FAT32 File System Analysis

F_active ch4nh33

1. What the File System
2. File Allocation Table
3. Carving

What the File System

What is the File System

-파일시스템은 자료를 보관하기 위한 하드웨어 장치를 사용하여 파일의 물리적인 정보를 관리하는 것

-최근, 네트워크를 통하여 서버상의 파일에 접근하고 관리하는 NFS,SMB 등도 파일시스템에 포함

What is the File System

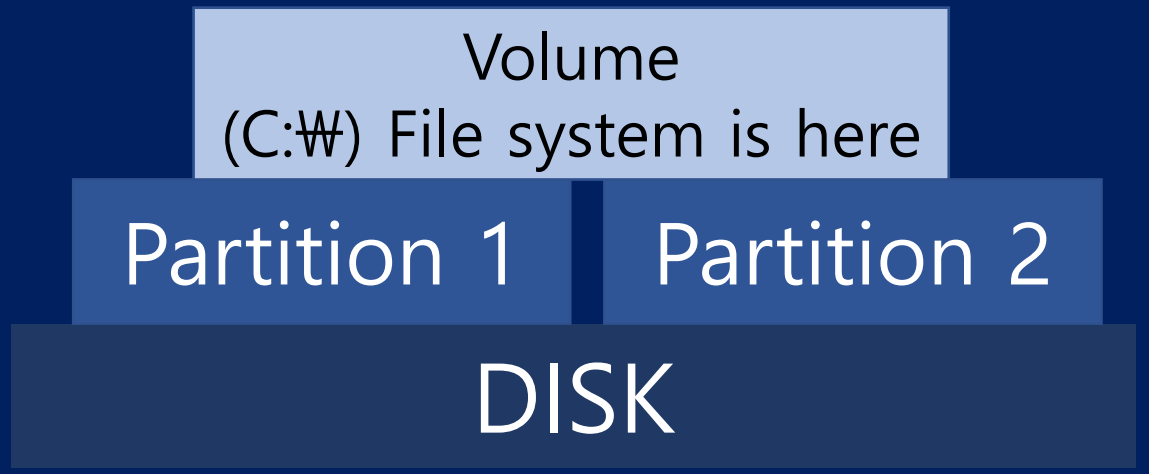
Basic Structure

What is the File System

Exception

Floppy disk

RAID



File Allocation Table

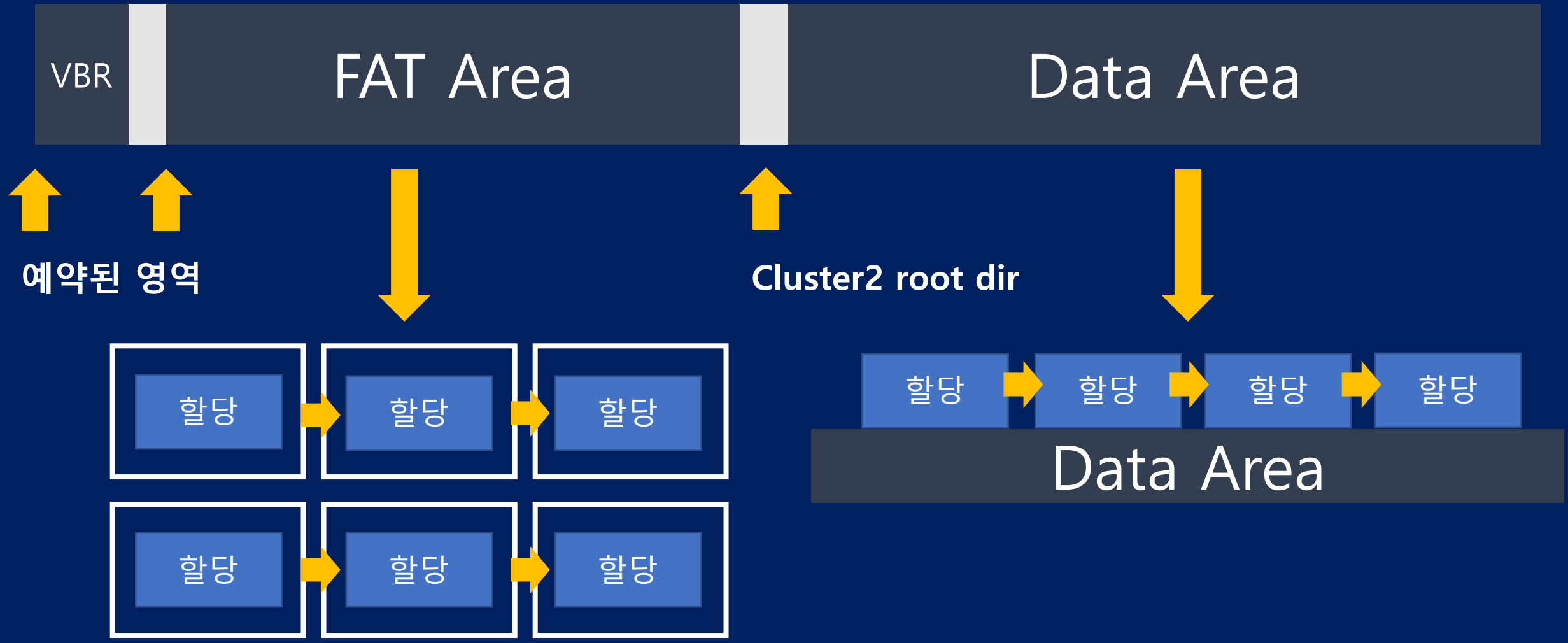
File Allocation Table

File Allocation Table

간단한 / 보편화된
운영체제에서 볼 수 있는 가장 간단한 파일 시스템 중 하나

그 외)NTFS , HFS , APFS , NNFS , Ext2 , Ext3 , Ext4 , ISO 9660...

