

Contribution

Marathon

Innovation

# 엔터프라이즈 서버관리

## 7주차 : 중간시험대비 총정리

---

2025년 1학기

# 목 차

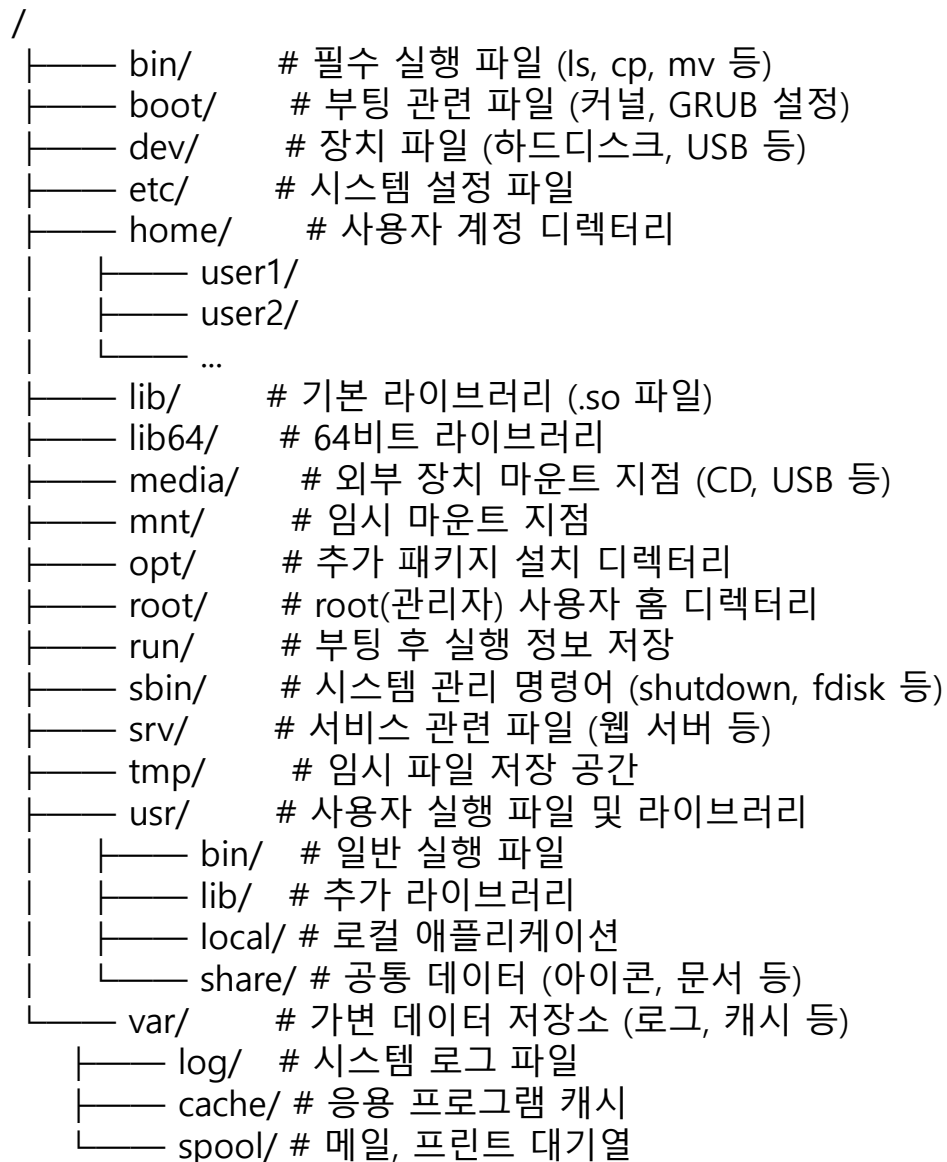
1. 리눅스의 디렉터리 구조
2. 리눅스 라이선스
3. 파티션
4. 사용자 생성 및 계정관리
5. 디렉터리 및 파일
6. 권한 및 그룹 설정
7. 파일시스템 관리

