Linux 저장장치

강의: 신 인 호 교수

LINUX (리눅스)

목 차

01. 저장장치 종류

LINUX (리눅스)

리눅스 지정된 저장장치명

플로피 디스크 : /dev/fd

첫번째 플로피디스크 : /dev/fd0

첫번째 플로피디스크 : /dev/fd1

SCSI 디스크 : /dev/sh

첫번째 SCSI디스크 : /dev/sha

첫번째 SCSI디스크 : /dev/shb

cd-rom(SCSI cd-rom) :

/dev/scd0, /dev/sr0

리눅스 지정된 저장장치명

cd-rom(SCSI cd-rom) :

/dev/scd0, /dev/sr0

IDE 디스크 : /dev/hd

Primary Master : /dev/hda

Primary Slave : /dev/hdb

Secendary Master: /dev/hdc

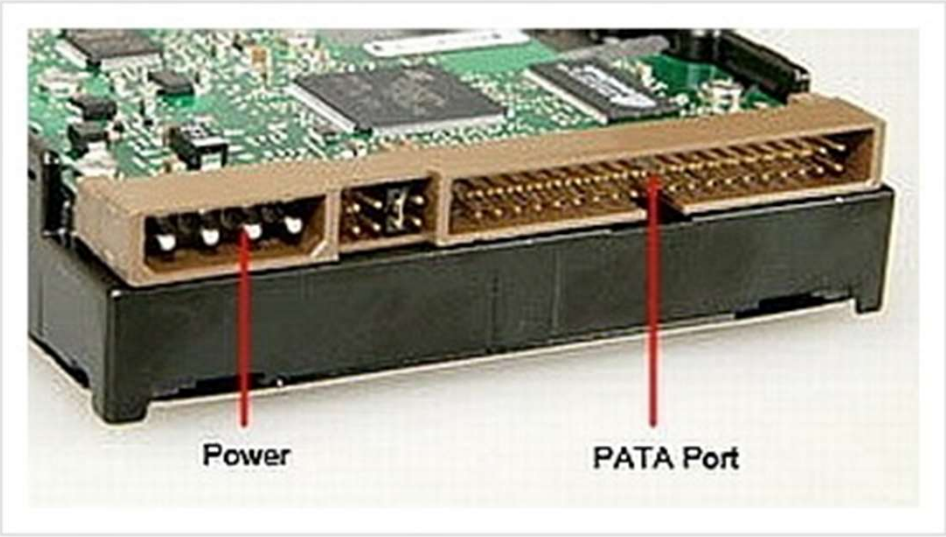
Secendary Slave : /dev/hdd

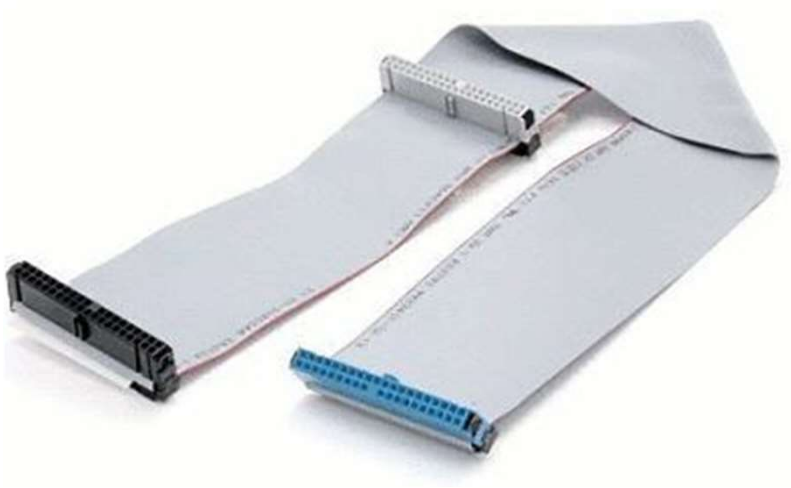
XT(씨게이트 모멘터스 XT) 디스크 : /dev/xd

첫번째 XT 디스크 : /dev/xda

첫번째 XT디스크 : /dev/xdb

IDE(Integrated Drive Electronics)





가장 오래된 규격으로 포트는 40개의 핀으로 구성된 직사각형 입니다.

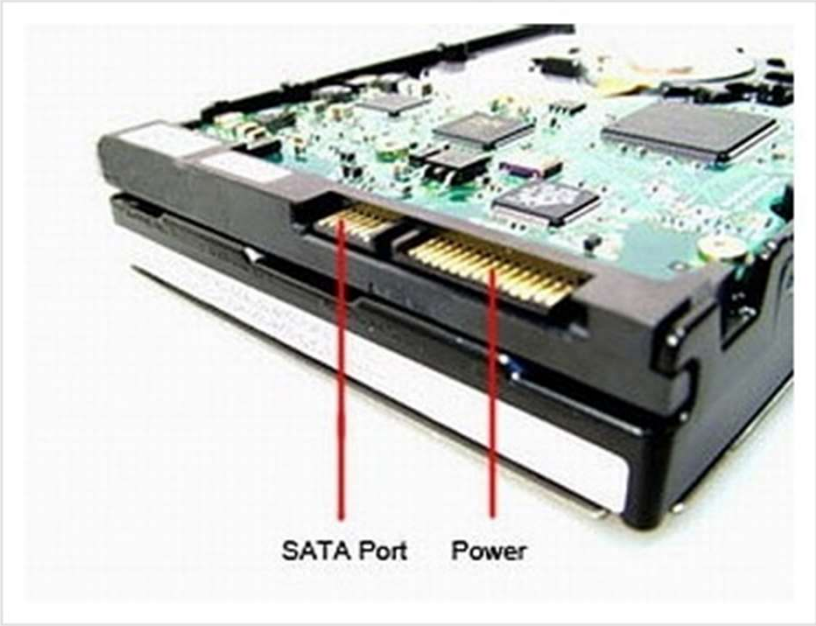
데이터를 병렬로 전송한다는 뜻에서 PATA(Parallel Advanced Tachnology Attachment) 인터페이스라고 부르기도 합니다.

