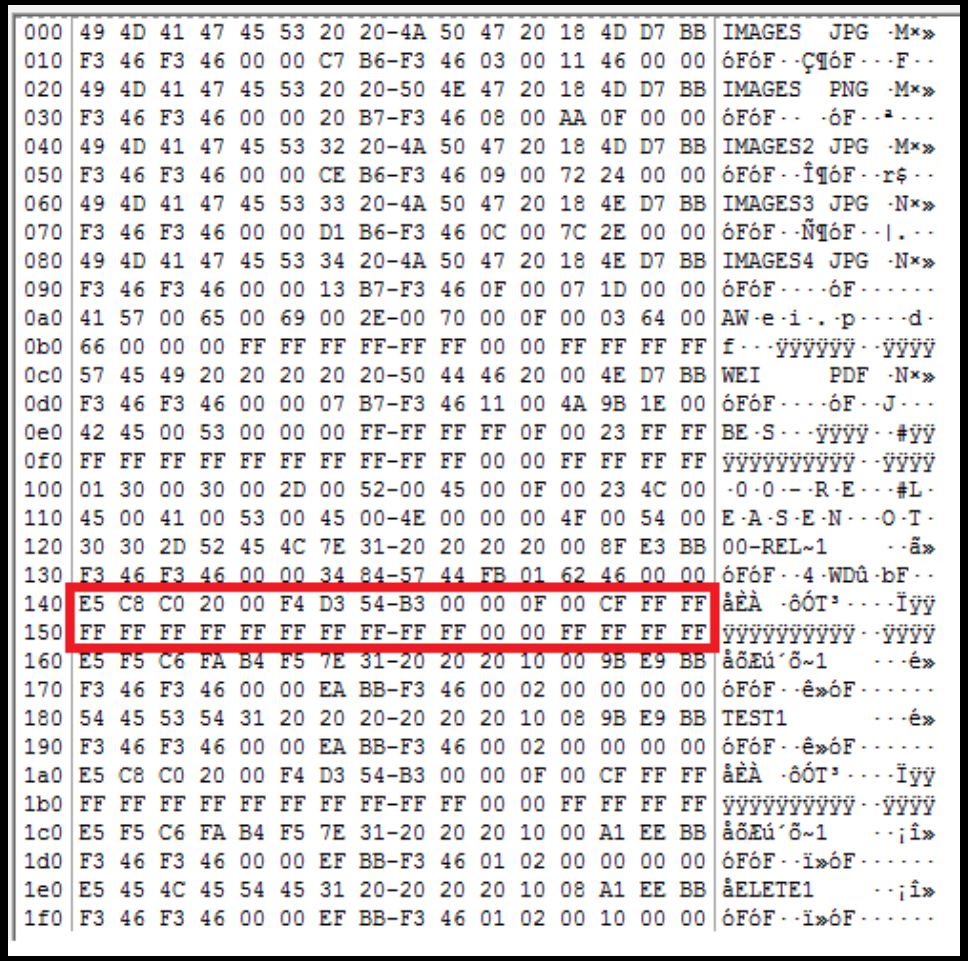
- Filename 8자, Ext 3자
  - 영어 대문자 A-Z
  - 아라비아 숫자 0~9
  - OS가 지원하는 문자(한글)

Filename								Ext		
F	O	O						B	A	R
F	I	L	E	D	A	T	A	D	O	C
F	O	O								
F	O	O						A		

FileName[0] 의 의미

값	내용
0xE5	삭제된 데이터라는 것을 의미. 파일을 삭제하면 해당 디렉토리 엔트리 Filename[0]을 단순히 0xE5로 바꾼다.
0x00	해당 디렉토리 엔트리가 비어있고, 뒤에도 모두 비어있다라는 의미, 그 뒤의 엔트리를 검색할 필요가 없음
0x05	실제로는 0XE5 값인데 일본 문자(간지)의 첫 바이트 값이 0xE5라서 모두 삭제된 파일로 취급받기 때문에 0x05를 일본어 문자의 0XE5를 표시하는데 사용함. 즉 실제로 삭제된 파일 이름의 [0] 은 0xE5