# 1. 리눅스의 디렉터리 구조

## ■ 디렉터리

- 파일 저장소를 의미
- 최상위 디렉터리(/)를 기준으로 하위 디렉터리들이 존재하는 계층적 트리구조

최상위 디렉터리 / 는 root이다  
root/는 루트 관리자 홈 디렉터리이다.  
헛갈리면 안됨



# 1. 리눅스의 디렉터리 구조

## ■ 리눅스의 디렉터리별 저장 내용

설명보고 어떤 디렉터리인지 알 수 있게 공부  
중요한 거 위주로 외우기  
bin,/sbin,dev, etc,

디렉터리	설명	저장되는 주요 파일/내용
/	최상위 디렉터리, 모든 파일과 디렉터리의 출발점	시스템의 모든 파일과 폴더 포함
/bin	필수 실행 파일(기본적인 명령어)이 저장된 디렉터리	ls, cp, mv, cat 등 기본 명령어
/sbin	시스템 관리 명령어 저장 (관리자용 시스템 표준 명령어)	shutdown, fdisk, reboot 등 관리자 명령어
/boot	부팅 관련 파일 저장(부팅 시 커널 이미지와 부팅 정보 저장)	커널 파일 (vmlinuz), GRUB 설정 파일
/dev	장치(Device) 파일 저장	하드디스크 (/dev/sda), 터미널 (/dev/tty)
/etc	시스템 환경 설정 파일 저장	passwd, shadow, fstab, hosts
/home	일반 사용자 계정의 홈 디렉터리	각 사용자 개인 파일 (/home/user)
/root	root(관리자) 사용자 홈 디렉터리	root 계정의 설정 파일, 개인 파일
/lib, /lib64	필수 시스템 라이브러리 저장 (프로그램 실행 지원)	.so 공유 라이브러리 파일
/media	외부 저장 장치 마운트 위치	USB, CD-ROM (/media/cdrom, /media/usb)
/mnt	파일시스템을 임시 마운트 디렉터리	외부 파일시스템 수동 마운트 위치
/opt	추가 소프트웨어 패키지 저장	타사 응용 프로그램 (/opt/google/chrome)
/run	시스템 시작 후 실행 정보 저장	PID 파일, 소켓 파일
/srv	서비스 관련 데이터 저장	웹 서버 (/srv/www), FTP 서버 데이터
/tmp	프로세스 작업 시 임시 파일 저장 (재부팅 시 삭제)	임시 파일, 세션 데이터
/usr	사용자용 애플리케이션 및 라이브러리	실행 파일(/usr/bin), 라이브러리(/usr/lib)
/var	시스템의 가변 데이터(로그, 캐시)와 메일 데이터 저장	로그 파일(/var/log), 메일(/var/mail)
/proc	가상 파일 시스템으로 커널과 시스템 정보를 실시간으로 제공	CPU 정보, 시스템 메모리 정보 등
/lost+found	파일 시스템의 이상 유무를 진단하고 복구하는 fsck에 의해 사용	손상된 파일의 복구된 부분 등

# 1. 리눅스의 디렉터리 구조 [예상문제]

1. /home 디렉터리에 대한 일반적인 설명으로 맞는 것은?

- ① 사용자 계정의 홈 디렉터리가 위치하는 디렉터리
- ② 시스템 환경 설정 파일을 저장하는 디렉터리
- ③ 파일 시스템을 일시적으로 마운트 시 사용하는 디렉터리
- ④ 기본적인 명령어가 저장되는 디렉터리

/home 디렉터리는 사용자 홈 디렉터리로서, login 시 처음으로 위치하게되는 디렉터리이다. 리눅스는 언어 지원이나 시스템 설치 등 여러 가지 면에서 윈도우에 비해 사용이 용이하지는 않다. ② /etc ③ /mnt ④ /bin

# 1. 리눅스의 디렉터리 구조 [예상문제]

## 2. 리눅스에서 파일 시스템은 어떠한 구조로 구성되어 있는가?

- ① 배열 구조
- ② 단층 구조
- ③ 네트워크 구조
- ④ 계층적 트리 구조

리눅스는 최상위 디렉터리를 기준으로 하위 디렉터리들이 존재하는 계층적 트리 구조로 구성되어 있다.