File Allocation Table



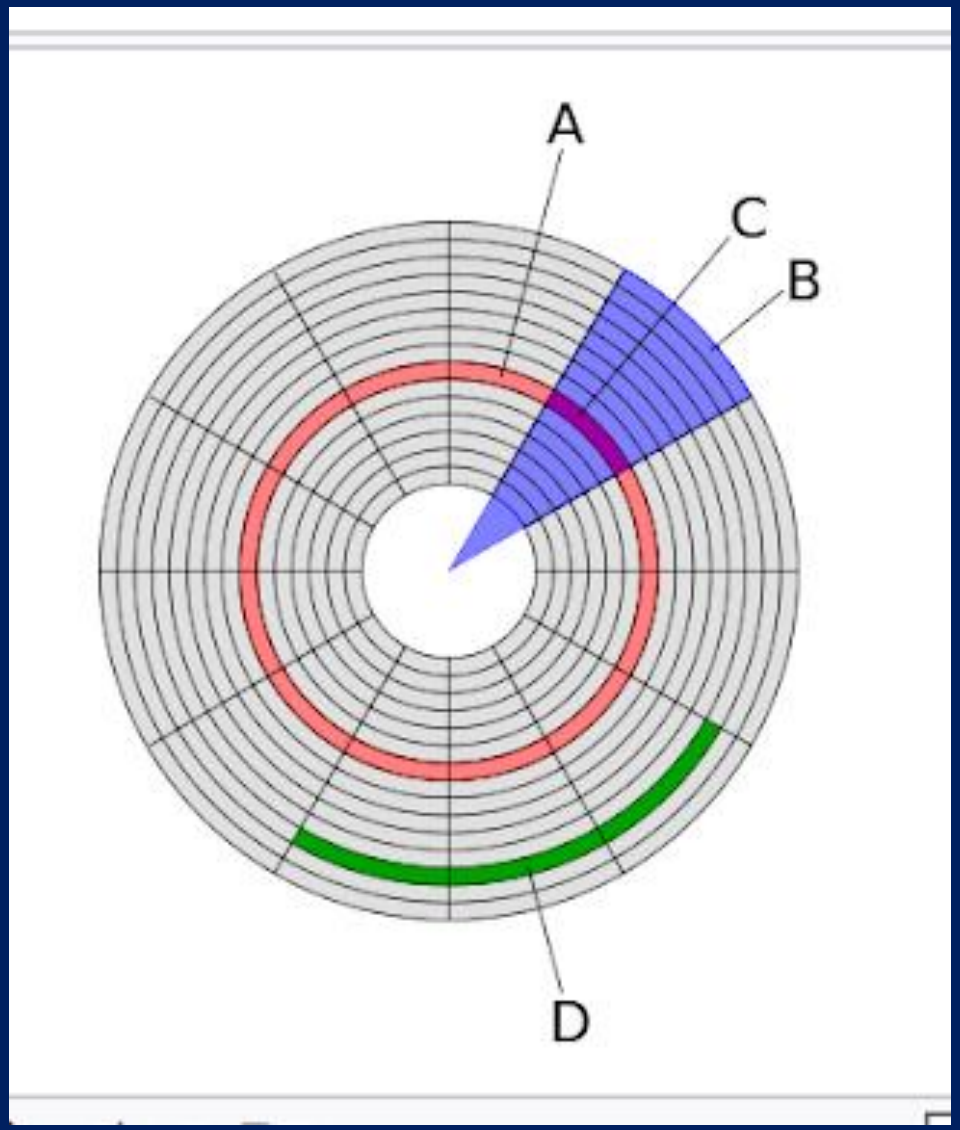
File Allocation Table

VBR – Volume Boot Record

```
uint8_t bootcode[3];
char OEMname[8];
uint16_t bytePerSector;
uint8_t sectorPercluster;
uint16_t sizeOfReservedSec;
uint8_t cntOfFAT;
uint16_t maxFileOfRoot;
uint16_t totalSec2;
uint8_t mediaTyp;
uint16_t cntOfFATSec;
uint16_t secOfTrack;
uint16_t headOnStor;
uint32_t secOfPrePart;
uint32_t totalSec4;
uint32_t secOfFAT;
```

```
uint16_t HowToAlloc;
uint8_t minorVersion;
uint8_t majorVersion;
uint32_t OffOfRootClust;
uint16_t OffOfFsinfo;
uint16_t OffOfCopyBootSector;
char reserved[12];
uint8_t driveNum;
uint8_t unused;
uint8_t extendSignature;
uint32_t volSereal;
char volabel[11];
char sysType[8];
char padding[420];
char signautre[2];
```

File Allocation Table



A : Track

B : Geomatrical Sector

C : Sector

D : Cluster

File Allocation Table

Find Offset

File Allocation Table

Entry

```
char FirstSig;
char filename[10];
uint8_t filetype;
uint8_t reserved;
uint8_t createTime;
uint16_t createTimeD;
uint16_t createDate;
uint16_t accessDate;
uint16_t parentTopClu;
uint16_t modifyTime;
uint16_t modifyDate;
uint16_t parentBotClu;
uint32_t filesize;
```

