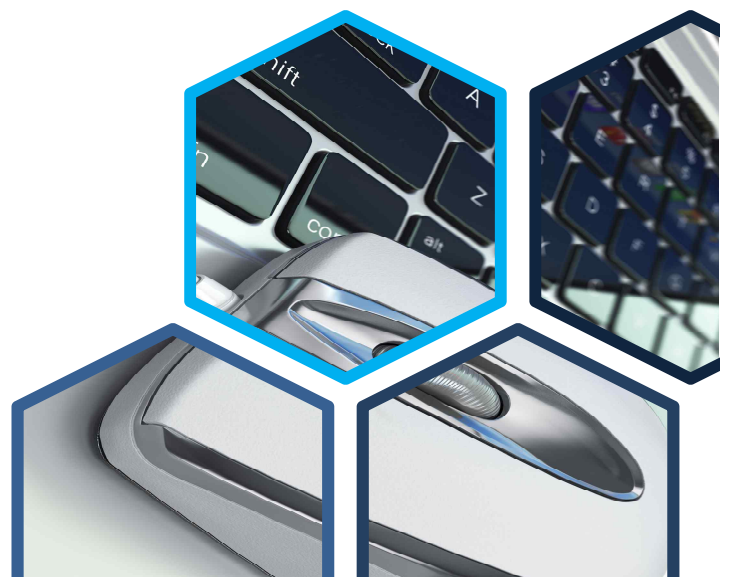




파일 시스템 마운트





학습목표

- 리눅스 파일 시스템의 종류를 설명할 수 있다.
- 리눅스 파일 시스템의 구조를 설명할 수 있다.
- 파일 시스템 마운트에 대해 설명할 수 있다.



학습내용

- 리눅스 파일 시스템의 종류
- 리눅스 파일 시스템의 구조
- 파일 시스템 마운트



리눅스 파일 시스템의 종류

1 디스크 기반 파일 시스템



파일 시스템

- ☁ 파일과 디렉터리의 집합을 **구조적으로** 관리하는 체계
- ☁ 어떤 구조를 구성하여 파일이나 디렉터리를 관리하는지에 따라 다양한 형식의 파일 시스템 존재



리눅스 파일 시스템의 종류

1 디스크 기반 파일 시스템



리눅스 고유의 디스크 기반 파일 시스템



ext(ext1)

- 'Extended File System'의 약자로 1992년 4월 리눅스 0.96c에 포함되어 발표
- 파일 시스템의 최대 크기는 2GB, 파일 이름의 길이는 255바이트까지 지원
- inode 수정과 데이터의 수정 시간 지원이 안 되고, 파일 시스템이 복잡해지고 파편화되는 문제
- 현재 리눅스에서는 ext 파일 시스템을 사용하지 않음



ext2

- ext 파일 시스템이 가지고 있던 문제를 해결하고자 1993년 1월 발표
- ext2는 ext3 파일 시스템이 도입되기 전까지 사실상 리눅스의 표준 파일 시스템으로 사용
- 이론적으로 32TB까지 가능(최대 볼륨크기 2TB~32TB)



리눅스 파일 시스템의 종류

1 디스크 기반 파일 시스템



리눅스 고유의 디스크 기반 파일 시스템



ext3

- ext2를 기반으로 개발되어 호환이 가능하며 2001년 11월 공개
- 가장 큰 장점은 **저널링(Journaling)** 기능을 도입 복구기능 강화
 - ① **저널**: 일련의 로그 기록
 - ① **저널링**: 데이터를 디스크에 기록하기 전에 저널에 수정사항을 기록하는 것으로 디스크에 기록되는 데이터의 복구 기능을 강화함
- 파일 시스템의 최대 크기는 블록의 크기에 따라 **2~32TB**까지 지원



ext4

- **1EB(엑사바이트, 1EB= 1,024X1,024TB)** 이상의 볼륨과 **16TB** 이상의 파일을 지원
- ext2 및 ext3와 호환성을 유지하며 2008년 12월 발표



