

---

# 파일시스템

---

## 파일 시스템이란

---

- 컴퓨터의 파일 시스템은 파일과 그 안에 든 자료를 저장하고 찾기 쉽도록 유지 관리하는 방법이다.
- 파일 시스템은 하드 디스크나 CD-ROM 과 같은 물리적 저장공간을 저장장치로 활용할 수도 있지만, NFS 와 같은 파일 시스템에서는 네트워크 상에 존재하는 파일에 접근하기 위한 가상적 인터페이스인 경우도 있다.
- 학술적으로, 파일 시스템은 자료를 계층적으로 저장, 탐색, 접근, 조작하기 위한 추상적 자료구조의 집합으로 정의된다.

## 파일 시스템의 종류

---

### minix

---

- 리눅스에서 처음 사용한 파일 시스템이다.
- 파일 이름은 최대 30 자, 파일 시스템당 최대 64MB 를 지원한다.

### ext2

---

- ext3 가 나오기 전까지 거의 대부분의 리눅스가 채택한 파일 시스템이다.
- 256 바이트 길이의 파일 이름이 가능하다.
- 2GB 의 단일 파일, 5TB 의 디스크를 사용할 수 있다.
- 예상치 못한 정전이나 기타 사유로 시스템이 비정상적으로 종료가 되었을 경우 부팅시 ext2 파일 시스템 검사 프로그램인 e2fsck 를 실행하여 파일 시스템의 일관성 검사를 하게 된다.
- 이러한 작업은 데이터가 많은 파일 시스템의 경우 검사하는 시간이 오래 걸리고, 그 동안 아무 작업도 할 수 없게 된다.

### ext3

---

- 커널 2.4 부터 지원하고 있는 파일 시스템
- ext2 의 파일 시스템에 저널링 기능을 추가한 파일 시스템이다.
- 저널링 기능을 이용한 ext3 는 ext2 파일 시스템보다 데이터 신뢰성이 강화 되었고, 보다 빠른 속도로 작업을 수행할 수 있다.
- 인식할 수 있는 최대 볼륨 16TB 까지이다.
- 삭제파일 복구가 용이하지 않다.

☆ 일반 파일시스템과 저널링 파일시스템

#### 기존 일반 파일 시스템의 단점

- 예기치 않은 시스템 장애가 발생했을 때, 디스크 상의 모든 파일이 잘못된 i-노드나 데이터 블록을 가리키는 i-노드를 갖고 있는 지를 확인하기 위해 fsck 프로그램을 수행해야 한다.
- 큰 용량의 파일 시스템에 대해 fsck 를 실행하는 작업은 오랜 시간이 걸린다.

#### 저널링 파일 시스템 원리

- 데이터를 디스크에 기록할 때 적절한 장소를 찾는 대신에 가장 빨리 저장할 수 있는 아무 곳이나 데이터를 기록한다.
- 그 때마다 이 데이터 블록의 위치를 로깅한다.
- 데이터를 기록하는데 걸리는 시간이 훨씬 적어지고, 데이터를 빨리 기록하는 안전성 때문에 시스템에 장애가 발생하더라도 매번 fsck 를 수행할 필요가 없다.

## ext4

---

- ext3 를 개선하여 개발한 파일 시스템이다.
- 파일시스템 점검속도가 현저하게 빨라졌다.
- 파일 복구가 용이하다.
- 16TB 이상의 큰 볼륨을 가질 수 있다.

## fat

---

MS-DOS 및 윈도우 운영 체제에서 쓰이는 파일 시스템이다.

## NTFS

---

- 윈도우 서버 및 XP, 비스타 운영체제에서 쓰이는 파일 시스템이다.

- 예전의 리눅스는 읽기 전용으로 지원되었지만, 최근에는 읽기/쓰기가 가능하다.

## iso9660

---

- CD-ROM 표준 파일 시스템이다.

## XFS (Red Hat Enterprise 7 의 새로운 기본 파일 시스템)

---

- 최대 파일 시스템의 크기가 16 엑사바이트(exabytes)이며 최대 파일 크기는 8 엑사바이트이다.