디렉토리 엔트리(Directory entry)란 디렉토리를 표현하는 데에 쓰이는 자료구조 파일 시스템에 따라서 이를 구성하는 항목도 달라짐

일반적으로는 파일이름, 파일속성 등 파일에 대한 여러가지 정보가 저장되는데, 유닉스 계열에서는 파일이름 번호만 저장

MS-DOS에서 디렉토리 엔트리는 파일 이름, 확장자, 속성, 시각, 날짜, 첫 번째 블록의 번호, 파일 크기 정보를 가지고 있음

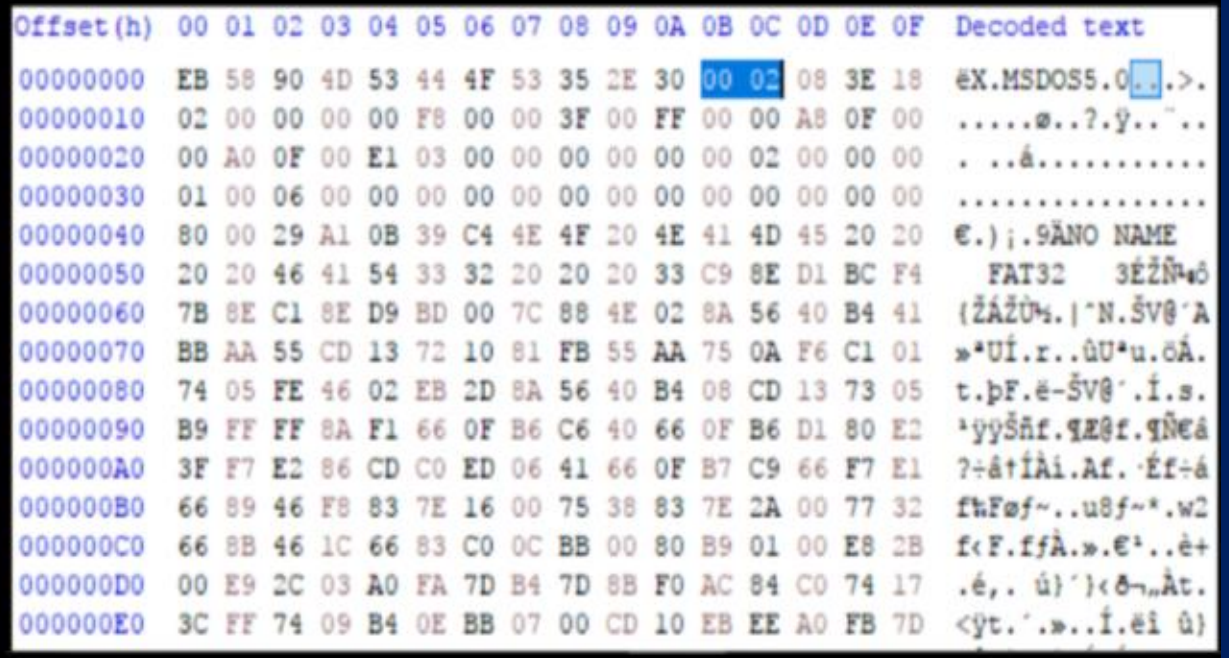
File Allocation Table

VBR - OEM name

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	08	3E	18	ëX.MSDOS5.0...>.
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	00	A8	0F	00	.....ø..?.ÿ..¨..
00000020	00	A0	0F	00	E1	03	00	00	00	00	00	00	02	00	00	00	. ..á...........
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000040	80	00	29	A1	0B	39	C4	4E	4F	20	4E	41	4D	45	20	20	€.)¡.9ÄNO NAME
00000050	20	20	46	41	54	33	32	20	20	20	33	C9	8E	D1	BC	F4	FAT32 3ÉŽÑ¼ô
00000060	7B	8E	C1	8E	D9	BD	00	7C	88	4E	02	8A	56	40	B4	41	{ŽÁŽÙ½.|ˆN.ŠV@´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-ŠV@´.Í.s.
00000090	B9	FF	FF	8A	F1	66	0F	B6	C6	40	66	0F	B6	D1	80	E2	¹ÿÿŠñf.¶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

FAT의 OEM name은 일반적으로 MSDOS5.0이다.