# 1. 리눅스의 디렉터리 구조 [예상문제]

## 3. Linux의 디렉터리에 대한 설명으로 옳지 않은 것은?

- ① /bin : 시스템의 각종 드라이버들이 저장
- ② /var : 가변 자료 저장 디렉터리로 로그 파일 저장
- ③ /lib : 프로그램 실행을 지원해 주는 라이브러리 저장
- ④ /root : 루트 사용자의 홈 디렉터리로 루트 사용자의 관련 파일들을 저장

/bin : 실행 파일이나 기본 명령어 저장

# 1. 리눅스의 디렉터리 구조 [예상문제]

4. Linux의 기본 디렉터리와 해당 디렉터리에 저장되는 파일들의 설명으로 옳지 않은 것은?

- ① /tmp : 프로세스 작업 시 임시로 생성되는 파일 저장
- ② /boot : 부팅 시 커널 이미지와 부팅 정보 저장
- ③ /var : 시스템의 로그 파일과 메일 데이터 저장
- ④ /sbin : 실행 파일이나 기본 명령어 저장

/sbin : 관리자용 시스템 표준 명령어 및 시스템 관리와 관련된 실행 명령어 저장

/bin : 실행 파일이나 기본 명령어 저장



# 1. 리눅스의 디렉터리 구조 [예상문제]

5. 다음 중 가상 파일 시스템으로 프로세스 및 하드웨어에 대한 정보를 담고 있는 디렉터리는?

- ① /tmp
- ② /proc
- ③ /var
- ④ /lost+found

① /tmp : 프로세스 작업 시 임시로 생성되는 파일 저장

③ /var : 시스템의 로그 파일과 메일 데이터 저장

④ /lost+found : 파일 시스템의 이상 유무를 진단하고 복구하는 fsck에 의해 사용

## 2. 리눅스 라이선스

### ■ GNU GPL(General Public License)

- 자유롭게 사용, 복제, 수정 후 재배포하는 자유 소프트웨어
- 소프트웨어를 변경할 경우 해당 소프트웨어도 GPL 라이선스로 배포되고 외부에 공표나 배포할 때에는 **전체 소스코드가 무료로 공개**되어야 함
- 리눅스도 GPL 라이선스를 적용함
- 자유는 금전적인 측면과 관계가 없기 때문에 **유료로 판매**한다고 해도 문제가 생기지는 않음
- 프리 소프트웨어 재단(FSF)에서 만든 GNU 소프트웨어를 누구나 자유롭게 사용하도록 하기 위해 만든 저작권 없는 **카피레프트(copyleft)** 라이선스



카피레프트(copyleft) : 저작권 없음

카피라이트(copyright) : 저작권 있음

카피라이트 표기법 : Copyright 2024, © 2024, (C) 2024

## 2. 리눅스 라이선스

### ■ LGPL(Lesser GPL)

- 1991년 프리 소프트웨어 재단(FSF)의 리처드 스톨먼은 GPL 라이선스의 강력한 카피레프트 조건을 절충하여 **단순한 사용**은 소스코드를 공개하지 않아도 되는 것을 허가하는 LGPL 발표
- **라이브러리에 응용프로그램을 링크** 시킬 경우에는 소스코드를 공개할 의무가 없기 때문에 독점 소프트웨어로 사용 가능함
- LGPL을 사용해서 개발한 뒤 GPL로 변경이 가능