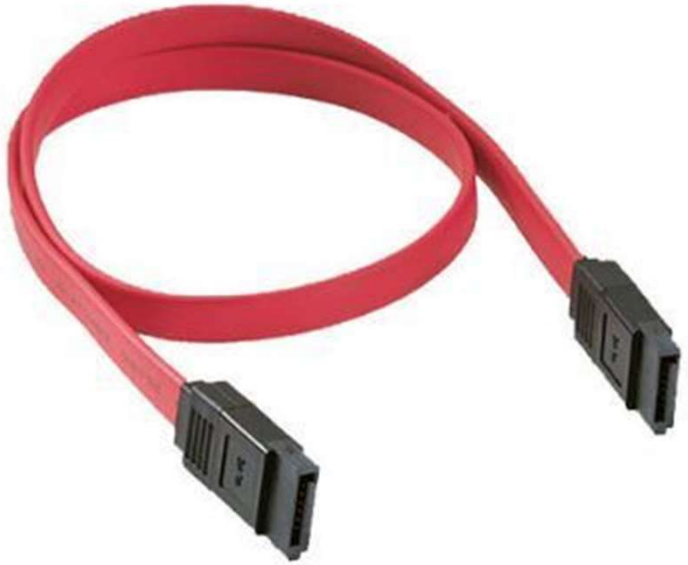
버전별로 데이터 전송속도가 다르며 최신 규격인 UDMA6 모드에서는 초당 133.3MB의 데이터를 전송한다.

현재는 초기형 IDE보다 성능이 향상된 E-IDE(Enhanced IDE) 규격이 사용 되고있다.

참고: https://m.blog.naver.com/bizblocklll/222188130854

SATA(Serial Advanced Technology Attachment)





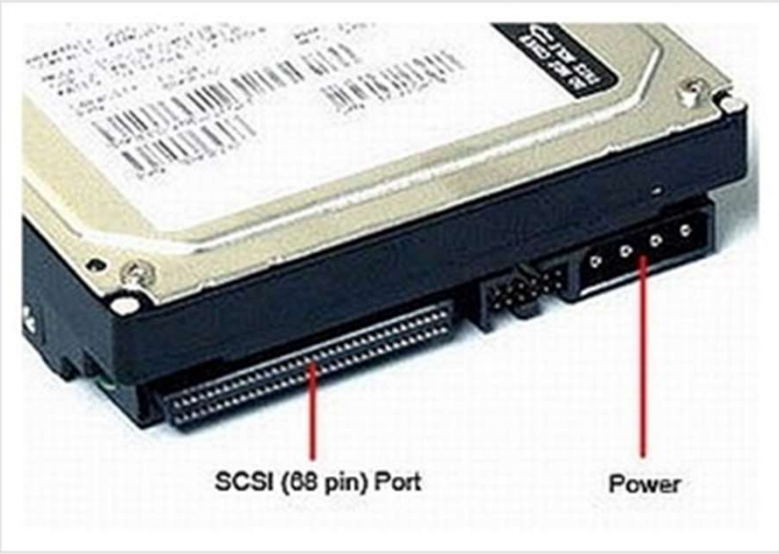
최근에 나온 인터페이스로 하드디스크 드라이브의 속도와 연결 방식 등을 개선하기 위해 나왔습니다.

SATA 1 은 초당 150MB, SATA 2는 초당 300MB의 전송 속도를,

SATA 2 는 USB처럼 허브를 이용해 하나의 포트에 여러 개의 하드디스크를 연결할 수 있고 PC를 끄지 않고 장치를 연결하는 핫 플러그 등의 기능이 추가되었습니다.

참고: https://m.blog.naver.com/bizblocklll/222188130854

SCSI(Small Computer System Interface)



서버나 워크스테이션 등에 쓰이는 고속 인터페이스로,

무엇보다 안정성이 높은 것이 최대의 장점이지만 가격이 고가입니다.

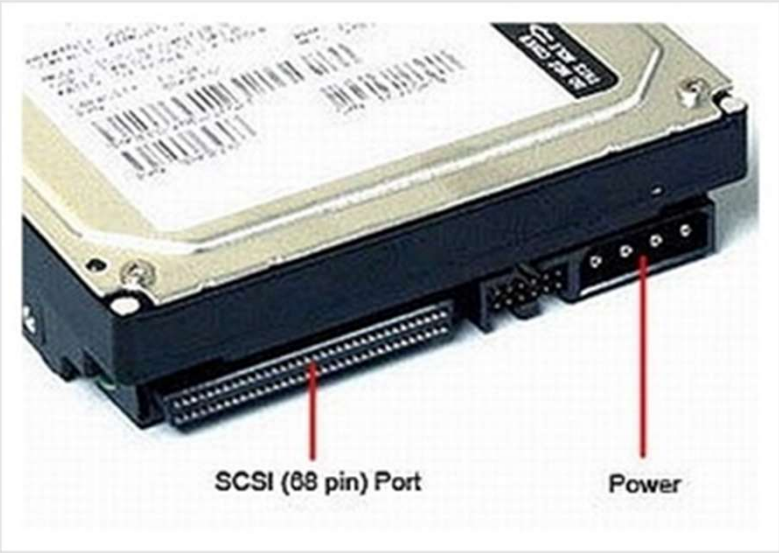
이 규격을 쓰려면 별도의 확장 카드를 달아야 해서 또 추가 비용이 들어갑니다.