FileName[0]



```
000 49 4D 41 47 45 53 20 20-4A 50 47 20 18 4D D7 BB  IMAGES  JPG ·M×»
010 F3 46 F3 46 00 00 C7 B6-F3 46 03 00 11 46 00 00  óFóF··Ç¶óF···F··
020 49 4D 41 47 45 53 20 20-50 4E 47 20 18 4D D7 BB  IMAGES  PNG ·M×»
030 F3 46 F3 46 00 00 20 B7-F3 46 08 00 AA 0F 00 00  óFóF·· ·óF··ª···
040 49 4D 41 47 45 53 32 20-4A 50 47 20 18 4D D7 BB  IMAGES2 JPG ·M×»
050 F3 46 F3 46 00 00 CE B6-F3 46 09 00 72 24 00 00  óFóF··Î¶óF··r$··
060 49 4D 41 47 45 53 33 20-4A 50 47 20 18 4E D7 BB  IMAGES3 JPG ·N×»
070 F3 46 F3 46 00 00 D1 B6-F3 46 0C 00 7C 2E 00 00  óFóF··Ñ¶óF··|.··
080 49 4D 41 47 45 53 34 20-4A 50 47 20 18 4E D7 BB  IMAGES4 JPG ·N×»
090 F3 46 F3 46 00 00 13 B7-F3 46 0F 00 07 1D 00 00  óFóF····óF······
0a0 41 57 00 65 00 69 00 2E-00 70 00 0F 00 03 64 00  AW·e·i·.·p····d·
0b0 66 00 00 00 FF FF FF FF-FF FF 00 00 FF FF FF FF  f···ÿÿÿÿÿÿ··ÿÿÿÿ
0c0 57 45 49 20 20 20 20 20-50 44 46 20 00 4E D7 BB  WEI     PDF ·N×»
0d0 F3 46 F3 46 00 00 07 B7-F3 46 11 00 4A 9B 1E 00  óFóF····óF··J···
0e0 42 45 00 53 00 00 00 FF-FF FF FF 0F 00 23 FF FF  BE·S···ÿÿÿÿ··#ÿÿ
0f0 FF FF FF FF FF FF FF FF-FF FF 00 00 FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿ··ÿÿÿÿ
100 01 30 00 30 00 2D 00 52-00 45 00 0F 00 23 4C 00  ·0·0·-·R·E···#L·
110 45 00 41 00 53 00 45 00-4E 00 00 00 4F 00 54 00  E·A·S·E·N···O·T·
120 30 30 2D 52 45 4C 7E 31-20 20 20 20 00 8F E3 BB  00-REL~1    ··ã»
130 F3 46 F3 46 00 00 34 84-57 44 FB 01 62 46 00 00  óFóF··4·WDû·bF··
140 E5 C8 C0 20 00 F4 D3 54-B3 00 00 0F 00 CF FF FF  åÈÀ ·ôÓT³····Ïÿÿ
150 FF FF FF FF FF FF FF FF-FF FF 00 00 FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿ··ÿÿÿÿ
160 E5 F5 C6 FA B4 F5 7E 31-20 20 20 10 00 9B E9 BB  åõÆú´õ~1   ···é»
170 F3 46 F3 46 00 00 EA BB-F3 46 00 02 00 00 00 00  óFóF··ê»óF······
180 54 45 53 54 31 20 20 20-20 20 20 10 08 9B E9 BB  TEST1      ···é»
190 F3 46 F3 46 00 00 EA BB-F3 46 00 02 00 00 00 00  óFóF··ê»óF······
1a0 E5 C8 C0 20 00 F4 D3 54-B3 00 00 0F 00 CF FF FF  åÈÀ ·ôÓT³····Ïÿÿ
1b0 FF FF FF FF FF FF FF FF-FF FF 00 00 FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿ··ÿÿÿÿ
1c0 E5 F5 C6 FA B4 F5 7E 31-20 20 20 10 00 A1 EE BB  åõÆú´õ~1   ··¡î»
1d0 F3 46 F3 46 00 00 EF BB-F3 46 01 02 00 00 00 00  óFóF··ï»óF······
1e0 E5 45 4C 45 54 45 31 20-20 20 20 10 08 A1 EE BB  åELETE1    ··¡î»
1f0 F3 46 F3 46 00 00 EF BB-F3 46 01 02 00 10 00 00  óFóF··ï»óF······
```

