forensic-proof.com

File System/MBR/GPT

Twitter : @pr0neer

Blog : forensic-proof.com

Email : proneer@gmail.com

Kim Jinkook

개요

1. File System

- ✓ Introduction
- ✓ Sector Addressing
- ✓ Cluster & Block
- ✓ Slack Space
- ✓ Partition vs. Volume

2. MBR (Master Boot Record)

- ✓ Boot Code
- ✓ DOS Partition Table
- ✓ Partition

3. GPT (GUID Partition Table)

- ✓ Introduction
- ✓ Extensible Firmware Interface (EFI)
- ✓ OS Support of GPT
- ✓ GPT Layout
- ✓ GPT Structure
- ✓ Acquiring GPT disks and partitions
- ✓ GPT header & entries analysis tools
- ✓ GPT artifacts and reconstruction
- ✓ Digital forensics point of view

File System

Security is a people problem…

File System

Introduction

- 데이터는 파일 형태로 저장매체에 저장
- 저장매체의 공간이 커질 수록(파일 수 증가) ➔ 파일시스템 필요
- 압축, 암호화, 저널, 동적 할당, 다국어 지원 등 다양한 추가적인 기능 지원

저장매체	운영체제	파일시스템
디스크 장치	Windows	FAT(FAT12/16/32, exFAT), NTFS
	Linux	ext2/3/4
	Unix-like	UFS
	OS-2	HPFS
	Mac OS	HFS, HFS+
	Solaris	ZFS
	AIS	JFS
	IRIX	XFS
	HP-UX	ODS-5, VxFS
광학장치		ISO 9660, UDF

File System

Abstract Structure

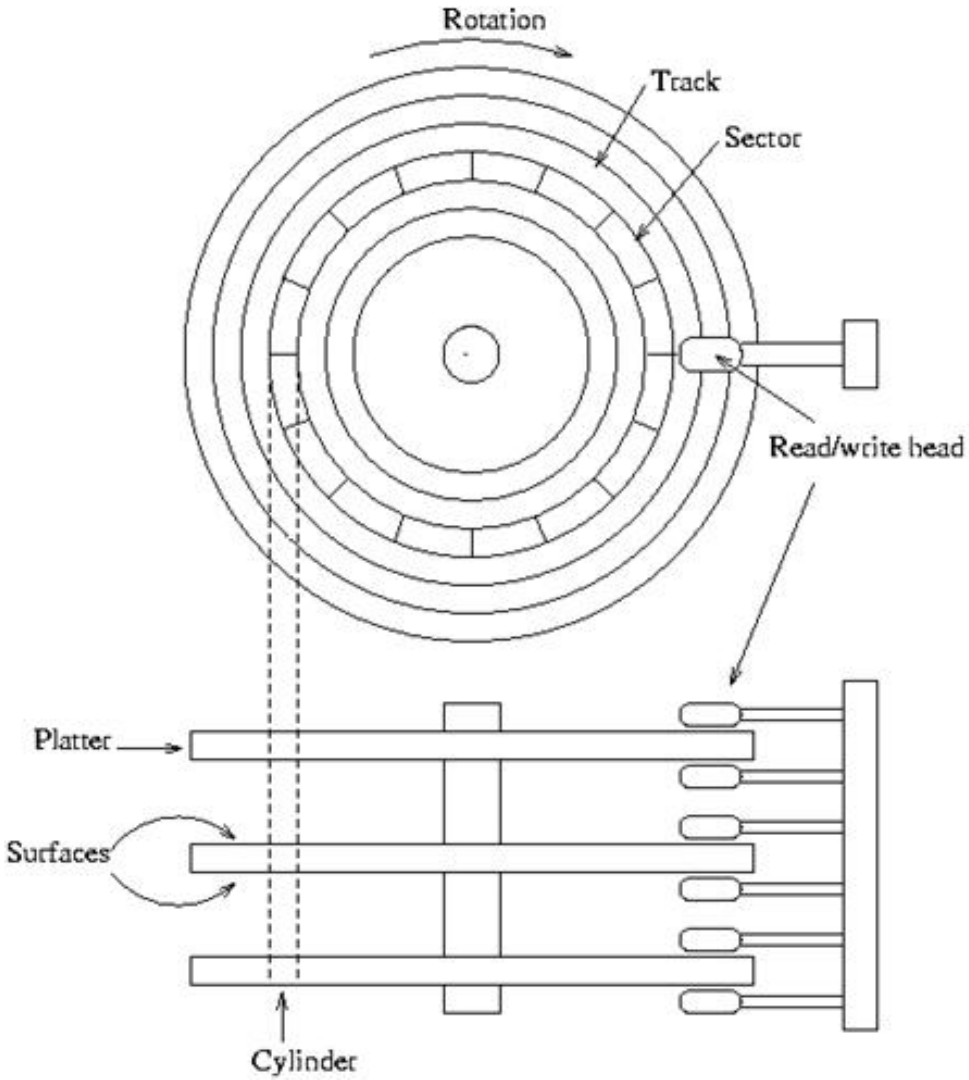
- 메타 영역과 데이터 영역으로 구분
- 메타 영역 : 파일의 속성, 이름, 크기, 시간 정보 등의 메타 정보
- 데이터 영역 : 파일의 실제 데이터

File System

Sector Addressing

- 하드디스크는 섹터라는 최소한의 데이터 저장단위 사용
- 섹터는 일반적으로 512 바이트
- 특정 파일에 접근하고자 할 경우 해당 파일이 위치한 섹터 주소로 접근해야 함
- **섹터 주소 지정 방식**
  - CHS (Cylinder-Head-Sector) 주소 지정 방식
  - LBA (Logical Block Addressing)

Sector Addressing – CHS cont.



Sector Addressing – CHS cont.

- 실린더(Cylinder), 헤드(Head), 섹터(Sector)의 물리적인 구조를 기반으로 주소 지정
- CHS(21, 3, 20) 주소에서 파일 읽기
  - 3번째 헤드를 21번째 실린더의 20번째 섹터로 이동 한 후 정해진 섹터만큼 읽기
- 용량 제한 발생

	할당비트 (실린더 수)	할당비트 (헤드 수)	할당비트 (섹터 수)	표현 가능한 최대 용량
IDE/ATA 표준	16 (65,536)	4 (16)	8 (256)	128 GB
BIOS INT 13h	10 (1,024)	8 (256)	6 (63)	7.88GB
최소 가능 비트	10 (1,024)	4 (16)	6 (63)	504 GB

- 2^{10} (1,024) (Cylinders) X 2^4 (16) (heads) X 2^6-1 (63) (Sectors) X 512 (sector size)= 528,482,304 (504 MB)
- 실린더, 헤드는 0부터 시작, 섹터는 1부터 시작

Sector Addressing – CHS

- BIOS보다 ATA 표준이 더 많은 수의 비트를 할당
- BIOS를 통해 전달되는 비트를 변환하여 지정함으로써 용량 증가
  - Large Mode 또는 Extended CHS (ECHS)
- 예) 웨스턴 디지털 (WD, Western Digital) 社의 Caviar AC33100

	실린더 수	헤드 수	섹터 수	표현 용량
IDE/ATA 표준	65,536	16	256	128 GB
Hard Disk Logical Geometry	6,136	16	63	2.95 GB
BIOS Translation Factor	Divide by 8	Multiply by 8	-	-
BIOS Translated Geometry	767	128	63	2.95 GB
BIOS INT 13h	1,024	256	63	7.88 GB

File System

Sector Addressing – LBA cont.

- HDD 용량 증가에 따라 CHS 방식을 대체하기 위한 방식
- CHS, LBA 모두 ATA-1 명세에 포함 ➔
  - CHS 가 먼저 사용되고 이후 LBA가 주목 받음
- 물리적인 구조와 상관없이 모든 섹터를 선형적으로 배열 (논리적인 주소)
  - 일렬로 늘어선 섹터의 주소는 0부터 시작
- 논리적인 주소로 특정 파일을 접근하고자 할 경우 물리적인 위치 값으로 변환 필요?
- LBA 등장으로 CHS 주소는 ATA-6 명세부터 사라짐
- 일부 소수의 임베디드 장비에서 CHS 주소 방식을 사용하기도 함

Sector Addressing – LBA cont.

- CHS ➔ LBA 변환

LBA = ((CYLINDER * heads per cylinder + HEAD) * sectors per track) + SECTOR -1

- LBA ➔ CHS

CYLINDER = LBA / (heads per cylinder * sectors per track)

HEAD = (LBA / sectors per track) % heads per cylinder

SECTOR = (LBA % sector per track) + 1

- 실제 변환을 위해서는 ZBR(Zone Bit Recording) 을 사용하는 환경도 고려

File System

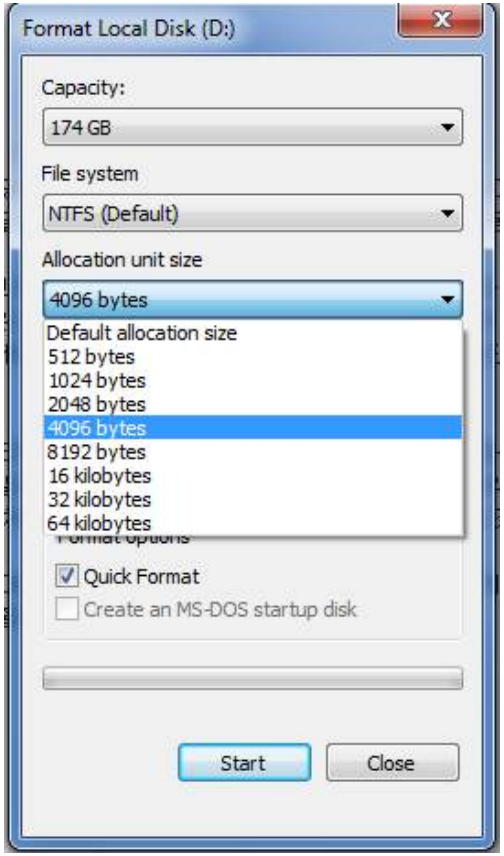
Sector Addressing – LBA

- LBA는 초기 ATA 표준에 28 비트를 할당 (4비트는 다른 용도로 사용)
- 2^{28} (268,435,456 sectors) X 512 (sector size) = 128 GB
- ATA-6 표준에서 48비트 LBA로 확장
  - 144 PB (Petabytes) = 144,000,000 GB
- 이 용량이 제약이 될 수 있을까?

File System

Cluster & Block cont.

- 데이터 관리의 효율을 위해 클러스터 또는 블록을 사용
- 디스크 I/O 명령을 줄이기 위해 ➔ 4MB 데이터를 쓰기 위해 4K(1,024 번), 512바이트 (8,192 번)



FAT32

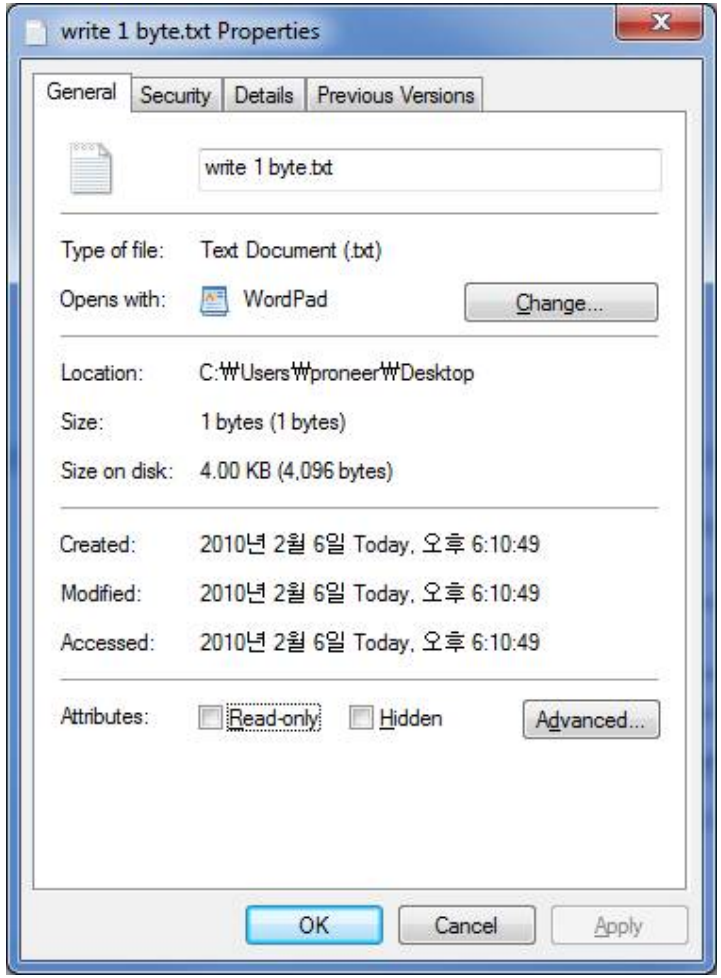
볼륨 크기	클러스터 크기
32MB – 8GB	4KB
8GB – 16GB	8KB
16GB – 32GB	16KB
32GB -	32KB