최신 규격인 울트라 320은 최대 320MB/초의 속도를 낸다.

최근 이를 대체하는 SAS 인터페이스가 등장하여 널리 사용되고 있다

참고: https://m.blog.naver.com/bizblocklll/222188130854

SAS(Serial Attached SCSI)



SCSI 규격을 한 단계 발전시킨 것으로 이 규격 역시 서버 등의 대형 컴퓨터에 주로 쓰입니다.

성능은 울트라 320 SCSI보다 좋구요 커넥터와 선은 SATA와 같은 것을 이용하지만 컨트롤러가 달라 SAS 규격의 하드디스크가 SATA 제품보다 훨씬 비쌉니다. SATA 규격 하드디스크 드라이브를 SAS 장치에 꽂아 쓸 수는 있지만 반대로 SAS 하드디스크를 일반 SATA 인터페이스에 연결하지는 못합니다.

참고: https://m.blog.naver.com/bizblocklll/222188130854

SSD(Solid State Drive. Solid State Disk)

하드디스크 드라이브(HDD)와 비슷하게 동작하면서도 기계적 장치인 HDD와는 달리 반도체를 이용하여 정보를 저장합니다. 고속으로 데이터를 입출력할 수 있으면서도 기계적 지연이나 실패율이 현저히 적고, 또 외부의 충격으로 데이터가 손상되지 않으며, 발열·소음 및 전력소모가 적고, 소형화·경량화할 수 있는 장점이 있습니다.

참고: https://m.blog.naver.com/bizblocklll/222188130854

SSD(Solid State Drive. Solid State Disk)

플래쉬 방식의 비휘발성 낸드 플래쉬 메모리 램(RAM) 방식의 휘발성 DRAM을 사용합니다.

플래시 방식은 RAM 방식에 비하면 느리지만 HDD보다는 속도가 빠르며, 비휘발성 메모리를 사용하여

갑자기 정전이 되더라도 데이터가 손상되지 않습니다.

DRAM 방식은 빠른 접근이 장점이지만 제품 규격이나 가격, 휘발성이라는 문제점이 있다.

2008년 128GB에 이어 256GB의 대용량 SSD가 개발되면서 노트북PC나 데스크톱PC에도 활용할 수 있게 되었습니다. 비싼 가격 문제를 해결한다면 HDD를 대체할 차세대 저장장치가 될 것으로 전망됩니다

참고: https://m.blog.naver.com/bizblocklll/222188130854

02. RAID 종류와 구성방법

LINUX (리눅스)

RAID*(Redundant Array of Independent Disk)*

1. RAID는 여러 개의 디스크를 묶어 하나의 디스크 처럼 사용하는 기술 입니다.
2. RAID를 사용하였을 때 기대 효과는
   - 대용량의 단일 볼륨을 사용하는 효과
   - 디스크 I/O 병렬화로 인한 성능 향상 (RAID 0, RAID 5, RAID 6 등)
   - 데이터 복제로 인한 안정성 향상 (RAID 1 등)이 있습니다
3. RAID 0 ~ RAID 6(1,2,5,6을 주로 사용)
4. RAID 2~5구성시 디스크 2개이상 오류시 복구불가

참고: https://devocean.sk.com/blog/techBoardDetail.do?ID=163608

참고: https://m.blog.naver.com/whentlr/120199172537

파티션 분할

fdisk –l 현재 디스크 및 파티션 보기

명령	설명	
a	파티션의 부트 가능 플래그를 전환합니다.	
c	파티션의 DOS 호환성 플래그를 전환합니다.	
d	파티션을 삭제합니다.	
l	fdisk에 알려진 파티션 유형을 나열합니다.	
m	모든 사용 가능한 메뉴 목록을 표시합니다.	
n	새 파티션을 추가합니다.	
p	현재 디스크에 대한 파티션 테이블을 프린트합니다.	
q	변경사항을 저장하지 않고 종료합니다.	
t	파티션에 대한 파일 시스템 유형을 변경합니다.	
u	표시/항목 단위를 변경합니다.	
v	파티션 테이블을 검증합니다.	
w	디스크에 테이블을 쓰고 종료합니다.	
x	다음의 전문가용 추가 기능을 나열합니다.	
	b---	파티션에서 데이터의 시작 위치를 이동시킵니다.
	c---	실린더 수를 변경합니다.
	d---	파티션 테이블에 있는 소스 데이터를 프린트합니다.
	e---	디스크의 확장 파티션을 나열합니다.
	h---	디스크의 헤드 수를 변경합니다.
	r---	메인 메뉴로 갑니다.
	s---	디스크의 섹터 수를 변경합니다.