리눅스 파일 시스템의 종류

2 기타 파일 시스템



리눅스에서 지원하는 기타 파일 시스템

msdos	MS-DOS 파티션을 사용하기 위한 파일 시스템
iso9660	CD-ROM, DVD의 표준 파일 시스템으로 읽기 전용으로 사용됨
nfs	Network file system으로 원격 서버의 디스크를 연결할 때 사용됨
ufs	Unix file system으로 유닉스의 표준 파일 시스템
vfat	윈도 95, 98, NT를 지원하기 위한 파일 시스템
hpfs	HPFS(High Performance File System)를 지원하기 위한 파일 시스템
ntfs	윈도 NTFS(New Technology File System)를 지원하기 위한 파일 시스템
sysv	유닉스 시스템V를 지원하기 위한 파일 시스템
hfs	맥 컴퓨터의 hfs 파일 시스템을 지원하기 위한 파일 시스템



리눅스 파일 시스템의 종류

2 기타 파일 시스템



리눅스의 가상 파일 시스템

swap	▶ 스왑 영역을 관리하기 위한 스왑 파일 시스템
tmpfs	▶ temporary file system으로 메모리에 임시 파일을 저장하기 위한 파일 시스템이며, 시스템이 재시작할 때마다 기존 내용이 없어짐 (예: /tmp 디렉터리 - 페도라 18부터)
proc	▶ proc file system으로 /proc 디렉터리 ▶ 커널의 현재 상태를 나타내는 파일을 가지고 있음
ramfs	▶ 램디스크를 지원하는 파일 시스템
rootfs	▶ root file system으로 / 디렉터리 ▶ 시스템 초기화 및 관리에 필요한 내용을 관리함



확인 방법

- ☁ /proc/filesystems는 현재 커널이 지원하는 파일 시스템의 종류를 알려줌



리눅스 파일 시스템의 구조

1 ext4 파일 시스템의 구조



리눅스 고유의 디스크 기반 파일 시스템



리눅스의 모든 파일 시스템의 기본 개념

- 파일은 **inode**로 관리됨
- 디렉터리는 단순히 파일의 목록을 가지고 있는 파일임
- 특수 파일을 통해 장치에 접근 가능



특징

- 효율적으로 디스크를 사용하기 위해 저장 장치를 논리적인 블록의 집합 (**블록 그룹**)으로 구분
- 일반적으로 **블록은 4KB**이고 실제 크기는 시스템의 설정에 따라 변경 가능



블록 그룹 유형

블록 그룹 0	블록 그룹 a	블록 그룹 b
<ul style="list-style-type: none"> 파일 시스템의 첫 번째 블록 그룹 특별하게 그룹 0 패딩과 슈퍼 블록, 그룹 디스크립터를 가지고 있음 	<ul style="list-style-type: none"> 파일 시스템에서 첫 번째 블록 그룹이 아닌 블록 그룹 그룹 0 패딩이 없으나 슈퍼 블록과 그룹 디스크립터에 대한 복사본을 가지고 있음 	<ul style="list-style-type: none"> 파일 시스템에서 첫 번째 블록 그룹이 아닌 블록 그룹 그룹 0 패딩, 슈퍼 블록, 그룹 디스크립터가 없고 바로 데이터 블록 비트맵으로 시작함



리눅스 파일 시스템의 구조

1 ext4 파일 시스템의 구조



블록 그룹별 구조

블록 그룹 0

그룹 0 패딩	ext4 슈퍼블록	그룹 디스크립터 블록	GDT 예약 블록	데이터 블록 비트맵	inode 비트맵	inode 테이블	데이터 블록
1,024B	1블록	n블록	n블록	1블록	1블록	n블록	n블록