## 2. 리눅스 라이선스

### ■ MPL(Mozilla Public License)

- GPL 라이선스와 BSD 라이선스의 혼합적인 성격
- MPL 소스코드 수정 시 수정한 2차 저작물의 소스코드는 공개해야 함
  - . 단, MPL 소스코드와 혼합하여 다른 프로그램을 개발할 경우 수정된 코드만 공개
  - . 전체 프로그램은 비공개 가능
- MPL 라이선스로 소스코드를 수정한 경우 수정한 소스코드를 반드시 공개하고 원 저작자에게 수정한 부분을 알려야 함

수정한 소스코드 공개 : GPL, LGPL, MPL pl로 끝나는 건 일부든 전체든 공개 해야 하는 것

수정한 소스코드 미공개 : BSD, Apache, MIT

## 2. 리눅스 라이선스

### ■ BSD, Apache, MIT 라이선스

- 관련 라이선스가 적용된 소스코드를 수정하여 만든 2차적 저작물에 대해 소스코드의 비공개가 허용

- . 해당 라이선스가 적용된 소프트웨어를 다운로드하여 부분 또는 전체를 개인적이나 상업용 목적으로 사용이 가능

- . BSD (캘리포니아 대학교 버클리 캠퍼스에서 배포)

- 수정본을 무료로 재배포하는 것은 의무적인 사항이 아니므로 상용 소프트웨어에서도 사용

- . MIT (MIT 대학교에서 배포)

- 원작자의 저작권 정보와 라이선스 내용을 제거해서는 안됨

- 적용된 소프트웨어로는 X Windows System, JQuery, Node.js 등이 있다

- . Apache 라이선스 (아파치 소프트웨어 재단에서 배포)

- 재배포할 경우 Apache License 2.0 전문을 포함시키고 관련 소프트웨어임을 밝혀야 함

## 2. 리눅스 라이선스 [예상문제]

### 1. 다음에서 설명하는 라이선스로 알맞은 것은?

해당 소프트웨어를 누구나 개작할 수 있고, 수정본의 재배포 시에 소스코드 비공개가 가능하다. 이 라이선스가 적용된 소프트웨어에는 X Windows System, JQuery, Node.js 등이 있다.

① BSD

② MPL

③ GPL

④ MIT

MIT 라이선스는 소프트웨어를 개조한 제품을 반드시 오픈 소스로 배포해야 한다는 규정이 없으며, 이에 GNU 일반 공중 사용 허가서의 엄격함을 피하려는 사용자들에게 인기가 있다.

## 2. 리눅스 라이선스 [예상문제]

### 2. 이 설명에 해당하는 라이선스로 알맞은 것은?

독립적인 저작물일 경우에는 독립 저작물 모듈의 개별적인 배포에는 적용되지 않지만 어떤 형태로든(유료든 무료든) 외부에 공표나 배포할 때에는 전체 소스코드를 공개해야 한다.

- ① BSD
- ② MPL
- ③ GPL
- ④ MIT

GPL을 가진 프로그램을 유료로 판매하는 것은 가능하지만, 반드시 전체 소스코드를 무료로 공개해야 한다.

## 2. 리눅스 라이선스 [예상문제]

3. 다음 중 소스코드를 수정해서 만든 2차적 저작물(수정된 코드)을 반드시 공개해야 하는 라이선스로 알맞은 것은?

① GPL

② MPL

③ BSD

④ MIT

수정한 2차 소스코드는 MPL로 공개하고 원저작자에게 수정한 부분에 대해 알려야 하지만, 실행 파일은 독점 라이선스로 배포할 수 있다.