NTFS

볼륨 크기	클러스터 크기
7MB – 512MB	512Byte
513MB – 1GB	1KB
1GB – 2GB	2KB
2GB -	4KB

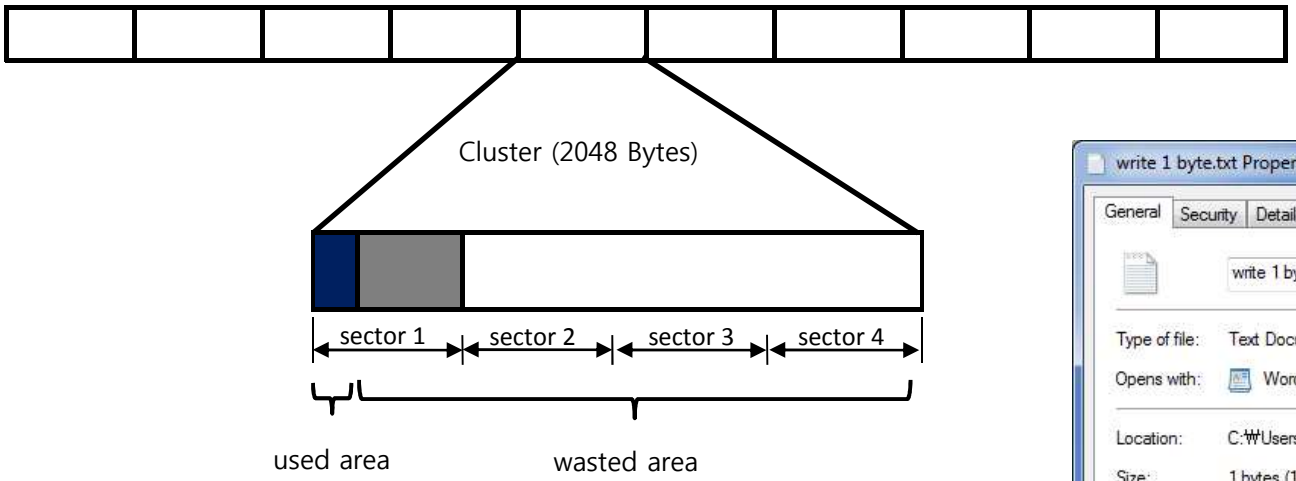
Cluster & Block cont.

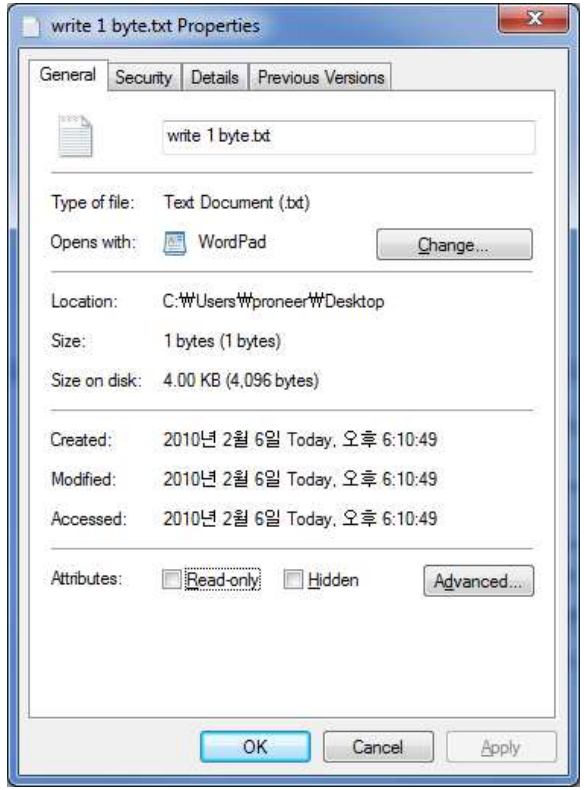
- 클러스터 크기를 알아보는 방법



Cluster & Block

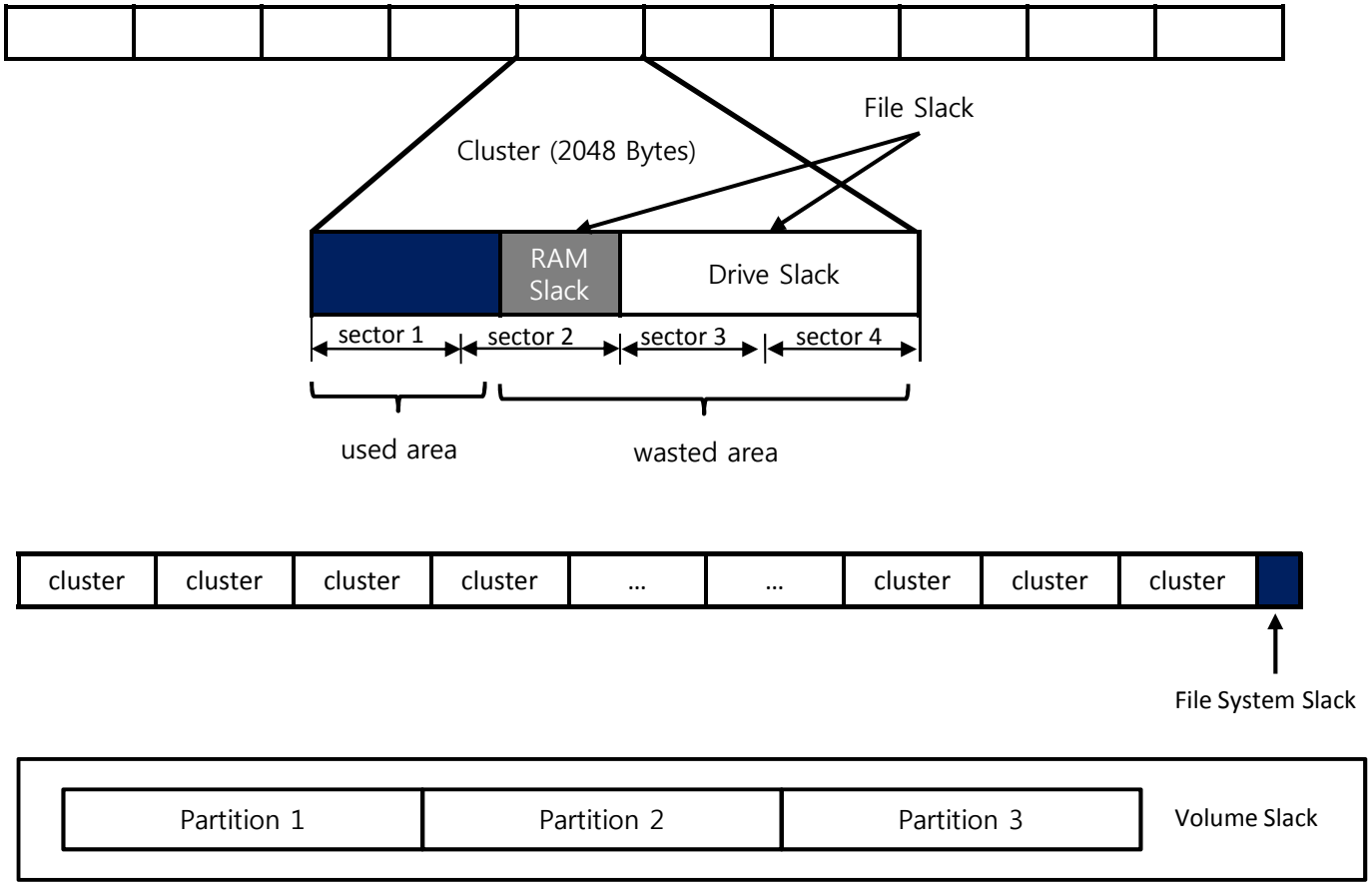
- 디스크 I/O의 효율 vs. 낭비되는 공간





Slack Space

- 물리적인 구조와 논리적인 구조의 차이로 인해 낭비되는 공간

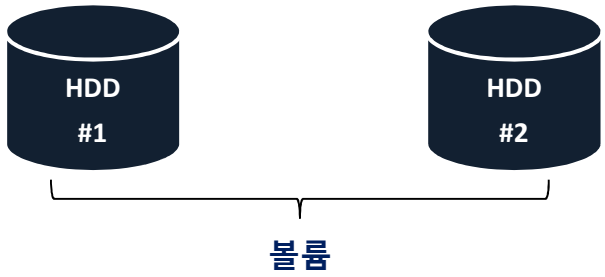


File System

Partition vs. Volume

- 파티션
  - 물리적으로 연속된 섹터들의 집합

- 볼륨
  - 논리적으로 연속된 섹터들의 집합



- 파티션 ⊂ 볼륨

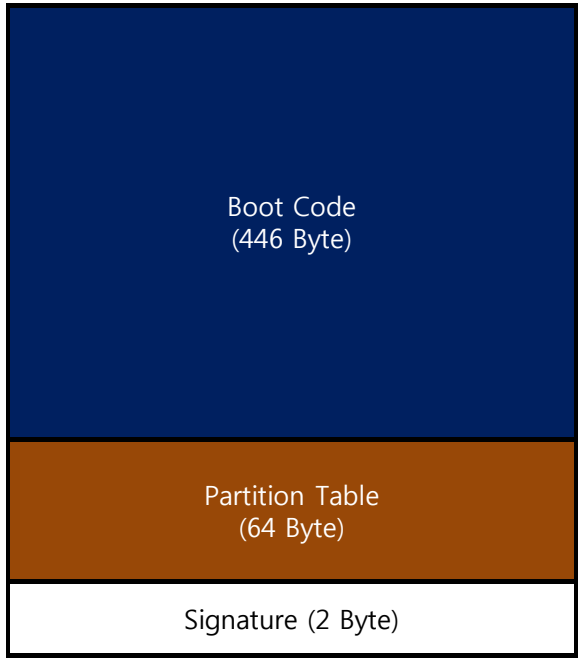
Master Boot Record

Security is a people problem…

Master Boot Record

MBR

- 저장매체 첫 번째 섹터 (LBA 0)에 위치하는 512 바이트 크기의 영역
- 부트 코드와 파티션 테이블로 구성



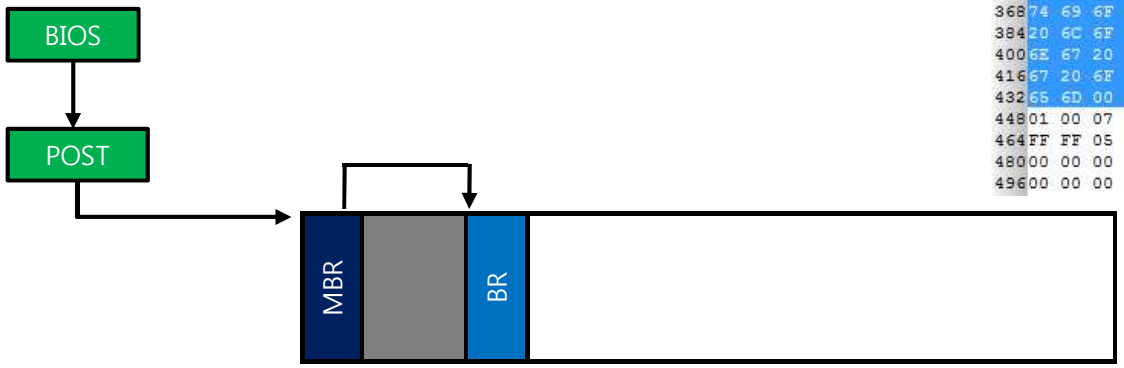
MBR 데이터 구조

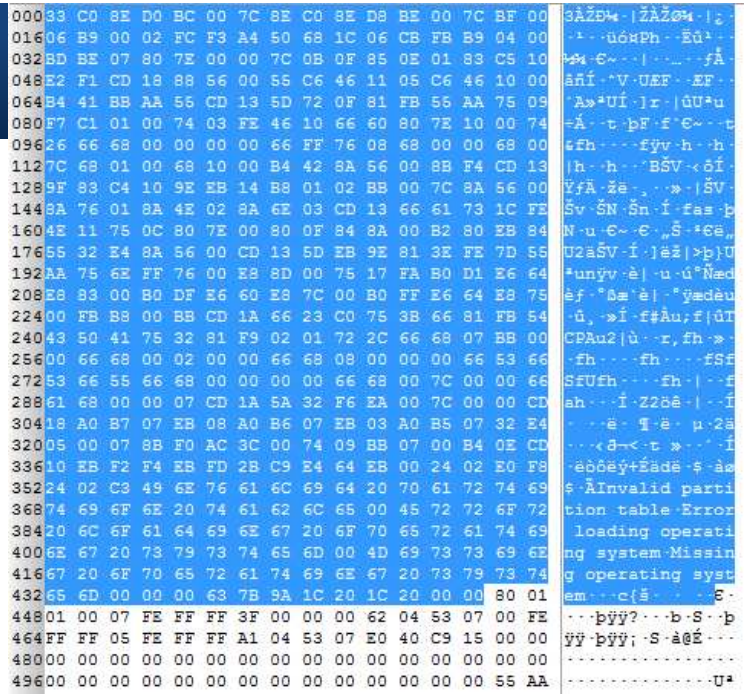
범 위		설명	크기
10 진수	16 진수		
0 – 445	0x0000 – 0x01BD	부트 코드	446 bytes
446 – 461	0x01BE – 0x01CD	파티션 테이블 엔트리 #1	16 bytes
462 – 477	0x01CE – 0X01DD	파티션 테이블 엔트리 #2	16 bytes
478 – 493	0x01DE – 0x01ED	파티션 테이블 엔트리 #3	16 bytes
494 – 509	0x01EE – 0x01FD	파티션 테이블 엔트리 #4	16 bytes
510 – 511	0x01FE – 0x01FF	시그니처 (0x55AA)	2 bytes

