

# Linux

Linux에서는 파티션이 되어 있어야 함(그냥 sda는 장치 연결만 되어 있는 상태)

파티션 작업시 파일시스템 생성후 마운트를 해 주어야 함

최상위 루트 디렉토리 안에 속해 연결되어 있기 때문

루트 안에서 사용하고 싶은 디렉토리를 지정시켜서 연결해 주어야 함.

마운트 = 연결이라고 생각하면 된다.

마운트 = 1회용

하드디스크 추가 → 파티션 작업 (fdisk) → 파일시스템 생성(포맷)작업 (mkfs, mkfs -t)

→ 마운트 작업 (mount) → 파일시스템테이블 파일에 등록(/etc/fstab)

- 파일 시스템을 전체 디렉터리 구조의 특정 디렉터리에 연결하는 것
- 마운트되는 위치를 마운트 지점이라고 함

```
1 █
2 #
3 # /etc/fstab
4 # Created by anaconda on Tue Jan  4 10:29:24 2022
5 #
6 # Accessible filesystems, by reference, are maintained under '/dev/disk'
7 # See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
8 #
9 /dev/mapper/centos-root / xfs defaults 0 0
10 UUID=9b25ed75-548f-4dd8-bf9e-5f4a1094fd3f /boot xfs defaults
   0 0
11 /dev/mapper/centos-swap swap swap defaults 0 0
~
```

※ CentOS7 Minimal 버전으로 생성

- 이름 : Disk
- ip 110
- 나머지는 기본 옵션
- 초기 스냅샷 찍기 ☆

```
[root@localhost ~]# ls /dev/sd*  
/dev/sda /dev/sda1 /dev/sda2 /dev/sdb  
[root@localhost ~]# fdisk /dev/sdb  
Welcome to fdisk (util-linux 2.23.2).
```

Changes will remain in memory only, until you decide to write them.  
Be careful before using the write command.

Device does not contain a recognized partition table  
Building a new DOS disklabel with disk identifier 0x4b86810c.

Command (m for help): m

Command action

- a toggle a bootable flag
- b edit bsd disklabel
- c toggle the dos compatibility flag
- d delete a partition
- g create a new empty GPT partition table
- G create an IRIX (SGI) partition table
- l list known partition types
- m print this menu
- n add a new partition
- o create a new empty DOS partition table
- p print the partition table
- q quit without saving changes
- s create a new empty Sun disklabel
- t change a partition's system id
- u change display/entry units
- v verify the partition table
- w write table to disk and exit
- x extra functionality (experts only)

Command (m for help): n

Partition type:

- p primary (0 primary, 0 extended, 4 free)
- e extended

주 파티션과 확장 파티션

Windows와 Linux의 차이 : 안정성

커널이 실제 하드웨어를 움직인다.

커널 = 운영체제로 생각(좁은 의미로의 운영체제)하면 됨

셸 = 커널로 보내는 명령어를 해석하는 명령어 해석기와 같음

파일시스템 생성

```
mkfs -t ext4( or mkfs.ext4) /dev/sdb1
```

마운트 하기

```
[root@localhost ~]# mkdir /data
[root@localhost ~]# touch /data/test1.txt
[root@localhost ~]# ls /data
test1.txt
[root@localhost ~]# mount /dev/sdb1 /data
```

60기가 디스크 1개 추가해서 파티션 1번(20G)는 dtest1, 2번(나머지) 파티션은 dtest2에 마운트 해보세요. 재부팅해도 계속 마운트 되도록 설정하세요.

```
mkdir /dtest1
```

```
mkdir /dtest2
```

mount point 지정

## RAID

### ◆ 정의

- 여러 개의 디스크를 하나의 디스크처럼 사용함
- 비용 절감 + 신뢰성 향상 + 성능 향상의 효과를 냄

### ◆ 하드웨어 RAID

- 하드웨어 제조업체에서 여러 개의 디스크를 하나의 디스크처럼 사용하도록 공급
- 소프트웨어 RAID보다 안정적이고 제조업체의 기술지원을 받을 수 있음
- 가격이 비쌈