참고: https://devocean.sk.com/blog/techBoardDetail.do?ID=163608

참고: https://m.blog.naver.com/whentlr/120199172537

Raid 구성을 위한 파티션 분할

fdisk /dev/sdb

command : n -> 새로운 파티션 분할

select : p -> Primary 파티션 선택

Partition number : 1 -> 파티션번호 1선택(Primary 파티션은 최대 4개까지 생성가능)

First sector : enter -> 시작 섹터 번호입력(1개의 파티션만 계획 중이므로 첫 섹터 설정)

Last sector : enter -> 마지막 섹터 번호입력(1개의 파티션만 계획 중이므로 마지막 섹터 설정)

command : t -> 파일시스템 유형선택

Hex Code : fd -> 'Linux raid AutoDetect' 유형번호

command : p -> 설정 내용확인

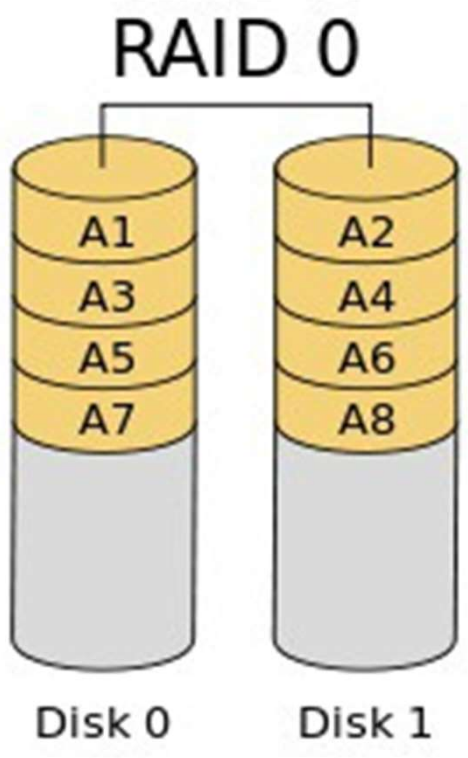
command : w -> 설정저장

참고: https://devocean.sk.com/blog/techBoardDetail.do?ID=163608

RAID 0

-.Striping (스트라이핑) 이라고도 부르는 방식,

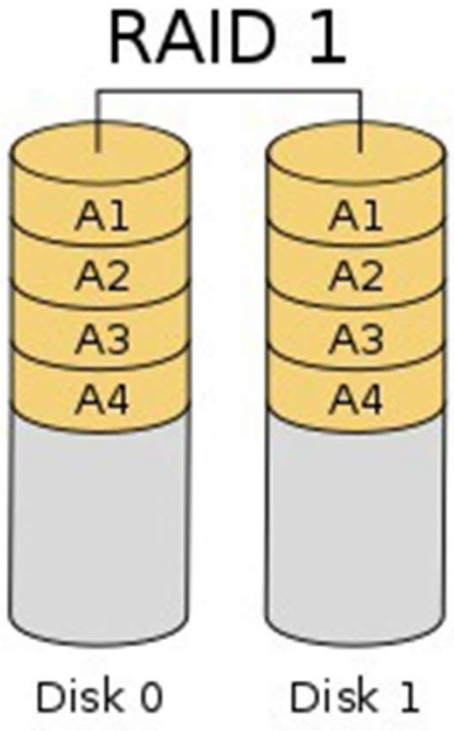
-.최소디스크 2개이상 필요



참고: https://devocean.sk.com/blog/techBoardDetail.do?ID=163608

RAID 1

-.Mirroring (미러링) 이라고도 부르는 방식
-.데이터를 복사하여 2개 가지고 있다.
-.N-1개의 디스크가 고장나도 데이터 사용이 가능
-.최소디스크 2개이상 필요



참고: https://devocean.sk.com/blog/techBoardDetail.do?ID=163608

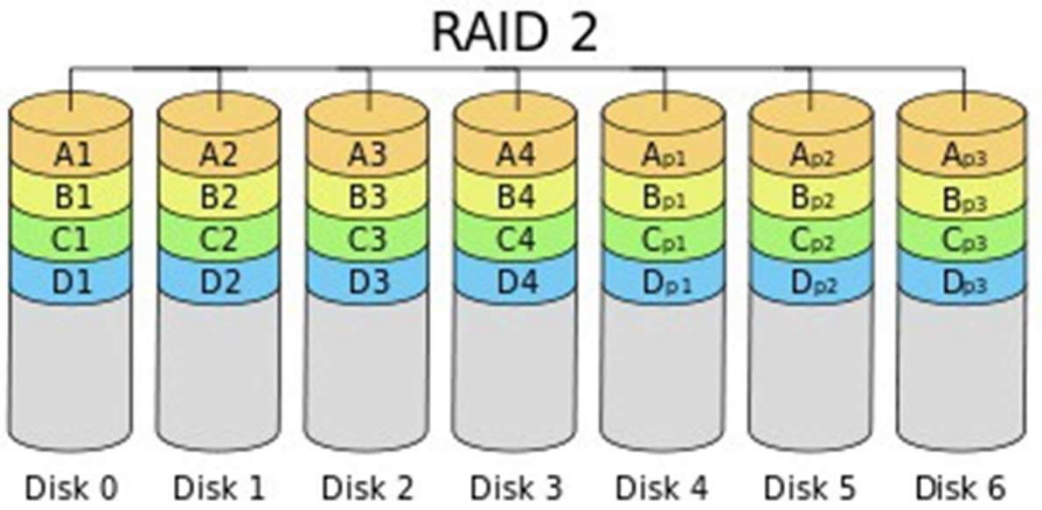
RAID 2

-. bit 단위로 striping을 하고, error correction을 위해 Hamming code 를 사용.

