NTFS - Part 2 (보고서)

날짜	@2024년 2월 11일
사람	ohnahee

Chapter 4. 볼륨 분석
- 4.1. 배경
  - 볼륨개념
  - 파티션의 전반적인 이론
  - 볼륨 조합의 전반적 이론
  - 섹터 주소 지정
- 4.2. 분석 기초
  - 분석 기술
  - 일관성 검사
  - 파티션 내용 추출
  - 지워진 파티션 복구

Chapter 5. PC 기반 파티션
- 5.1. 도스 파티션
  - 전체 개요
  - 데이터 구조체
- 5.2. 애플 파티션
  - 전체 개요
  - 데이터 구조체
  - 분석 고려 사항
- 5.3. 이동식 매체
  - 플로피 디스크
  - USB
  - ZIP
  - 디지털 카메라
  - CD-ROM

Chapter 6. 서버 기반 파티션
- 6.1. BSD 파티션
  - 전체 개요
  - 데이터 구조체
  - 분석 고려 사항
- 6.2. 썬 솔라리스 슬라이스
  - 전체 개요
  - **스팍 데이터 구조체**
  - i386 데이터 구조체
  - 분석 고려 사항
- 6.3. GPT 파티션
  - 전체 개념
  - 데이터 구조체
  - 분석 고려 사항

Chapter 7. 다중 디스크 볼륨
- 7.1. RAID (**R**edundant **A**rray of **I**nexpensive/**I**ndependent **D**isk)
  - RAID 레벨
  - 하드웨어 RAID
  - 소프트웨어 RAID
  - 전반적 분석 의견
  - 요약
- 7.2. 디스크 스패닝

개요
리눅스 MD

Chapter 4. 볼륨 분석

4.1. 배경

💡 볼륨과 파티션

볼륨개념

- OS나 응용 프로그램이 데이터를 저장해 사용하도록 주소가 지정된 섹터들의 집합
- 여러개의 저장 볼륨을 한 개의 볼륨으로 조합하는 것
- 한 개의 저장 볼륨을 여러개의 파티션으로 분할하는 것

파티션의 전반적인 이론

- 볼륨 안 연속된 섹터들의 집합
- 일반적인 파티션 시스템들은 한 개 이상의 테이블을 보유한다 → 파티션 테이블
  - 각 테이블 엔트리는 하나의 파티션을 설명한다
  - 각 엔트리는 시작 섹터 위치, 마지막 섹터 위치, 파티션 유형 정보 등을 가짐

볼륨 조합의 전반적 이론

- 큰 시스템들은 여러 디스크를 하나처럼 보이게 볼륨 조합 기술을 사용한다
  - 한 디스크가 손상되는 것을 대비해서 백업해두는 것
- 하드웨어 장치는 단지 큰 볼륨을 제공하는 것일뿐

섹터 주소 지정

- LBA는 첫 섹터를 0으로 시작하는 섹터의 물리적 주소

- 볼륨 안에서 시작으로부터 상대적인 주소를 나타낸다

4.2. 분석 기초

분석 기술

- 파일 시스템 분석 도구를 이용

1) 파티션 테이블 위치를 알아내고

2) 레이아웃 식별하여 파티션 테이블 내용을 알아낸다

3) 파티션 오프셋을 알아내고 파티션의 정보를 출력한다

*데이터 구조체 위치를 알아내고 처리할 필요가 있다

일관성 검사

- 여러 파티션의 위치를 확인하는 것

상태검사 시나리오 (두 파티션이 구성될 수 있는 방법 5가지)

- 두 파티션이 붙어있고 끝이 비어있는 경우
- 두 파티션이 떨어져서 할당되어있는 경우
  - 숨겨진 공간의 검사 필요
- 두 파티션이 붙어있는 경우

제외하고는 전부 유효하지 않음

파티션 내용 추출

- 리눅스 도구 dd 기반
  - Unix 및 Unix 계열 운영 체제에서 사용되는 명령줄 유틸리티
  - 데이터 복사 Data Duplicator 의 약어
  - 주로 데이터를 복사하고 변환하는데 사용

- 옵션
  - if : 읽을 디스크 이미지
  - of : 저장될 출력 파일

- bs : 한 번에 읽어들일 블록 크기 (기본 512byte)
- skip : 읽기 전에 건너 뛸 블록 개수 (크기 bs)
- count : 입력에서 출력으로 복사할 블록 개수 (크기 bs)

#dd if=disk1.dd of=part1.dd bs=512 skip=63 count=1028097
→ dd를 사용하여 disk1.dd 파일에서 데이터를 읽어와서 part1.dd 파일에 쓰는 추출작업

- 이러한 파일 레이아웃을 보여주는 도구가 파티션 시작, 끝 섹터를 알려주면 계산이 필요
  - 마지막 섹터 - 시작섹터 + 1

지워진 파티션 복구

- 파일시스템이 파티션에 위치해 있다는 가정하에 동작
- 파일 시스템은 일적한 매직이나 시그니처 값을 갖는 데이터 구조체로 시작
  - FAT : 첫번째 섹터의 바이트 510-511에 0x55, 0xAA 값을 가짐
- 파티션 복구 도구 gpart (리눅스)
  - 섹터를 검사하고 어떤 파일시스템 타입이 가장 가능성이 있는지 측정
  - 파일시스템 타입 번호를 구분
  - 자세한 내용을 보여주지 않기 때문에 -v 플래그 사용해야함