Master Boot Record

MBR Boot Code

- 부팅 시 POST 과정 후 저장매체 첫 섹터 호출
- 첫 섹터인 MBR은 자신의 부트 코드 수행

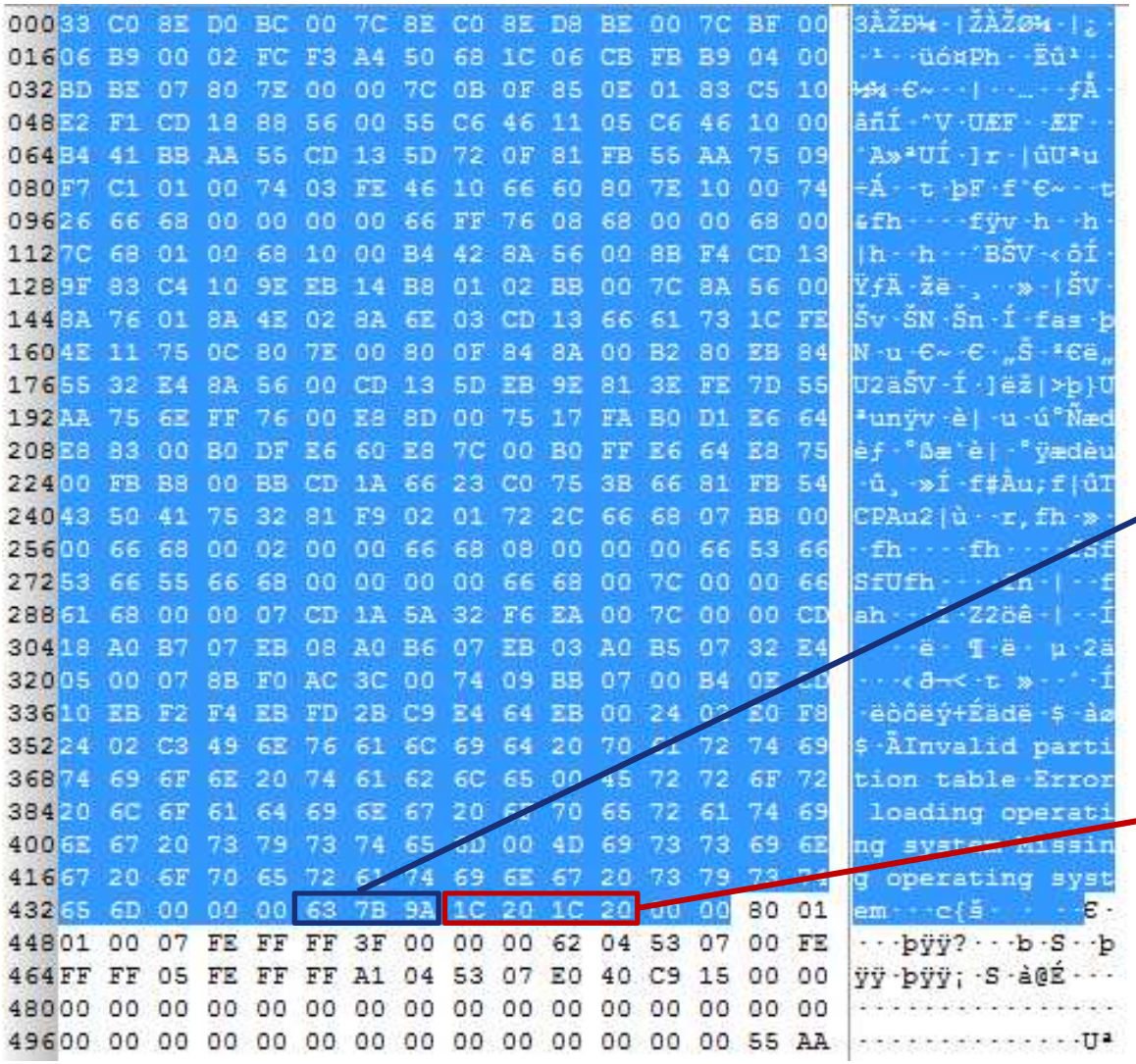




- 부트 코드 역할
  1. MBR 파티션 테이블에서 부팅 가능한 파티션 검색
  2. 부팅 가능한 파티션이 있을 경우, 해당 파티션의 BR(Boot Record)의 부트 섹터 호출
  3. 부팅 가능한 파티션이 없을 경우, 오류 메시지 출력
     1. Invalid partition table.
     2. Error loading operating system.
     3. Missing operating system.

Master Boot Record

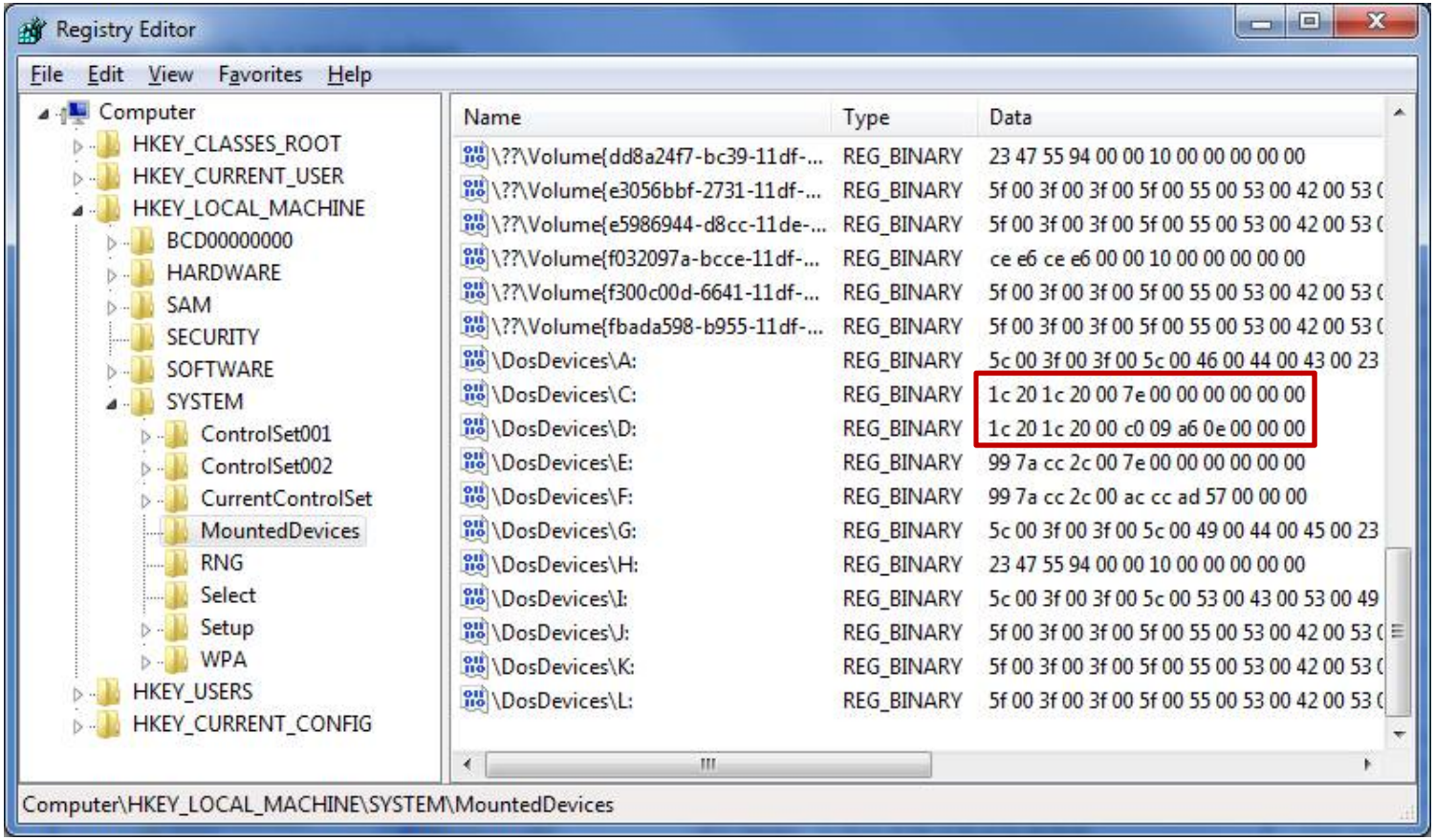
MBR Boot Code – Device GUID cont.



Master Boot Record

MBR Boot Code – Device GUID cont.

- 시스템에 어떤 장치가 마운트되면 레지스트리에 해당 장치의 GUID(Globally Unique ID) 저장
- HKEY_LOCAL_MACHINE\SYSTEM\MountedDevices



Master Boot Record

MBR Boot Code – Device GUID

- HKEY_LOCAL_MACHINE\SYSTEM\MountedDevices
  - **\DosDevices\C:** ➔ 1c 20 1c 20 00 7e 00 00 00 00 00 00
  - **\DosDevices\D:** ➔ 1c 20 1c 20 00 c0 09 a6 0e 00 00 00
- 나머지 8 바이트의 의미는?

Master Boot Record

MBR DOS Partition Table cont.

```
000 33 C0 8E D0 BC 00 7C FB 50 07 50 1F FC BE 1B 7C  3ÀŽÐ¼·|ûP·P·ü¾·|
016 BF 1B 06 50 57 B9 E5 01 F3 A4 CB BD BE 07 B1 04  ¿··PW¹å·ó¤Ë½¾·±·
032 38 6E 00 7C 09 75 13 83 C5 10 E2 F4 CD 18 8B F5  8n·| u·ƒÅ·âôÍ·‹õ
048 83 C6 10 49 74 19 38 2C 74 F6 A0 B5 07 B4 07 8B  ƒÆ·It·8,tö µ·´·‹
064 F0 AC 3C 00 74 FC BB 07 00 B4 0E CD 10 EB F2 88  ð¬<·tü»··´·Í·ëò^
080 4E 10 E8 46 00 73 2A FE 46 10 80 7E 04 0B 74 0B  N·èF·s*þF·€~··t·
096 80 7E 04 0C 74 05 A0 B6 07 75 D2 80 46 02 06 83  €~··t· ¶·uÒ€F··ƒ
112 46 08 06 83 56 0A 00 E8 21 00 73 05 A0 B6 07 EB  F··ƒV ·è!·s· ¶·ë
128 BC 81 3E FE 7D 55 AA 74 0B 80 7E 10 00 74 C8 A0  ¼|>þ}Uªt·€~··tÈ
144 B7 07 EB A9 8B FC 1E 57 8B F5 CB BF 05 00 8A 56  ··ë©‹ü·W‹õË¿··ŠV
160 00 B4 08 CD 13 72 23 8A C1 24 3F 98 8A DE 8A FC  ·´·Í·r#ŠÁ$?ŠÞŠü
176 43 F7 E3 8B D1 86 D6 B1 06 D2 EE 42 F7 E2 39 56  C÷ã‹Ñ†Ö±·ÒîB÷â9V
192 0A 77 23 72 05 39 46 08 73 1C B8 01 02 BB 00 7C   w#r·9F·s·¸··»·|
208 8B 4E 02 8B 56 00 CD 13 73 51 4F 74 4E 32 E4 8A  ‹N·‹V·Í·sQOtN2äŠ
224 56 00 CD 13 EB E4 8A 56 00 60 BB AA 55 B4 41 CD  V·Í·ëäŠV·`»ªU´AÍ
240 13 72 36 81 FB 55 AA 75 30 F6 C1 01 74 2B 61 60  ·r6|ûUªu0öÁ·t+a`
256 6A 00 6A 00 FF 76 0A FF 76 08 6A 00 68 00 7C 6A  j·j·ÿv ÿv·j·h·|j
272 01 6A 10 B4 42 8B F4 CD 13 61 61 73 0E 4F 74 0B  ·j·´B‹ôÍ·aas·Ot·
288 32 E4 8A 56 00 CD 13 EB D6 61 F9 C3 49 6E 76 61  2äŠV·Í·ëÖaùÃInva
304 6C 69 64 20 70 61 72 74 69 74 69 6F 6E 20 74 61  lid partition ta
320 62 6C 65 00 45 72 72 6F 72 20 6C 6F 61 64 69 6E  ble·Error loadin
336 67 20 6F 70 65 72 61 74 69 6E 67 20 73 79 73 74  g operating syst
352 65 6D 00 4D 69 73 73 69 6E 67 20 6F 70 65 72 61  em·Missing opera
368 74 69 6E 67 20 73 79 73 74 65 6D 00 00 00 00 00  ting system·····
384 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ················
400 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ················
416 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ················
432 00 00 00 00 00 2C 44 63 99 7A CC 2C 00 00 80 01  ·····,Dc™zÌ,··€·
448 01 00 07 FE FF FF 3F 00 00 00 D8 E5 D6 2B 00 00  ···þÿÿ?···ØåÖ+··
464 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ················
480 C1 FF 05 FE FF FF 17 E6 D6 2B 2A 66 61 0E 00 00  Áÿ·þÿÿ·æÖ+*fa···
496 00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA  ··············Uª
```

