저장방식

1 저장장치구조

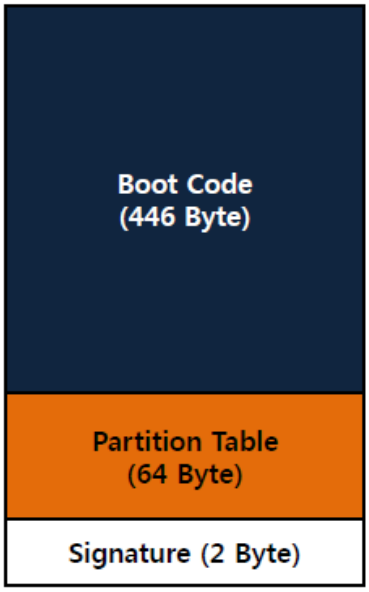
1.1

- 순서 : 부팅 -> 롬에있는 것을 불러옴 -> 주변 디바이스 장치 잘 연결되어있는지 확인 -> 보조기억장치의 첫 512바이트를 읽음
- 이 512바이트가 MBR(master boot record)이다. 즉 OS가 어느 드라이브에 깔려 있는지 확인하는 절차이다. 그 볼륨 처음에 가면 VBR이 있는데 이것은 부트로더와 커널이 올라갈 때 DLL을 올리는 용도이다.
- 섹터 크기는 512인데 이것은 정해진 것이 아니다. 그냥 단지 처음 512바이트를 걍 MBR이라고 한다. 요즘은 2TB이상은 인식 못하기 때문에 GPP라는 방식도 많이 쓴다.