Chapter 5. PC 기반 파티션

5.1. 도스 파티션

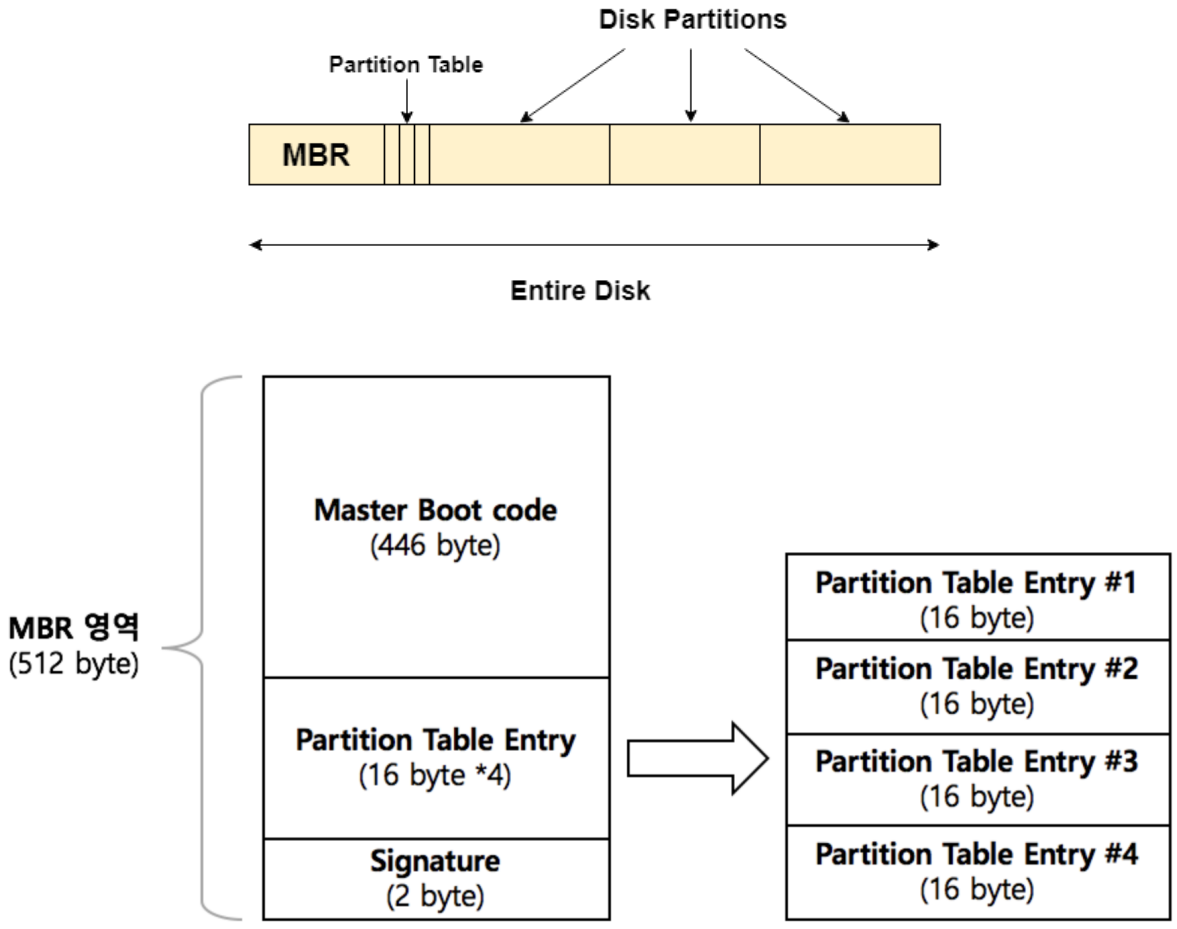
도스파티션 개념
부트 코드 위치

전체 개요

도스파티션

- MS, Window, 리눅스 등이 사용하는 분할 시스템

MBR 개념

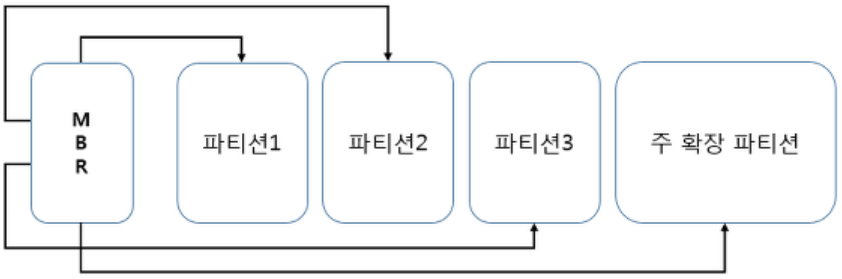


- 도스파티션을 사용하는 디스크는 512바이트 섹터 내 MBR을 가짐
- MBR내에는 부트코드, 시그니처, 파티션 테이블이 있음
  - 부트코드 : 파티션 테이블 처리 명령어, 운영체제 위치 확인 명령어
  - 파티션 테이블 : 도스파티션을 설명하는 4개의 엔트리
    - 파티션 시작의 CHS/LBA 주소
    - 파티션 끝의 CHS 주소
    - 파티션 섹터 수

- 파티션 타입
- 플래그

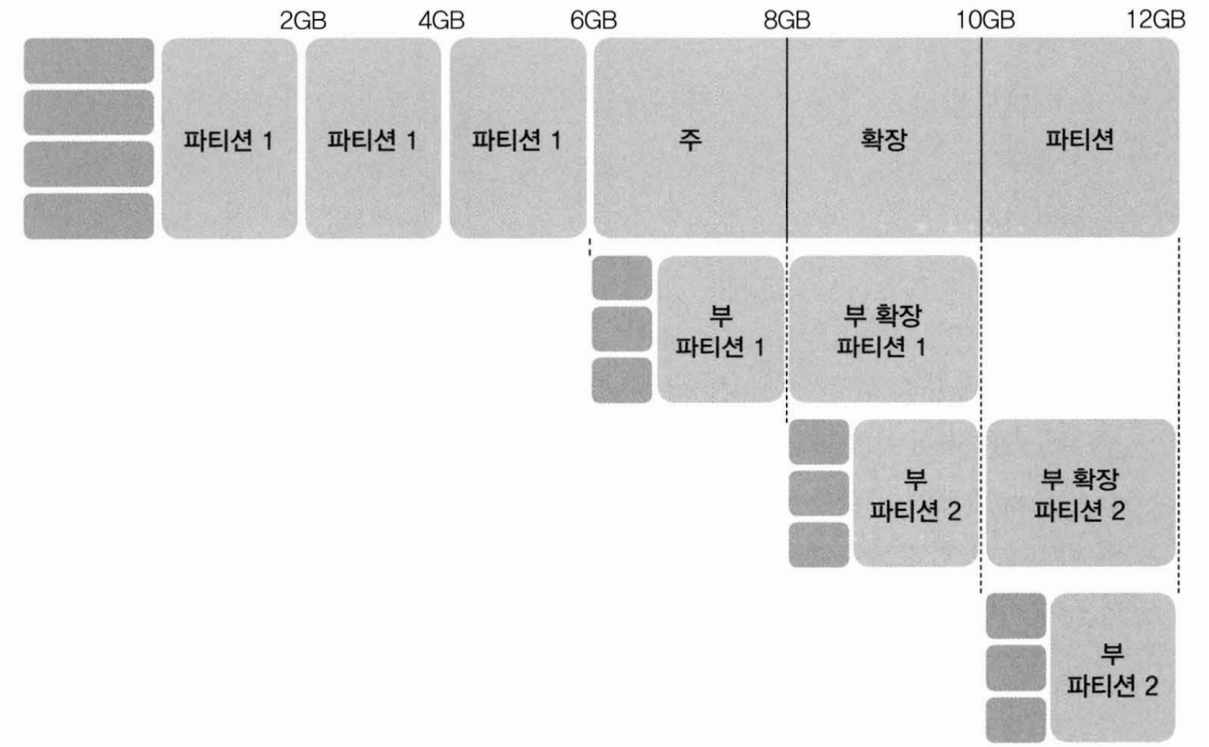
확장 파티션 개념

- MBR은 4개의 파티션 포함



기본 확장 파티션 구조

- 더 많은 파티션을 포함하고 싶다면 도스파티션을 자르면 됨 → 도스 파티션이 복잡해지는 이유



[그림 2.101] 예문 확장 파티션

- 사용자가 12GB 디스크를 2GB 파티션 6개로 나누길 원함

1. MBR 내 3개의 엔트리를 사용해서 2GB 파티션 3개 지정
2. 나머지 6GB 부 확장 파티션에 할당
3. 연결리스트 방법을 이용해 3개의 파티션을 할당
4. 주 확장 파티션의 첫번째 섹터에 있는 파티션 테이블을 이용해서 6-8GB 범위에 부 파일 파티션 생성
5. 8-10GB 부 확장 파티션

*파티션 테이블은 부 확장 파티션내에 존재하며 10-12GB 범위인 다른 부 파일 파티션을 위한 엔트리를 가짐

부트코드

- MBR 512byte 섹터에서 1-446 바이트에 존재
- 부트섹터 바이러스는 MBR의 1-446 바이트에 코드 삽입 후 부팅될 때 마다 매번 실행 됨
- 여러 OS 사용하고 싶은 경우
    1. 이 코드에 사용자가 OS 선택 코드를 삽입
    2. MBR 부트 실행시 OS가 실행되고 윈도우 부트 파티션에서 선택모드 삽입 가능

데이터 구조체

MBR 데이터 구조체

- 0~455 까지는 부트코드이며 각 파티션의 엔트리는 16바이트씩 가진다

Address Range	Size	Field Name	Description
0x0000~0x01BD	446 Byte	Boot Code	부트 코드
0x01BE~0x01CD	16 Byte	Partition Table Entry #1	파티션 테이블 엔트리 1번
0x01CE~0x01DD	16 Byte	Partition Table Entry #2	파티션 테이블 엔트리 2번
0x01DE~0x01ED	16 Byte	Partition Table Entry #3	파티션 테이블 엔트리 3번
0x01EE~0x01FD	16 Byte	Partition Table Entry #4	파티션 테이블 엔트리 4번
0x01FE~0x01FF	2 Byte	Signature	시그니처 (0x55AA)
MBR Structure 0x0000~0x01FF (Size : 512 Byte)			