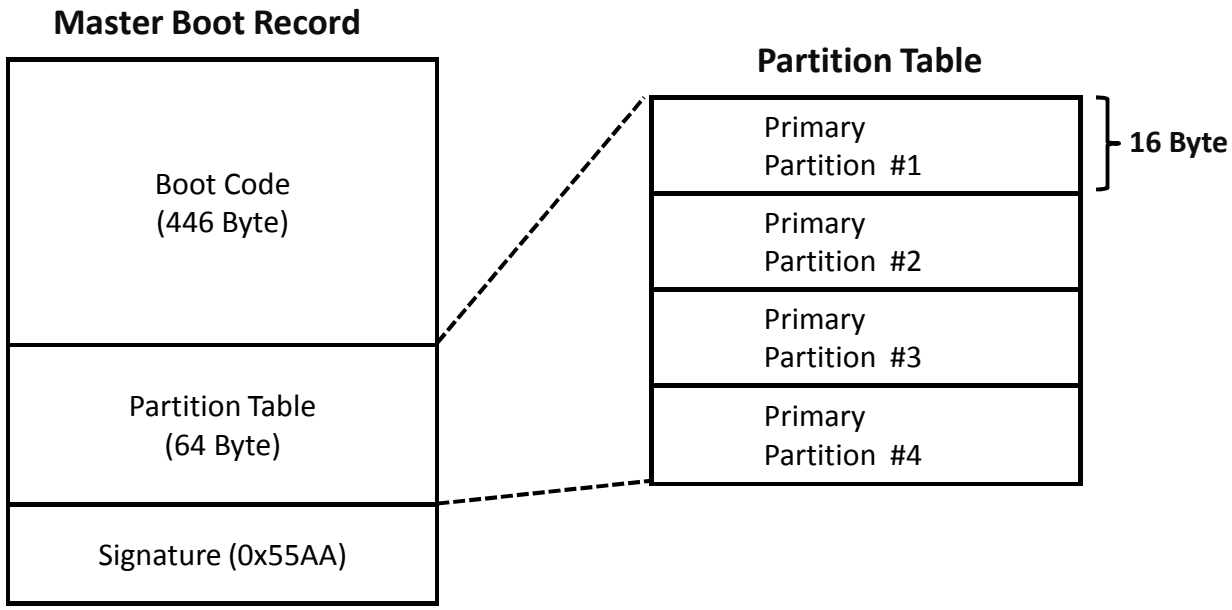
DOS Partition Table

Master Boot Record

MBR DOS Partition Table cont.

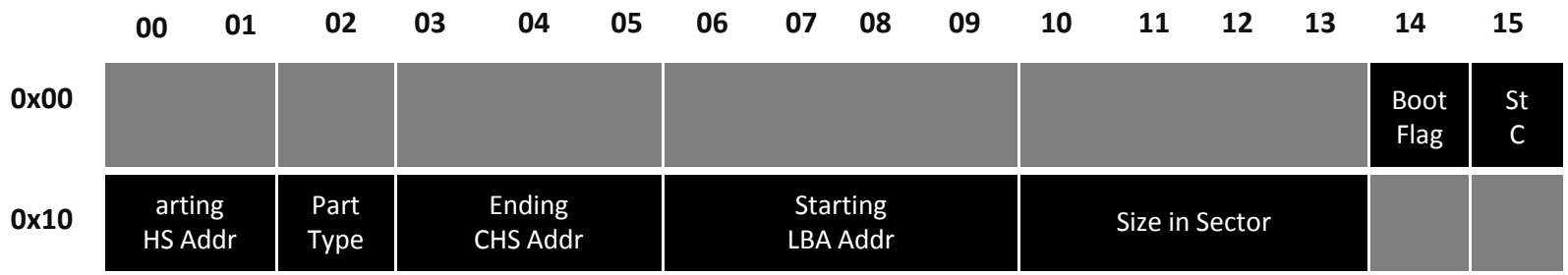
- ***Primary Partition***
- ***Extended Partition***
- ***Logical Partition***

위치	크기	설명
0	446	Boot Code
446	16	Partition #1
462	16	Partition #2
478	16	Partition #3
494	16	Partition #4
510	2	Signature (0x55AA)



Master Boot Record

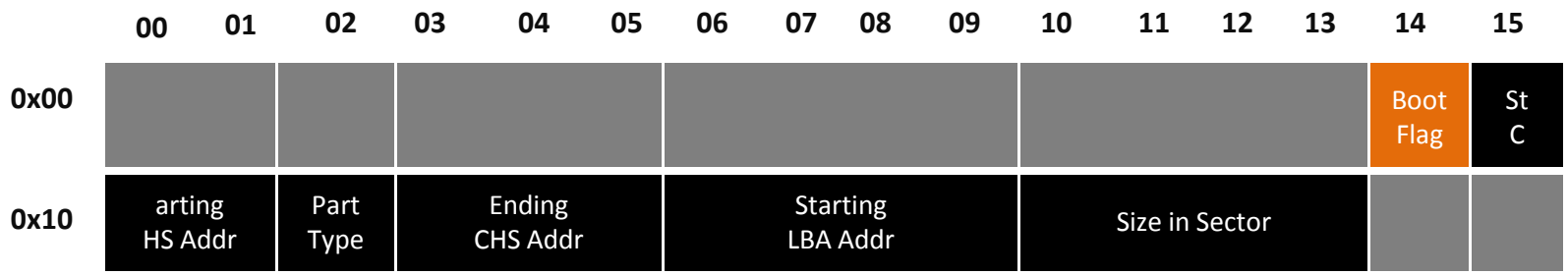
MBR DOS Partition Table cont.

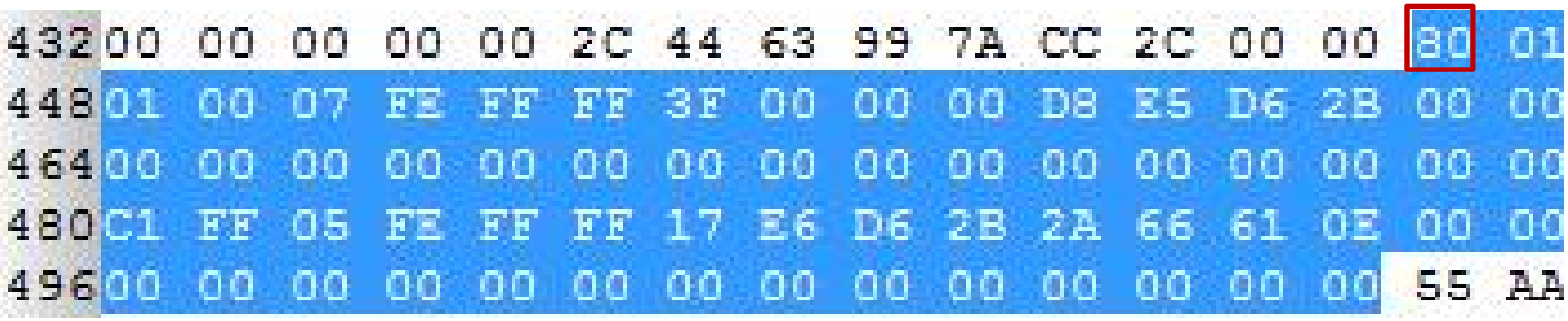



```
432 00 00 00 00 00 2C 44 63 99 7A CC 2C 00 00 80 01
448 01 00 07 FE FF FF 3F 00 00 00 D8 E5 D6 2B 00 00
464 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
480 C1 FF 05 FE FF FF 17 E6 D6 2B 2A 66 61 0E 00 00
496 00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA
```

Master Boot Record

MBR DOS Partition Table cont.

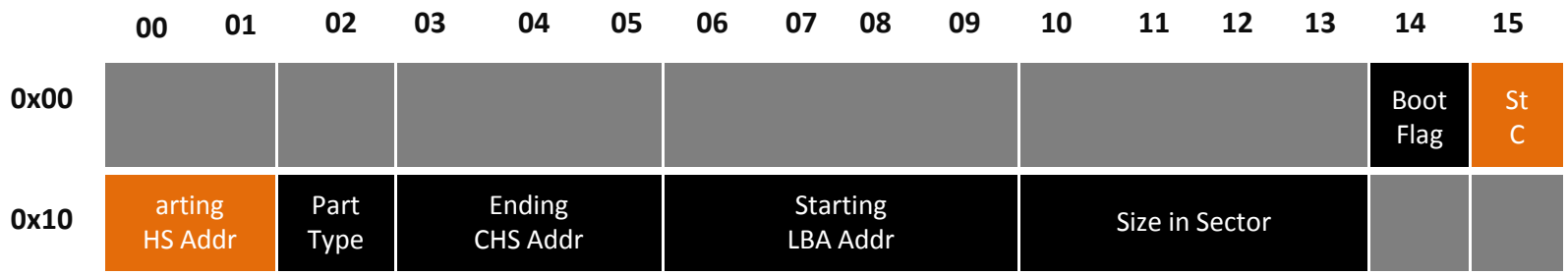




- 부트 플래그(Boot Flag) : 부팅 가능한 저장매체인지를 여부
  - 0x80 : 부팅 가능
  - 0x00 : 부팅 불가능

Master Boot Record

MBR DOS Partition Table cont.

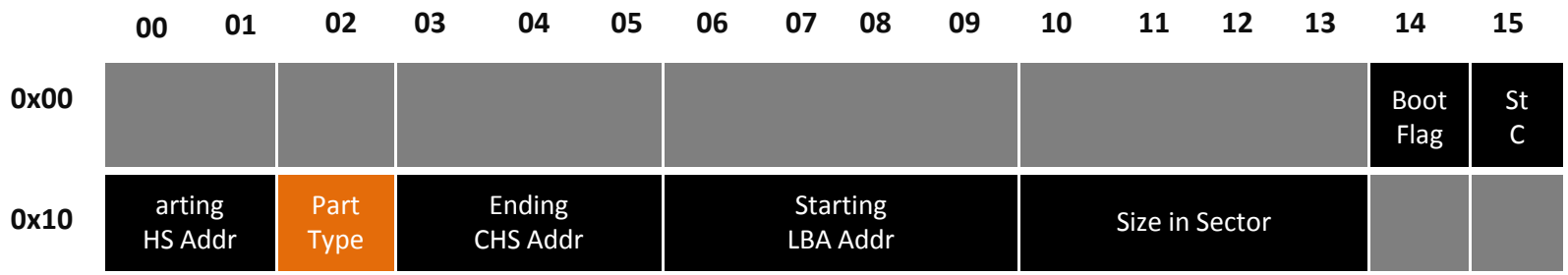


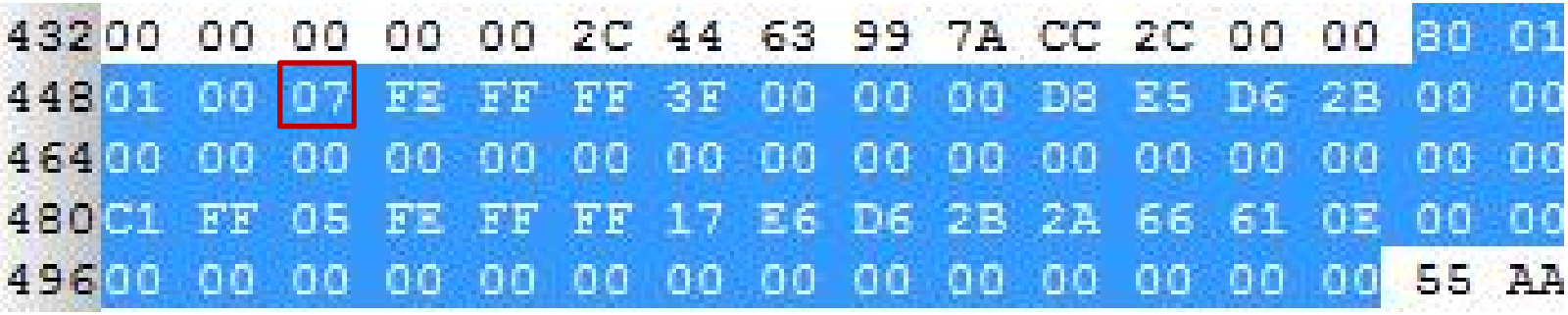

```
432 00 00 00 00 00 2C 44 63 99 7A CC 2C 00 00 80 01
448 01 00 07 FE FF FF 3F 00 00 00 D8 E5 D6 2B 00 00
464 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
480 C1 FF 05 FE FF FF 17 E6 D6 2B 2A 66 61 0E 00 00
496 00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA
```

- 시작 CHS 주소 (Starting CHS Address) : 주소지정방식이 CHS일 경우 파티션의 시작 위치
  - 0x000101

Master Boot Record

MBR DOS Partition Table cont.