1Sector 당 차지하는 Byte

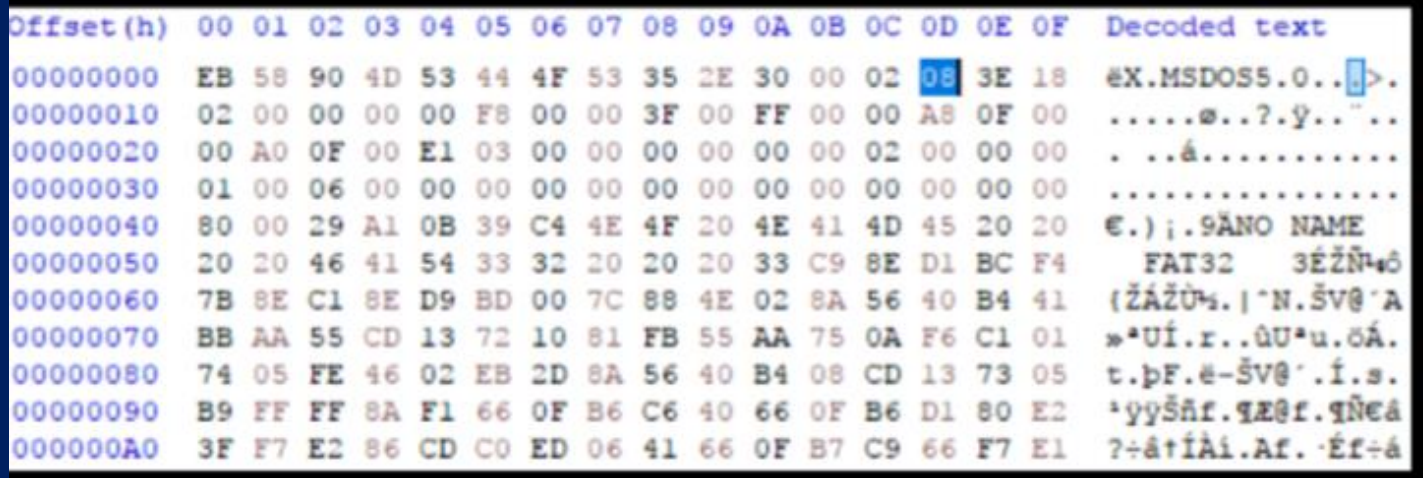


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	08	3E	18	ëX.MSDOS5.0...>.
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	00	A8	0F	00	.....ø..?.ÿ..¨..
00000020	00	A0	0F	00	E1	03	00	00	00	00	00	00	02	00	00	00	. ..á...........
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000040	80	00	29	A1	0B	39	C4	4E	4F	20	4E	41	4D	45	20	20	€.)¡.9ÄNO NAME
00000050	20	20	46	41	54	33	32	20	20	20	33	C9	8E	D1	BC	F4	FAT32 3ÉŽÑ¼ô
00000060	7B	8E	C1	8E	D9	BD	00	7C	88	4E	02	8A	56	40	B4	41	{ŽÁŽÙ½.|ˆN.ŠV@´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-ŠV@´.Í.s.
00000090	B9	FF	FF	8A	F1	66	0F	B6	C6	40	66	0F	B6	D1	80	E2	¹ÿÿŠñf.¶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‰Føƒ~..u8ƒ~*.w2
000000C0	66	8B	46	1C	66	83	C0	0C	BB	00	80	B9	01	00	E8	2B	f‹F.fƒÀ.».€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}