도스 파티션 엔트리 구조

범위(Byte)	설명	범위(Byte)	설명

0 ~ 0	부팅 플래그	**5 ~ 7**	파티션 끝 CHS 주소
1 ~ 3	파티션 시작 CHS 주소	**8 ~ 11**	파티션 시작 LBA 주소
4 ~ 4	파티션 타입	**12 ~ 15**	섹터 크기

도스 파티션 타입

타입	설명	타입	설명
0x00	Empty	**0x83**	Linux
0x01	FAT12, CHS	**0x84**	Hibernation Mode
0x04	FAT16, 16 ~ 32MB, CHS	**0x85**	Linux Extended
0x05	Microsoft Extended, CHS	**0x86**	NTFS Volume Set
0x06	FAT 32MB ~ 2GB, CHS	**0x87**	NTFS Volume Set
0x07	NTFS	**0xA0**	Hibernation Mode
0x0B	FAT32, CHS	**0xA1**	Hibernation Mode
0x0C	FAT32, LBA	**0xA5**	FreeBSD
0x0E	FAT16, 32MB ~ 2GB, LBA	**0xA6**	OpenBSD
0x0F	Microsoft Extended, LBA	**0xA8**	Mac OS X
0x11	Hidden FAT12, CHS	**0xA9**	NetBSD
0x14	Hidden FAT16, 16 ~ 32MB, CHS	**0xAB**	Mac OS X Boot
0x16	Hidden FAT16, 32MB ~ 2GB, CHS	**0xB7**	BSDI
0x1B	Hidden FAT32, CHS	**0xB8**	BSDI Swap
0x1C	Hidden FAT16, 16 ~ 32MB, LBA	**0xEE**	EFI GPT Disk
0x42	Microsoft MBR, Dynamic Disk	**0xEF**	EFI System Partition
0x82	Solaris x86	**0xFB**	VMware File System
0x82	Linux Swap	**0xFC**	VMware Swap

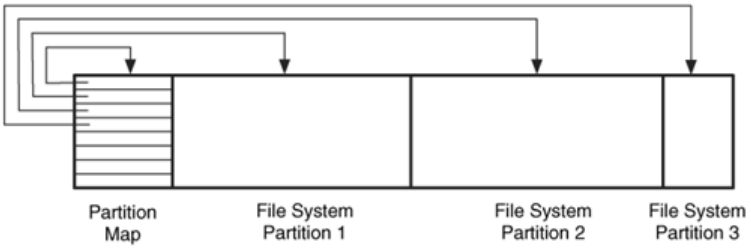
확장 파티션 데이터 구조체

- MBR과 구조는 같지만 시작 섹터 주소가 디스크 기준이 아니다
- 부 확장 파티션이 시작하는 섹터와 부 파일 시스템이 시작하는 섹터의 상대 주소가 다르다

5.2. 애플 파티션

- 애플 노트북, 아이패드 같은 장비들에게서 볼 수 있는 파티
- 매킨토시 시스템이 파일들을 전송하기 위해 사용하는 디스크 이미지는 파티션 맵을 이용
    - 이 디스크 이미지는 윈도우 ZIP과 유닉스 TAR과 유사
- 파티션에 파일시스템이 존재하고 그 안에 파일들이 저장

전체 개요



- 디스크 시작에 위치한 파티션 맵 구조체에서 애플파티션 설명
- 펌웨어가 구조체를 처리하는 코드를 포함하기때문에 맵은 부트코드를 포함하지 않음
    - 펌웨어 : 하드웨어이면서 소프트웨어의 기능도 하는 하드웨어
- 파티션 맵의 첫엔트리는 자기자신을 위한 엔트리
    - 전체 저장공간이 가질 수 있는 최대 크기를 나타냄

데이터 구조체

파티션 맵 엔트리

Byte Range	Description	Essential
01	Signature value (0x504D)	No
23	Reserved	No
47	Total Number of partitions	Yes
811	Starting sector of partition	Yes
1215	Size of partition in sectors	Yes
1647	Name of partition in ASCII	No
4879	Type of partition in ASCII	No
8083	Starting sector of data area in partition	No
8487	Size of data area in sectors	No
8891	Status of partition (see table 5-8)	No
9295	Starting sector of boot code	No
9699	Size of boot code in sectors	No
100103	Address of boot loader code	No
104107	Reserved	No
108111	Boot code entry point	No
112115	Reserved	No
116119	Boot code checksum	No
120135	Processor type	No
136511	Reserved	No

애플 파티션 상태 값

- 다른 파티션과 같이 정수를 사용하지 않고 아스키를 사용
- 각 파티션의 상태 값은 오래된 A./UX(애플 구형 운영체제) 시스템들과 현재 매킨토시 시스템 둘 다 적용

Type	Description
0x00000001	Entry is valid (A/UX only)
0x00000002	Entry is allocated (A/UX only)
0x00000004	Entry in use (A/UX only)
0x00000008	Entry contains boot information (A/UX only)
0x00000010	Partition is readable (A/UX only)
0x00000020	Partition is writable (Macintosh & A/UX)
0x00000040	Boot code is position independent (A/UX only)
0x00000100	Partition contains chain-compatible driver (Macintosh only)
0x00000200	Partition contains a real driver (Macintosh only)
0x00000400	Partition contains a chain driver (Macintosh only)
0x40000000	Automatically mount at startup (Macintosh only)
0x80000000	The startup partition (Macintosh only)

분석 고려 사항

- 데이터 구조체에서 사용디지 않은 필드들이 있어 적은 양의 데이터를 숨길 수 있다
- 파티션 구조체와 맵 끝 사이 섹터들에 데이터를 숨길 수 있다

5.3. 이동식 매체

💡 이동식 매체 또한 파티션을 가지고 하드디스크와 사용하는 구조가 같다

플로피 디스크

- FAT12로 포맷하는 플로피 디스크의 경우는 파티션이 존재하지 않는다
  - 플로피 디스크 전체를 하나의 파티션으로 취급하기 때문

USB

- 파티션이 있는 것도 존재하고 없는 것도 존재

ZIP

- IOMEGA ZIP 디스크 같은 대용량 이동식 매체는 파티션 테이블 사용
  - 사용하는 파티션 테이블은 맥, PC시스템에 따라 다름
- PC에서 사용하는 디스크는 도스기반파티션
  - 네번째 슬롯에는 하나의 파티션만 가진다

디지털 카메라

- 카메라 내 플래시 카드로 전형적인 파티션 사용
- FAT 파일시스템 사용

CD-ROM

- 종류가 방대하여 설명 복잡
- 대부분의 CD들은 **ISO 9660 포맷**을 사용
  - 운영체제들은 그러한 내용 읽기 가능

Chapter 6. 서버 기반 파티션

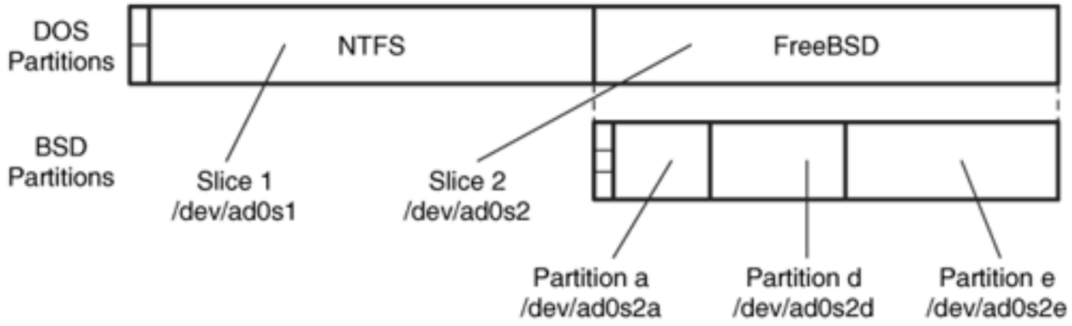
6.1. BSD 파티션

*도스파티션에 대한 이해 먼저 하고 읽을 것

- 운영체제 시작할 때 사용자가 접근할 파티션을 선택할 수 있음을 유의

전체 개요

- 도스파티션보다 간단하지만 애플파티션보다는 제한적
- 같은 디스크에서 BSD, 윈도우를 둘 다 운영하기 위해서는 **BSD 파티션이 도스 파티션 내**에 있어야 함
- BSD 파티션은 도스 파티션의 '주 도스 파티션' 이 될 수 있다



