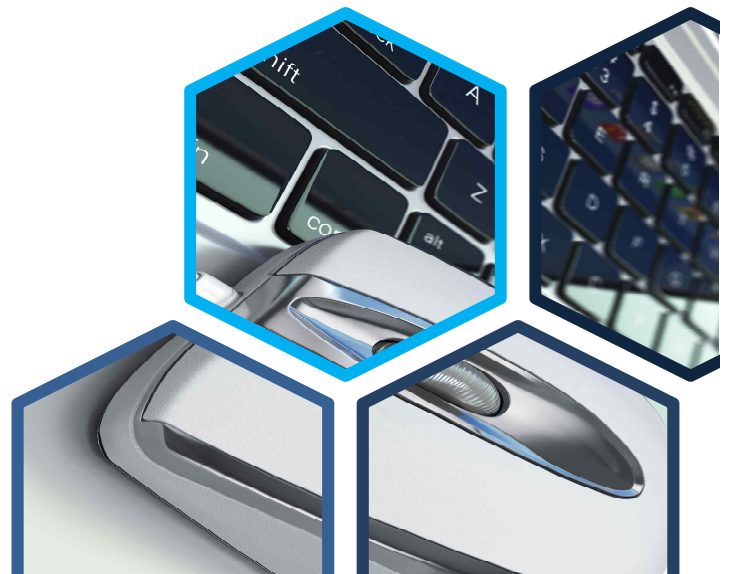


컴퓨터시스템

디스크 관리





학습목표

- 파일 시스템을 생성할 수 있다.
- 디스크 마운트와 LVM에 대해 설명할 수 있다.
- 디스크를 관리할 수 있다.