512 Byte 이다

File Allocation Table

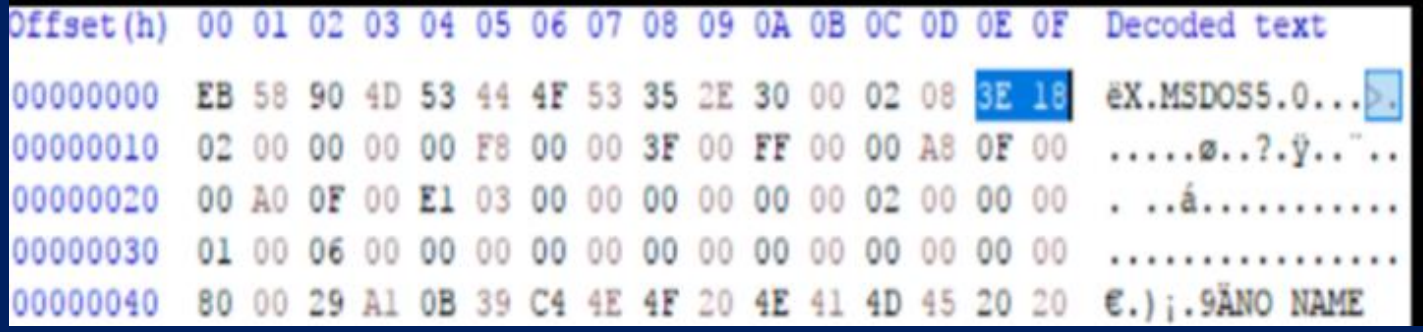
1Cluster 당 차지하는 Sector



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	08	3E	18	ëX.MSDOS5.0...>.
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	00	A8	0F	00	.....ø..?.ÿ..¨..
00000020	00	A0	0F	00	E1	03	00	00	00	00	00	00	02	00	00	00	. ..á...........
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000040	80	00	29	A1	0B	39	C4	4E	4F	20	4E	41	4D	45	20	20	€.)¡.9ÄNO NAME
00000050	20	20	46	41	54	33	32	20	20	20	33	C9	8E	D1	BC	F4	FAT32 3ÉŽÑ¼ô
00000060	7B	8E	C1	8E	D9	BD	00	7C	88	4E	02	8A	56	40	B4	41	{ŽÁŽÙ½.|ˆN.ŠV@´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-ŠV@´.Í.s.
00000090	B9	FF	FF	8A	F1	66	0F	B6	C6	40	66	0F	B6	D1	80	E2	¹ÿÿŠñf.¶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

그럼 1클러스터당 바이트 수는
8 * 512 == 4096 Byte

예약된 영역의 Cluster 개수 (6206)