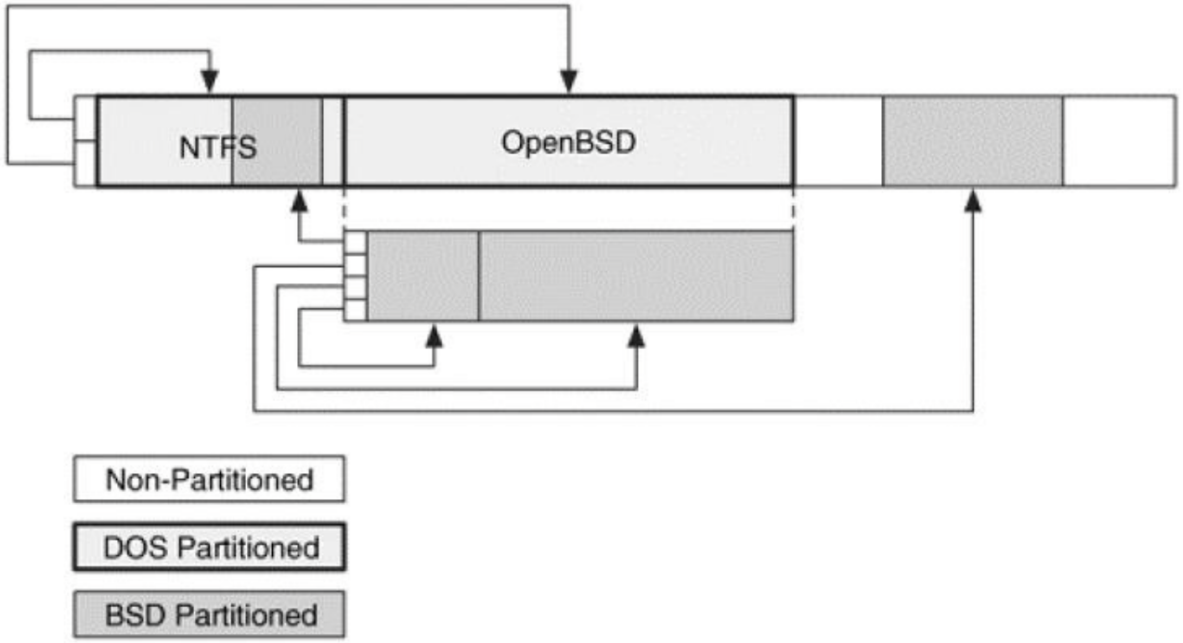
두 도스 파티션과 3개의 BSD파티션을 가지는 디스크

- 시작 섹터 주소는 디스크 시작에서 상대적
  - UFS, FAT, 비할당, 스왑공간 같이 BSD 파티션이 있는 파일시스템 타입
- 구조체를 먼저 읽어서 전체 파티션을 파악 후 운영체제가 두 개 이상인 시스템에서 사용자가 어떤 파티션에 주로 접근하는지에 대한 이해가 필요

Free BSD 개요

- 하나의 디스크에서 도스, BSD 둘 다 선택해서 사용 가능

OpenBSD/NetBSD 개요



- BSD디스크 레이블 구조체 엔트리만 접근이 가능
- 디스크레이블은 **모든 파티션을 설명**할 수 있어야 한다

데이터 구조체

BSD 디스크 레이블 데이터 구조체

Byte Range	Description	Essential
03	Signature value (0x82564557)	No
45	Drive type	No
67	Drive subtype	No
823	Drive type name	No
2439	Pack identifier name	No
4043	Size of a sector in bytes	Yes
4447	Number of sectors per track	No
4851	Number of tracks per cylinder	No
5255	Number of cylinders per unit	No
5659	Number of sectors per cylinder	No
6063	Number of sectors per unit	No
6465	Number of spare sectors per track	No
6667	Number of spare sectors per cylinder	No
6871	Number of alternate cylinders per unit	No

7273	Rotational speed of disk	No
7475	Hardware sector interleave	No
7677	Track skew	No
7879	Cylinder skew	No
8083	Head switch time in microseconds	No
8487	Track-to-track seek time in microseconds	No
8891	Flags	No
92111	Drive specific information	No
112131	Reserved	No
132135	Signature value (0x82564557)	No
136137	Checksum	No
138139	Number of partitions	Yes
140143	Size of boot area	No
144147	Maximum size of file system boot super block	No
148163	BSD Partition #1 (see Table 6.2)	Yes
164179	BSD Partition #2 (see Table 6.2)	Yes
180195	BSD Partition #3 (see Table 6.2)	Yes
196211	BSD Partition #4 (see Table 6.2)	Yes
212227	BSD Partition #5 (see Table 6.2)	Yes
228243	BSD Partition #6 (see Table 6.2)	Yes
244259	BSD Partition #7 (see Table 6.2)	Yes
260275	BSD Partition #8 (see Table 6.2)	Yes
276291	BSD Partition #9 (see Table 6.2)	Yes
292307	BSD Partition #10 (see Table 6.2)	Yes
308323	BSD Partition #11 (see Table 6.2)	Yes
324339	BSD Partition #12 (see Table 6.2)	Yes
340355	BSD Partition #13 (see Table 6.2)	Yes
356371	BSD Partition #14 (see Table 6.2)	Yes
372387	BSD Partition #15 (see Table 6.2)	Yes
388403	BSD Partition #16 (see Table 6.2)	Yes
404511	Unused	No

16바이트의 BSD 디스크 레이블 엔트리 데이터 구조

Byte Range	Description	Essential
03	Size of BSD partition in sectors	Yes
47	Starting sector of BSD partition	Yes
811	File system fragment size	No
1212	File system type (see Table 6.3)	No
1313	File system fragments per block	No
1415	File system cylinders per group	No

BSD 파티션 타입 값

Type	Description
0	Unused Slot
1	Swap space
2	Version 6
3	Version 7
4	System V
5	4.1BSD
6	Eighth edition
7	4.2BSD fast file system (FFS)
8	MSDOS file system (FAT)
9	4.4BSD log-structured file system (4.4LFS)
10	In use, but unknown or unsupported
11	OS/2 HPFS
12	CD-ROM (ISO9660)
13	Bootstrap
14	Vinum drive

예제 이미지

BSD Partitions | Server-based Partitions
BSD Partitions / Server-based Partitions from File System Forensic Analysis
https://flylib.com/books/en/2.48.1/bsd_partitions.html

참고

분석 고려 사항

- 디스크 레이블 구조체에는 가가 BSD 파티션의 타입 필드가 있지만 필수는 아님
  - MS 윈도우에서는 드라이브 문자를 얻을 수 있는지 판단 위해 필요
- 디스크 레이블 크기는 대부분 404 바이트
- 8개 엔트리를 가지는 레이블은 276 바이트
- 총 512 바이트로 구성된 섹터에서 남는 공간은 필수적으로 생기는데 이곳에 데이터 은닉가능
  - 시그니처를 통해 엔트리 크기를 알아내고 공간 분석 피요
- Free BSD는 도스와 BSD 파티션을 분석해야 한다
  - Open BSD는 BSD 파티션만을 볼 수 있기 때문에 도스 파티션 테이블과 BSD 디스크 레이블 비교가 필요하다
  - 겹치는 공간, 빈 공간 확인 필요