- 파티션 유형 (Partition Type) : 해당 파티션의 유형 (0x00 – 0xFF)
  - 0x07
- 파티션 유형 값 조작으로 숨긴 파티션 생성 가능 (실습)

Master Boot Record

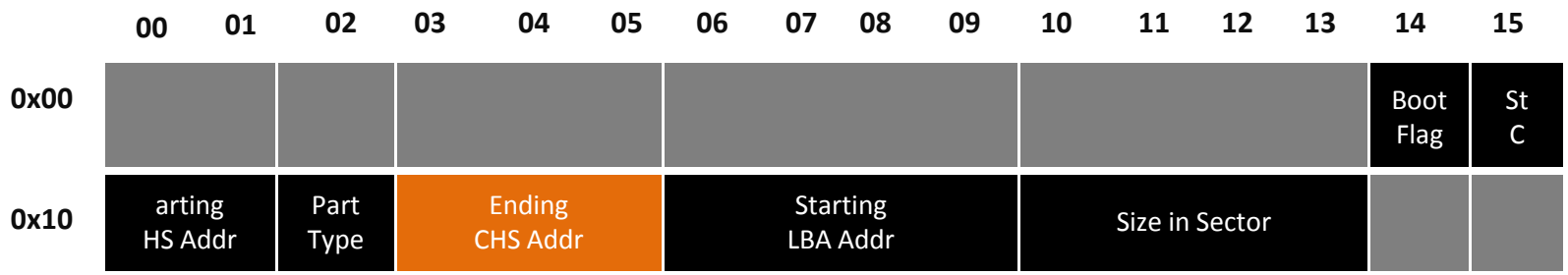
MBR DOS Partition Table cont.

- 파티션 유형 (Partition Type)

값 (16진수)	설명
00h	Empty
01h	DOS 12-bit FAT, CHS
02h	XENIX root file system, CHS
03h	XENIX /usr file system (obsolete)
04h	DOS 16-bit FAT (up to 32M), CHS
05h	DOS 3.3+ extended partition, CHS
06h	DOS 3.31+ Large File System (16-bit FAT, over 32M), CHS
07h	Advanced Unix, exFAT, NTFS
08h	OS/2 (V1.0 – 1.3 only), AIX bootable partition, Commodore DOS, DELL partition spanning multiple drives
09h	AIX data partition
0Ah	OPUS, Coherent swap partition, OS/2 Boot Manager
0Bh	Windows 95 with 32-bit FAT, CHS
0Ch	Windows 95 with 32-bit FAT (using LBA-mode INT 13 extensions), LBA
0Dh	-
…	…
FEh	LANstep, IBM PS/2 IML
FFh	XENIX bad block table

Master Boot Record

MBR DOS Partition Table cont.

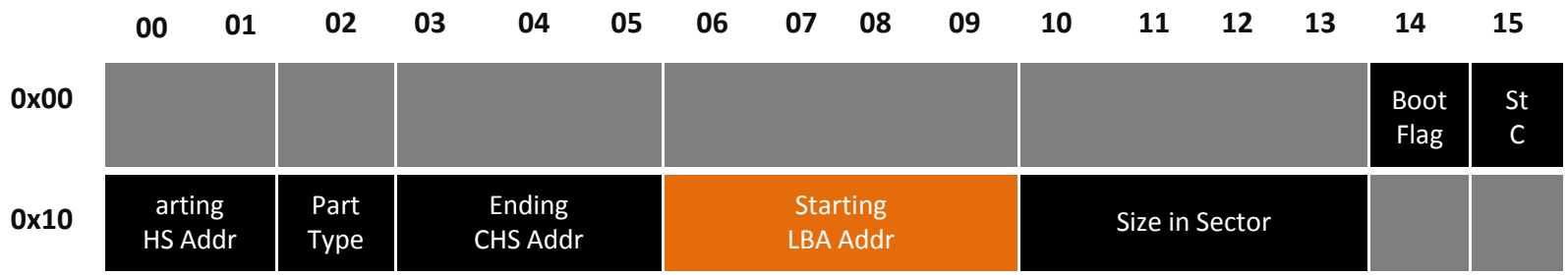



```
432 00 00 00 00 00 2C 44 63 99 7A CC 2C 00 00 80 01
448 01 00 07 FE FF FF 3F 00 00 00 D8 E5 D6 2B 00 00
464 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
480 C1 FF 05 FE FF FF 17 E6 D6 2B 2A 66 61 0E 00 00
496 00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA
```

- 마지막 CHS 주소 (Ending CHS Address) : 주소지정방식이 CHS일 경우 파티션의 끝 위치
  - 0xFFFFFE

Master Boot Record

MBR DOS Partition Table cont.

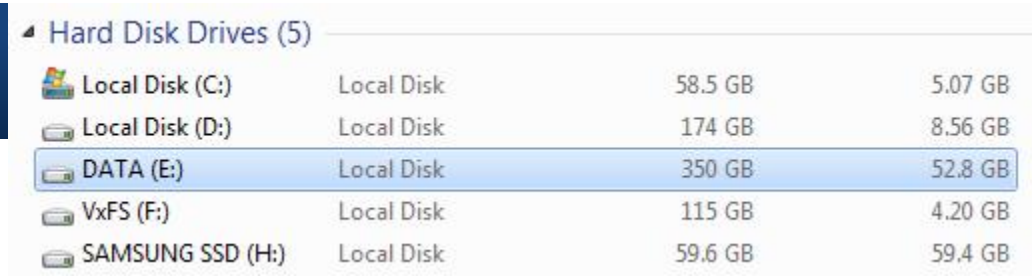


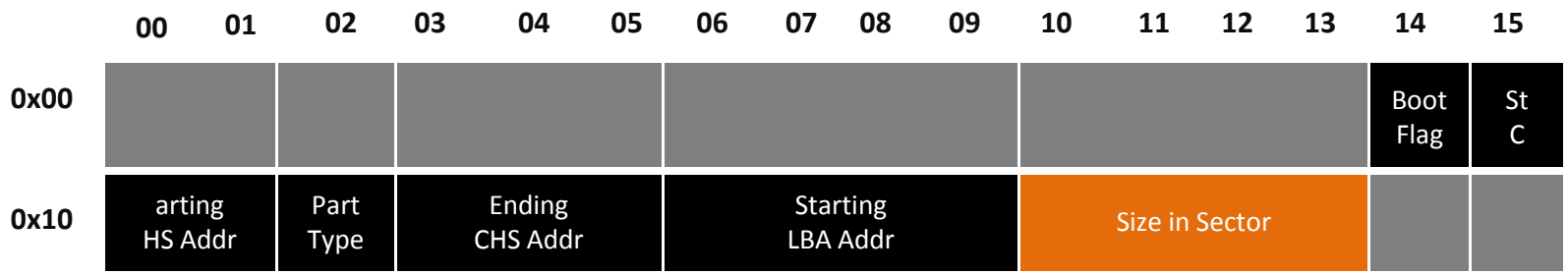

```
432 00 00 00 00 00 2C 44 63 99 7A CC 2C 00 00 80 01
448 01 00 07 FE FF FF 3F 00 00 00 D8 E5 D6 2B 00 00
464 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
480 C1 FF 05 FE FF FF 17 E6 D6 2B 2A 66 61 0E 00 00
496 00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA
```

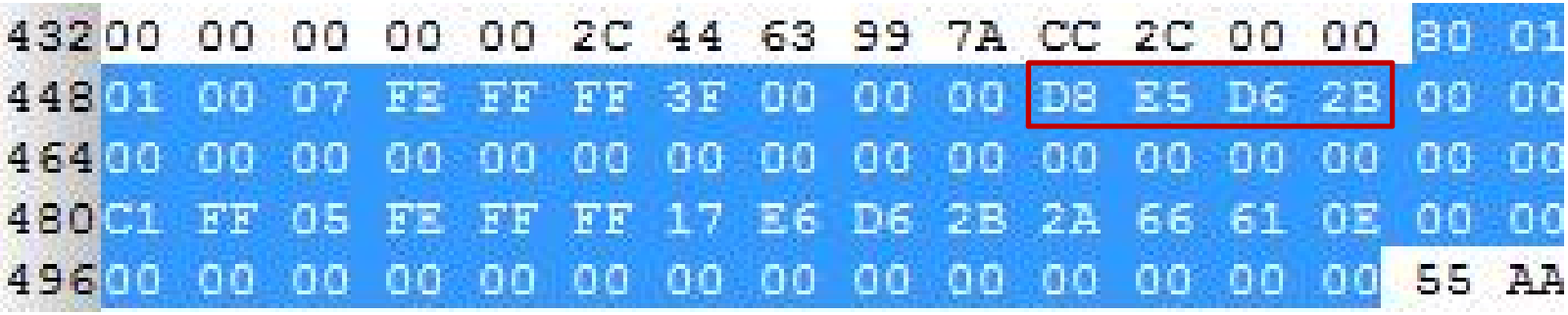
- 시작 LBA 주소 (Starting LBA Address) : 주소지정방식이 LBA일 경우, 파티션의 시작 섹터 위치
  - 0x0000003F : 63
- MBR을 사용하는 모든 저장매체의 첫 파티션의 시작 위치는 63 섹터, 이유는?

Master Boot Record

MBR DOS Partition Table cont.







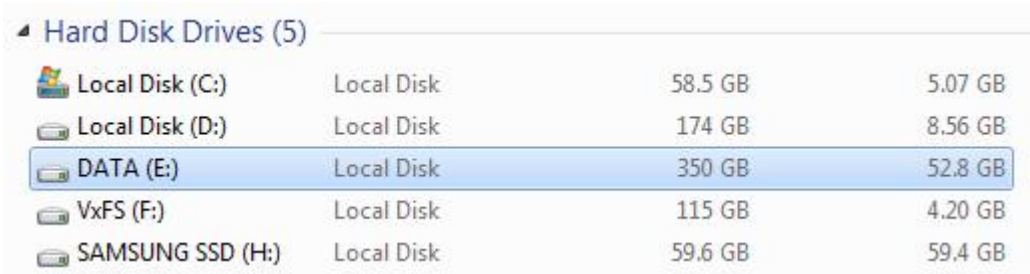
- 파티션 섹터 수 (Size in Sector) : 파티션(LBA)에 할당된 섹터의 총 수
  - 0x2BD6E5D8 X 512 (sector size) = 376,577,961,984 (350 GB)
- DOS 파티션이 인식할 수 있는 파티션 최대 크기는?
  - 2^{32} (4,294,967,295) X 512 = 2,199,023,255,552 = **2 TB**

Master Boot Record

MBR DOS Partition Table

```
432 00 00 00 00 00 2C 44 63 99 7A CC 2C 00 00 80 01
448 01 00 07 FE FF FF 3F 00 00 00 D8 E5 D6 2B 00 00
464 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
480 C1 FF 05 FE FF FF 17 E6 D6 2B 2A 66 61 0E 00 00
496 00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA
```