MPL 라이선스는 수정된 코드만 공개 (전체 프로그램은 비공개 가능)

GPL 라이선스는 수정된 소스 코드 전체 공개 필수



## 2. 리눅스 라이선스 [예상문제]

### 4. 다음 중 아파치 라이선스에 대한 설명으로 틀린 것은?

- ① 아파치 소프트웨어 재단에서 자체적으로 만든 소프트웨어 라이선스 규정이다.
- ② GPL보다 완화된 조건의 공개 소프트웨어 라이선스이다.
- ③ 누구든 자유롭게 아파치 소프트웨어를 다운받아 부분 또는 전체를 개인적 혹은 상업적 목적으로 이용할 수 있다.
- ④ 재배포 시 아파치 라이선스 버전 2.0을 포함시켜야 한다.

GPL보다 완화된 조건의 소프트웨어는 LGPL(Lesser General Public License)이다.

## 2. 리눅스 라이선스 [예상문제]

5. 다음 중 라이선스 특성이 다른 하나는?

① Apache

② MIT

③ BSD

④ MPL

MPL은 Apache, MIT, BSD 라이선스가 적용된 프로그램들의 소스코드를 수정해서 사용할 경우에도 반드시 별도의 소스코드와 실행파일을 공개할 필요는 없다.

## 2. 리눅스 라이선스 [예상문제]

6. 다음 중 BSD 라이선스에 대한 설명으로 틀린 것은?

- ① 소스코드 공개의 의무가 없다.
- ② 상업적 소프트웨어에서도 무제한 사용 가능하다.
- ③ 수정본의 재배포는 의무사항이다.
- ④ 수정본은 소스코드로 공개하지 않아도 된다.

BSD 라이선스는 수정본의 재배포가 의무적인 사항이 아니므로 상용 소프트웨어에서도 사용할 수 있다.

# 3. 파티션

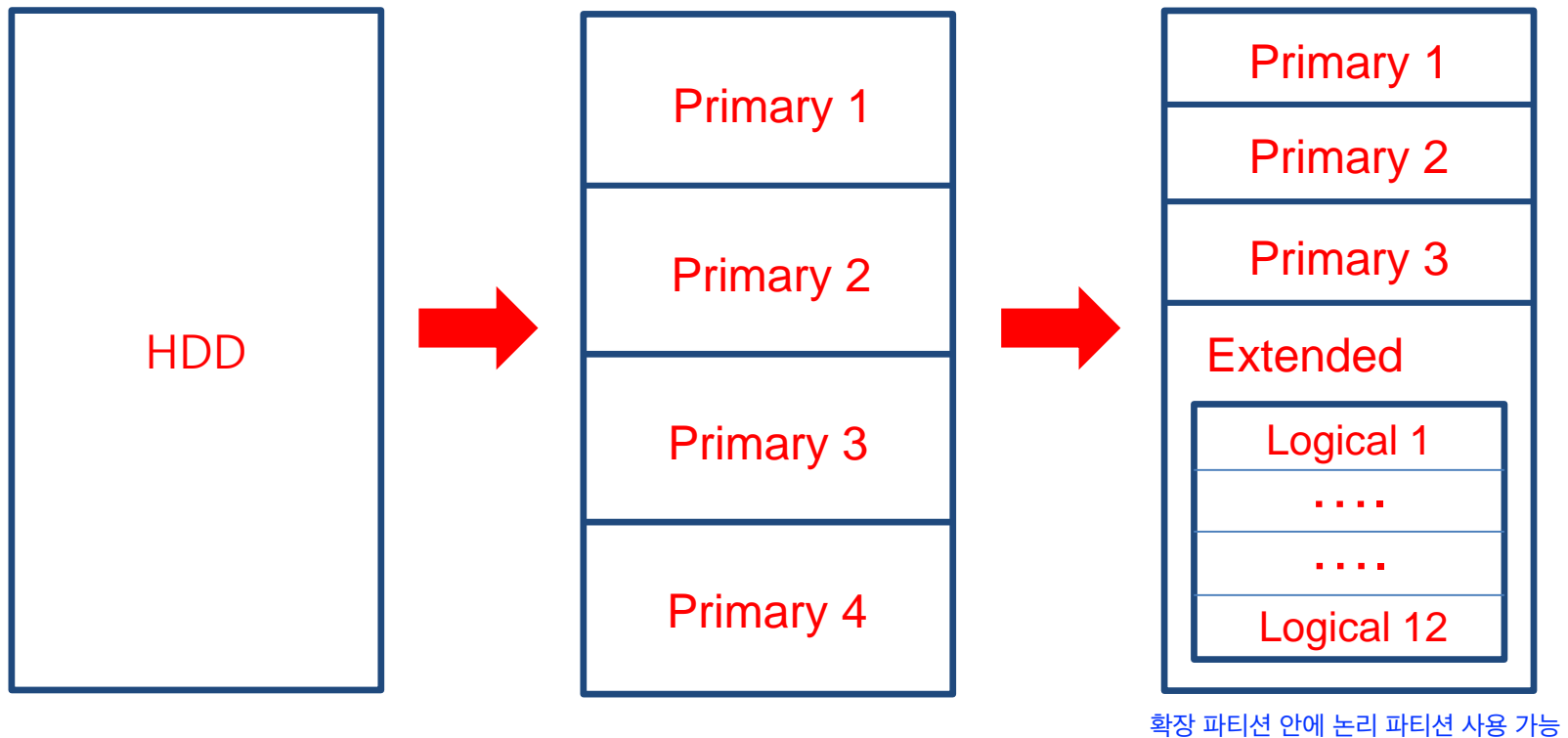
## ■ 파티션 (Partition)