블록 그룹 a 중복 저장

ext4 슈퍼블록	그룹 디스크립터 블록	GDT 예약 블록	데이터 블록 비트맵	inode 비트맵	inode 테이블	데이터 블록
1블록	n블록	n블록	1블록	1블록	n블록	n블록

블록 그룹 b

데이터 블록 비트맵	inode 비트맵	inode 테이블	데이터 블록
1블록	1블록	n블록	n블록



그룹 0 패딩

- 블록 그룹 0의 첫 1,024 바이트는 특별한 용도로 사용됨
- x86 부트 섹터와 부가 정보를 저장



리눅스 파일 시스템의 구조

1 ext4 파일 시스템의 구조



슈퍼 블록

☁ 파일 시스템과 관련된 다양한 정보가 저장

- ① 전체 inode개수와 블록의 개수
- ① 할당되지 않은 블록(Free block)와 inode(Free inode)의 개수
- ① 첫 번째 데이터 블록의 주소
- ① 블록의 크기
- ① 그룹당 블록의 개수
- ① 마운트 시간
- ① 파일 시스템의 상태
- ① 그룹 디스크립터의 크기

슈퍼 블록에 문제가 생길 경우

전체 파일 시스템을 사용할 수 없게 됨

슈퍼 블록을 다른 블록 그룹에 복사,
블록 그룹 0의 슈퍼 블록을 읽을 수
없을 경우

복사본을 사용하여 복구 가능



리눅스 파일 시스템의 구조

1 ext4 파일 시스템의 구조



그룹 디스크립터와 GDT 예약 블록

- 그룹 디스크립터도 **블록 그룹 0에 있는 것으로** 슈퍼 블록의 다음에 위치
- 그룹 디스크립터에 저장되는 정보
 - 블록 비트맵의 주소, inode 비트맵의 주소, inode 테이블의 주소
 - 할당되지 않은 블록의 개수, 할당되지 않은 inode의 개수, 디렉터리의 개수
 - 블록 비트맵, inode 비트맵 체크섬



데이터 블록 비트맵과 inode 비트맵

데이터 블록 비트맵	inode 비트맵
블록 그룹에 포함된 데이터 블록의 사용 여부를 확인하는 데 사용	inode 테이블의 항목(inode) 사용 여부 표시