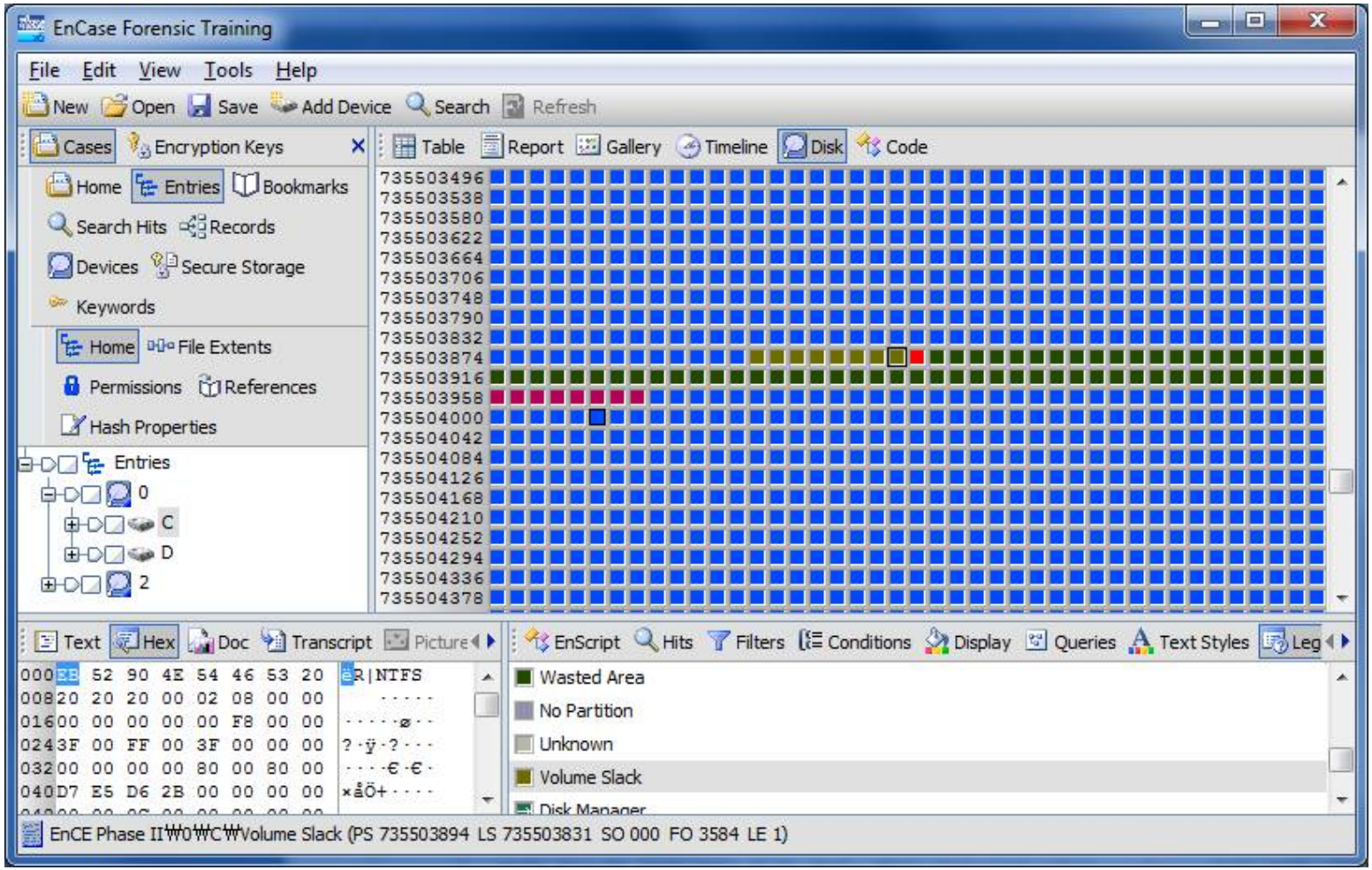
Partition	Boot Flag	Starting CHS Address	Partition Type	Ending CHS Address	Starting LBA Address	Size in Sector
#1	0x80	0x000101	0x07	0xFFFFFE	0x0000003F (63)	0x2BD6E5D8 (735,503,832; 350 GB)
#2	0x00	0x000000	0x00	0x000000	0x00000000 (00)	0x00000000
#3	0x00	0xFFC100	0x05	0xFFFFFE	0x2BD6E617 (735,503,895)	0x0E61662A (241,264,170; 115 GB)
#4	0x00	0x000000	0x00	0x000000	0x00000000 (00)	0x00000000



Master Boot Record

EnCase Disk View

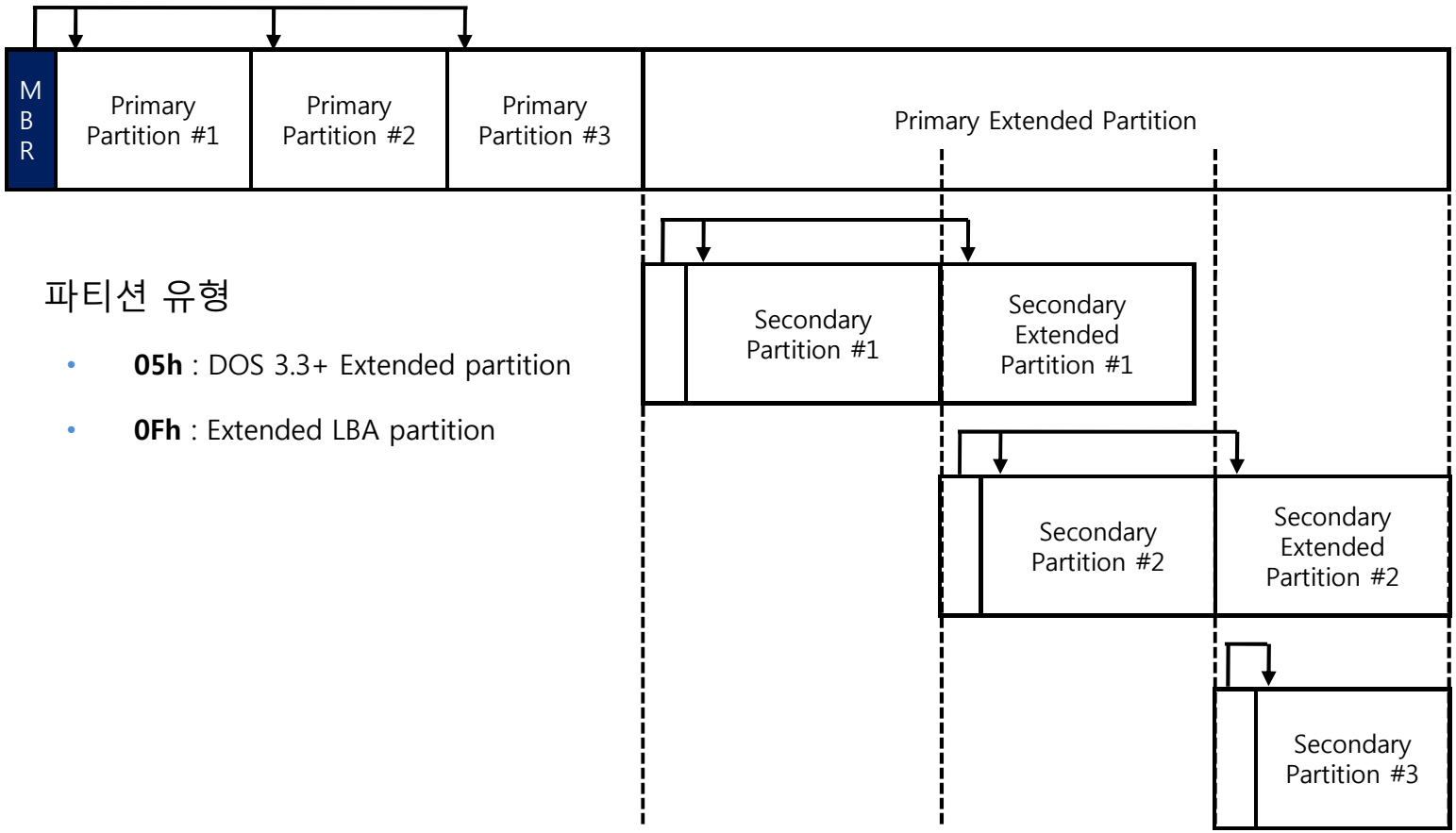
- 파티션 테이블에는 파티션이 연속인데 볼륨 슬랙(BR 포함)이 존재하는 이유는?



Master Boot Record

Partition cont.

- 주 파티션과 확장, 논리 파티션과의 관계



- 파티션 유형
  - **05h** : DOS 3.3+ Extended partition
  - **0Fh** : Extended LBA partition

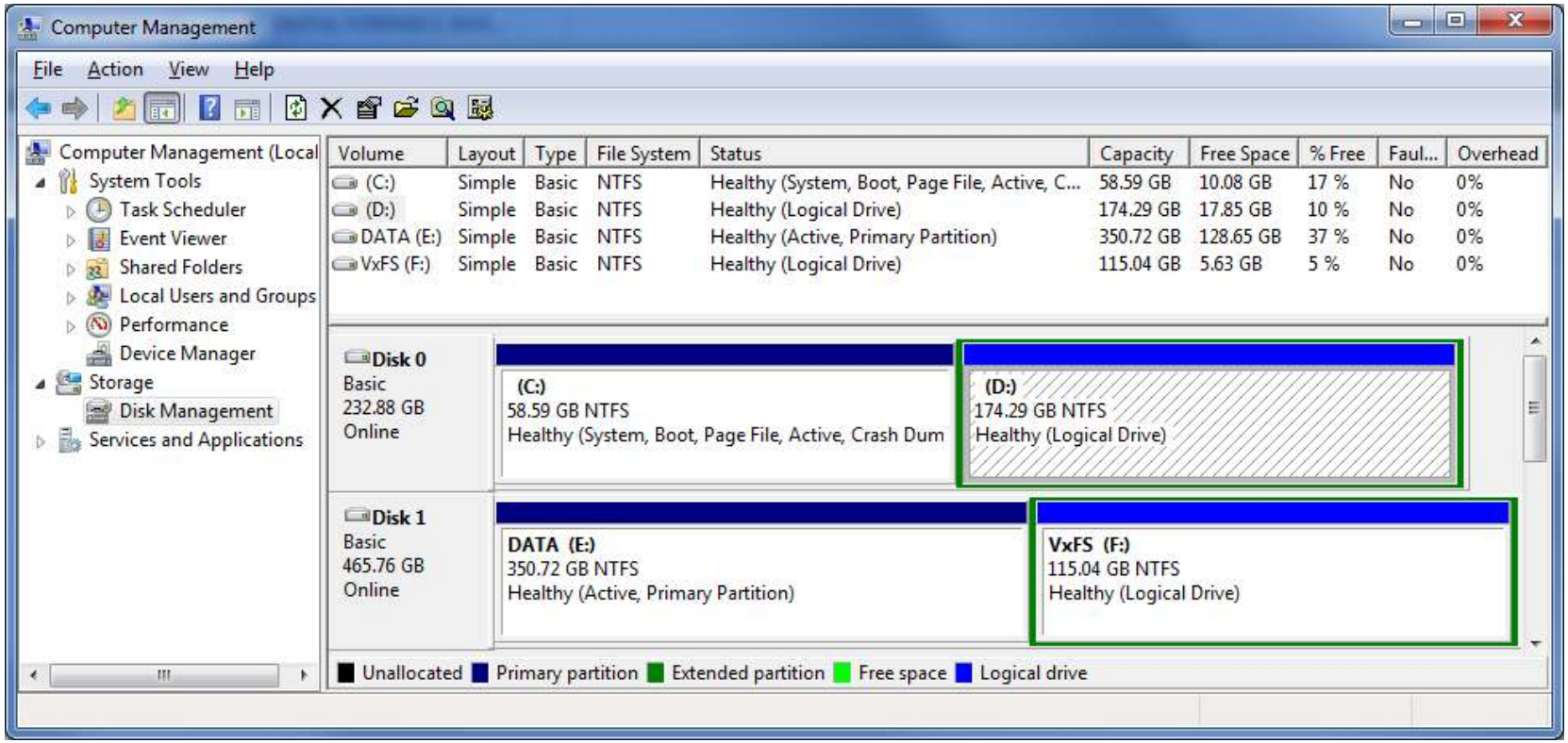
Master Boot Record

Partition cont.

- 파티션 사용의 이점
  - 시스템 파티션과 분할하여 데이터 저장 용도나 백업용으로 사용 ➔ 시스템 파티션만 포맷
  - 하나의 시스템에 다양한 운영체제를 설치 (멀티부팅)
  - NTFS의 경우 MFT 크기를 감소시켜 성능 향상 (경우에 따라 다름)
  - 파일 탐색의 경우, 헤드 움직임 감소 ➔ 탐색 시간 향상

Master Boot Record

Partition



GUID Partition Table

Security is a people problem…

GUID Partition Table

Introduction

- BIOS를 통한 DOS/MBR 파티션 테이블의 한계
  - 2^{32} (4,294,967,295) X 512 = 2,199,023,255,552 = 2 TB
- 인텔에서 BIOS를 대체할 수단으로 Extensible Firmware Interface (EFI) 표준 제안
- 개선된 EFI 펌웨어에서 지원하는 파티션 테이블 형식 ➔ GPT
- 단순한 파티션 테이블 외에 다양한 디스크 정보 저장
- 1980년 대 : MBR 파티션 발표
- 1990년대 후반 : Unified EFI(UEFI)의 부분으로 새로운 파티션 테이블 방법 개발
- 현재 : GPT가 UEFI 세부 명세에 포함