삭제된 데이터

분류	내용
Deleted	사용자가 디렉토리를 볼 때 파일 네임 엔트리가 표시되지 않는 상태. 삭제되었다는 표시만 된 상태, 파일 이름과 메타데이터가 일치하는 상태, 단순히 사용자가 Delete를 한 상태
Orphan	파일 이름과 메타데이터 구조 사이의 관계가 더 이상 정확하지 않다는 점을 제외하고는 Deleted 상태와 유사.
Unallocated	한 번이라도 할당된 적이 있는 파일 네임 엔트리를 가지고 있고 연관된 메타데이터 구조의 링크는 끊기거나 재사용된 상태, 복구는 볼륨의 미할당 영역에서 아직 재사용되지 않은 영역을 카빙함으로써 복구를 시도한다.
Overwritten	다른 파일에 재할당된 하나 이상의 데이터 유닛을 가지고 있음. 완전한 복구는 더 이상 불가능 하지만, 부분 복구는 가능할 수도 있다.

Attribute 값

속성	이름	내용
0x01	Read Only	읽기 전용
0x02	Hidden	파일을 숨긴다.
0x04	System	운영체제에서 사용하는 파일
0x08	Volume Label	이 파일의 이름이 볼륨 레이블이 됨
0x10	Directory	디렉토리
0x20	Archive	파일
0xF0	LongFileName	Long File Name Entry

Create Time

속성	유효 범위	내용
초(second)	0 ~ 29	초를 기록한다. 2초당 1을 증가
분(minute)	0 ~ 59	분
시간(Hour)	0 ~ 23	시간

Create Date

속성	유효 범위	내용
Day	1 ~ 31	
Month	1 ~ 12	
Year	0 ~ 127	1980년 부터 시작 : 최대 1980 + 127 = 2107년

Directory Entry Sample

000	49	4D	41	47	45	53	20	20-4A	50	47	20	18	4D	D7	BB
010	F3	46	F3	46	00	00	C7	B6-F3	46	03	00	11	46	00	00
020	49	4D	41	47	45	53	20	20-50	4E	47	20	18	4D	D7	BB
030	F3	46	F3	46	00	00	20	B7-F3	46	08	00	AA	0F	00	00
040	49	4D	41	47	45	53	32	20-4A	50	47	20	18	4D	D7	BB
050	F3	46	F3	46	00	00	CE	B6-F3	46	09	00	72	24	00	00
060	49	4D	41	47	45	53	33	20-4A	50	47	20	18	4E	D7	BB
070	F3	46	F3	46	00	00	D1	B6-F3	46	0C	00	7C	2E	00	00
080	49	4D	41	47	45	53	34	20-4A	50	47	20	18	4E	D7	BB
090	F3	46	F3	46	00	00	13	B7-F3	46	0F	00	07	1D	00	00
0a0	41	57	00	65	00	69	00	2E-00	70	00	0F	00	03	64	00
0b0	66	00	00	00	FF	FF	FF	FF-FF	FF	00	00	FF	FF	FF	FF
0c0	57	45	49	20	20	20	20	20-50	44	46	20	00	4E	D7	BB
0d0	F3	46	F3	46	00	00	07	B7-F3	46	11	00	4A	9B	1E	00
0e0	42	45	00	53	00	00	00	FF-FF	FF	FF	0F	00	23	FF	FF
0f0	FF	FF	FF	FF	FF	FF	FF	FF-FF	FF	00	00	FF	FF	FF	FF
100	01	30	00	30	00	2D	00	52-00	45	00	0F	00	23	4C	00
110	45	00	41	00	53	00	45	00-4E	00	00	00	4F	00	54	00
120	30	30	2D	52	45	4C	7E	31-20	20	20	20	00	8F	E3	BB
130	F3	46	F3	46	00	00	34	84-57	44	FB	01	62	46	00	00
140	E5	C8	C0	20	00	F4	D3	54-B3	00	00	0F	00	CF	FF	FF
150	FF	FF	FF	FF	FF	FF	FF	FF-FF	FF	00	00	FF	FF	FF	FF
160	E5	F5	C6	FA	B4	F5	7E	31-20	20	20	10	00	9B	E9	BB
170	F3	46	F3	46	00	00	EA	BB-F3	46	00	02	00	00	00	00
180	54	45	53	54	31	20	20	20-20	20	20	10	08	9B	E9	BB
190	F3	46	F3	46	00	00	EA	BB-F3	46	00	02	00	00	00	00
1a0	E5	C8	C0	20	00	F4	D3	54-B3	00	00	0F	00	CF	FF	FF
1b0	FF	FF	FF	FF	FF	FF	FF	FF-FF	FF	00	00	FF	FF	FF	FF
1c0	E5	F5	C6	FA	B4	F5	7E	31-20	20	20	10	00	A1	EE	BB
1d0	F3	46	F3	46	00	00	EF	BB-F3	46	01	02	00	00	00	00
1e0	E5	45	4C	45	54	45	31	20-20	20	20	10	08	A1	EE	BB
1f0	F3	46	F3	46	00	00	EF	BB-F3	46	01	02	00	10	00	00