inode 테이블과 데이터 블록

inode 테이블	데이터 블록
파일 정보 저장	실제 데이터 저장

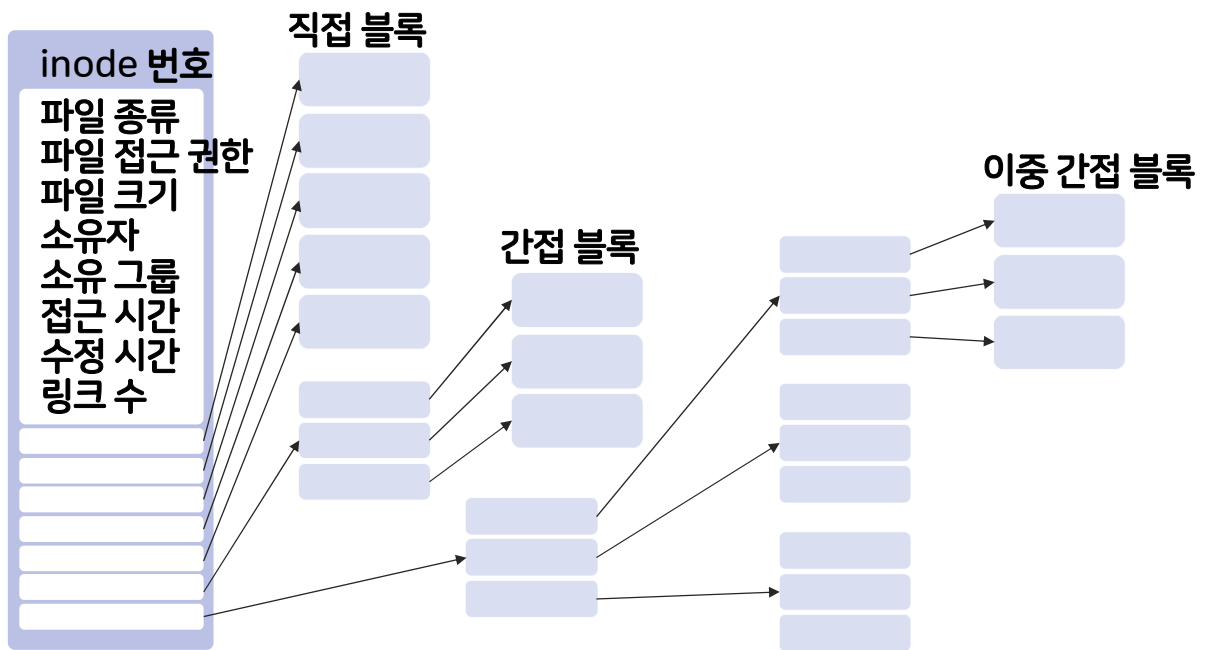


리눅스 파일 시스템의 구조

2 inode의 구조



inode의 구조





리눅스 파일 시스템의 구조

3 파일 시스템과 디렉터리 계층 구조



특징

- 디렉터리 계층 구조는 리눅스의 전체 파일과 디렉터리를 어떤 구조로 정리하고 관리할 것인지를 정의한 것
- 실제 파일이 저장되어 있는 파일 시스템은 디렉터리 계층 구조에 연결되어야 사용자가 접근 가능