쓸 만큼만 분할하는 것

- 하나의 물리적 디스크를 여러 개의 논리적 디스크로 분할 하는 것
  - 파티션마다 독립적인 파일 시스템이 운영
  - 파티션 정보를 확인할 수 파일 : `/proc/partitions`
  - 주(Primary) 파티션, 확장 파티션, 논리 파티션, 스왑 파티션으로 구분
    - . 주 파티션은 4개까지 사용 가능 (파티션 번호 1~4번 사용)
    - . 4개 이상의 파티션을 사용해야 할 때 하나의 주 파티션 안에 확장 파티션 설정
    - . 확장 파티션은 하나의 물리적 디스크에 1개만 선언 가능
    - . 확장 파티션 안에 여러 개의 논리 파티션(5번 이후)을 분할하여 데이터를 저장
    - . 스왑파티션은 가상메모리 영역을 담당하며 RAM 용량의 2배를 권장
- [스왑과 반대 개념] RamDrive(Ram Disk) : RAM의 일부를 저장 공간으로 활용

# 3. 파티션

- 파티션 (Partition)



# 3. 파티션

## ■ 디바이스 명명 규칙

### ① 플로피 디스크

- . 첫 번째 플로피 : /dev/fd0
- . 두 번째 플로피 : /dev/fd1

### ② SCSI 디스크

- . 첫 번째 SCSI 디스크 : /dev/sda
- . 두 번째 SCSI 디스크 : /dev/sdb

### ③ IDE 디스크

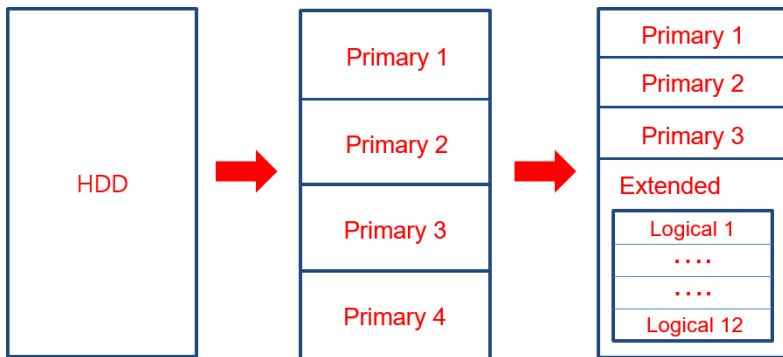
- . Primary Master : /dev/hda
- . Primary Slave : /dev/hdb
- . Scondary Master : /dev/hdc
- . Scondary Slave : /dev/hdd