학습내용

- 파일 시스템 생성
- 디스크 마운트 및 LVM
- 디스크 관리

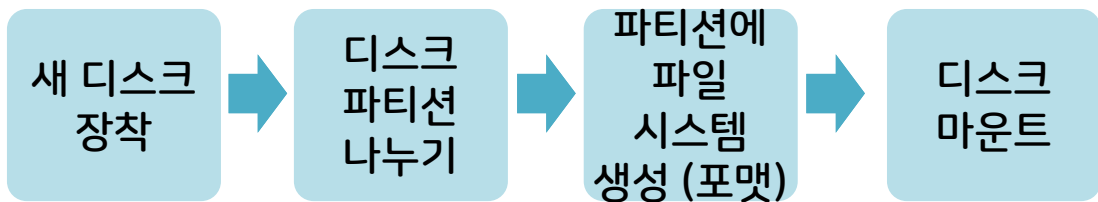


파일 시스템 생성

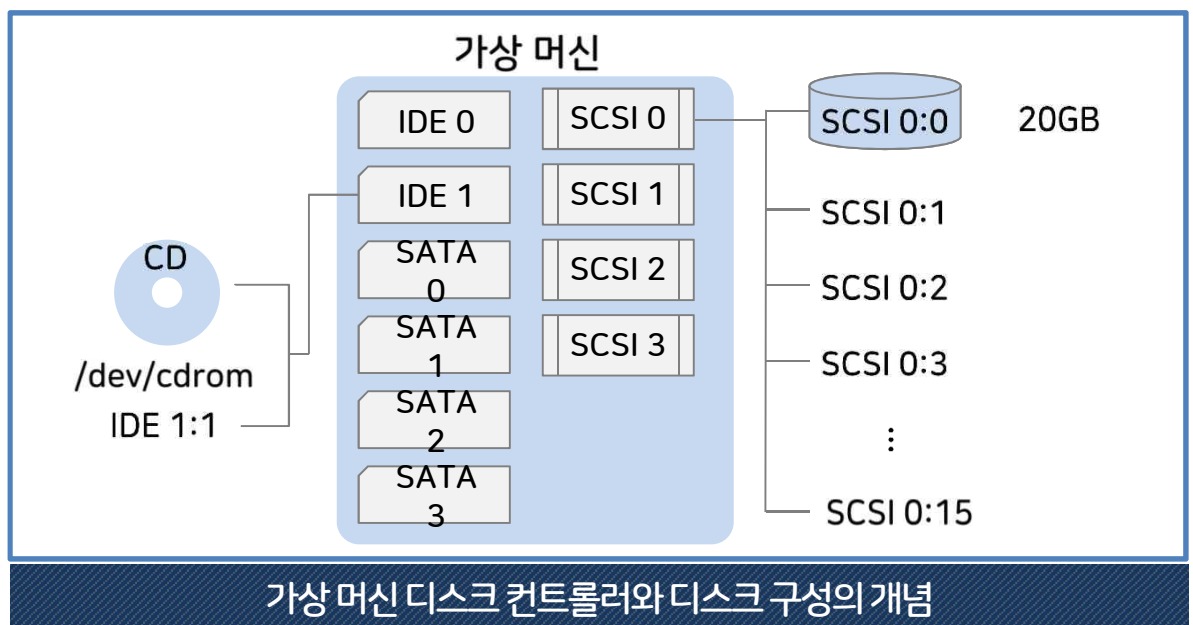
1 가상 머신 디스크 추가



디스크 추가 단계



가상 머신의 디스크 구성도



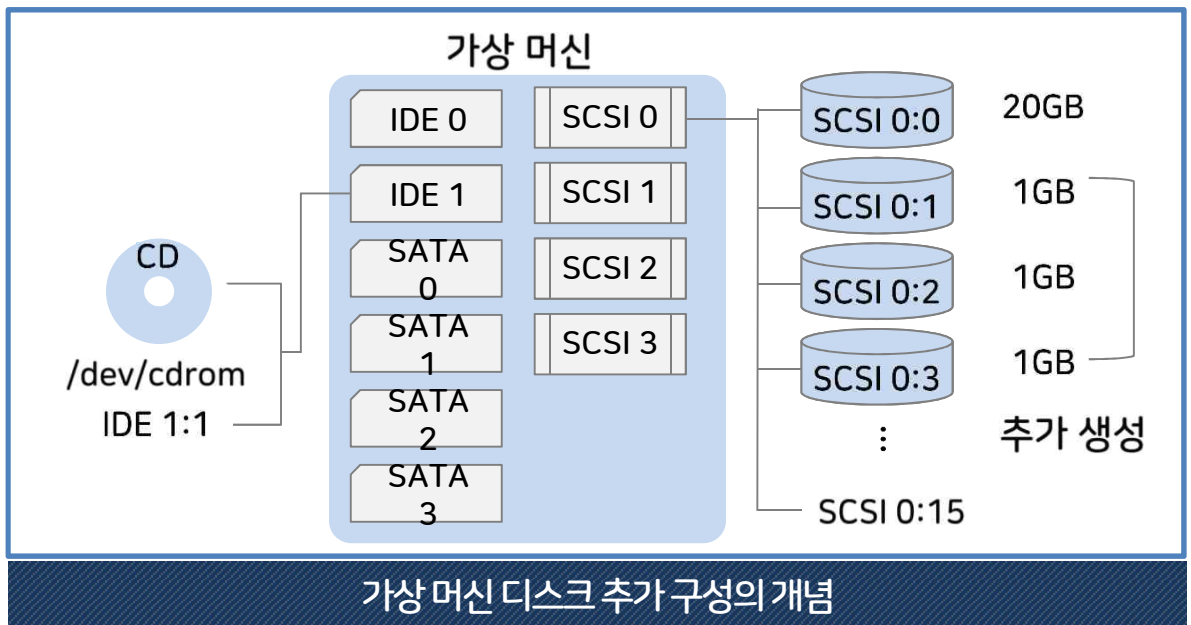


파일 시스템 생성

1 가상 머신 디스크 추가



가상 머신에 디스크 추가하기





파일 시스템 생성

2 디스크 파티션 나누기



디스크 파티션



파티션

- ☁ 하나의 디스크를 독립된 영역으로 구분하는 것
- ☁ 디스크 전체를 하나의 파티션으로 사용할 수도 있고 여러 개의 파티션으로 나누어 사용할 수도 있음



디스크 장치의 이름

- ☁ 리눅스에서 IDE 컨트롤러에 연결된 디스크는 `/dev/hd`로 시작
- ☁ SCSI나 SATA 컨트롤러에 장착된 디스크는 `/dev/sd`로 시작하는 이름을 주로 사용
- ☁ 최근에는 IDE, SCSI 등 구분 없이 모두 `/dev/sd`로 시작하는 이름을 사용
- ☁ 컨트롤러에 연결되는 디스크의 순서에 따라 다음과 같이 알파벳 추가:
`/dev/sda`, `/dev/sdb`