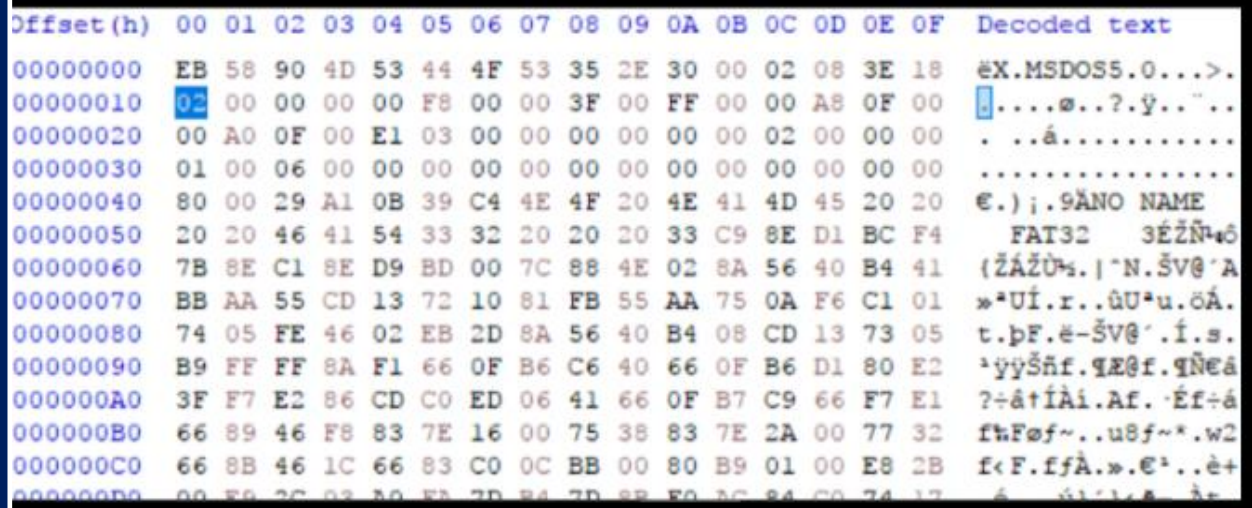


Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	08	3E	18	ëX.MSDOS5.0...>.
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	00	A8	0F	00	.....ø..?.ÿ..¨..
00000020	00	A0	0F	00	E1	03	00	00	00	00	00	00	02	00	00	00	. ..á...........
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000040	80	00	29	A1	0B	39	C4	4E	4F	20	4E	41	4D	45	20	20	€.)¡.9ÄNO NAME

6206 * 512 = 3,177,472 Byte는
예약된 영역으로
FAT 영역의 시작 부분이 된다.

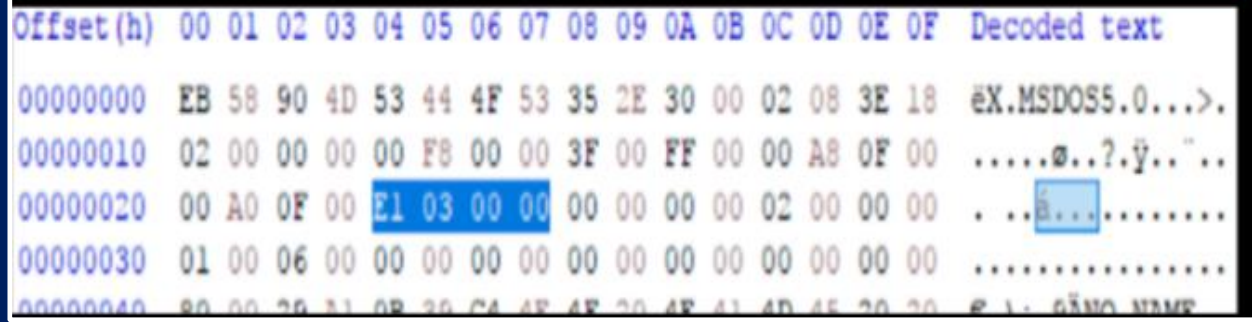
File Allocation Table

존재하는 FAT 영역의 개수 2개



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	08	3E	18	ëX.MSDOS5.0...>.
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	00	A8	0F	00	.....ø..?.ÿ..¨..
00000020	00	A0	0F	00	E1	03	00	00	00	00	00	00	02	00	00	00	. ..á...........
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000040	80	00	29	A1	0B	39	C4	4E	4F	20	4E	41	4D	45	20	20	€.)¡.9ÄNO 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.|ˆN.ŠV@´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-ŠV@´.Í.s.
00000090	B9	FF	FF	8A	F1	66	0F	B6	C6	40	66	0F	B6	D1	80	E2	¹ÿÿŠñf.¶Æ@f.¶Ñ€â
000000A0	3F	F7	E2	86	CD	C0	ED	06	41	66	0F	B7	C9	66	F7	E1	?÷â†ÍÀi.Af.·Éf÷á
000000B0	66	89	46	F8	83	7E	16	00	75	38	83	7E	2A	00	77	32	f‰Føƒ~..u8ƒ~*.w2
000000C0	66	8B	46	1C	66	83	C0	0C	BB	00	80	B9	01	00	E8	2B	f‹F.fƒÀ.».€¹..è+