6.2. 썬 솔라리스 슬라이스

💡 일반적인 솔라리스 파티션의 구조를 설명
Spac과 i386의 구조체를 분석

- 대용량 서버, 데스크톱 시스템에서 사용
- 디스크 크기와 솔라리스 버전에 따라 2가지의 다른 타입의 분할 시스템 사용
- UFS 파일 시스템 사용
- 솔라리스 9는 64비트의 주소필드를 지원하는데 1KB 이상의 파일시스템을 저장 할 수 있다
- EFI 파티션 테이블 사용
- BSD 디스크 레이블과 비슷한 데이터 구조
- 각 파티션을 위해 슬라이스 라는 용어 사용
- 스팍(Spac)솔라리스와 i386솔라리스 파티션의 데이터 구조가 다름

전체 개요

- 설치 시 디스크(하드웨어 플랫폼 기반)에 디스크 레이블 생성
- 디스크 레이블에는 최대 파티션 개수가 있음
- 블록장치들은 /dev/dsk/에 존
- 미가용장치는 /dev/rdsk/에 존재
- 장치가 존재하고 그 안에 파티션(슬라이스)는 다음과 같은 이름 형식을 가짐
    - **Spac: cWtXdYsZ**
    - **i386: cWdYsZ**

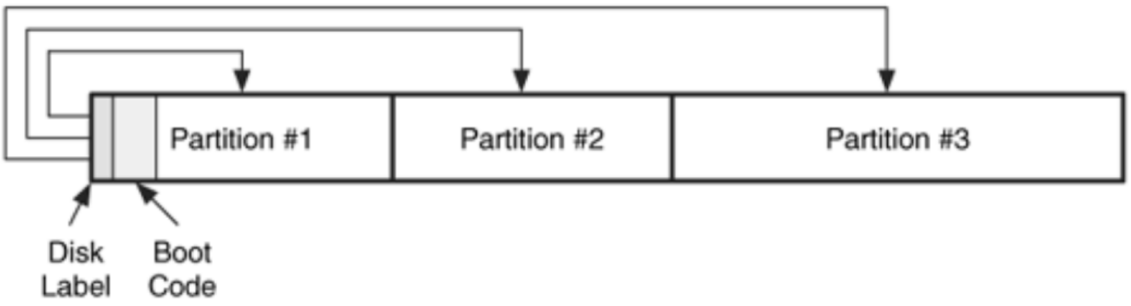
- W : 컨트롤번호
- X : SCSI ID(Physical bus target Number)
- Y : 버스드라이브 번호
- Z : 드라이브 슬라이스 번호

테이블 엔트리에 생성되는 파티션 이름 규칙

Table 6.8. The typical partition that is created in each table entry.

Table Entry	Description
0	/root/partitionThe operating system and kernel
1	Swap space
2	The entire disk, including the disk label and all partitions
3	/export/ partition
4	/export/swap/ partition
5	/opt/ partition
6	/usr/ partition
7	/home/ partition

스팍 데이터 구조체



썬 스팍 디스크 레이아웃

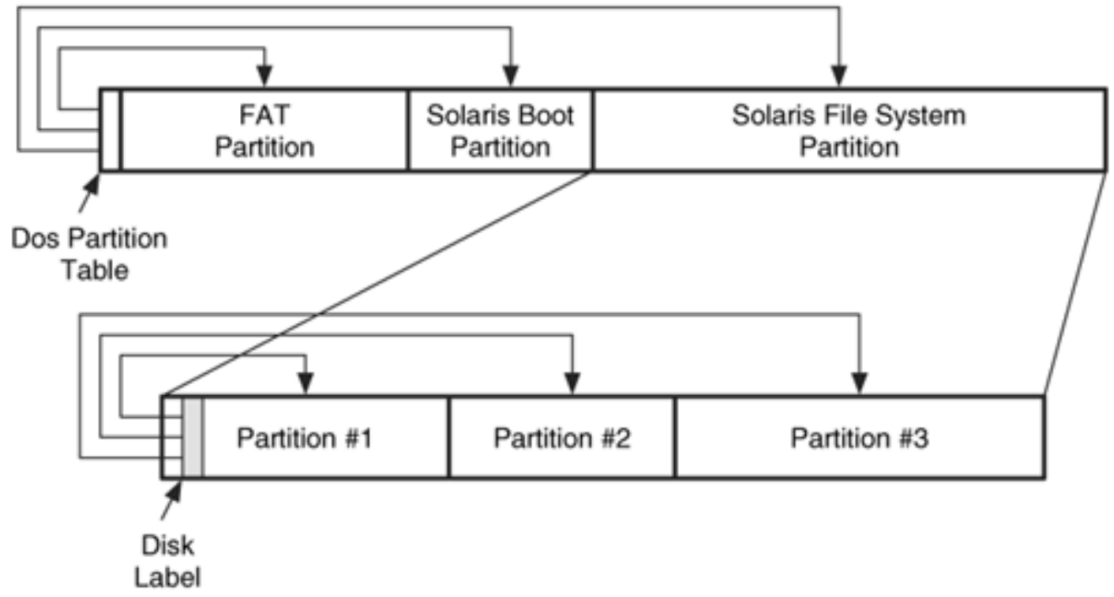
- 디스크 레이블은 디스크 첫 번째 섹터 0에 생성

썬 스팍 디스크 레이블 구조체

Byte Range	Description	Essential
0127	ASCII Label	No
128261	Sparc VTOC (see Table 6.10)	Yes
262263	Sectors to skip, writing	No
264265	Sectors to skip, reading	No
266419	Reserved	No
420421	Disk speed	No
422423	Number of physical cylinders	No
424425	Alternates per cylinder	No
426429	Reserved	No
430431	Interleave	No
432433	Number of data cylinders	No
434435	Number of alternate cylinders	No
436437	Number of heads	Yes
438439	Number of sectors per track	Yes
440443	Reserved	No
444451	Partition #1 disk map (see Table 6.13)	Yes
452459	Partition #2 disk map (see Table 6.13)	Yes
460467	Partition #3 disk map (see Table 6.13)	Yes
468475	Partition #4 disk map (see Table 6.13)	Yes
476483	Partition #5 disk map (see Table 6.13)	Yes
484491	Partition #6 disk map (see Table 6.13)	Yes
492499	Partition #7 disk map (see Table 6.13)	Yes
500507	Partition #8 disk map (see Table 6.13)	Yes
508509	Signature Value (0xDABE)	No
510511	Checksum	No

i386 데이터 구조체

- 솔라리스를 i386 시스템에 설치시 한 개 이상의 도스 기반 파티션 필요
- 일반적으로 한 개의 부트파티션과 한개의 파일시스템을 사용하는 파티션으로 설치



i386 썬 디스크 예

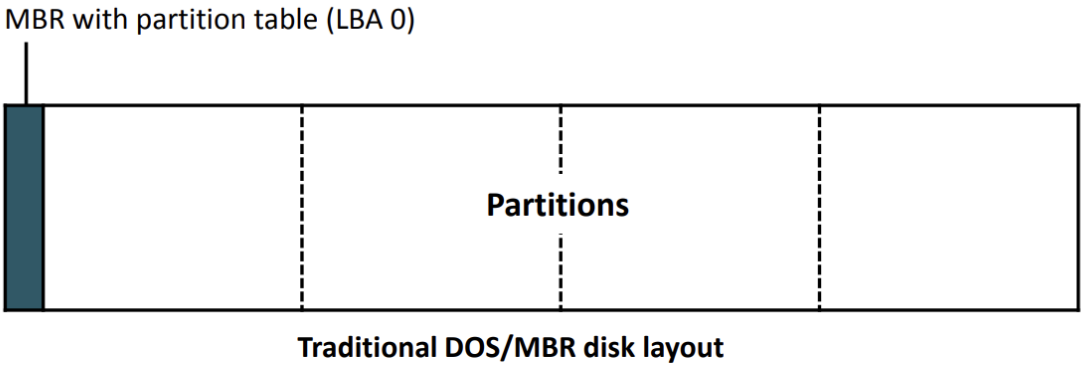
분석 고려 사항

- 다른 파티션 조사 필요
  - 비사용 공간에 데이터가 숨어있을 수 있음
- 타입 필드가 항상 필요한 것은 아니다
- 디스크 레이블 위치 결정을 못하면 시그니처 확인