구성 방법

한 파일 시스템으로 구성

- 디렉터리 계층 구조에서 보이는 모든 디렉터리와 파일을 하나의 파일 시스템으로 구성

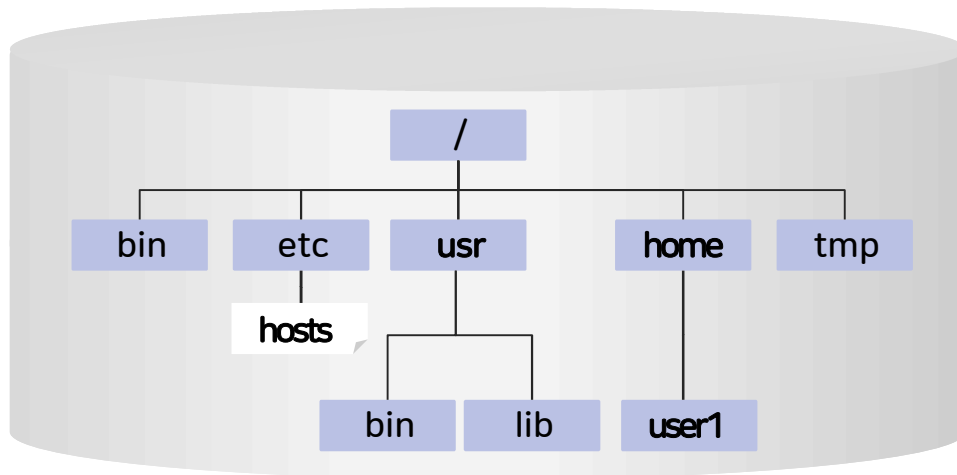
여러 파일 시스템으로 구성

- 디렉터리 계층 구조를 여러 파일 시스템으로 구분하여 구성
- 일부 파일 시스템에 문제가 생기더라도 다른 파일 시스템의 파일은 안전함

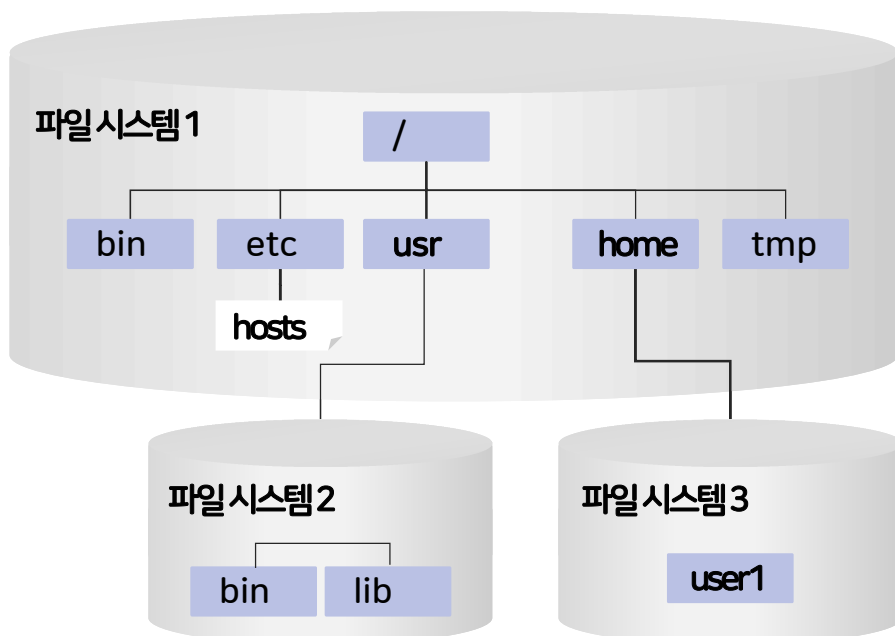


리눅스 파일 시스템의 구조

3 파일 시스템과 디렉터리 계층 구조



한 파일 시스템으로 구성하기



여러 파일 시스템으로 구성하기



파일 시스템 마운트

1 /etc/fstab 파일



마운트

- 파일 시스템을 디렉터리 계층 구조의 **특정 디렉터리와 연결하는 것**



마운트 포인트

- 디렉터리 계층 구조에서 **파일 시스템이 연결되는 디렉터리**



파일 시스템 마운트 설정 파일

- 리눅스에서 시스템이 부팅될 때 자동으로 파일 시스템이 마운트가 되게 하려면 /etc/fstab 파일에 설정
- /etc/fstab 파일의 기능
 - **파일 시스템의 마운트 설정 정보 저장**



파일 시스템 마운트

1 /etc/fstab 파일



/etc/fstab 파일의 구조

장치명

마운트
포인트

파일
시스템의
종류

옵션

덤프 관련
설정

파일 점검
옵션

🔗 /etc/fstab 파일의 예

```
user1@myubuntu:~$ cat /etc/fstab
```

```
# <file system> <mount point>    <type> <options>      <dump> <pass>
# / was on /dev/sda1 during installation
UUID=7009cb18-dbd5-4ffc-af86-599cee765454 /          ext4  errors=remount-ro
0          1
/swapfile                                none          swap      sw
0          0
```

장치명 : UUID=7009cb18-dbd5-4ffc-af86-599cee765454

마운트 포인트 : /

파일 시스템의 종류 : ext4

옵션 : errors=remount-ro

덤프 관련 설정 : 1

파일 점검 옵션 : 1

- 🔗 UUID는 'Universally Unique Identifier'의 약자로 로컬 시스템과 다른 시스템에서 **파일 시스템을 유일하게 구분해주는 128비트의 숫자**
- 🕒 UUID는 시스템의 하드웨어 정보와 시간 정보를 조합하여 **랜덤으로 생성**
- 🕒 UUID로 지정된 장치는 **/dev/disk/by-uuid** 디렉터리에서 찾을 수 있음