1개의 FAT 영역에 차지하는 Sector(993)



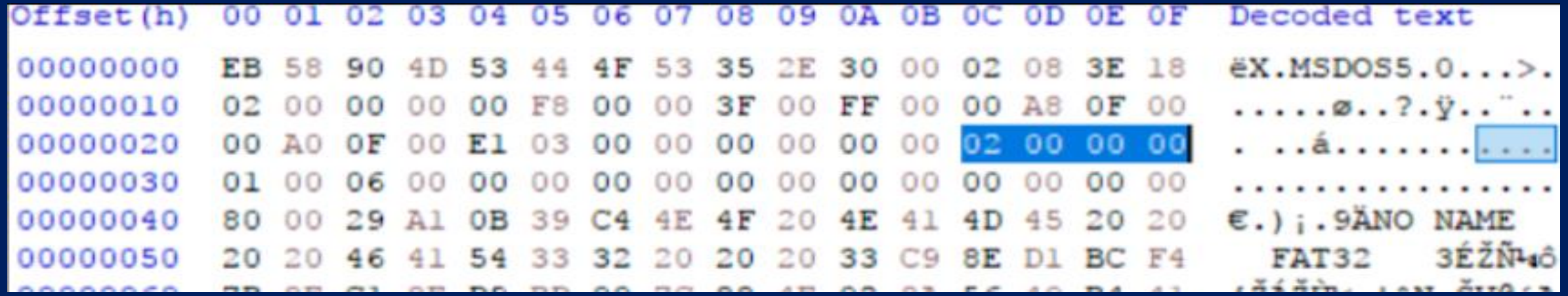
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	08	3E	18	ëX.MSDOS5.0...>.
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	00	A8	0F	00	.....ø..?.ÿ..¨..
00000020	00	A0	0F	00	E1	03	00	00	00	00	00	00	02	00	00	00	. ..á...........
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	....

두개의 FAT가 있고, 1개당 가지는 Sector의 수가 993개로

2 * 993 * 512 = 1,016,832 Byte 가 된다.

File Allocation Table

Root Directory Cluster number



Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	08	3E	18	ëX.MSDOS5.0...>.
00000010	02	00	00	00	00	F8	00	00	3F	00	FF	00	00	A8	0F	00	.....ø..?.ÿ..¨..
00000020	00	A0	0F	00	E1	03	00	00	00	00	00	00	02	00	00	00	. ..á...........
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000040	80	00	29	A1	0B	39	C4	4E	4F	20	4E	41	4D	45	20	20	€.)¡.9ÄNO NAME
00000050	20	20	46	41	54	33	32	20	20	20	33	C9	8E	D1	BC	F4	FAT32 3ÉŽÑ¼ô