-. m+1개의 데이터 디스크와 m개의 패리티 디스크로 구성되어있다.(N == (m+1) + m))

-. 1개의 디스크 에러 시 복구 가능합니다.

-. 최소디스크 3개 이상 필요



참고: https://devocean.sk.com/blog/techBoardDetail.do?ID=163608

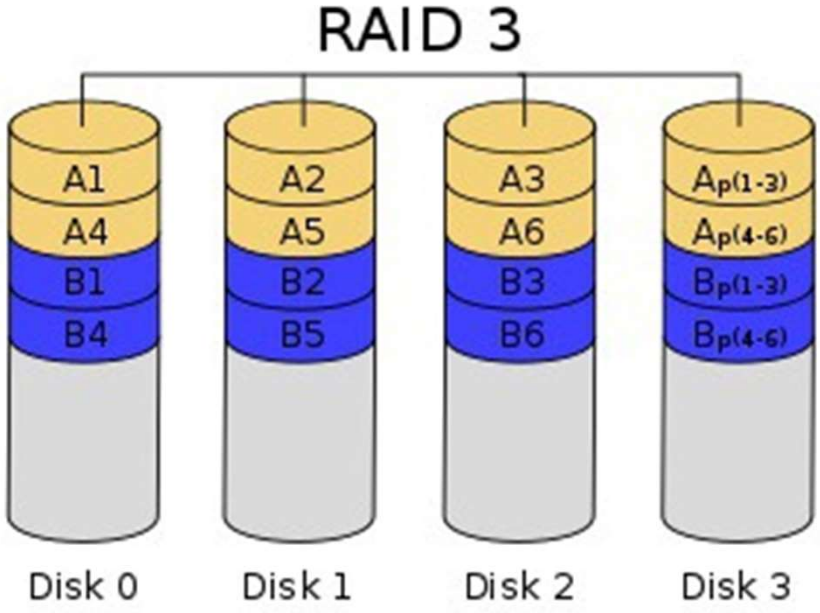
RAID 3

-.Byte 단위로 striping을 하고, error correction을 위해 패리티 디스크를 1개 사용.

-.용량 및 성능이 단일 디스크 대비 (N-1) 배 증가

-.1개의 디스크 에러 시 복구 가능합니다.

-.최소디스크 3개이상 필요



참고: https://devocean.sk.com/blog/techBoardDetail.do?ID=163608

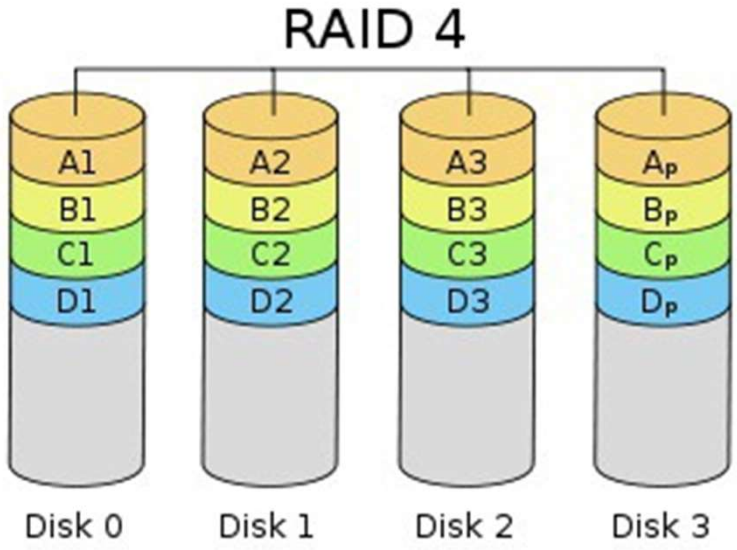
RAID 4

-.Block 단위로 striping을 하고, error correction을 위해 패리티 디스크를 1개 사용.

-.용량 및 성능이 단일 디스크 대비 (N-1) 배 증가

-.1개의 디스크 에러 시 복구 가능합니다.

-.최소 3개의 디스크로 구성



참고: https://devocean.sk.com/blog/techBoardDetail.do?ID=163608

RAID 5

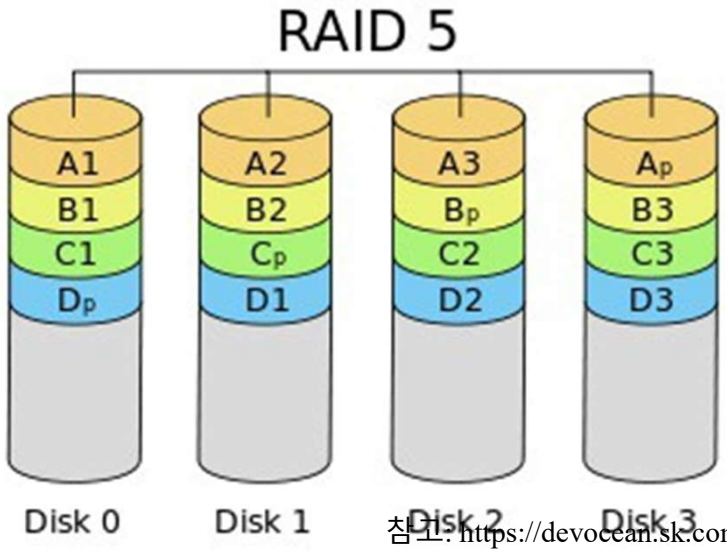
-.제일 사용 빈도가 높은 RAID Level

-.Block 단위로 striping을 하고, error correction을 위해 패리티를 1개의 디스크에 저장하고, 패리티 저장 하는 디스크를 고정하지 않고, 매 번 다른 디스크에 저장

-.용량 및 성능이 단일 디스크 대비 (N-1) 배 증가

-.최소 3개의 디스크로 구성 가능하고, 1개의 디스크 에러 시 복구 가능합니다.