000	I	M	A	G	E	S			J	P	G			M	»
010	ó	F	ó	F			Ç	¶	ó	F			F		
020	I	M	A	G	E	S			P	N	G			M	»
030	ó	F	ó	F					ó	F			ª		
040	I	M	A	G	E	S	2		J	P	G			M	»
050	ó	F	ó	F			Î	¶	ó	F			r	$	
060	I	M	A	G	E	S	3		J	P	G			N	»
070	ó	F	ó	F			Ñ	¶	ó	F				.	
080	I	M	A	G	E	S	4		J	P	G			N	»
090	ó	F	ó	F					ó	F					
0a0	A	W		e		i		.		p				d	
0b0	f				ÿ	ÿ	ÿ	ÿ				ÿ	ÿ	ÿ	ÿ
0c0	W	E	I						P	D	F			N	»
0d0	ó	F	ó	F					ó	F			J		
0e0	B	E		S				ÿ	ÿ	ÿ	ÿ			#	ÿ
0f0	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ
100		0		0		-		R		E			#	L	
110	E		A		S		E		N			O		T	
120	0	0	-	R	E	L	~	1						ã	»
130	ó	F	ó	F			4		W	D	û		b	F	
140	å	È	À			ô	Ó	T	³				Ï	ÿ	ÿ
150	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ
160	å	õ	Æ	ú	´	õ	~	1						é	»
170	ó	F	ó	F			ê	»	ó	F					
180	T	E	S	T	1									é	»
190	ó	F	ó	F			ê	»	ó	F					
1a0	å	È	À			ô	Ó	T	³				Ï	ÿ	ÿ
1b0	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ	ÿ
1c0	å	õ	Æ	ú	´	õ	~	1					¡	î	»
1d0	ó	F	ó	F			ï	»	ó	F					
1e0	å	E	L	E	T	E	1						¡	î	»
1f0	ó	F	ó	F			ï	»	ó	F					

Long File Name(LFN)

- 유니코드(UTF-16) 방식으로 인코딩 됨
- 최대 255자
- 확장자가 3자 이상 가능
- 기존의 Short File Name 과 호환
- 기본의 Short File Name 보다 특수문자 허용 범위가 넓음
- 하나의 LFN에 최대 13자를 저장할 수 있다.
  - 255자를 채울려면?

