

# 16-1. File System

## Chapter 16 File System & Disk Management

### 리눅스 파일 시스템 종류

#### 리눅스 고유의 디스크 파일 시스템

- 파일 시스템은 다양한 정보를 가지고 있는 파일과 디렉터리의 집합을 구조적으로 관리하는 체계이다.
  - 즉, 파일과 디렉터리를 관리하려면 파일시스템이 필요하다.
- **ext4(Fourth Extended File System)**
  - 1EB(엑사바이트, 1EB=1,024x1,024TB) 이상의 볼륨과 16TB의 파일을 지원하며, ext2 및 ext3와 호환성을 유지하고 있다.
  - 온라인 조각 모음 기능도 지원한다.

#### 리눅스에서 지원하는 다른 디스크 기반 파일 시스템

- 표준 파일 시스템이라고 할 수 있는 ext2, ext3, ext4 외에 유닉스나 윈도우 시스템 등 다른 시스템과 호환되도록, 또한 CD-ROM, DVD 같은 외부 저장 장치를 사용할 수 있도록 다양한 파일 시스템을 지원한다.

파일 시스템	기능
msdos	MS-DOS 파티션을 사용하기 위한 파일 시스템
iso9660	CD-ROM, DVD의 표준 파일 시스템으로 읽기 전용으로 사용
nfs	Network File System으로 원격 서버의 디스크를 연결할 때 사용
ufs	Unix File System으로 유닉스의 표준 파일 시스템
vfat	Windows 95, 98, NT를 지원하기 위한 파일 시스템
hpfs	HPFS를 지원하기 위한 파일 시스템
ntfs	Windows의 NTFS를 지원하기 위한 파일 시스템
sysv	유닉스 시스템 V를 지원하기 위한 파일 시스템
hfs	맥의 hfs 파일 시스템을 지원하기 위한 파일 시스템

#### 특수 용도의 가상 파일 시스템

- 리눅스에는 디스크가 아니라 메모리에서 생성되어 사용되는 가상 파일 시스템이 있다.

파일 시스템	기능
--------	----

swap	<ul style="list-style-type: none"> <li>스왑 영역을 관리하기 위한 스왑 파일 시스템</li> </ul>
tmpfs	<ul style="list-style-type: none"> <li>Temporary File System으로 메모리에 임시 파일을 저장하기 위한 파일 시스템이며, 시스템이 재시작할 때마다 기존 내용이 없어진다.</li> <li>예를 들면 /run 디렉터리이다.</li> </ul>
proc	<ul style="list-style-type: none"> <li>proc 파일 시스템으로 /proc 디렉터리이다.</li> <li>커널의 현재 상태를 나타내는 파일을 가지고 있다.</li> </ul>
ramfs	<ul style="list-style-type: none"> <li>램 디스크를 지원하는 파일 시스템</li> </ul>
rootfs	<ul style="list-style-type: none"> <li>Root File System으로 /디렉터리이다.</li> <li>시스템 초기화와 관리에 필요한 내용을 관리한다.</li> </ul>

### 현재 시스템이 지원하는 파일 시스템 확인하기

- /proc/filesystems는 현재 커널이 지원하는 파일 시스템의 종류를 알려준다.
- nodev: 해당 파일 시스템이 블록장치(예: 디스크)에 연결되지 않다는 것으로 가상 파일 시스템임을 의미
- fuseblk: ntfs 파일 시스템을 연결할 때 사용하는 파일 시스템

```

lima@itserver: ~
lima@itserver:~$ more /proc/filesystems
nodev    sysfs
nodev    rootfs
nodev    ramfs
nodev    bdev
nodev    proc
nodev    cpuset
nodev    cgroup
nodev    cgroup2
nodev    tmpfs
nodev    devtmpfs
nodev    configfs
nodev    debugfs
nodev    tracefs
nodev    securityfs
nodev    sockfs
nodev    dax
nodev    bpf
nodev    pipefs
nodev    hugetlbfs
nodev    devpts
nodev    ext3
nodev    ext2
nodev    ext4
nodev    squashfs
nodev    vfat
nodev    ecryptfs
nodev    fuseblk
nodev    fuse
nodev    fusectl
nodev    pstore
nodev    mqueue
nodev    autofs