-.RAID 0에서 성능, 용량을 조금 줄이는 대신 안정성을 높인 RAID Level



참고: https://devocean.sk.com/blog/techBoardDetail.do?ID=163608

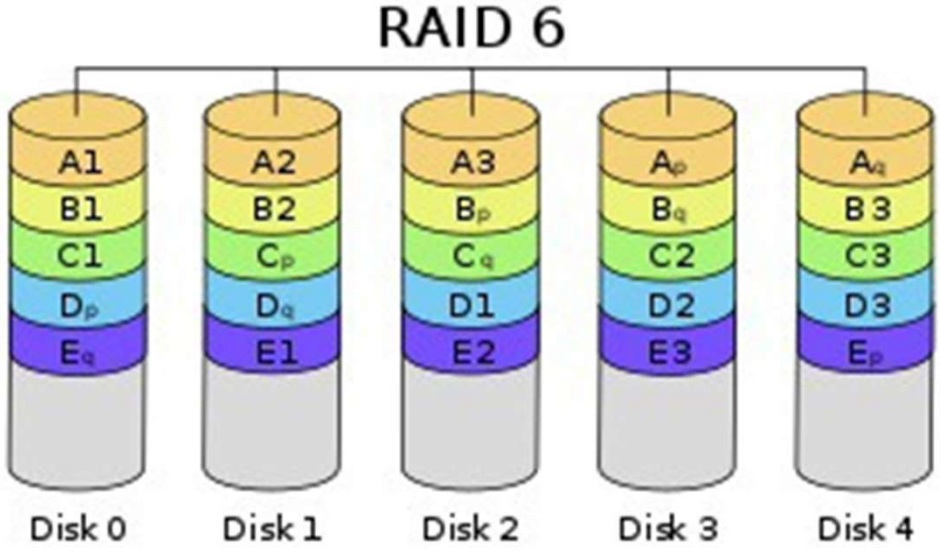
RAID 6

-.Block 단위로 striping을 하고, error correction을 위해 패리티를 2개의 디스크에 저장하는데, 패리티 저장 하는 디스크를 고정하지 않고, 매 번 다른 디스크에 저장합니다.

-.용량 및 성능이 단일 디스크 대비 (N-2) 배 증가

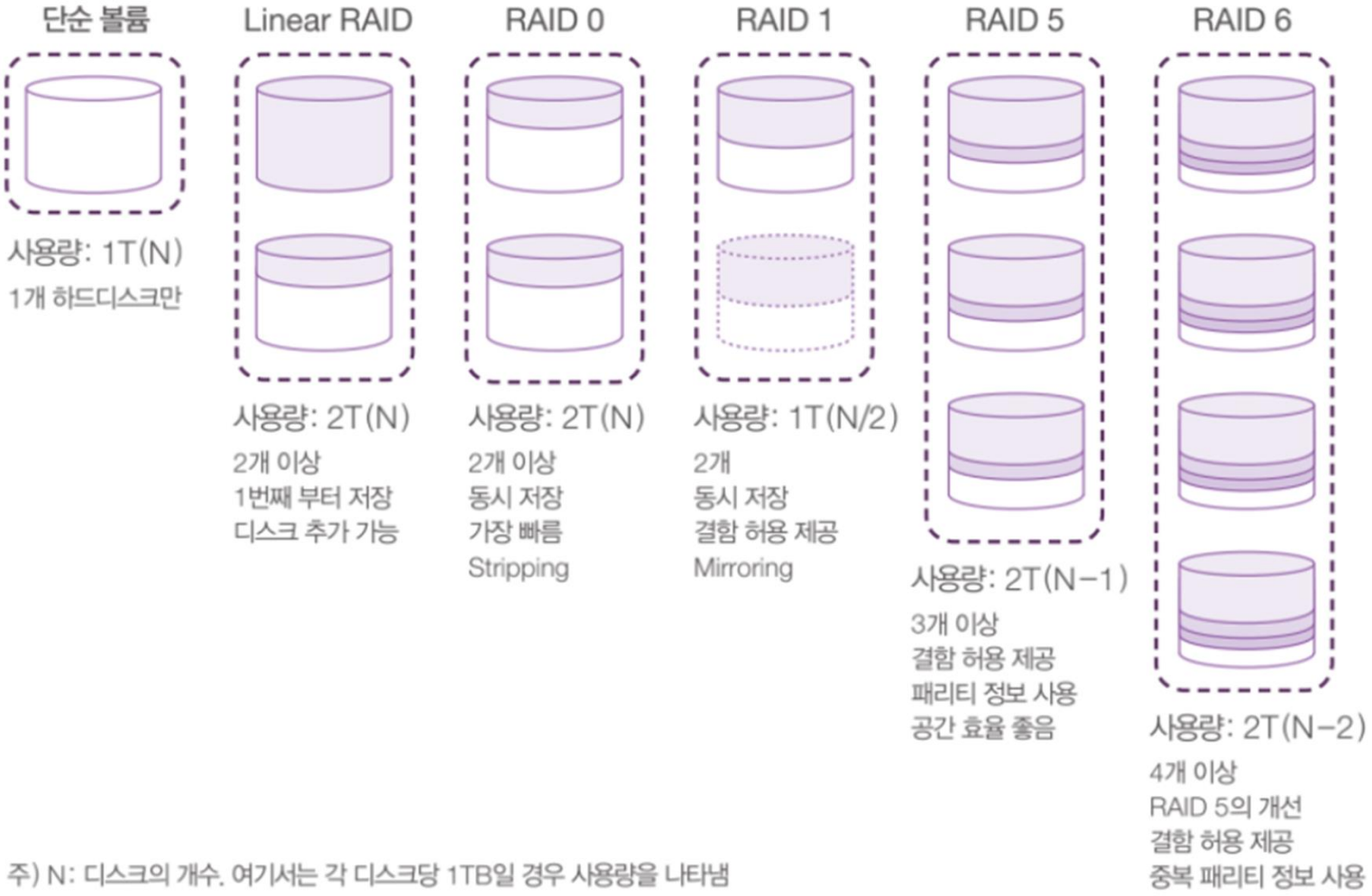
-.최소 4개의 디스크로 구성 가능, 2개의 디스크 에러 시 복구 가능합니다.