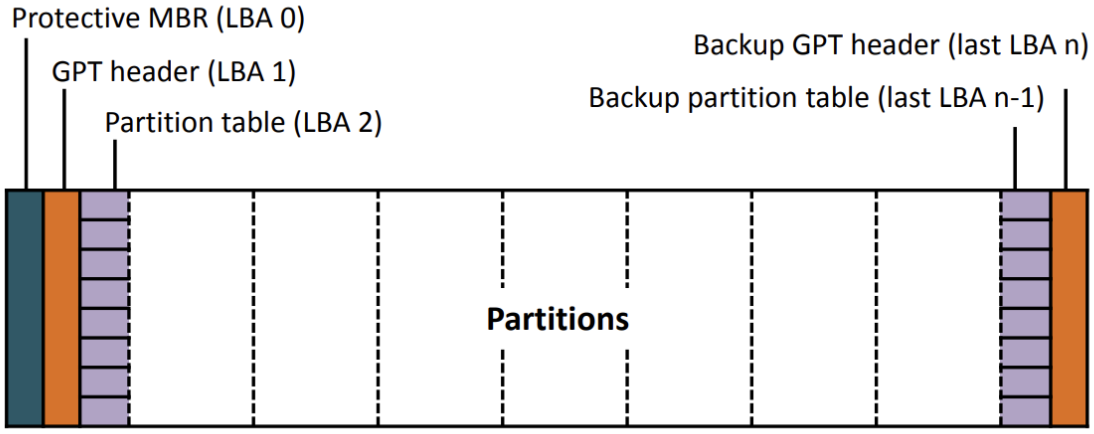
6.3. GPT 파티션

- 고성능 서버에서 사용
- 데이터 구조체 복사본 관리
- IA64 시스템들은 IA32가 가지고 있는 BIOS가 없는 대신 EFI(Extensible Firmware Interface)가 존재

전체 개념



Traditional DOS/MBR disk layout



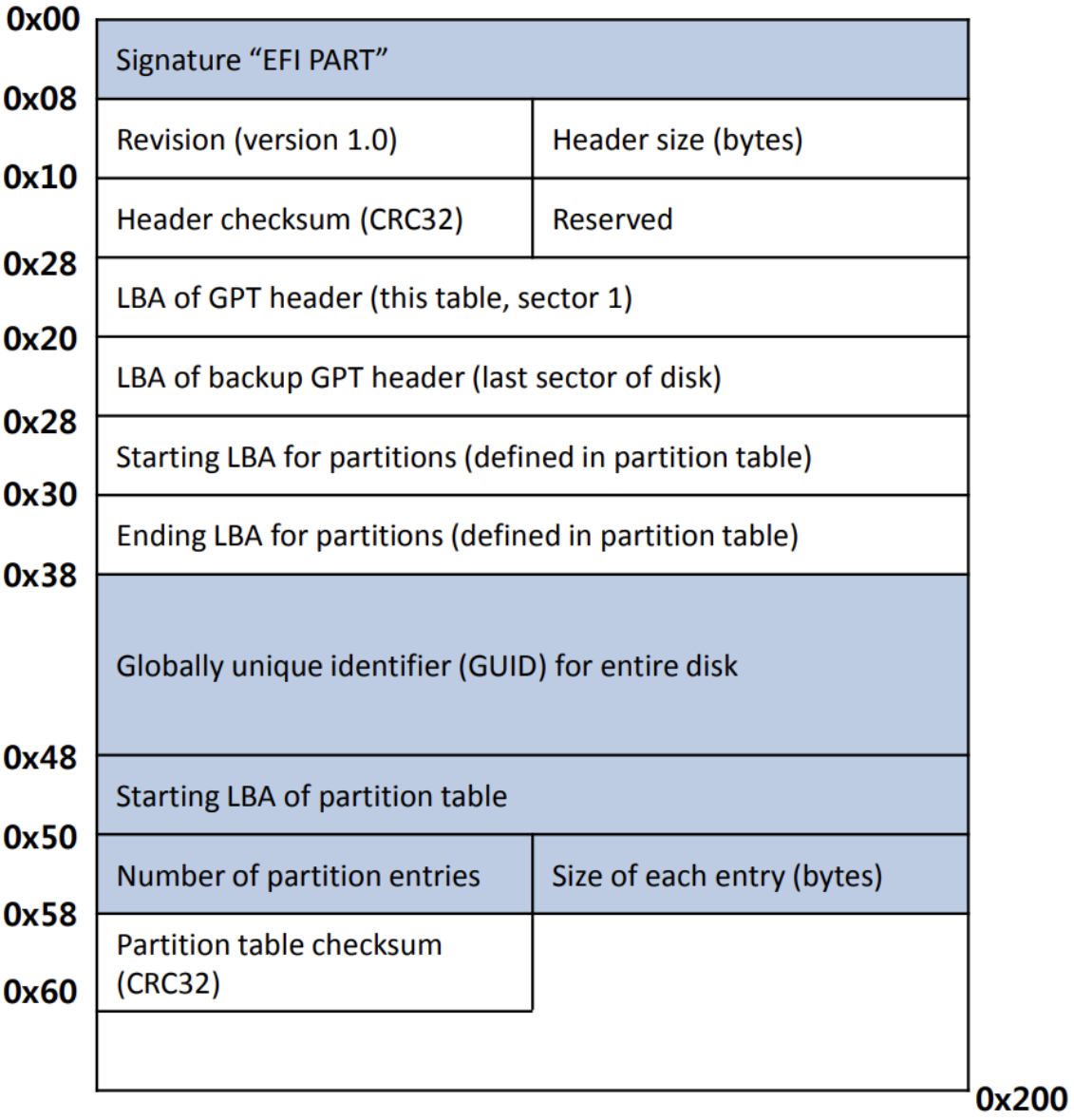
GPT disk layout

- 5개의 영역
- 첫번째 보호용 MBR
  - 섹터 0에 존재
  - 한 개의 엔트리가 있는 도스 파티션이 존재
  - 전체 디스크에 있는 0xEE 타입 파티션을 위함
- 두번째 GPT 헤더
  - 섹터 1에 존재
  - GPT 디스크 생성시 고정된 파티션 테이블 위치와 크기를 저으이
- 파티션 테이블
  - 시작과 끝 주소
  - 유형 값
  - 이름
  - 속성 플래그
  - GUID 값