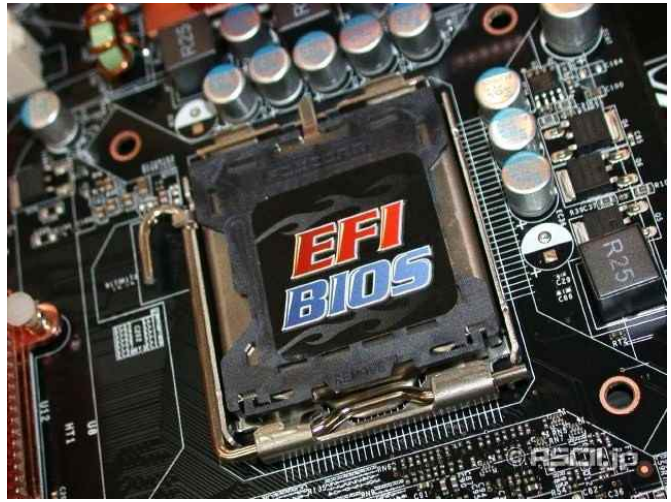
GUID Partition Table

GPT Disk benefits

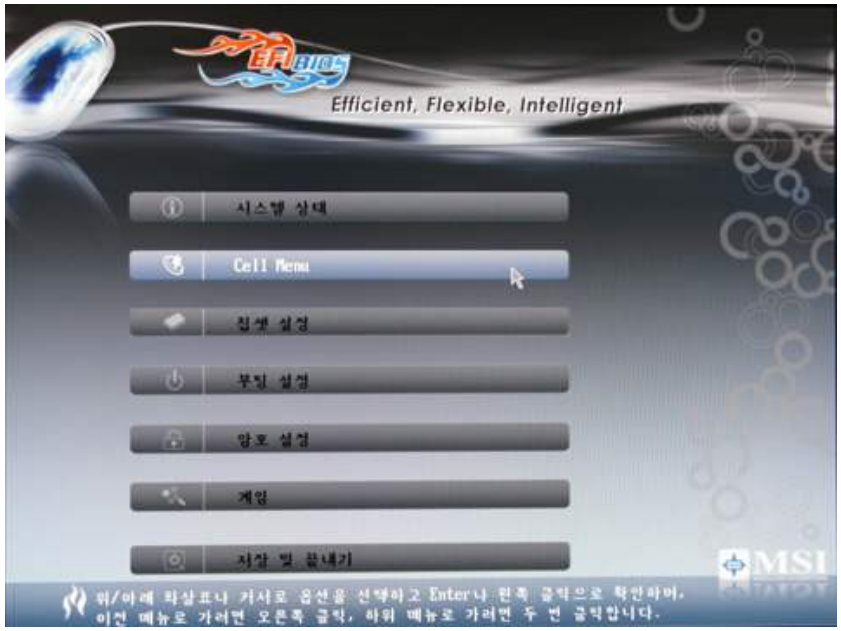
- 128개의 주(primary) 파티션 생성 가능 (MBR은 4개만 가능)
- 대용량의 볼륨 지원
  - MBR은 32비트 파티션 크기 지원 : (0xFFFF FFFF) = 2 Terabytes (2^{40})
  - GPT는 64비트 파티션 크기 지원 : (0xFFFF FFFF FFFF FFFF) = 8 Zettabytes (20^{70})
- CRC (cyclical Redundancy Check)를 이용해 파티션 테이블 보호 ➔ 신뢰성 확보
- x64 기반의 플랫폼에서 사용 가능
- GPT의 중요 데이터 구조는 볼륨의 끝에 복제본 저장 ➔ 장애 복구 가능

GUID Partition Table

Extensible Firmware Interface (EFI)



- 운영체제와 하드웨어 펌웨어 사이의 새로운 인터페이스
  - BIOS (Basic Input/Output System) 대체
- 초기에는 인텔에서 개발, 현재는 통합 (Unified) EFI로 발전
- 주요 특징
  - GUI 인터페이스
  - 마우스 사용 가능
  - Pre-OS 소프트웨어 구동 가능
    - 시스템 복구 기능
    - 인터넷 브라우저 등
  - 네트워크 기능
  - 다국어(한국어 포함) 지원



GUID Partition Table

OS support of GPT

- 대부분의 유닉스 기반 운영체제에서 GPT 기반 부팅을 지원

Unix-class Operating System

OS	Version/Edition	Platform	Boot from GPT on PC/BIOS	Boot from GPT on EFI
FreeBSD	Since 7.0	x86, x86-64	Yes	Yes
Linux	Fedora 8+, Ubuntu 8.04+	x86, x86-64, IA-64	Yes	Yes
Mac OS X	Since 10.4.0	x86, x86-64	Yes with bootloader (hackintosh)	Yes
Solaris	Since Solaris 10	x86, x86-64, SPARC	No	No

http://en.wikipedia.org/wiki/GUID_Partition_Table#cite_note-2

GUID Partition Table

OS support of GPT

- 마이크로소프트는 32비트 플랫폼에 EFI를 지원하지 않고, GPT를 이용한 부팅도 지원하지 않음

Windows 32-bit version

OS	Version/Edition	Platform	Boot from GPT on PC/BIOS	Boot from GPT on EFI
Windows XP	(2001-10-25)	x86	No	No
Windows Server 2003	(2003-04-24)	x86	No	No
Windows Server 2003	Service Pack 1 (2005-03-30)	x86	No	No
Windows Vista	(2005-09-22)	x86	No	No
Windows Server 2008	(2008-02-27)	x86	No	No
Windows 7	(2009-10-22)	x86	No	No

http://en.wikipedia.org/wiki/GUID_Partition_Table#cite_note-2

GUID Partition Table

OS support of GPT

- 마이크로소프트는 32비트 플랫폼에 EFI를 지원하지 않고, GPT를 이용한 부팅도 지원하지 않음
- GPT 부팅이 지원되지 않는 경우 데이터 디스크로만 사용 가능

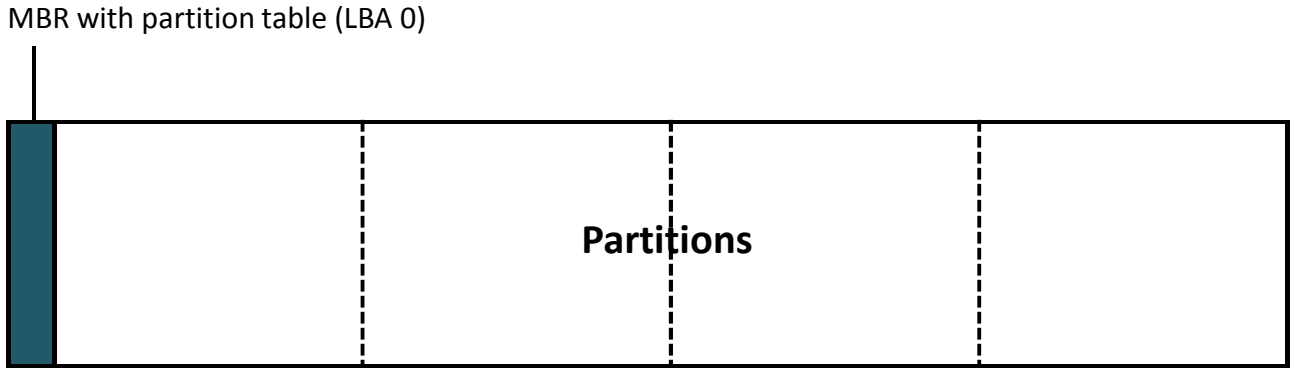
Windows 64-bit version

OS	Version/Edition	Platform	Boot from GPT on PC/BIOS	Boot from GPT on EFI
Windows XP	(2001-10-25)	IA-64	No	Yes
Windows XP	(2003-03-28)	IA-64	No	Yes
Windows Server 2003	(2003-04-24)	IA-64	No	Yes
Windows Server 2003	Service Pack 1 (2005-03-30)	x86-64	No	No
Windows XP	Professional x64 (2005-04-25)	x86-64	No	No
Windows Vista	(2005-09-22)	x86-64	No	Yes
Windows Server 2008	(2008-02-27)	x86-64, IA-64	No	Yes
Windows 7	(2009-10-22)	x86-64	No	Yes
Windows Server 2008 R2	(2009-10-22)	x86-64, IA-64	No	Yes

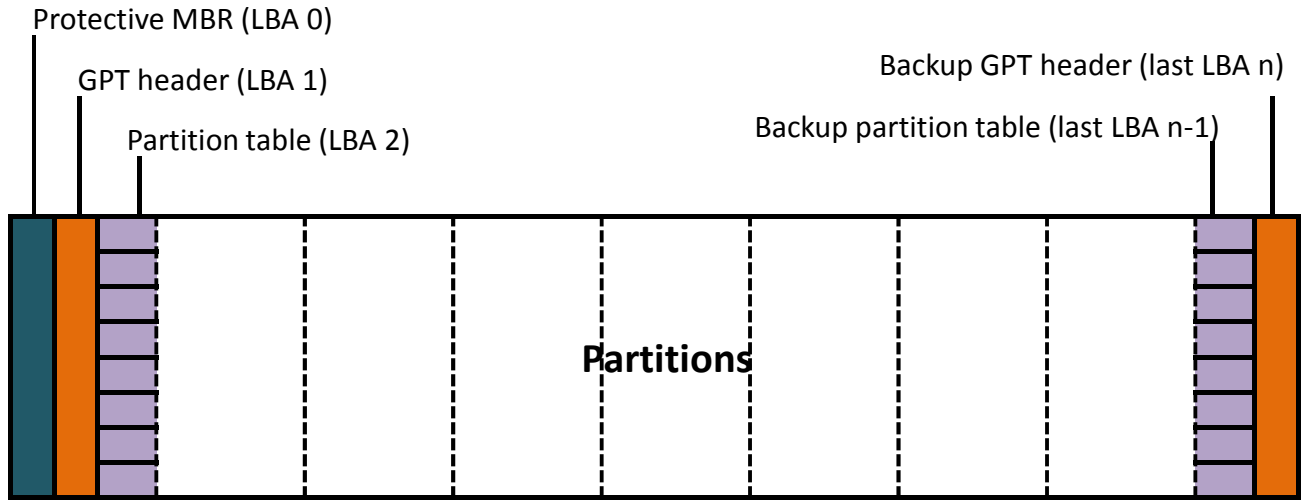
http://en.wikipedia.org/wiki/GUID_Partition_Table#cite_note-2