-.RAID 5에서 성능, 용량을 좀 더 줄이고, 안정성을 좀 더 높인 RAID Level



참고: https://devocean.sk.com/blog/techBoardDetail.do?ID=163608

Disk RAID 구성방식



주) N: 디스크의 개수. 여기서는 각 디스크당 1TB일 경우 사용량을 나타냄

[그림 6-23] 각 RAID 방식 비교

참고: 이것이 리눅스다.(365 Page)

LVM(Logical Volume Mnager) 개념

여러 개의 디스크를 합쳐서 1개의 파티션으로 구성한 후 다시 필요에 따라서 다시 나눠야 하는 상황에서 사용한다.

예를 들어 2TB 용량의 디스크 2개를 합친 후에 다시 1TB와 3 TB로 나눠야 할 때이다.

여러 개의 디스크를 합치지 않더라도 1개의 디스크로 LVM으로 구성하고 다시 파티션을 구분할 수도 있다.

LVM(Logical Volume Mnager)

물리볼륨(Physical Volume): /dev/hdb, /dev/hdc등의 파티션을 의미한다.

볼륨그룹(Volume Group): 물리볼륨를 합쳐서 1개의 물리 그르룹을 만든것이다.

논리볼륨(Logical Volume): 볼륨그룹를 1개이상으로 나눈 것으로 논리적그룹이라고도 한다.

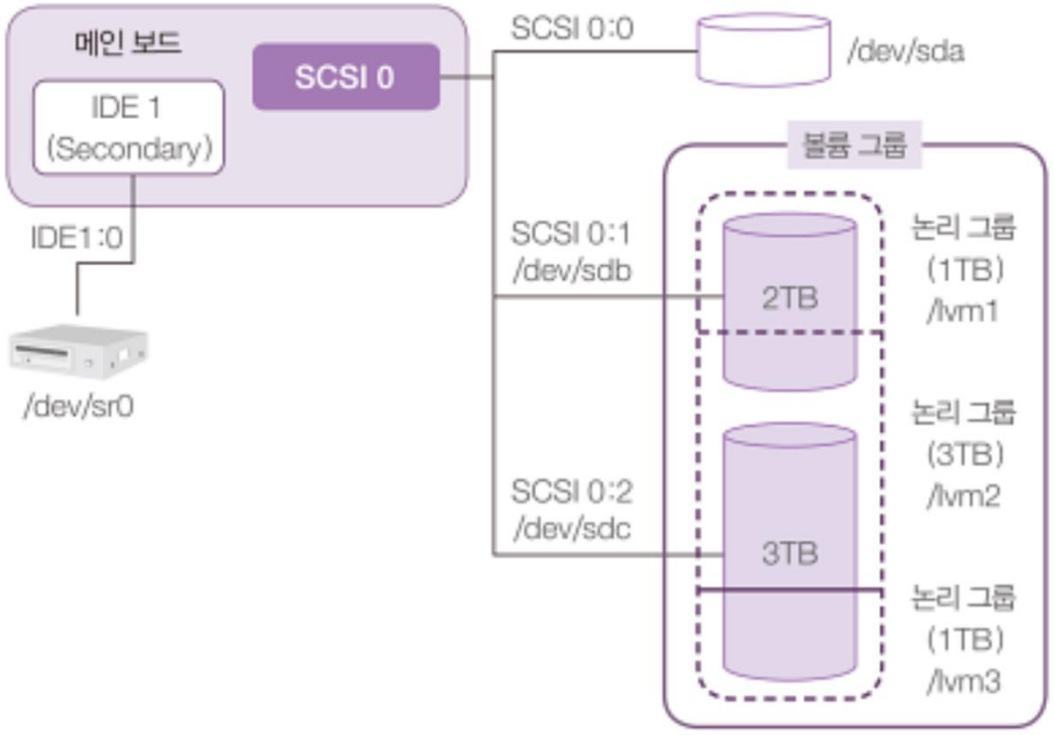


그림 6-98 LVM구현용 디스크 2개추가한 구성도

참고: 이것이 리눅스다.(422 Page)