Long File Name Entry

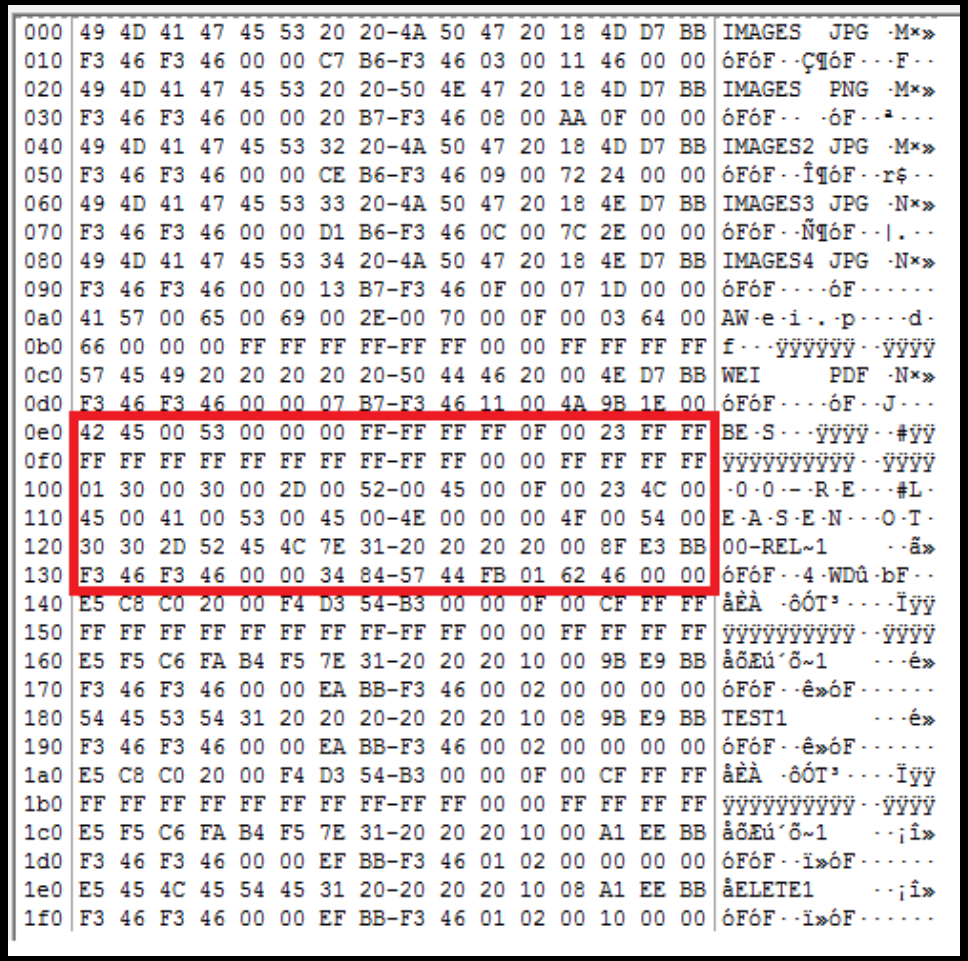
내용	시작 위치	사이즈	비고
Order	0	1	LFN의 정렬된 순번이 저장됨. 6번째 비트(0x40)이 1이면 마지막 해당 파일의 마지막 LFN Entry임
Name1	1	10	UTF-16으로 한글자당 2 bytes를 차지
Attribute	11	1	0x0F
Type	12	1	0
Check Sum	13	1	Short File Name의 CheckSum
Name2	14	12	
First Cluster Low	26	2	0
Name3	28	4	10 + 12 + 4 = 26 으로 총 13자가 가능

SFN 과 LFN

Directory Entry 순서	Order 항목 값
N번째 LFN Entry(마지막)	0x40 | N
...	
2nd LFN Entry	0x02
1st LFN Entry	0x01
위의 LFN을 가지는 SFN	해당 사항 없음

- LFN이 거꾸로 저장되어 있다.
- 즉 제대로 된 파일 이름은 1, 2, ... , N으로 문자열을 배열해야 한다.