파일 시스템 마운트

1 /etc/fstab 파일



/etc/fstab 파일의 구조

장치명	파일 시스템 장치명 예) /dev/hda1, /dev/sda1과 같이 특정 디스크를 지정
마운트 포인트	파일 시스템이 마운트될 마운트 포인트를 설정
파일 시스템의 종류	파일 시스템의 종류를 설정, ext2, ext3, ext4 등
옵션	파일 시스템의 속성을 지정
덤프 관련 설정	0(dump 불가), 1(dump 가능)
파일 점검 옵션	0(부팅 시 fsck 명령 실행 안함), 1(루트 파일 시스템으로 점검), 2(루트 파일 시스템 이외)

- **덤프**: 컴퓨터 프로그램이 특정 시점에 작업 중이던 메모리 상태를 기록한 것으로, 보통 프로그램이 비정상적으로 종료했을 때 만들어짐




파일 시스템 마운트

1 /etc/fstab 파일



파일 시스템 속성 설정 옵션

defaults	일반적인 파일 시스템에 지정하는 속성으로 rw, nouser, auto, exec, suid 속성을 모두 포함
auto	부팅 시 자동으로 마운트
exec	실행 파일이 실행되는 것을 허용
suid	setuid, setgid 의 사용을 허용
ro	읽기 전용 파일 시스템
rw	읽기, 쓰기가 가능한 파일 시스템
user	일반 사용자도 마운트 가능
nouser	일반 사용자의 마운트가 불가능하며, root만 마운트 가능
noauto	부팅 시 자동으로 마운트하지 않음
noexec	실행 파일이 실행되는 것을 허용하지 않음
nosuid	setuid, setgid 의 사용을 금지
usrquota	사용자별로 디스크 쿼터 설정 가능
grpquota	그룹별로 디스크 쿼터 설정 가능

 **쿼터**: 디스크의 크기 및 파일의 용량을 사용자 또는 그룹 별로 제한



파일 시스템 마운트

2 마운트 관련 명령



마운트: mount

- 기능: 파일 시스템을 마운트함
- 형식: mount [옵션] [장치명 또는 마운트 포인트]
- 옵션

-t 파일 시스템 종류	파일 시스템 종류를 지정함
-o 마운트 옵션	마운트 옵션을 지정함
-f	마운트할 수 있는지 점검만 함
-r	읽기만 가능하게 마운트함(-o ro와 동일)

사용 예

```
mount
mount /dev/sdb1 /
mount -t iso9660 /dev/cdrom /mnt/cdrom
```



파일 시스템 마운트

2 마운트 관련 명령



언마운트: umount

- 기능: 파일 시스템을 언마운트함
- 형식: umount [옵션] [장치명 또는 마운트 포인트]
- 옵션

-t 파일 시스템 종류 파일 시스템 종류를 지정함

- 사용 예

```
umount /dev/sdb1
umount /mnt
```



마운트 명령만 사용하는 경우

- 옵션이나 인자를 지정하지 않고 마운트 명령을 사용하면
현재 마운트 되어 있는 정보를 출력
- 마운트 명령으로 출력되는 정보는 **/etc/mtab** 파일의 내용과 동일
 - 장치명
 - 마운트 포인트
 - 파일 시스템의 종류
 - 마운트 옵션
 - 사용하지 않는 항목 두 개(0 0)



파일 시스템 마운트

2 마운트 관련 명령



마운트 명령으로 장치를 연결하는 방법

- 하드디스크를 디렉터리 계층 구조에 연결하는 법

```
mount /dev/sdb1 /mnt
```

- 다양한 장치 마운트의 예

장치	마운트 명령 형식의 예
ext2 파일 시스템	mount -t ext2 /dev/sdb1 /mnt
ext3 파일 시스템	mount -t ext3 /dev/sdb1 /mnt
ext4 파일 시스템	mount -t ext4 /dev/sdb1 /mnt mount /dev/sdb1 /mnt
CD-ROM	mount -t iso9660 /dev/cdrom /mnt/cdrom
윈도 디스크	mount -t vfat /dev/hdc /mnt
USB 메모리	mount /dev/sdc1 /mnt → 리눅스용 USB 메모리의 경우 mount -t vfat /dev/sdc1 /mnt → 윈도우용 USB 메모리의 경우
읽기 전용 마운트	mount -r /dev/sdb1 /mnt
읽기/쓰기 마운트	mount -w /dev/sdb1 /mnt
원격 디스크 마운트	mount -t nfs 서버주소:/NFS 서버 측 디렉터리 /mnt