파일 시스템 생성

2 디스크 파티션 나누기



디스크 장치의 이름과 파티션 표시하기

- 하나의 디스크를 여러 개의 파티션으로 구분할 경우 파티션은 **디스크 장치 이름의 뒤에 숫자를 붙여서 표시**

/dev/sda	첫째 디스크 전체를 의미하는 장치 이름
/dev/sda0	디스크의 첫째 파티션
/dev/sda1	디스크의 둘째 파티션



파일 시스템 생성

3 fdisk 명령



fdisk

- 기능: 디스크의 파티션 **생성, 삭제, 보기** 등 파티션을 관리함
- 형식: fdisk [옵션] [장치명]
- 옵션

-b 크기	섹터 크기를 지정함(512, 1024, 2048, 4096)
-l	파티션 테이블을 출력함

- 사용 예

```
fdisk /dev/sdb      fdisk -l
```



fdisk 명령의 내부 명령

명령	기능	명령	기능
a	부팅 파티션을 설정함	p	파티션 테이블을 출력함
b	BSD 디스크 라벨을 편집함	q	작업 내용을 저장하지 않고 종료함
c	도스 호환성을 설정함	s	새로운 빈 Sun 디스크 라벨을 생성함
d	파티션을 삭제함	t	파티션의 시스템 ID를 변경함 (파일 시스템 종류 변경)
l	사용 가능한 파티션을 종류를 출력함	u	항목 정보를 변경·출력함
m	도움말을 출력함	v	파티션 테이블을 검사함
n	새로운 파티션을 추가함	w	파티션 정보를 디스크에 저장하고 종료함
o	새로운 빈 DOS 파티션을 생성함	x	실린더 개수 변경 등 전문가를 위한 부가적 기능임



파일 시스템 생성

4 파티션에 파일 시스템 생성



파일 시스템 생성하기

- 파일 시스템은 이 파티션에서 **파일과 디렉터리**를 관리하기 위한 구조를 만드는 것



파일 시스템 생성 명령

- mkfs, mke2fs**



mkfs

- 기능: **리눅스** 파일 시스템을 만듦
- 형식: **mkfs** [옵션] [장치명]
- 옵션

-t 종류

파일 시스템의 종류를 지정함(기본값은 ext2)

- 사용 예

```
mkfs /dev/sdb1  
mkfs -t ext4 /dev/sdb1
```




파일 시스템 생성

4 파티션에 파일 시스템 생성



mke2fs

- 기능: 리눅스 개정판 확장 파일 시스템(ext2, ext3, ext4)을 만듦
- 형식: mke2fs [옵션] [장치명]
- 옵션

-t 종류	파일 시스템의 종류를 지정함(기본값은 ext2)
-b 블록 크기	블록 크기를 바이트 수로 지정함
-c	배드 블록을 체크함
-f 프래그먼트 크기	프래그먼트 크기를 바이트 수로 지정함
-i inode당 바이트 수	inode당 바이트 수를 지정함(기본값은 4,096B)
-m 예약 블록 퍼센트	슈퍼 유저에게 예약해둘 블록의 퍼센트를 지정함(기본값은 5%)

사용 예

```
mkfs /dev/sdb1
mkfs -t ext4 /dev/sdb1
```



가상 머신 디스크 추가하기 실습 영상은
학습 콘텐츠에서 확인하실 수 있습니다.



디스크 마운트 및 LVM

1 디스크 마운트



디스크 마운트

- ☁ 파일 시스템을 **디렉터리 계층 구조에 마운트함**



마운트 포인트 준비하기

```
[root@localhost ~]# mkdir /mnt/hdd1  
[root@localhost ~]# mkdir /mnt/hdd2  
[root@localhost ~]# mkdir /mnt/hdd3
```



파일 시스템 마운트하기

- ☁ /dev/sdb1을 **/mnt/hdd1** 디렉터리에 마운트하기

```
[root@localhost ~]# mount /dev/sdb1 /mnt/hdd1  
[root@localhost ~]#
```