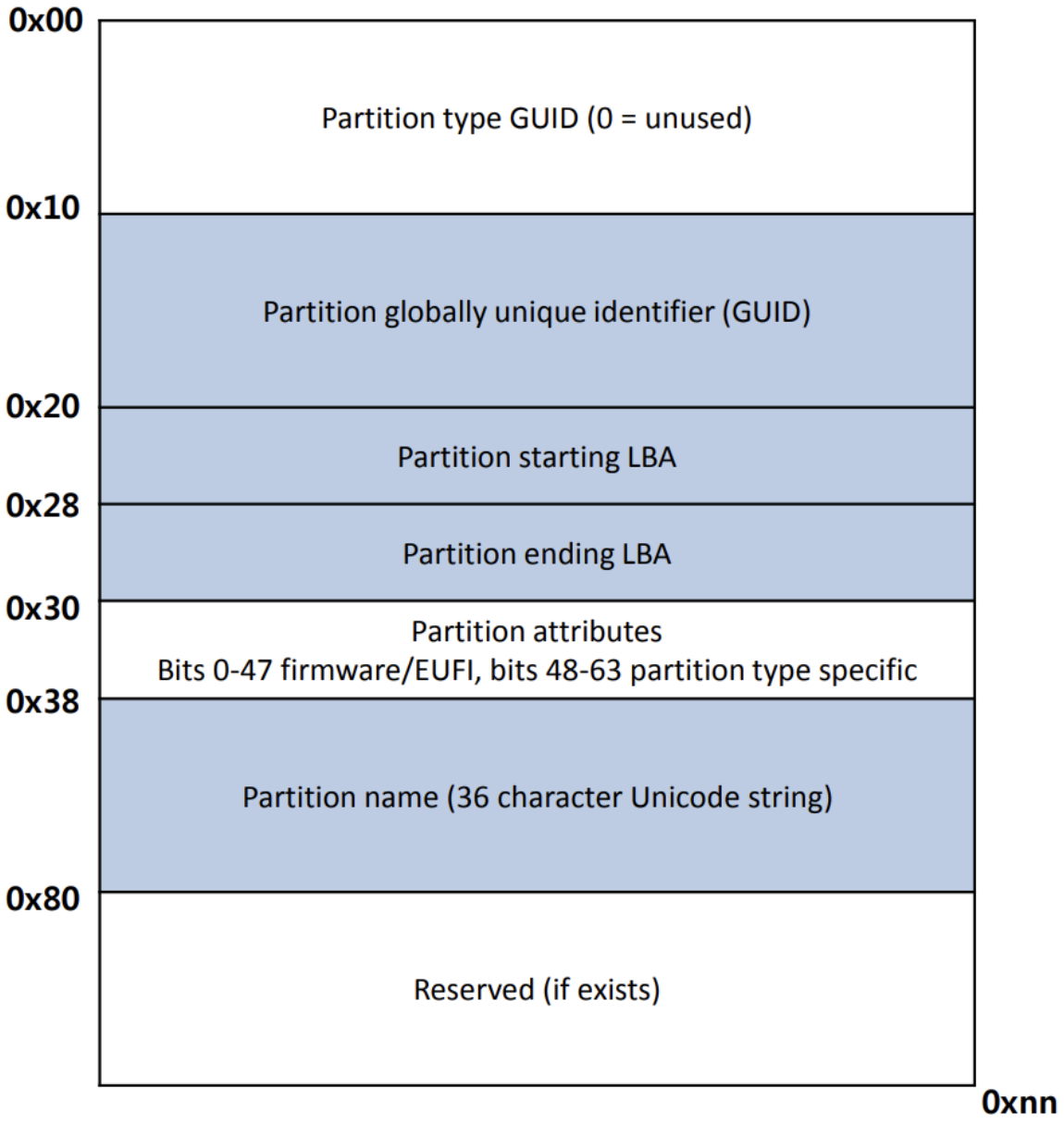
데이터 구조체

- GPT 디스크의 첫 영역은 도스 파티션 테이블 사용
- 두번째 영역에 존재하는 GPT 헤더는 디스크 레이아웃을 설명하고 다음과 같

GPT 헤더 구조체



각 테이블 엔트리



분석 고려 사항

- GPT 디스크를 접할 일들은 거의 없음
- 하지만 리눅스를 GPT 디스크에 파티셔닝 하는데 사용할 수 있다
- GPT는 복구의 용이를 위해 파티션 테이블 백업본을 갖는다
- 섹터 0, 섹터1 과 같은 비사용 공간, 엔트리에 데이터 숨김 가능

Chapter 7. 다중 디스크 볼륨

💡 중요서버에서 성능, 신뢰성, 확장성을 위해 사용하는 다중 디스크에 대한 이해

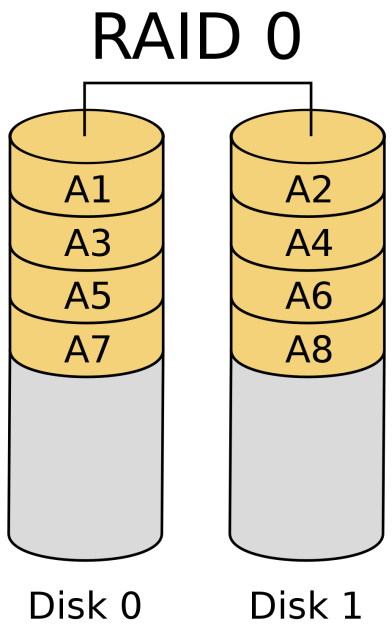
7.1. RAID (Redundant Array of Inexpensive/Independent Disk)

- 고성능, 중요 시스템에서 자주 사용
- 저비용 디스크로 고비용과 고성능 디스크와 유사한 저장능력
- 하드웨어 컨트롤러나 소프트웨어 드라이버는 여러 디스크를 한번에 병합하고 이는 단일 큰 볼륨으로 인식 함

RAID 레벨

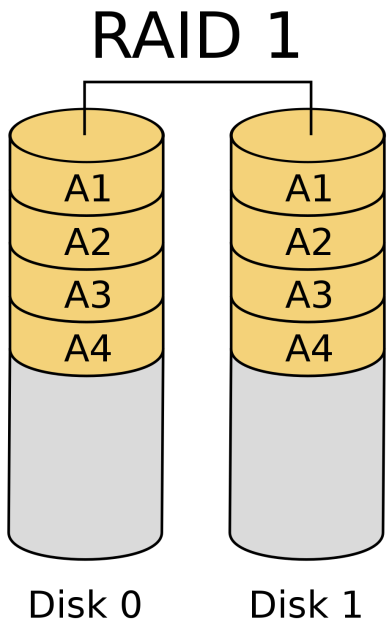
- RAID 볼륨은 하드디스크와 결합하는 하드웨어나 소프트웨어에 의해 만들어지는 볼륨