파일 시스템 마운트

3 USB 메모리 연결 및 해제



리눅스용 USB 메모리 연결하기

1 USB 메모리를 USB 슬롯에 꽂고 리눅스 시스템에 인식시킴

①

VMware Player의 Player 메뉴에서 'Removable KM

Poinchips
를 선택

②

USB 장치를 호스트 OS에서 분리하여 가상 머신에 연결한다는 메시지 출력

③

기존에 사용하던 USB 메모리이면 자동으로 디렉터리에 마운트

④

마운트 명령을 실행해보면 마지막에 장치가 추가되었는지 확인 가능함

⑤

마운트를 해제하고 파일 시스템 생성 작업을 해야 함



파일 시스템 마운트

4 CD-ROM 연결 및 해제



CD-ROM 연결하기

- 1 CD-ROM 장치를 USB로 연결하여 리눅스 시스템에 인식시킴
- 2 CD-ROM를 마운트함

```
[root@localhost /]# mount -t iso9660 /dev/cdrom /mnt
mount: /dev/sr0 is write-protected, mounting read-only
[root@localhost /]#
```

```
[root@localhost /]# ls -l /dev/cdrom
lrwxrwxrwx. 1 root root 3 5월 12 16:51 /dev/cdrom ->
[root@localhost /]#
```



파일 시스템 마운트

4 CD-ROM 연결 및 해제



CD-ROM 해제하기

- 1 CD-ROM의 내용을 확인하고 사용할 수 있음

```
[root@localhost /]# ls /mnt  
EFI LivesOS images isolinux  
[root@localhost /]#
```

- 2 CD-ROM 장치를 사용하고 나면 언마운트 명령을 사용하여 마운트를 해제함

```
[root@localhost /]# ls -l /dev/cdrom  
lrwxrwxrwx. 1 root root 3 5월 12 16:51 /dev/cdrom →  
[root@localhost /]#
```



파일 시스템 마운트 실습 영상은
학습 콘텐츠에서 확인하실 수 있습니다.



📌 핵심요약

1 리눅스 파일 시스템의 종류

- 📁 파일 시스템은 파일과 디렉터리의 집합을 구조적으로 관리하는 체계임
- 📁 어떤 구조를 구성하여 파일이나 디렉터리를 관리하느냐에 따라 다양한 형식의 파일 시스템이 존재함

2 리눅스 파일 시스템의 구조

- 📁 ext4 파일 시스템의 구조
 - 효율적으로 디스크를 사용하기 위해 저장 장치를 논리적인 블록의 집합(블록 그룹)으로 구분
 - 일반적으로 블록은 4KB이고 실제 크기는 시스템의 설정에 따라 변경 가능
- 📁 inode 구조
 - 디렉터리 계층 구조는 리눅스의 전체 파일과 디렉터리를 어떤 구조로 정리하고 관리할 것인지를 정의함
 - 실제 파일이 저장되어 있는 파일 시스템은 디렉터리 계층 구조에 연결되어야 사용자가 접근 가능



핵심요약

3 파일 시스템 마운트

- 📚 마운트: 파일 시스템을 디렉터리 계층 구조의 특정 디렉터리와 연결하는 것
- 📚 리눅스에서 시스템이 부팅될 때 자동으로 파일 시스템이 마운트되게 하려면 `/etc/fstab` 파일에 설정함
- 📚 `/etc/fstab` 파일의 기능: 파일 시스템의 마운트 설정 정보 저장