- ☁ /dev/sdb2 파티션을 **/mnt/hdd2** 디렉터리에 마운트하기

```
[root@localhost ~]# mount -t ext3 /dev/sdb2 /mnt/hdd2  
[root@localhost ~]#
```

- ☁ 마운트 결과

```
[root@localhost ~]# mount  
(생략)  
/dev/sdb1 on /mnt/hdd1 type ext2  
(rw,relatime,seclabel,block_validity,barrier,user_xattr,acl)  
/dev/sdb2 on /mnt/hdd2 type ext3  
(rw,relatime,seclabel,data=ordered)  
[root@localhost ~]#
```



디스크 마운트 및 LVM

1 디스크 마운트



파일 시스템 사용하기

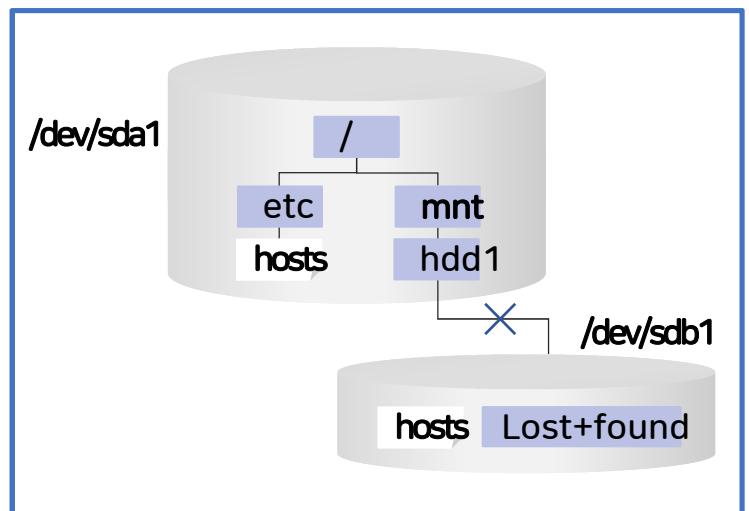
```
[root@localhost ~]# cp /etc/hosts /mnt/hdd1
[root@localhost ~]# ls /mnt/hdd1
hosts  lost+found
[root@localhost ~]#
```



마운트를 해제하는 경우

```
[root@localhost ~]# umount /mnt/hdd1
[root@localhost ~]# ls /mnt/hdd1
[root@localhost ~]#
```

- 마운트가 해제된 뒤
/mnt/hdd1 디렉터리에
아무 파일도 없음
- 파일 시스템의 마운트가
해제되면 이 파티션과의
연결이 끊어지므로
/mnt/hdd1에서 hosts
파일을 볼 수 없음



디스크 마운트 해제와 파일의 관계



디스크 마운트 및 LVM

2 LVM



여러 디스크를 하나처럼 사용하기

- ☁ **디스크의 용량이 부족할 때** 여러 개의 디스크를 하나의 디스크처럼 사용



LVM의 기본 개념

- ☁ LVM은 독립적으로 구성된 디스크 파티션을 하나로 연결하여 **한 파티션처럼 사용할 수 있도록 해줌**



LVM 관련 용어

PV(Physical Volume, 물리 볼륨)	/dev/sdb1, /dev/sdb2 같은 실제 하드 디스크의 파티션
VG(Volume Group, 볼륨 그룹)	여러 개의 PV를 그룹으로 묶은 것 (예: /dev/sdb1, /dev/sdb2가 GRP1이라는 그룹을 만들 때 GRP1을 VG라고 함)
LV(Logical Volume, 논리 볼륨)	VG를 다시 적절한 크기의 파티션으로 나눔
PE(Physical Extent)	PV가 가진 일정한 블록을 의미
LE(Logical Extent)	LV가 가진 일정한 블록을 의미