### ◆ 소프트웨어 RAID

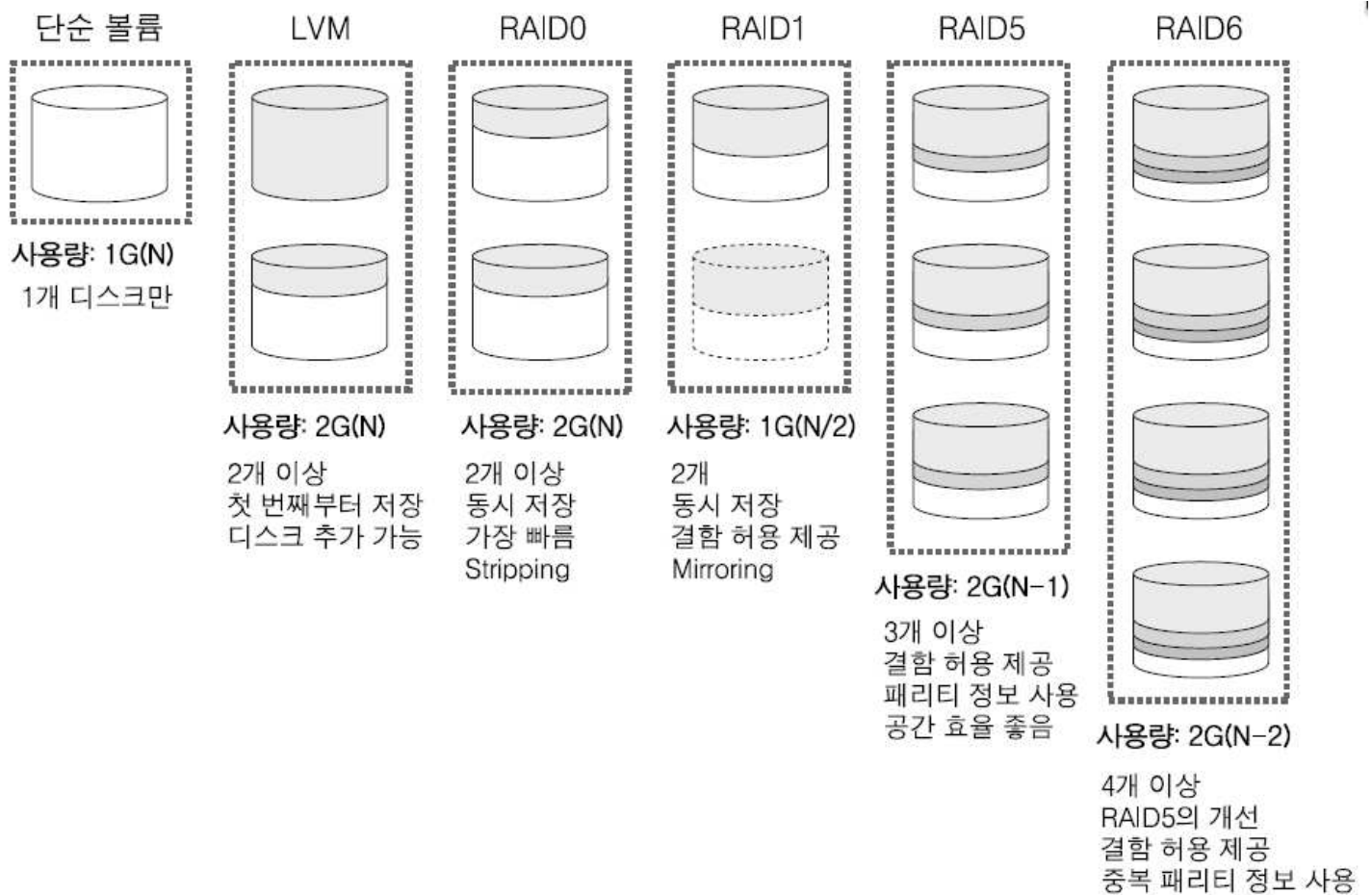
- 고가의 하드웨어 RAID의 대안
- 하드디스크만 있으면 운영체제에서 지원하는 방식으로 구성
- 하드웨어 RAID에 비해서 신뢰성이나 속도 등이 떨어질 수 있음
- 저렴한 비용으로 좀 더 안전한 데이터의 저장이 가능

### ◆ 두 방식은 물리적인 차이만 있을 뿐 동일한 개념

◆ RAID는 하드디스크가 묶이는 방식에 따라 레벨로 나뉘어지며 효율성과 안정성을 고려하여 레벨을 결정 (성능이나 중요도 순으로 숫자가 정해지는 것이 아님)



## < 하드웨어 RAID >



### LVM

#### ◆ 기능

- LVM이란 여러 개의 하드디스크를 합쳐서 한 개의 파일시스템으로 사용하는 것



◆ 특징

- 작은 용량의 하드디스크 여러 개를 큰 용량의 하드디스크 한 개 처럼 사용
  - 각각 파티션을 생성할 수 있다.
- LinearRAID의 경우 하나의 디스크로만 사용 가능.

◆ 용어

- Volume : 크기가 재조정될 수 있는 파티션
- Physical Volume (물리 볼륨) : 하나의 물리적 디스크, /dev/sda1, /dev/sdb1
- Volume Group (볼륨 그룹) : 여러 물리 볼륨을 하나로 묶은 것으로 파티션의 한 종류
- Logical Volume (논리 볼륨) : 볼륨 그룹에서 가용한 공간을 분할한 것 (파티션의 개념)

◆ 논리 볼륨 만들기

- 하드디스크에서 파티션을 만들
- 파티션을 물리 볼륨으로 만들
- 여러 개의 물리 볼륨을 볼륨 그룹으로 만들
- 볼륨 그룹안에서 논리 볼륨을 생성
- 파일 시스템 생성 후 마운트하여 사용

LVM 명령어

LVM 관련 명령	설명
<b>pvcreate</b>	파티션을 초기화하여 물리 볼륨으로 만들
<b>pvdisplay</b>	물리 볼륨의 속성을 출력
<b>vgcreate</b>	명령 pvcreate로 만든 블록 디바이스로부터 새로운 볼륨 그룹을 생성
<b>vgextend</b>	기존 볼륨 그룹의 크기를 늘리기 위해 물리 볼륨을 추가
<b>vgdisplay</b>	볼륨 그룹의 속성을 출력
<b>lvcreate</b>	볼륨 그룹에서 새로운 논리 볼륨을 생성
<b>lvdisplay</b>	논리 볼륨의 속성을 출력

lvextend 논리 볼륨의 크기를 늘린다.