1.2 MBR

- 모든저장장치의가장처음에존재하는구조
- 최근에는 MBR의 단점을 보완한 GPT (GUID Partition Table)가 사용됨.
- 구조

- 저장장치 첫번째 섹터(LBA 0)에 위치하는 512 바이트크기의 영역
- 부트코드와 파티션테이블로 구성

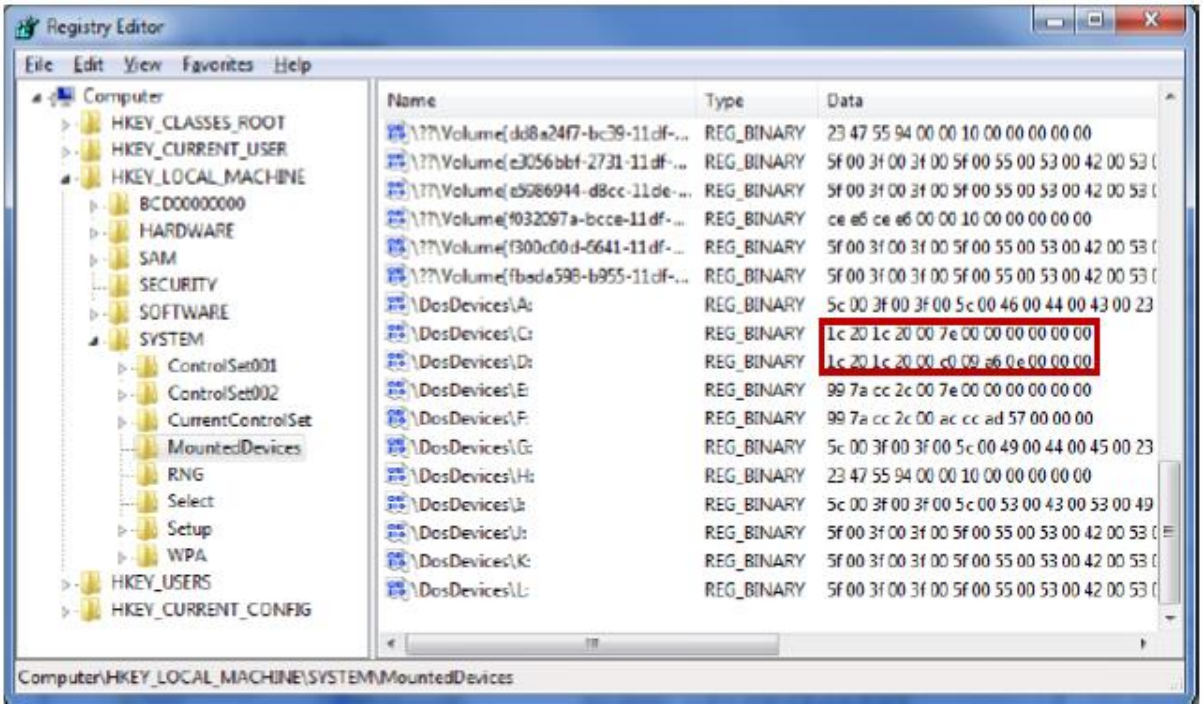


MBR 데이터 구조

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

- 처음 446은 부트코드, 그 다음 64는 파티션 테이블. 각 파티션 당 16바이트를 써서 파티션을 표현한다. 따라서 총 4개의 파티션을 만들 수 있다. 그다음 시그니처는 고정된 값이다.
- 부트코드
  - 파티션 테이블을 해석해서 부팅가능한 볼륨이 있나를 확인해서 있으면 부팅시켜준다.
  - 부팅 시 POST 과정 후 저장매체 첫 섹터호출
  - 첫 섹터인 MBR은 자신의 부트코드를 수행
- 역할
  - MBR 파티션테이블에서부팅가능한파티션검색
  - 부팅가능한파티션이있을경우, 해당파티션의VBR로점프
  - 부팅가능한파티션이없을경우, 오류메시지출력(처음 이 3개의 메세지가 뜬다면 MBR에 부팅 가능한 하드가 없기 때문이다. 이것도 안뜬다면 MBR문제도 아니고 그 이전 문제이다)
    - Invalid partition table.
    - Error loading operating system.
    - Missing operating system.
  - BIOS -> POST -> MBR -> VBR

Device GUID



- 장치가 마운트 되면 레지스트리에 장치 GUID(Globally Unique ID) 저장
- HKEY_LOCAL_MACHINE₩SYSTEM₩MountedDevices
- 일부러 지우지 않는 이상은 한번이라도 마운트가 되었다면 여기에 모두 기록이 된다. 데이터가 5f, 3f로 나오는 것은 usb인데 32기가 이상은 디스크로 인식해버리므로 추적이 안된다.
- 앞에 MBR의 시그니처가 4바이트 나머지는 볼륨의 시작위치를 뜻 하고 나머지 8바이트는 각 파티션의 시작 섹터 위치를 뜻한다. USB를 초기화 하지 않는 이상 이 4바이트 값은 바뀌지 않는다. 맨마지막에 있는 것들은 각 드라이브에 가장 최근에 꽂힌 드라이브이다. DATA 앞이 같을 경우 하나의 드라이브가 C드라이브 D드라이브로 나뉘어질 수 있다.
- 가장 최근에 꽂힌 드라이브가 F드라이브면 위(긴 줄) 아래(짧은 줄) 2개가 남고 만약 다른 USB가 꽂혔는제 F드라이브면 아래에 있는 정보는 최근 유에스비로 내용이 바뀌고 위에는 새로 꽂힌 USb의 정보가 새로 기록된다.

1.3 MBR 슬랙

- 저장장치의 시작인 MBR과 볼륨의 시작인 VBR 사이에 존재 하는 낭비되는 공간
- 루트킷, 랜섬웨어 등의 악성코드가 악용 vs. 보안 솔루션등이 선용
- 윈도우XP/2K3

- 63섹터(FDISK 트랙할당방식)

- 윈도우Vista/7/8
  - 2,048섹터(1MiB 할당방식)

- 이 슬랙을 악성코드가 많이 이용한다.

- 운영체제가 떠야 백신이 올라가는데. 백신이 올라가기전에 악성코드를 올리면 보안프로그램을 우회할 수 있다. 부팅 때 올라가므로 이 악성코드를 부트킷이라고 한다.

1.4 VBR

- 볼륨의시작에위치하는구조로볼륨의클러스터크기만큼할당

- 파일시스템의메타정보(BPB) + 부트로더로딩코드

- 볼륨의부트로더를로딩하여운영체제를부팅시키는코드

- 이 vbr부터는 섹터 단위가 아니라 클러스터 단위이다.

- 이것을 넘어가면 볼륨데이터.. 메타데이터. 파일데이터 등 여러 데이터가 있다.

1.5 Volume Data

- 파일시스템에 의해 할당된 볼륨데이터

- [메타데이터+ 파일데이터]로 구성

2 파일시스템

2.1 클러스터와 블록

- 데이터관리와CPU 성능효율을위해클러스터또는블록을사용

- 4MB 데이터를쓰기위해4K라면1,024번, 512바이트라면8,192번

- 리눅스나 유닉스는 블록이라는 단위를 쓰고 윈도우는 클러스터 단위를 쓴다. 하나로 묶어서 입출력단위로 쓰는데 데이터를 할당하거나 삭제. 해제하는 행위를 한다. 왜이렇게 하냐면 물론 100퍼센트 이렇게 동작을 하지는 않겠지만 CPU는 인터럽트방식으로 동작을 한다. 디스크한테도 명령을 내릴 때 만약 다 쓰고 나면 인터럽트를 날려야한다. 이때 CPU는 그 때에 맞게 처리를 한다. 하지만 모두 쓸때 알려주면 굉장히 비효율적이므로 중간중간알려주는데 그것이 좋은데 그것이 클러스터 단위이다. 그래서 섹터를 묶은 것이다.

2.2 파티션과 볼륨

- 파티션과 볼륨이라는 말을 쓰는데 파티션은 물리적으로 연속된 부분을 말한다. 볼륨은 논리적으로 연속된 부분을 말한다. 두개의 디스크를 붙여 하나의 볼륨으로 쓰는데 디스크를 이때의 동적디스크를 하나의 볼륨으로 쓰는 것이다. 이것이 OS에 모두 들어가 있으므로 쓰는 것이 가능하다. 따라서 보통은 연속되어있으므로 파티션이고 2개의 드라이브를 하나로도 쓸 수있으므로 볼륨이 더 큰 의미이다.