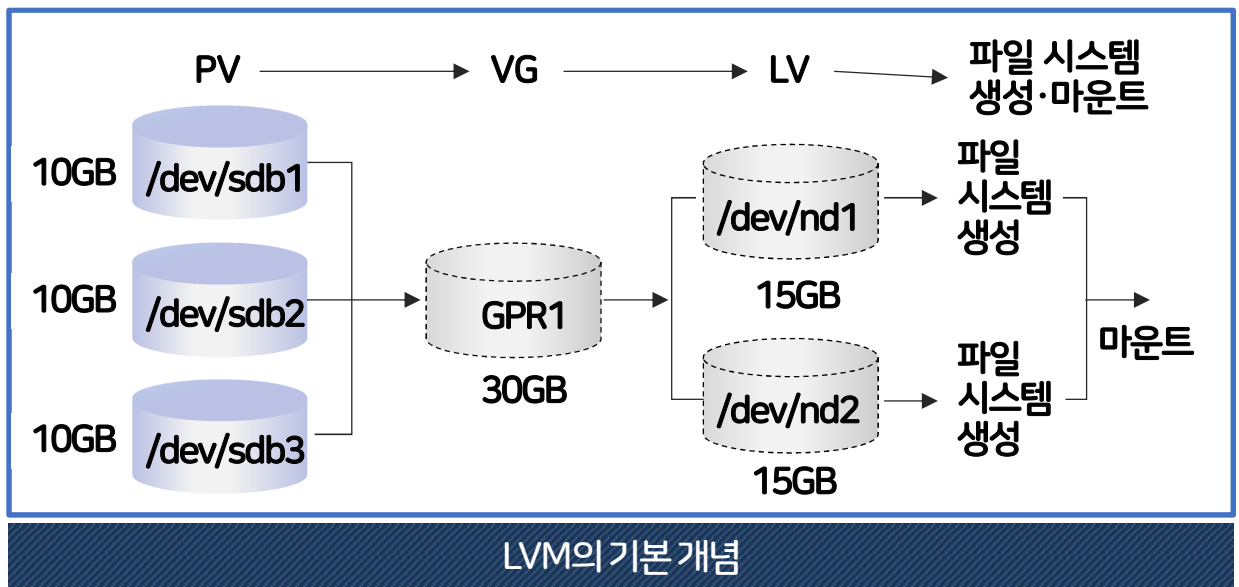


디스크 마운트 및 LVM

2 LVM



LVM의 기본 개념





디스크 마운트 및 LVM

2 LVM



LVM 관련 명칭

구분	기능	명령
PV	PV 생성	pvcreate 파티션 이름
	PV 상태 확인	pvscan
VG	VG 생성	vgcreate VG명 파티션(PV)명1 파티션(PV)명2...
	VG 활성화	vgchange -a y VG명
	VG 비활성화	vgchange -a n VG명
	VG 삭제	vgremove VG명
	VG 정보 확인	vgdisplay -v VG명
	VG에 PV 추가	vgextend VG명 PV명
구분	기능	명령
VG	VG에서 PV 삭제	vgreduce VG명 PV명
	VG명 변경	vgrename 기존 VG명 새 VG명
LV	LV 생성	lvcreate -l PE 수 VG명 -n LV명
	LV 삭제	lvremove LV명
	LV 상태 확인	lvscan
	LV 용량 확대	lvextend -l +PE 수 LV명
	LV 용량 축소	lvextend -l -PE 수 LV명