레벨 0



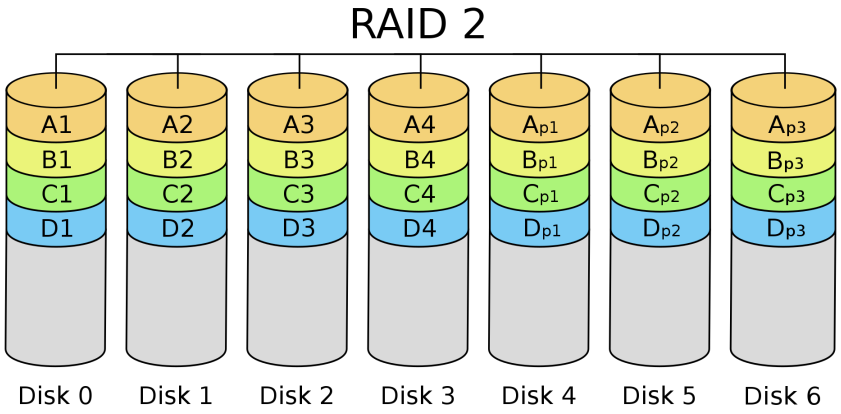
- 2개 이상의 디스크 사용
- 두 디스크를 교차

레벨 1



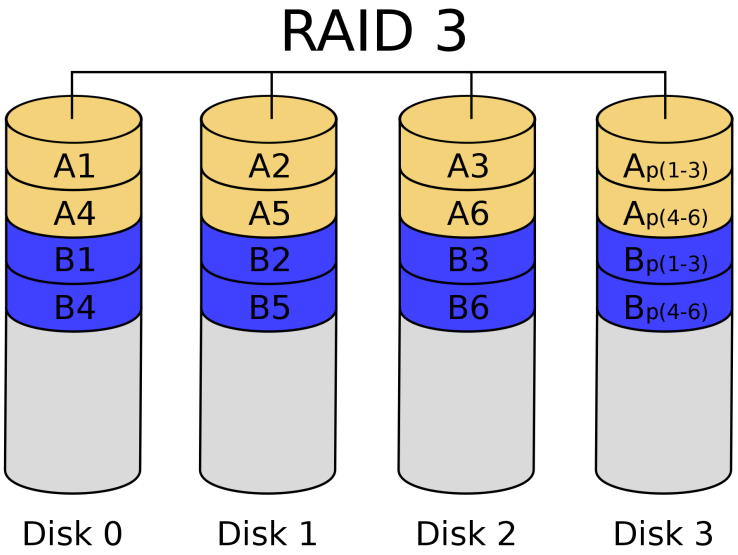
- 2개 이상의 디스크를 사용해서 데이터 미러링
- 한 데이터에 문제가 생기면 다른 디스크를 복구용으로 사용

레벨 2



- 디스크에서 데이터를 읽을 때 정확하지 않은 데이터를 여러 디스크에 분배
- 별도의 오류 수정 코드 값들을 포함하게 함
- 비트 크기의 데이터 묶음으로 나눔

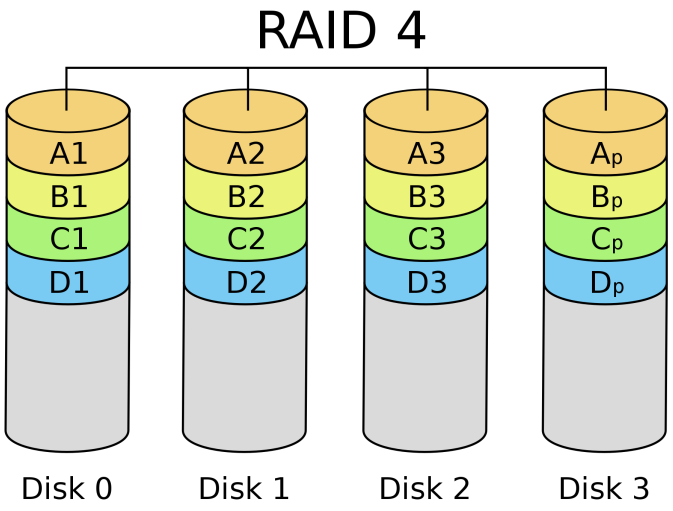
레벨 3



- 최소 3개의 디스크 필요

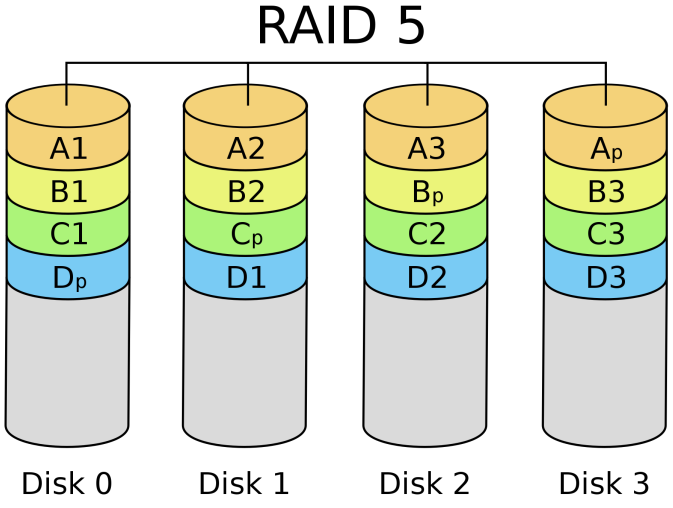
- 전용 패리티 디스크를 가짐
  - 다른 두 디스크에서 오류 발생시 그 내용을 생성하기 위해 사용

레벨 4



- 볼륨을 블록 크기 데이터 묶음으로 배분하는 차이 외에는 레벨 3과 동일

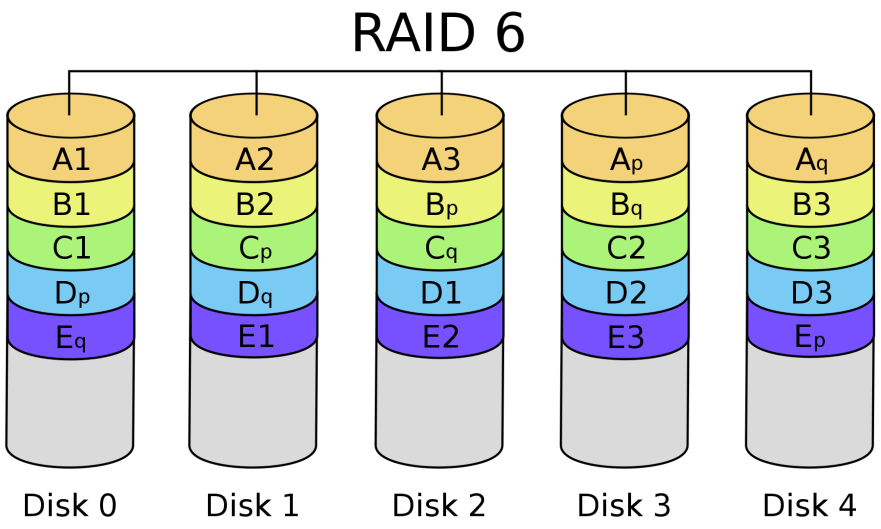
레벨5



- 레벨 4와 유사

- 패리티 병목 현상 제거
- 전용 패리티 디스크가 없음

레벨 6 (가장 최근)



- 각 디스크에 두 개의 패리티(더블 패리티) 데이터를 기록
- 디스크가 최대 2개까지 고장나도 데이터 손실이 발생하지 않음

하드웨어 RAID

- RAID 볼륨을 만드는 방법은 특정 하드웨어를 사용하는 것

배경

- 버스들 중 한개에 삽입하는 특별한 컨트롤러
- ATA/SCSI/Firewire 같은 일반적인 디스크 컨트롤러에 삽입하는 형태

소프트웨어 RAID

- RAID를 소프트웨어로 구분한 방식

전반적 분석 의견

- 이 시스템을 조사하는 일은 많지않음
- 구형 방법의 다양성 때문에 조사하기 쉽지 않음
- 원본디스크 수정하지 않도록 매우 주의

요약

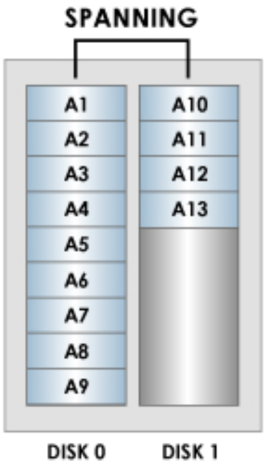
- RAID 시스템 수집의 연습 필요
- 개별적 디스크 수집은 매우 안전해야함

7.2. 디스크 스패닝

- 여러개의 디스크를 하나의 큰 디스크로 보여주도록 하는 것

개요

- 메모를 수집하기 위해 일반 노트 대신 여러개의 링 바인더를 사용하는 것과 같음
    - 하나가 다 차면 새로운 것을 사는 것이 아니라 추가적으로 끼워서 커스텀 하는
- 디스크 스패닝은 새로운 저장공간을 기존 저장 공간 끝에 추가 가능하다



리눅스 MD

- 2가지 방법 존재
  - MD 드라이버는 기본 스패닝을 수행
  - LVM 은 더욱 개선된 시스템

*두 시스템은 모두 주요 리눅스 배포판에 존재