예약된 영역

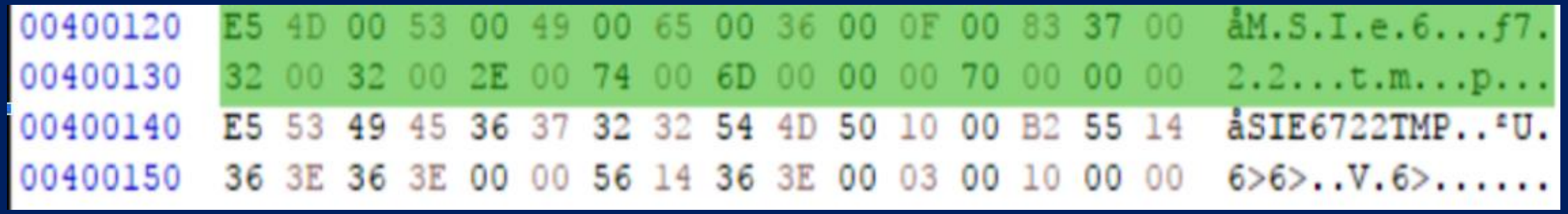
대부분은 여기가 Cluster2

Carving

Carving

Carving

Delete File



```
00400120  E5 4D 00 53 00 49 00 65 00 36 00 0F 00 83 37 00  åM.S.I.e.6...ƒ7.
00400130  32 00 32 00 2E 00 74 00 6D 00 00 00 70 00 00 00  2.2...t.m...p...
00400140  E5 53 49 45 36 37 32 32 54 4D 50 10 00 B2 55 14  åSIE6722TMP..²U.
00400150  36 3E 36 3E 00 00 56 14 36 3E 00 03 00 10 00 00  6>6>..V.6>......
```

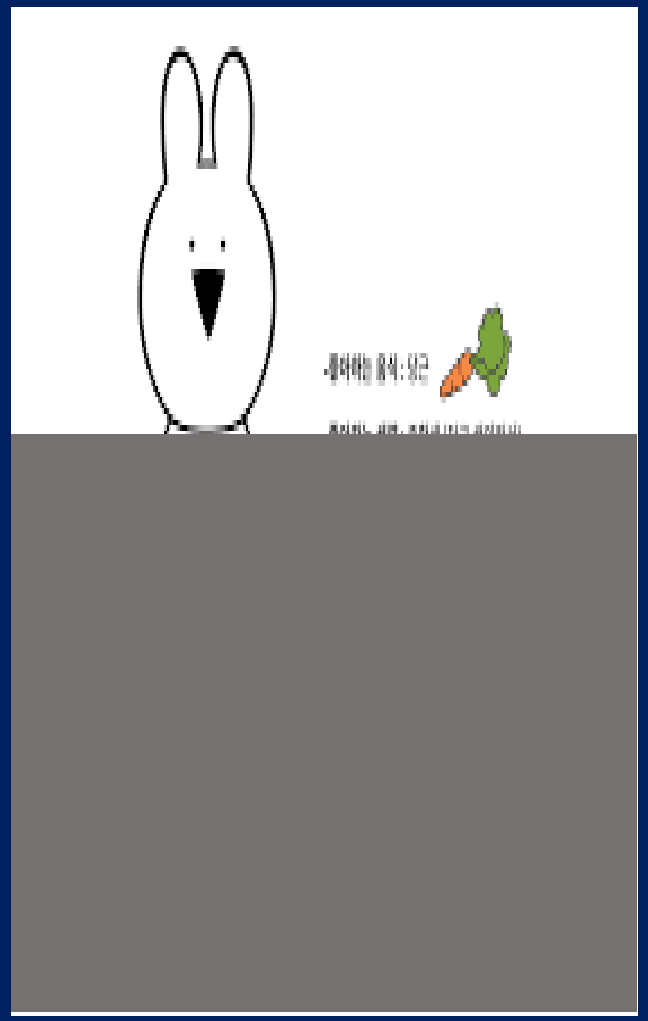
E5 로 첫 시그니처가 생긴다.
이를 이용해서 지워진 데이터의 Cluster를 알아내고 해당 Cluster를 복구해 내면 된다

Carving

확장자	시그니처
JPG	FF D8
PNG	89 50 4E 47
EXE (MS PE structure)	4D 5A
압축파일 (ZIP 등)	50 4B

시그니처 데이터데이터데이
터데이터데이터데이터데이
터 데이터데이터데이터데이
터데이터데이터데이터데이
터데이터데이터데이터데이
터데이터 데이터 데이터데
이터데이터데이터데 이터데
이터데이터데이터데이터데
이 터데이터데이터데이터데
이터데이터 데이터데이터데
이터데이터데이터데 이터데
이터데이터데이터 푸터 **의
미 없는 데이터 의미 없는
데이터 의미 없는 데이터 의
미 없는 데이터 의미 없는
데이터 의미 없는 데이터 의
미 없는 데이터 의미 없는
데이터**

Carving



데이터가 잘렸을 경우.
이럴 경우는 사실,,,
잘린 부분의 데이터는 못 살리지만
JPG 나 일부 파일의 경우 시그니처 와 상단부분
데이터만 볼 수 있는 경우가 있다.

Carving

중간에 데이터가 바뀐 경우
복구가 불가능 할 수도 있지만 가끔가
다가 예외의 경우가 나올 때 도 있다.

이상 허접한 발표 였습니다.