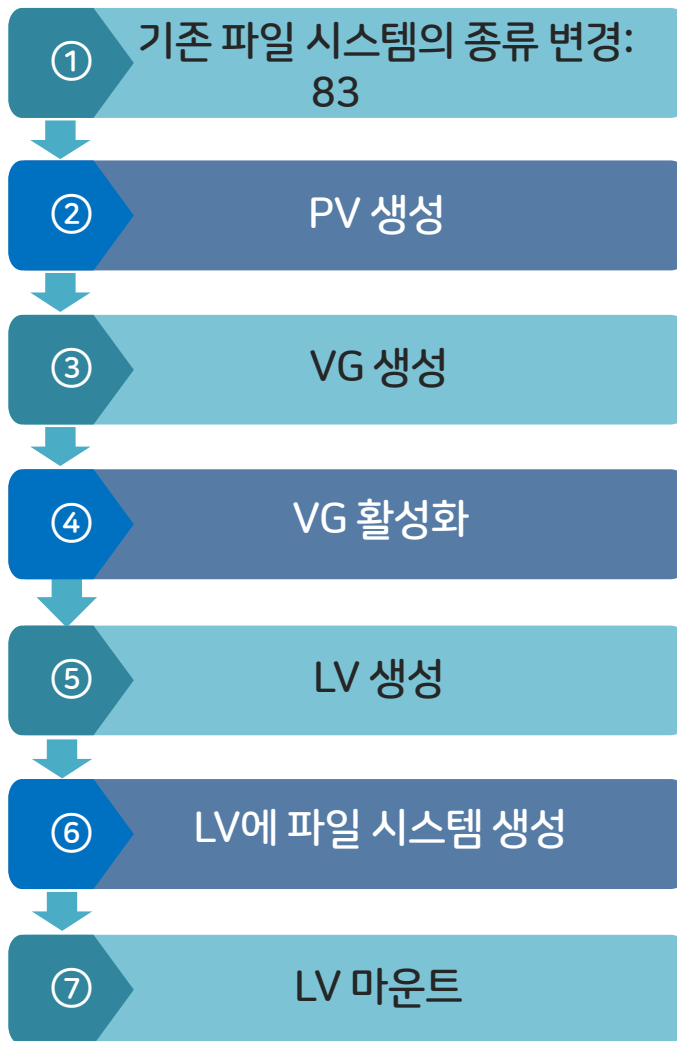


디스크 마운트 및 LVM

2 LVM



LVM 생성 과정





디스크 관리

1 디스크 용량 확인 및 출력



파일 시스템별 디스크 사용량 확인하기: df

- 기능: 디스크의 남은 공간에 대한 정보를 출력함
- 형식: df [옵션] [파일 시스템]
- 옵션

-a	모든 파일 시스템을 대상으로 디스크 사용량을 확인함
-k	디스크 사용량을 KB 단위로 출력함
-m	디스크 사용량을 MB 단위로 출력함
-h	디스크 사용량을 알기 쉬운 단위(GB, MB, KB 등)로 출력함
-t 파일 시스템 종류	지정한 파일 시스템 종류에 해당하는 디스크의 사용량을 출력함
-T	파일 시스템 종류도 출력함

사용 예

```
df
df -h
```



df 명령으로 출력되는 항목

- 파일 시스템 장치명
- 파일 시스템의 전체 용량
- 파일 시스템의 사용량
- 파일 시스템의 사용 가능한 남은 용량
- 사용량을 퍼센트로 표시
- 마운트 포인트



디스크 관리

1 디스크 용량 확인 및 출력



파일 시스템 사용량을 이해하기 쉬운 단위로 표시하기: **-h 옵션**

- 각 파일 **시스템별**로 이해하기 쉬운 단위로 사용량을 표시



파일 시스템의 종류 정보 출력하기: **-T 옵션**



디렉터리나 사용자별 디스크 사용량 확인하기: **du**

- 기능: 디스크의 **사용 공간에 대한 정보**를 출력함
- 형식: du [옵션] [디렉터리]
- 옵션

-s	특정 디렉터리 전체 사용량을 출력함
-h	디스크 사용량을 알기 쉬운 단위(GB, MB, KB 등)로 출력함

- 사용 예

```
du
du -s ~user1
```



디스크 관리

1 디스크 용량 확인 및 출력



du 명령만 사용하는 경우



현재 디렉터리의 **디스크 사용량**을 출력



전체 디스크 사용량 출력하기: **-s 옵션**



특정 사용자의 디스크 사용량 출력하기: **-sh 옵션**



디스크 관리

2 파일 시스템 검사하고 복구하기



파일 시스템 검사하고 복구하기

☁ 파일 시스템 손상의 이유

- ❶ 부적절한 시스템 종료
- ❷ 전원의 불안정
- ❸ 소프트웨어 오류
- ❹ 하드웨어 오작동 등

- ☁ 손상된 파일 시스템의 용량을 확인할 뿐만 아니라 **파일 시스템의 상태를 점검하고 문제가 있을 때 복구해야 함**



디스크 관리

2 파일 시스템 검사하고 복구하기



fsck 명령으로 파일 시스템 검사하기

- inode 및 블록, 디렉터리, 파일 링크 등을 검사하고 필요 시 복구 작업도 수행
- 기능: 리눅스의 파일 시스템을 점검함
- 형식: fsck [옵션] [장치명]
- 옵션