```

## 리눅스 파일 시스템의 구조

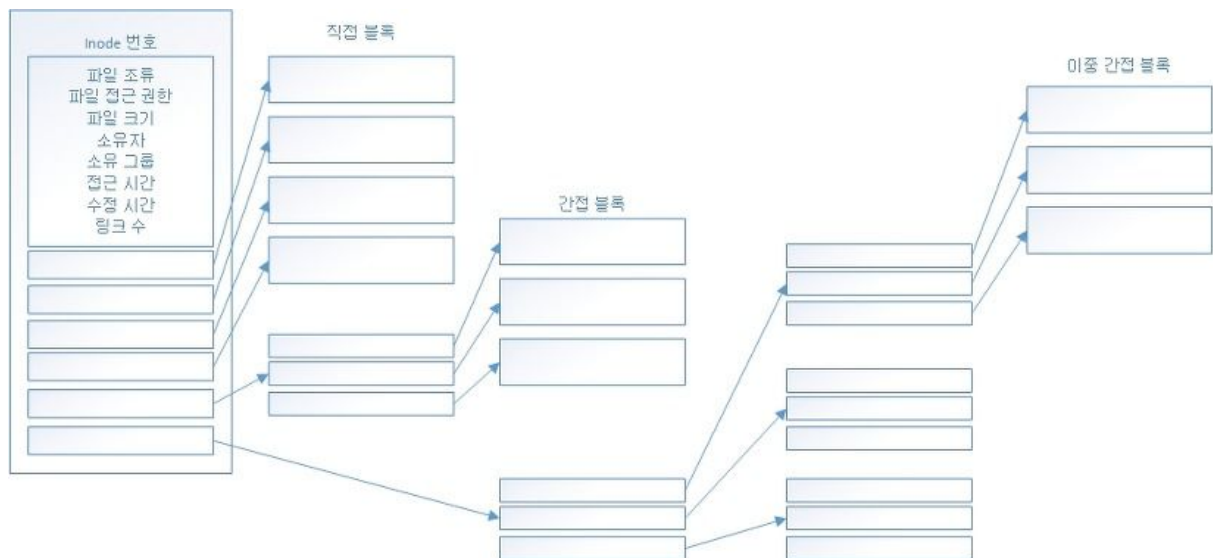
- 리눅스의 모든 파일 시스템은 기본적으로 유닉스 운영체제에서 유래된 공통의 3가지 개념을 바탕으로 구현되었다.

- 3가지 개념
  - 파일은 inode로 관리한다.
  - 디렉터리는 단순히 파일의 목록을 가지고 있는 파일일 뿐이다.
  - 특수 파일을 통해 장치에 접근할 수 있다.

#### ext4 파일 시스템의 구조

- 효율적으로 디스크를 사용하기 위해 저장 장치를 논리적 블록의 집합(블록 그룹)으로 구분한다.
- 블록 그룹은 3가지 유형이 있다.
  - **블록 그룹 0**
    - 파일 시스템의 첫번째 블록 그룹
    - 그룹 0 패딩과 수퍼블록, 그룹 디스크립터를 가지고 있다.
  - **블록 그룹 a**
    - 첫번째 블록 그룹이 아닌 블록 그룹
    - 그룹 0 패딩이 없지만 수퍼블록과 그룹 디스크립터에 대한 복사본을 가지고 있다.
  - **블록 그룹 b**
    - 첫번째 블록 그룹이 아닌 블록 그룹
    - 그룹 0 패딩, 수퍼 블록, 그룹 디스크립터가 없고 바로 데이터 블록 비트맵으로 시작한다.

#### inode 구조



- inode는 크게 2 부분, 즉 파일 정보를 저장하는 부분과 파일 내용이 실제로 저장되어 있는 데이터 블록의 주소를 저장하는 부분으로 나뉘어진다.
  - **파일 정보**
    - 파일 종류, 파일 접근 권한, 파일 크기, 소유자, 접근 및 수정 시간 등
    - 사용자가 ls -l 명령으로 확인하는 정보
  - **데이터 블록 주소**
    - 직접 블록: 데이터 블록에 대한 주소를 직접 저장
    - 간접 블록: 데이터 블록에 대한 주소를 가지고 있는 블록에 대한 주소를 저장
    - 이중 간접 블록: //