LFN Sample



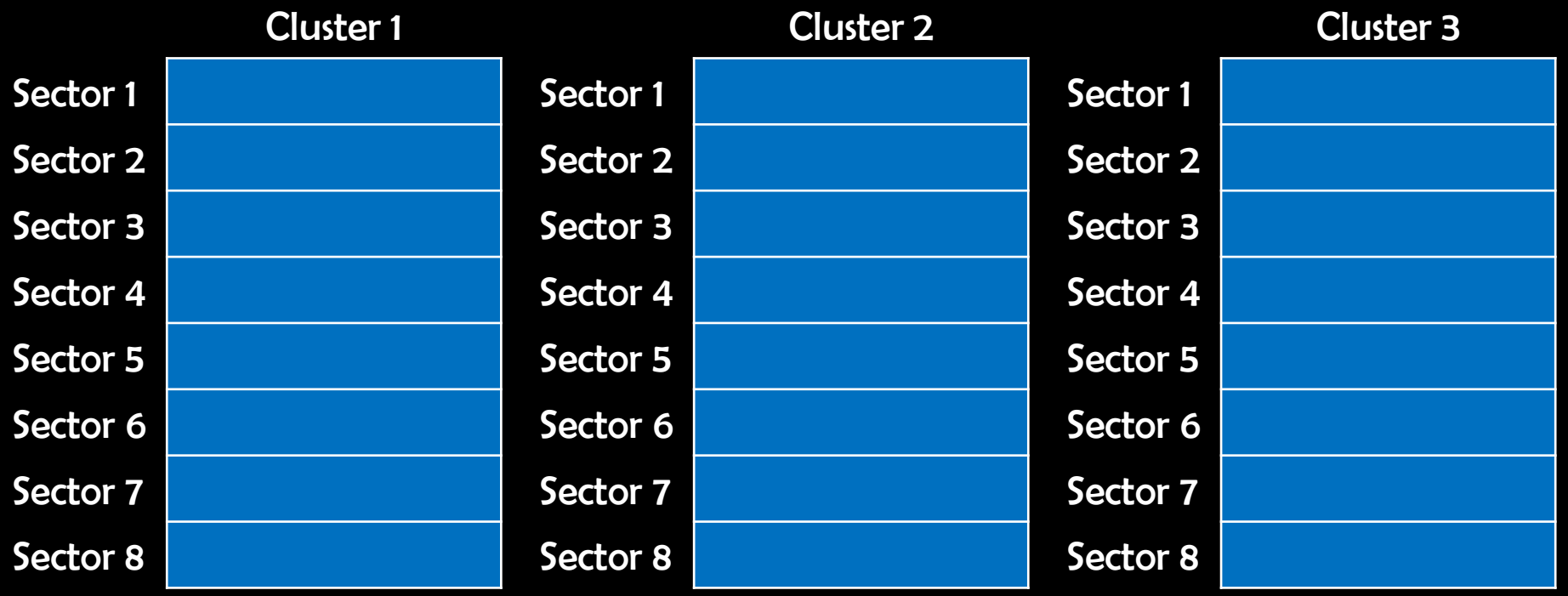
```
000 49 4D 41 47 45 53 20 20-4A 50 47 20 18 4D D7 BB  IMAGES  JPG ·M×»
010 F3 46 F3 46 00 00 C7 B6-F3 46 03 00 11 46 00 00  óFóF··Ç¶óF···F··
020 49 4D 41 47 45 53 20 20-50 4E 47 20 18 4D D7 BB  IMAGES  PNG ·M×»
030 F3 46 F3 46 00 00 20 B7-F3 46 08 00 AA 0F 00 00  óFóF·· ·óF··ª···
040 49 4D 41 47 45 53 32 20-4A 50 47 20 18 4D D7 BB  IMAGES2 JPG ·M×»
050 F3 46 F3 46 00 00 CE B6-F3 46 09 00 72 24 00 00  óFóF··Î¶óF··r$··
060 49 4D 41 47 45 53 33 20-4A 50 47 20 18 4E D7 BB  IMAGES3 JPG ·N×»
070 F3 46 F3 46 00 00 D1 B6-F3 46 0C 00 7C 2E 00 00  óFóF··Ñ¶óF··|.··
080 49 4D 41 47 45 53 34 20-4A 50 47 20 18 4E D7 BB  IMAGES4 JPG ·N×»
090 F3 46 F3 46 00 00 13 B7-F3 46 0F 00 07 1D 00 00  óFóF····óF······
0a0 41 57 00 65 00 69 00 2E-00 70 00 0F 00 03 64 00  AW·e·i·.·p····d·
0b0 66 00 00 00 FF FF FF FF-FF FF 00 00 FF FF FF FF  f···ÿÿÿÿÿÿ··ÿÿÿÿ
0c0 57 45 49 20 20 20 20 20-50 44 46 20 00 4E D7 BB  WEI     PDF ·N×»
0d0 F3 46 F3 46 00 00 07 B7-F3 46 11 00 4A 9B 1E 00  óFóF····óF··J···
0e0 42 45 00 53 00 00 00 FF-FF FF FF 0F 00 23 FF FF  BE·S···ÿÿÿÿ··#ÿÿ
0f0 FF FF FF FF FF FF FF FF-FF FF 00 00 FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿ··ÿÿÿÿ