## 파일 시스템관련 설정 파일과 명령어

---

파일 시스템과 관련된 설정 파일은 /etc/fstab 파일과 /etc/mtab 파일이 있다.

- /etc/fstab : 부팅시 마운트 할 파일 시스템 설정
- /etc/mtab : 현재 파일 시스템의 마운트 상황

## /etc/fstab

---

부팅 시 마운트 할 파일 시스템 설정

```
# cat /etc/fstab
# cat /etc/fstab
#
# /etc/fstab
# Created by anaconda on Sun Mar  3 09:09:21 2019
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=effd12ec-9fa1-4aea-917b-bcb3b524fa84 / ext4 defaults 1 1
UUID=2db662af-6786-4fec-bea8-63d84505424d /boot ext4 defaults 1 2
UUID=ad823036-d40f-4a5c-81a4-73d62a0cbee0 swap swap defaults 0 0
```

➤ 위에서 보는 것처럼 한 행이 마운트 할 파일 시스템이 있고 6 개의 항목으로 되어 있다.

장치(device)	마운트포인트	파일시스템종류	마운트옵션	백업	검사(우선 순위)
/LABEL=/ 	/	ext2	defaults	1	1

- 장치(devices) : 마운트 할 파일 시스템의 장치 파일 이름을 입력한다.  
(예) /dev/hda1, /dev/cdrom
- 마운트 포인트 : 파일 시스템이 마운트될 디렉토리를 입력한다.
- 파일 시스템 종류: 마운트 할 파일 시스템의 종류를 입력한다.  
(예) ext4, swap, iso9660
- 마운트 옵션 : 이 항목에는 다양한 옵션들이 존재하는데 각각의 옵션들을 콤마(,)를 이용해서 중복 입력이 가능하다. 대부분 defaults 를 이용한다.

옵 션	설 명
Defaults	rw, suid, dev, exec, auto, nouser, async, 옵션을 모두 선택한 것과 같다.
Auto	부팅시 자동으로 마운트
Noauto	부팅시 자동으로 마운트 하지 않는다.
Nosuid	uid 와 gid 가 적용되지 않는다.
ro	읽기 전용
rw	읽기/쓰기 허용

- 백업 : 백업 여부를 선택한다.  
0 : 백업하지 않는다.  
1 : 백업 가능한 파일 시스템
- 검사(우선 순위) : fsck 를 이용한 파일 시스템 검사 옵션  
0: 부팅시에 파일 시스템을 점검하지 않는다.  
1: 루트 파일 시스템으로서 부팅 시에 파일 시스템 점검을 한다.  
2: 루트 파일 시스템 이외의 파일 시스템으로서 부팅 시에 파일 시스템 점검을 한다.

## [/etc/mtab](#)

현재 파일 시스템의 마운트 상황(mount 명령어로 확인할 수 있다)

```
# cat /etc/mtab
rootfs / rootfs rw 0 0
sysfs /sys sysfs rw,seclabel,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
```

```
devtmpfs /dev devtmpfs rw,seclabel,nosuid,size=919144k,nr_inodes=229786,mode=755 0 0
....
# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime,seclabel)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
devtmpfs on /dev type devtmpfs
(rw,nosuid,seclabel,size=919144k,nr_inodes=229786,mode=755)
...
```

- 마운트 상황이 변경되면 /etc/mtab 파일의 내용도 변경되므로 특별히 수정할 필요는 없다.  
파일의 내용은 /etc/fstab 파일과 유사하다.

## 리눅스 파일시스템 생성(포맷) 작업

---

### mkfs 명령

---

- 리눅스에서 파일시스템을 생성하는 작업을 하기 위한 유틸리티이다.
- 새로운 디스크를 추가로 장착하였을 경우에는 fdisk 로 먼저 파티션을 생성한 다음 mkfs 로 파일시스템을 생성해야 한다.
- ext2 파일시스템을 생성할 때 사용하는 전용명령어로 mke2fs 명령어가 있다.
- Syntax : mkfs [-V] [-t 파일시스템 종류] [파일시스템 옵션] [장치이름] [용량]
- xfs 타입의 파일시스템 생성하기 예  
# mkfs -t xfs /dev/sdj1