-f	강제로 점검함
-b 슈퍼 블록	슈퍼 블록으로 지정한 백업 슈퍼 블록을 사용함
-y	모든 질문에 yes로 대답하게 함
-a	파일 시스템 검사에서 문제를 발견했을 때 자동으로 복구함

- 사용 예

```
fsck /dev/sdb1
fsck -f /dev/sdb1
```

- 일반적인 파일 시스템 검사: **fsck**
- 파일 시스템 강제 검사: **fsck -f**
- 파일 시스템 종류를 지정해 검사

```
[root@localhost ~]# fsck.ext4 /dev/sdd1
```



디스크 관리

2 파일 시스템 검사하고 복구하기



e2fsck 명령으로 파일 시스템 검사하기

- 기능: 리눅스의 확장 파일 시스템(ext2, ext3, ext4)을 점검함
- 형식: e2fsck [옵션] [장치명]
- 옵션

-f	강제로 점검함
-b 슈퍼 블록	슈퍼 블록으로 지정한 백업 슈퍼 블록을 사용함
-y	모든 질문에 yes로 대답하게 함
-j ext3/ext4	ext3나 ext4 파일 시스템을 검사할 때 지정함

- 사용 예

```
e2fsck /dev/sdb1  
e2fsck -f /dev/sdb1
```



디스크 관리

2 파일 시스템 검사하고 복구하기



배드 블록 검사하기: **badblocks**

- 기능: 장치의 배드 블록을 검색함
- 형식: badblocks [옵션] [장치명]
- 옵션

-v	검색 결과를 자세하게 출력함
-o 출력 파일	검색한 배드 블록 목록을 지정한 출력 파일에 저장함

- 사용 예

```
badblocks -v /dev/sdb1  
badblocks -v -o bad.out /dev/sdb1
```




3 파일 시스템 복구



백업 슈퍼 블록을 이용해 파일 시스템 복구하기

- 파일 시스템의 기본 슈퍼 블록에 문제가 있으면 해당 파일 시스템을 사용할 수 없음
- 백업 슈퍼 블록 중 하나를 사용하여 파일 시스템을 복구
- 백업 슈퍼 블록의 위치 파악하기: **dumpe2fs**
 - ① **dumpe2fs** 명령으로 /dev/sdd1 파일 시스템의 정보 출력
 - ① **dumpe2fs /dev/sdd1**: 슈퍼 블록에 관한 정보만 추출

```
[root@localhost ~]# dumpe2fs /dev/sdd1 | grep superblock
```

- 파일 시스템 복구하기: **-b 옵션**
- 예시: 파일 시스템의 기본 슈퍼 블록 /dev/sdd1 bk일 시스템 일부를 **dd** 명령으로 삭제 후 마운트하면 오류 메시지 발생
 - ① **sudo dd if=/dev/zero of=/dev/sdd1 bs=4096 count=20**
 - ① **sudo mount /dev/sdd1 /mnt/hdd1**
 - ① **sudo e2fsck -b 8193 -y /dev/sdd1**



📌 핵심요약

1 파일 시스템 생성

- 📌 파티션이란 하나의 디스크를 독립된 영역으로 구분하는 것을 말함
- 📌 디스크 전체를 하나의 파티션으로 사용할 수도 있고 여러 개의 파티션으로 나누어 사용할 수도 있음
- 📌 파일 시스템은 이 파티션에서 파일과 디렉터리를 관리하기 위한 구조를 만드는 것임
- 📌 파일 시스템 생성 명령: mkfs, mke2fs

2 디스크 마운트 및 LVM

- 📌 디스크 마운트란 파일 시스템을 디렉터리 계층 구조에 마운트하는 것을 의미함
- 📌 LVM은 독립적으로 구성된 디스크 파티션을 하나로 연결하여 한 파티션처럼 사용할 수 있도록 해줌



📌 핵심요약

3 디스크 관리

- 📊 파일 시스템별 디스크 사용량 확인하기: `df`
- 📊 디렉터리나 사용자별 디스크 사용량 확인하기: `du`
- 📊 파일 시스템 검사하기: `fsck` 명령
- 📊 파일 시스템 복구하기: 백업 슈퍼 블록 이용
- 📊 `/dev/sdd1` 파일 시스템의 정보 출력: `dumpe2fs` 명령