

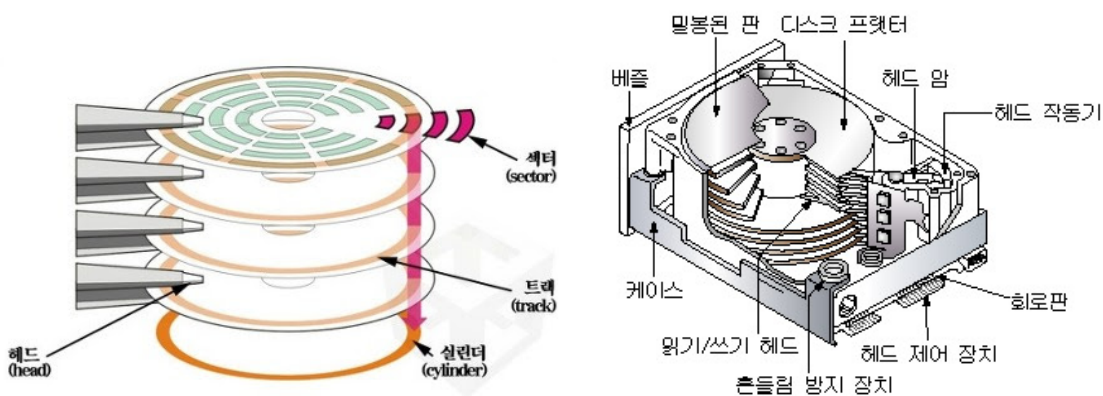
7.보조기억장치

7.보조기억장치

7.1. 하드디스크

- HDD: Hard Disk Drive
- 자기적인 방식으로 데이터를 저장함. 때문에 magnetic disk의 일조응로 지칭하기도 함.

7.1.1. 구조



- 플래터(platter): 자기 물질로 덮혀 있으며, N/S극을 저장하는 방식으로 데이터를 저장함.
 - 트랙(track): 플래터를 동심원으로 나눈 부분.
 - 섹터(sector): 가장 작은 전송 단위. 일반적으로 512byte이며, 하드 크기에 따라 차이남. 여러 섹터를 묶어 블록(block)이라고 하기도 함.
 - 실린더(cylinder): 여러 겹의 플래터 상에서 같은 트랙이 위치한 곳을 모아 연결한 논리적 단위.^[1] 연속된 정보는 각 플래터 위/아래에 저장하며, 이런 방식으로 디스크 암을 움직이지 않고 바로 데이터에 접근할 수 있다.
- 스피indle(spindle): 플래터를 회전시켜 데이터를 검색. 돌리는 속도는 RPM(Revolution Per Minute)으로 나타냄.
- 헤드(head): 플래터를 대상으로 데이터를 읽고 씴. 플래터 위에 떠 있는 바늘같은 부분.
 - single-head disk: 플래터 한 면당 하나의 헤드. 헤드가 이동하기 때문에 movable-head disk라고도 한다.
 - multiple-head disk: 트랙마다 헤드가 있다. 탐색시간이 따로 필요하지 않다. 때문에 fixed-head disk라고도 불린다.
- 디스크 암(disk arm): 헤드의 위치를 조정함.

7.1.2. 데이터 접근

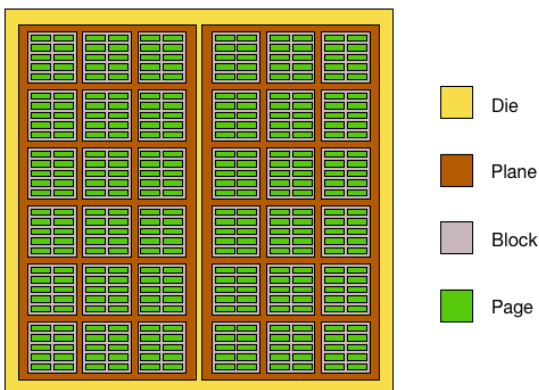
- 탐색 시간(seek time): 데이터를 가져오기 위해 헤드가 이동하는 시간.
- 회전 지연(rotational latency): 헤드가 있는 곳으로 플래터를 회전시키는 시간
- 전송 시간(transfer time): 하드 디스크와 컴퓨터 간 데이터를 전송하는 시간

참조 영역	시간
L1 cache	0.5ns
L2 cache	5ns
memory	7ns
reading 1mb from memory sequentially	250.000ns

참조 영역	시간
HDD searching time	10.000.000ns
reading 1mb from HDD sequentially	30.000.000ns
packet traversal time from California to Nederland	150.000.000ns