LVM(Logical Volume Mnager) 구현순서

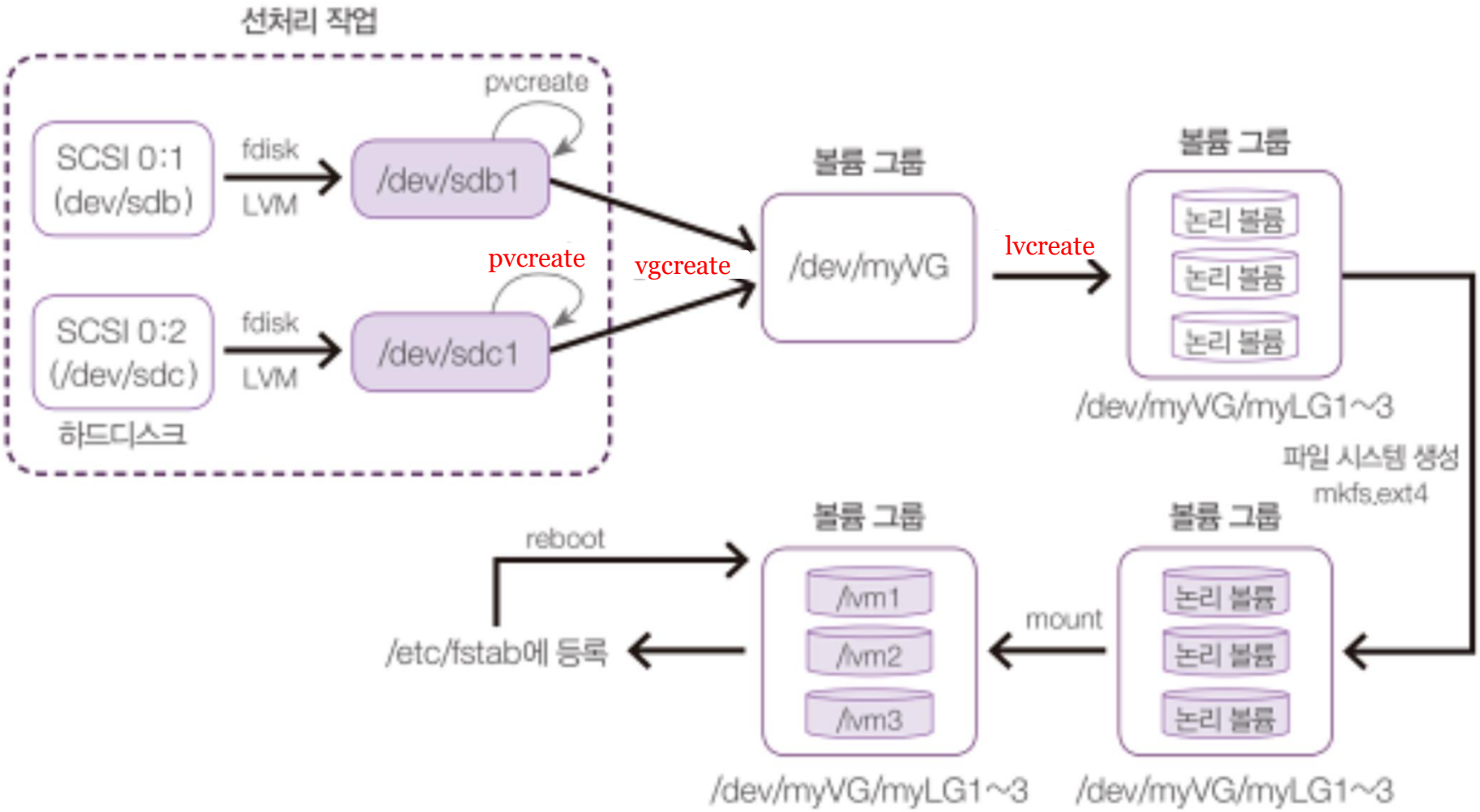


그림 6-99 LVM 구현 순서도

참고: 이것이 리눅스다.(423 Page)

참고: https://m.blog.naver.com/whentlr/120199172537

LVM 구성을 위한 파티션 분할

fdisk /dev/sdb

command : n -> 새로운 파티션 분할

select : p -> Primary 파티션 선택

Partition number : 1 -> 파티션번호 1선택(Primary 파티션은 최대 4개까지 생성가능)

First sector : enter -> 시작 섹터 번호입력(기본 설정)

Last sector : enter -> 마지막 섹터 번호입력(기본 설정)

command : t -> 파일시스템 유형선택

Hex Code : 8e -> 선택한 파일 시스템 유형번호('L'을 입력하면 유형번호 확인)

command : p -> 설정된 내용확인 (Linux LVM이 확인되어야 함)

command : w -> 설정저장

참고: https://devocean.sk.com/blog/techBoardDetail.do?ID=163608

참고: https://m.blog.naver.com/whentlr/120199172537

물리볼륨 및 볼륨그룹, 논리볼륨 생성

pvcreate /dev/sdc1 : 물리 볼륨생성

vgcreate /mtVG /dev/sdb1 /dev/sdc1 : 볼륨 그룹생성

vgdisplay : 볼륨그룹 생성확인

lvcreate --size 1G --name myLG1 myVG -> myVG아래 myLG1을 1GB 크기로 생성

lvcreate --size 3G --name myLG2 myVG -> myVG아래 myLG1을 1GB 크기로 생성

lvcreate --extents 100%FREE --name myLG3 myVG -> 나머지 모두 할당

참고: https://devocean.sk.com/blog/techBoardDetail.do?ID=163608

수고하셨습니다.

LINUX (리눅스)