100 01 30 00 30 00 2D 00 52-00 45 00 0F 00 23 4C 00  ·0·0·-·R·E···#L·
110 45 00 41 00 53 00 45 00-4E 00 00 00 4F 00 54 00  E·A·S·E·N···O·T·
120 30 30 2D 52 45 4C 7E 31-20 20 20 20 00 8F E3 BB  00-REL~1    ··ã»
130 F3 46 F3 46 00 00 34 84-57 44 FB 01 62 46 00 00  óFóF··4·WDû·bF··
140 E5 C8 C0 20 00 F4 D3 54-B3 00 00 0F 00 CF FF FF  åÈÀ ·ôÓT³····Ïÿÿ
150 FF FF FF FF FF FF FF FF-FF FF 00 00 FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿ··ÿÿÿÿ
160 E5 F5 C6 FA B4 F5 7E 31-20 20 20 10 00 9B E9 BB  åõÆú´õ~1   ···é»
170 F3 46 F3 46 00 00 EA BB-F3 46 00 02 00 00 00 00  óFóF··ê»óF······
180 54 45 53 54 31 20 20 20-20 20 20 10 08 9B E9 BB  TEST1      ···é»
190 F3 46 F3 46 00 00 EA BB-F3 46 00 02 00 00 00 00  óFóF··ê»óF······
1a0 E5 C8 C0 20 00 F4 D3 54-B3 00 00 0F 00 CF FF FF  åÈÀ ·ôÓT³····Ïÿÿ
1b0 FF FF FF FF FF FF FF FF-FF FF 00 00 FF FF FF FF  ÿÿÿÿÿÿÿÿÿÿ··ÿÿÿÿ
1c0 E5 F5 C6 FA B4 F5 7E 31-20 20 20 10 00 A1 EE BB  åõÆú´õ~1   ··¡î»
1d0 F3 46 F3 46 00 00 EF BB-F3 46 01 02 00 00 00 00  óFóF··ï»óF······
1e0 E5 45 4C 45 54 45 31 20-20 20 20 10 08 A1 EE BB  åELETE1    ··¡î»
1f0 F3 46 F3 46 00 00 EF BB-F3 46 01 02 00 10 00 00  óFóF··ï»óF······
```