7.2. 플래시 메모리

- USB, SD card, SSD 등
- NAND, NOR 플래시 메모리가 있으며, 주로 전자가 쓰인다.
- 전기적으로 데이터를 읽고 쓸 수 있는 반도체 기반 저장 장치.
- 보조기억 장치 외에도, ROM 등 거의 모든 전자 제품에 사용한다.
- cell: 데이터를 저장하는 가장 작은 단위. 이것들이 모여 MB, GB, TB 용량이 된다.
- SLC(single level cell):
 - 하나의 셀에 1비트를 저장
 - 읽기/쓰기가 빈번하며 고성능의 빠른 저장 장치가 필요한 경우
 - 빠른 비트 입출력, 긴 수명이 장점
 - 용량 대비 높은 가격
- MLC(multiple level cell):
 - 하나의 셀에 2비트 저장
 - 대용량화에 유리함
 - 용량대비 가격이 저렴
 - 수명과 속도가 SLC 보다 비교적 짧고 느림
- TLC(triple level cell):
 - 한 셀에 3비트 저장
 - 용량대비 가격이 저렴함
 - 수명과 속도가 MLC 보다 짧고 느림
- QLC(quadruple level cell)



- cell → page → block → plane → die 순으로 단위가 커짐
- 읽기와 쓰기는 page 단위에서 이루어짐
- 삭제는 페이지보다 큰 단위, block 부터 가능
- page state:
 - free: 어떠한 데이터도 저장하지 않은 상태
 - valid: 유효한 데이터를 저장한 상태. 이 상태인 페이지는 다른 데이터를 저장할 수 없음
 - invalid: 유효하지 않은 데이터(trash)를 저장한 상태
- valid page의 특성 때문에 블록에 A, B, C를 저장하고 A를 수정하기 위해서는 A를 trash로 변경 후 A'을 입력해야 한다.
- 이 처럼 낭비되는 메모리 공간을 최소화하기 위해 GB(garbage collection) 기능을 제공한다.
- GB는 유효한 페이지만 새 블록에 복사하고 기존의 블록은 삭제한다.

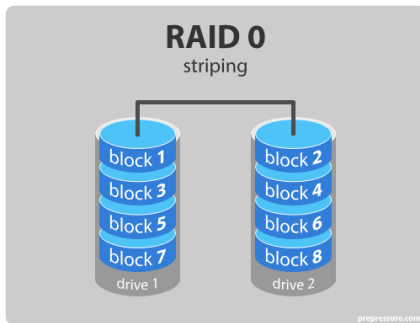
7.3. RAID의 정의와 종류

- Redundant Array of Independent Disk
- HDD, SSD를 사용하는 기술로, 데이터의 안전성, 높은 성능을 위해 여러 개의 물리적 보조기억장치를 하나의 논리적 보조기억장치처럼 사용하는 기술.

7.3.1. 종류

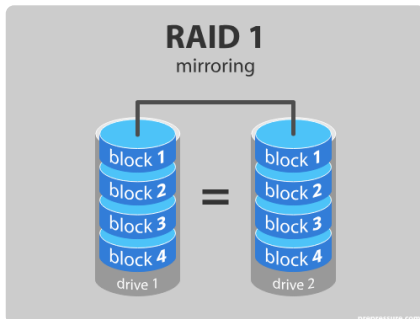
- RAID Level

- 0:



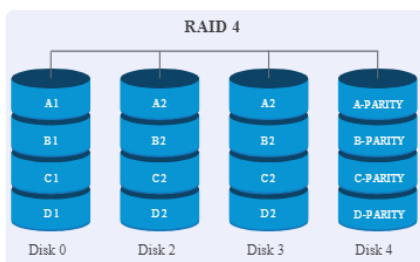
- 여러 보조기억 장치에 데이터를 단순히 나누어 저장/구성하는 방식
- 마치 줄무늬처럼 분산되어 저장된다 해서 이런 데이터를 **stripe**이라 하고, 저장하는 것을 **striping**이라고도 한다.
- 데이터는 디스크 개수만큼 나뉘어 저장된다.
- 하나의 저장 장치를 이용했을 때와는 달리, 병렬 처리가 가능해지기에 읽기/쓰기가 빨라진다.^[2]
- 디스크 중 하나에 문제가 생긴다면 정보를 읽는 데 문제가 생길 수 있다.