GUID Partition Table

GPT Layout



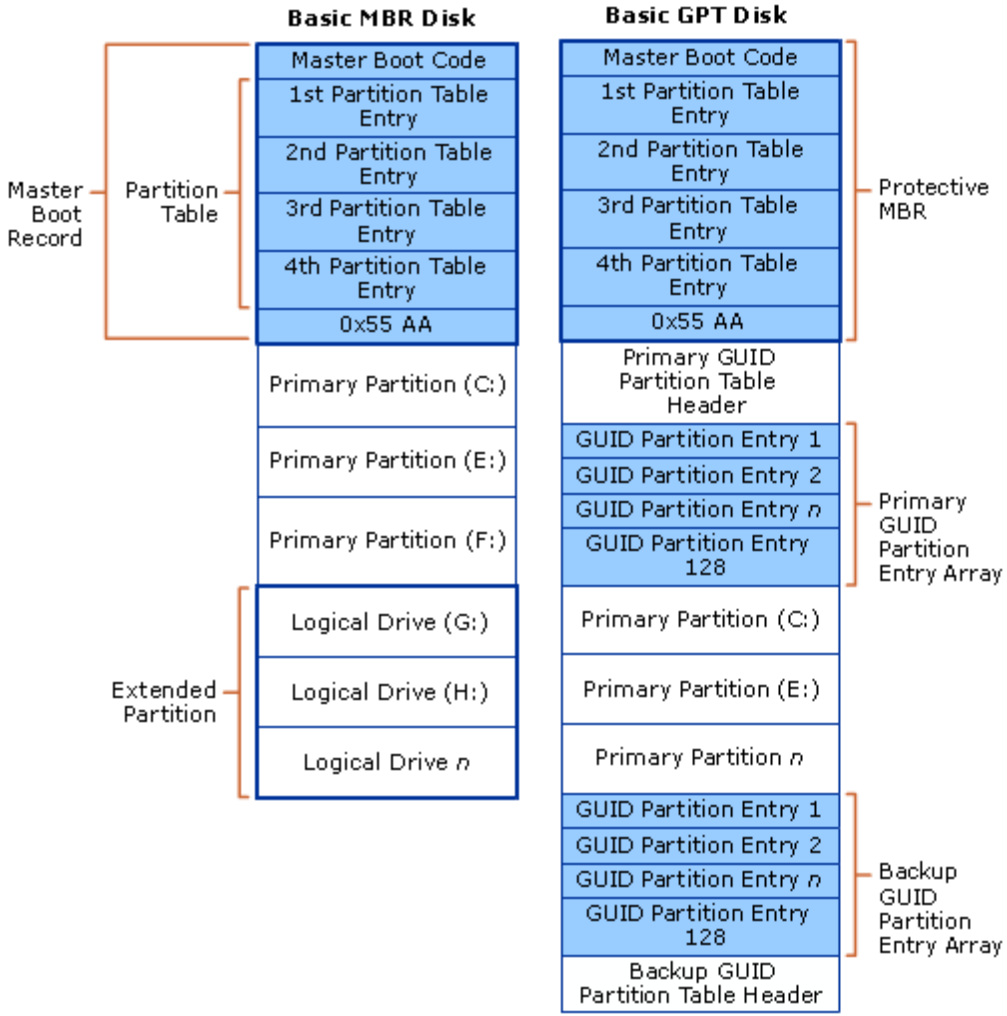
Traditional DOS/MBR disk layout



GPT disk layout

GUID Partition Table

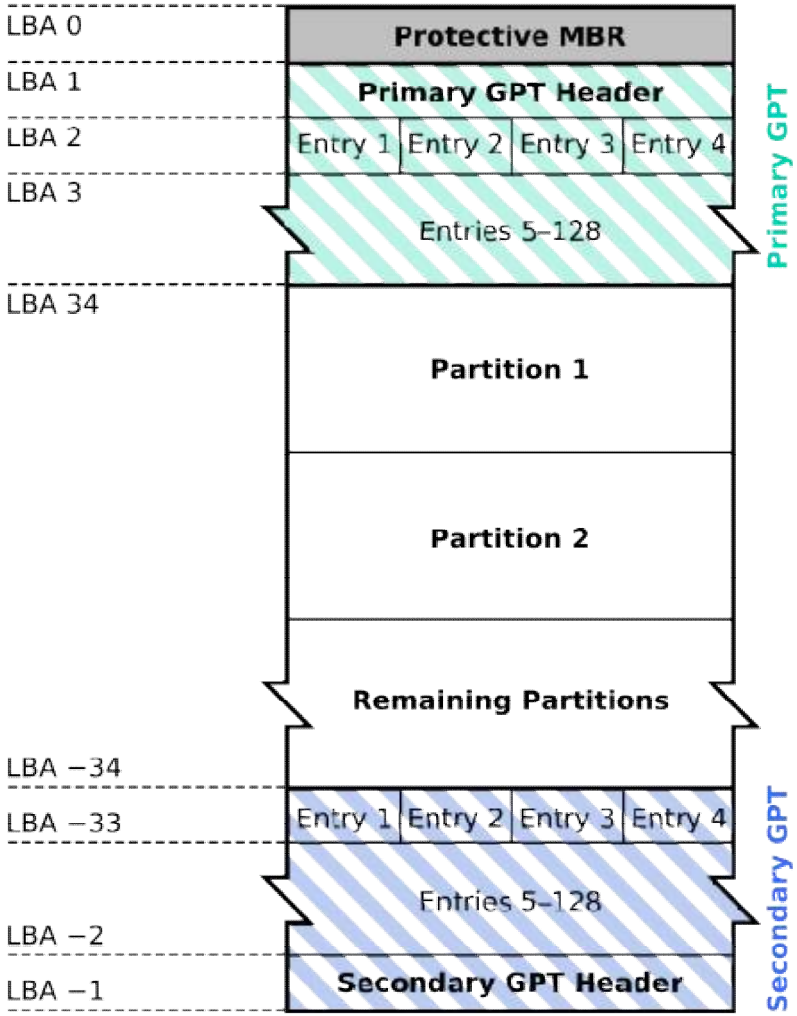
GPT Structure



- **Protective MBR**
  - 기존 MBR과 호환
  - 2 TB 단일 파티션으로 인식
  - GPT에서는 사용되지 않음
- **GPT header**
- **GPT entries**
  - 최대 128개(파티션) 지원
- **Backup**
  - GPT header
  - GPT entries

GUID Partition Table

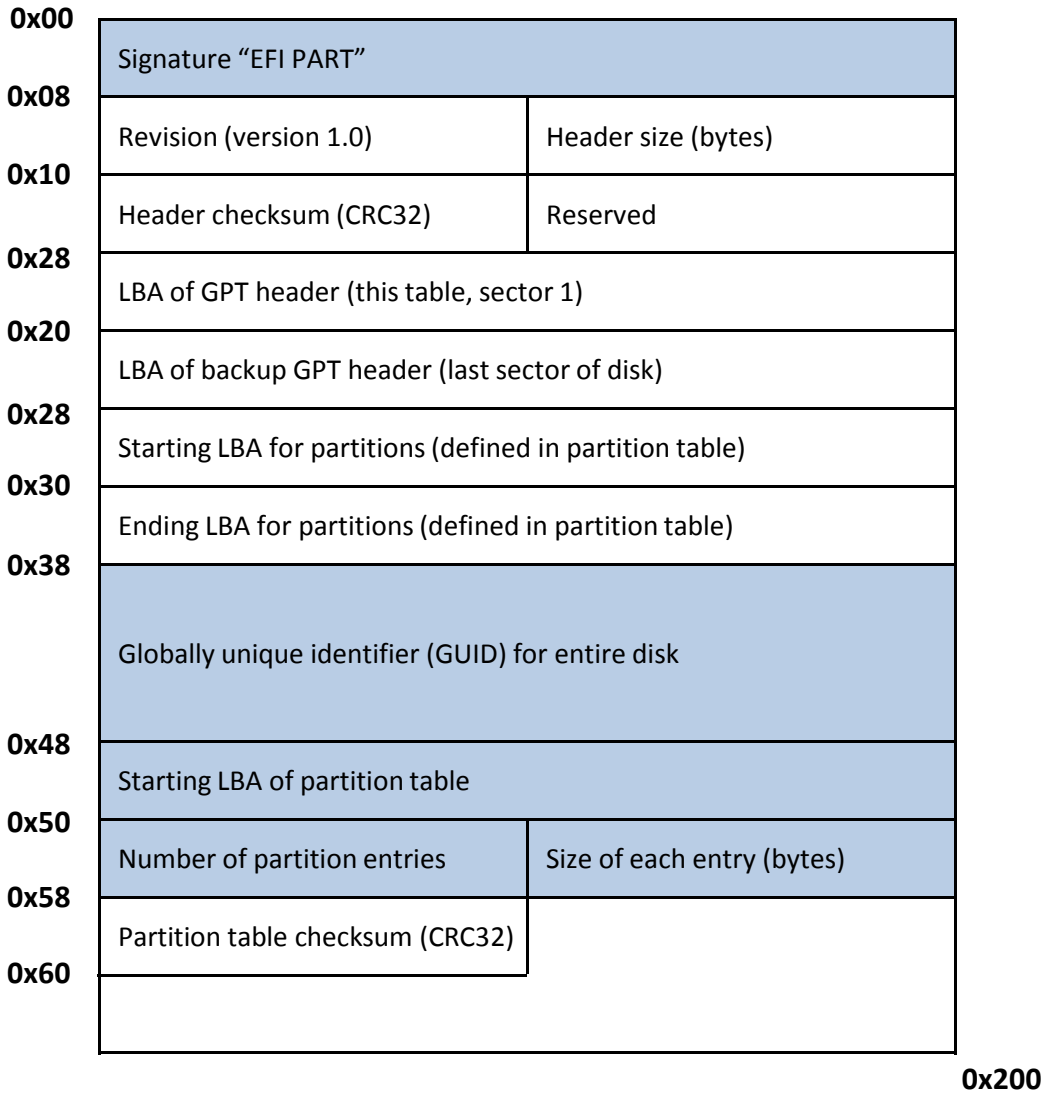
GUID Partition Table Scheme



GUID Partition Table

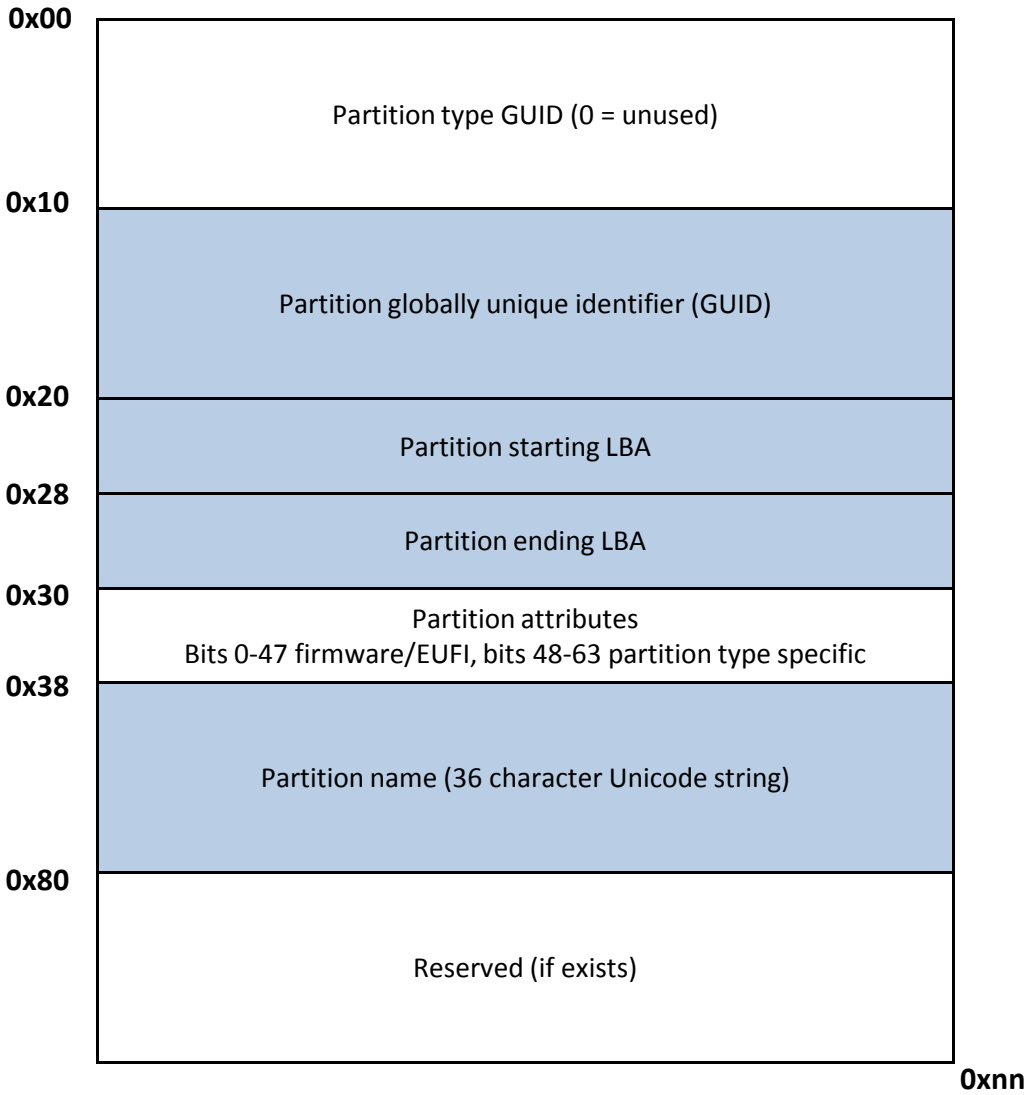
GPT Structure – GPT Header

- **Signature**
  - "EFI PART"
- **Header size**
  - 0x5c (92)
- **GUID for entire disk**
- **Number of partition entries**
  - 128
- **Size of each entry**
  - 128



GUID Partition Table

GPT Structure – GPT Entry



GUID Partition Table

Acquiring GPT disks and partitions

- **OSX partition tool**
  - *OSX disk utility*
  - *gpt*
- **Linux partition tool**
  - *Parted*
- **MS Windows partition tool**
  - *DISKPART*
- **Sleuth kit**
  - *mmls*
- **Hex editor**

GUID Partition Table

GPT header & entries analysis tools

- 전체 저장매체 이미징은 기존 방식과 차이가 없음
- 개별 GPT 파티션 추출
  - GPT 파티션 추출 기능을 제공하는 도구 이용 (dlfldd, mmcat)
    - #dcfldd if=/dev/sdb12 of=partition.dd
    - #mmcat –t gpt /dev/sdb18 > partition.dd
- HPA(Host Protected Area) & DCO (Device Configuration Overlay)
  - HPA, DCO 환경에서는 GPT 백업본이 저장매체 마지막이 아닐 수 있으므로 고려해서 분석

GUID Partition Table

GPT artifacts and reconstruction

- **MBR과 GPT 간의 변환**
  - MBR ➔ GPT (repartitioned or converted)
    - 이전 파티션 테이블 정보 삭제됨
    - 복구 방법 : 전체 디스크에서 이전 파티션 정보 검색
  - GPT ➔ MBR
    - 사용 도구에 따라 GPT 헤더 및 엔트리 정보가 남아 있을 가능성

- **GUID 값**
  - EFI 표준에서는 GUIDv1 사용
    - 타임스탬프, MAC 정보 사용
  - 개인정보의 이유로 윈도우에서는 GUIDv4 사용
    - 랜덤 데이터 사용
    - xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxx (y = 8, 9, A, B)

GUID Partition Table

Digital forensics point of view

- GPT 인식이 가능한 도구 사용
  - EnCase는 6.0 부터 지원
- MBR ➔ GPT, GPT➔ MBR 변환 시 이전 정보 복구
- GUIDv1 사용시 타임스탬프, MAC 정보 활용
- GPT 헤더와 엔트리 영역 (정상, 백업)에 데이터 은닉 가능성

Quiz!

Security is a people problem…

Quiz!

File system

- 파일시스템을 크게 메타 영역과 데이터 영역으로 구분하는 이유는?
- 섹터 주소 지정 방식 중 CHS 장점과 단점은?
- LBA 주조 지정 방식을 사용하는 이유는?
- 파일시스템에서 데이터 입/출력 시 클러스터나 블록 단위를 사용하는 이유는?
- 클러스터 크기는 4K일 때, 2K 크기의 파일을 저장한 경우 램슬랙과 드라이브 슬랙의 크기는?
- 파일 슬랙(램슬랙, 드라이브슬랙)의 특징은?
- 파티션과 볼륨의 차이는?

Quiz!

Master Boot Record & GUID Partition Table

- MBR 부트 코드의 역할?
- Device GUID의 역할은?
- MBR 파티션 테이블에 생성 가능한 최대 논리 파티션/확장파티션의 개수는?
- MBR 섹터 시그니처 값은 ?
- NTFS의 파티션 타입은?
- 파티션 사용의 이점은?
- GPT가 등장한 배경은?
- GPT에서 생성 가능한 주 파티션 개수는?

Question & Answer