문제2: 증거 이미지에서 Root Dir의 내용을 분류하시오.

Sector 와 Cluster

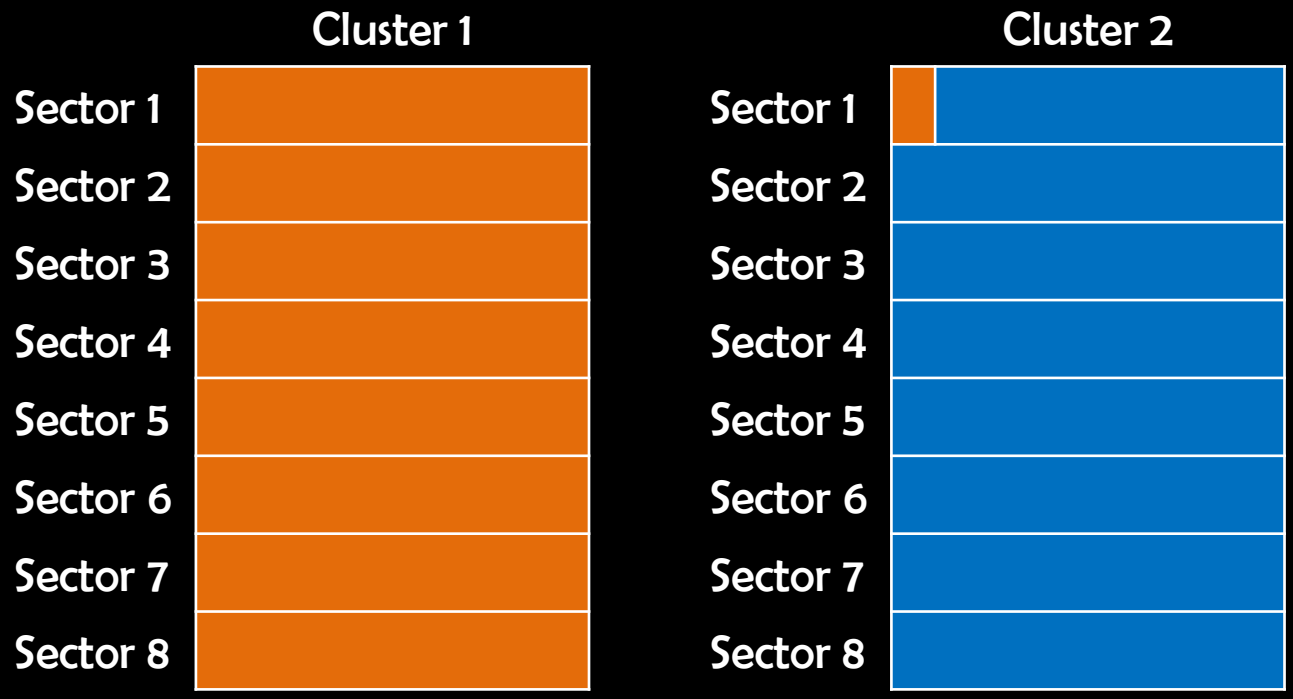
- Sector의 크기는 512
  - 그렇지 않은 장비도 있을 수 있음.
- Cluster 는 여러 개의 Sector의 모음
  - 쓰기나 읽기 효율성을 높이기 위해서 좀 더 큰 블록 단위를 이용함.

Cluster


Cluster 1
Sector 1
Sector 2
Sector 3
Sector 4
Sector 5
Sector 6
Sector 7
Sector 8
Cluster 2
Sector 1
Sector 2
Sector 3
Sector 4
Sector 5
Sector 6
Sector 7
Sector 8
Cluster 3
Sector 1
Sector 2
Sector 3
Sector 4
Sector 5
Sector 6
Sector 7
Sector 8

파일 데이터(2048 bytes)

파일 데이터(4097 bytes)



Unallocated Cluster

- 아직 파일등에서 사용하지 않는 Cluster

예약된 섹터

- 기본적으로 사용되지 않는 영역.
- 데이터를 은닉할 수 있는 영역.
  - 일반적인 경로로는 볼 수 없음.

FAT 에서 파일이 지워질 때

1. FAT Table을 따라서 할당된 FAT 영역을 0으로 변경
2. Directory Entry에서 파일명의 [0]번째 글자를 0xE5 로 바꾼다.
3. Directory Entry 에서 Cluster 위치의 상위 2 바이트를 0을 채운다.
    1. 파일 복구를 어렵게 하기 위한 정책
    2. 그래서 0xFFFF 보다 큰 위치에 있는 클러스터 부분은 찾기가 어렵다.

파일의 복구 원리

- 기본적으로 디스크 읽기/쓰기 성능을 높이기 위해서, 가능한 순차적인 클러스터로 할당됨
  - 사이즈는 Directory Entry 에 남아있음.
  - 사이즈만큼 순차적으로 읽어서 복구 가능한 경우가 많음

Fragment 된 파일의 복구는?

- 첫번째 클러스터내 크기라면 복구가 가능
  - 대용량 파일은?
- 결론은 잘 안된다는 얘기...

미할당 클러스터내의 데이터의 복구?

- 파일 복구의 원리가 그대로 적용된다.
- 카빙으로 데이터 영역을 통한 복구 시도가 가능
- 카빙
  - 모든 클러스터를 돌면서 파일 헤더를 발견해서 데이터 복구를 시도함.
  - 대부분의 툴에서 이런 복구를 지원함.
  - 같은 이슈로 Fragment 된 파일의 복구는 힘듬.