- 1:



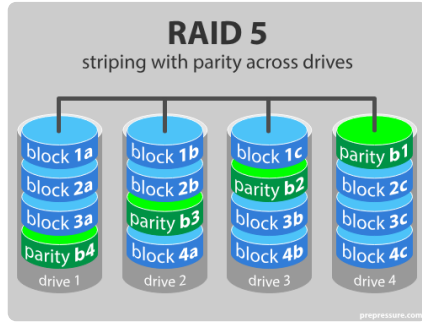
- 복사본을 만드는 방식. **mirroring**이라고도 한다.
- 두 번 써야 하기에 쓰기 시간이 더 걸리지만 복구가 간단하다.
- 필요한 용량이 크기 때문에 비용도 증가한다.

- 4:



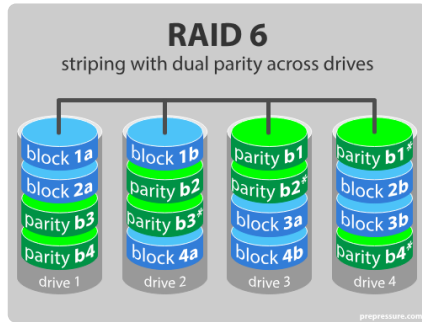
- 완전한 복사본 대신 오류를 검출/복구하기 위한 정보(**parity bit**^[3])를 저장한다.
- 이로써 1보다 더 적은 용량으로 데이터를 안전하게 보관할 수 있다.
- 새로운 데이터가 저장될 때마다 **parity bit**도 데이터를 써야하므로, **parity bit**를 쓰는 디스크가 하나인 level 4에서는 병목 현상이 생길 수 있다.

• 5:



- 4와 달리 각 디스크에 자신의 디스크가 아닌 parity bit를 저장하므로써 병목 현상을 해소한다.

• 6:



- 기본적인 구성은 5와 같지만, 하나의 디스크에 서로 다른 두 개의 패리티를 둔다.
- 위 방식으로, 오류를 검출/복구할 수 있는 수단이 두 개가 생긴다.
- 때문에 보다 안전하지만 쓰기 속도가 느려진다.
- 이외, 0과 1을 혼합한 10, 0과 5를 혼합한 50 방식도 있다. 이들을 Nested RAID라 칭한다.

1. 트랙은 하나의 플래터를 동심원으로 나눴다면, 실린더는 각각의 플래터에서 같은 위치에 있는 트랙을 모아놓은 원통형 공간이다.↵
2. 같은 양의 데이터를 읽기 위해 하나의 4TB 저장장치를 오가는 것 보다 빠르다.↵
3. 오류 검출만 가능하고 복구는 불가능하다. 다만 RAID에서는 오류 수정이 가능한데, 계산식 예시는 다음과 같다.

RAID array

Parity data is used by RAID arrays (redundant array of independent/inexpensive disks) to achieve redundancy. If a drive in the array fails, remaining data on the other drives can be combined with the parity data (using the Boolean XOR function) to reconstruct the missing data.

For example, suppose two drives in a three-drive RAID 5 array contained the following data:

Drive 1: 01101101

Drive 2: 11010100

To calculate parity data for the two drives, an XOR is performed on their data:

01101101

XOR 11010100

10111001

The resulting parity data, 10111001, is then stored on Drive 3.

Should any of the three drives fail, the contents of the failed drive can be reconstructed on a replacement drive by subjecting the data from the remaining drives to the same XOR operation. If Drive 2 were to fail, its data could be rebuilt using the XOR results of the contents of the two remaining drives, Drive 1 and Drive 3:

Drive 1: 01101101

Drive 3: 10111001

as follows:

01101101 Drive 1

XOR 10111001 Drive 3

11010100 Reconstructed Drive 2

The result of that XOR calculation yields Drive 2's contents. 11010100 is then stored on Drive 2, fully repairing the array.

XOR logic is also equivalent to even parity (because $a \text{ XOR } b \text{ XOR } c \text{ XOR } \dots$ may be treated as $\text{XOR}(a,b,c,\dots)$, which is an n-ary operator that is true if and only if an odd number of arguments is true). So the same XOR concept above applies similarly to larger RAID arrays with parity, using any number of disks. In the case of a RAID 3 array of 12 drives, 11 drives participate in the XOR calculation shown above and yield a value that is then stored on the dedicated parity drive.

Extensions and variations on the parity bit mechanism "double," "dual," or "diagonal" parity, are used in RAID-DP.

[Wiki: Parity bit # RAID array.](#)