# 3. 파티션 [예상문제]

## 1. 다음 중 파티션의 유형에 대한 설명으로 틀린 것은?

- ① 주 파티션은 4개까지 사용 가능하다.
- ② 주 파티션 4개 사용 후 확장 파티션의 선언이 가능하다.
- ③ 5개 이상의 파티션이 필요한 경우 확장 파티션의 선언이 필요하다.
- ④ 부팅 가능한 파티션은 디스크에 하나 이상 존재해야 한다.

4개 이상의 파티션을 사용해야 할 때 하나의 주 파티션 안에 확장 파티션을 설정한다.  
확장 파티션 안에 여러 개의 논리 파티션을 분할하여 데이터를 저장한다.



### 3. 파티션 [예상문제]

2. 물리적 시스템의 메모리의 크기가 512MB일때 권장하는 스왑 공간 용량으로 알맞은 것은?

- ① 256MB
- ② 512MB
- ③ 1024MB
- ④ 1536MB

스왑 영역은 메모리의 2배이므로  $512 \times 2 = 1024\text{MB}$ 이다.



### 3. 파티션 [예상문제]

3. 다음 중 E-IDE 타입의 디스크를 Secondary Slave에 연결했을 경우에 인식하는 장치 파일명으로 알맞은 것은?

- ① /dev/hda
- ② /dev/hdb
- ③ /dev/hdc
- ④ /dev/hdd

500MB를 초과하는 하드디스크를 지원하는 E-IDE 타입의 디스크는 ATA-2로 규격화하였다. 리눅스에서 HDE의 디바이스명은 다음과 같다.

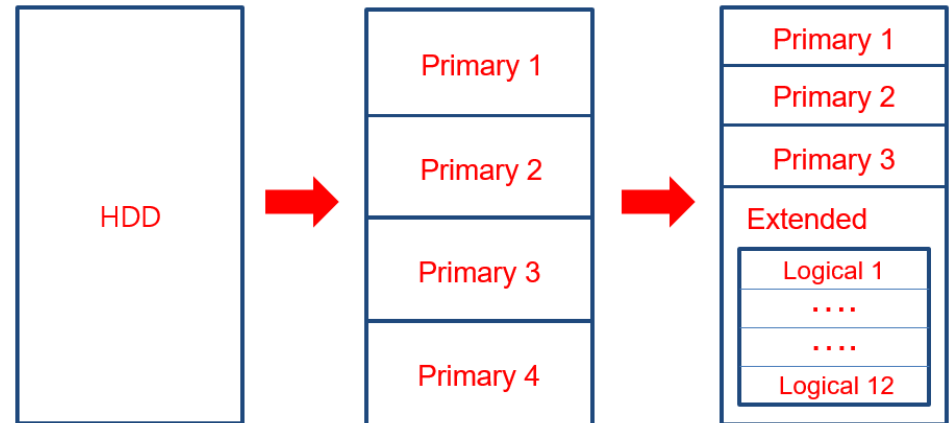
- ① /dev/hda : Primary Master
- ② /dev/hdb : Primary Slave
- ③ /dev/hdc : Secondary Master
- ④ /dev/hdd : Secondary Slave

### 3. 파티션 [예상문제]

4. 다음 중 /dev/sdb6에 대한 설명으로 틀린 것은?

- ① 첫 번째 확장 파티션이다.
- ② 디스크 유형은 IDE이다.
- ③ 해당 파티션은 확장 파티션에 속해 있다.
- ④ 두 번째 논리 파티션이다.

디스크 유형은 SCSI 또는 S-ATA이다.

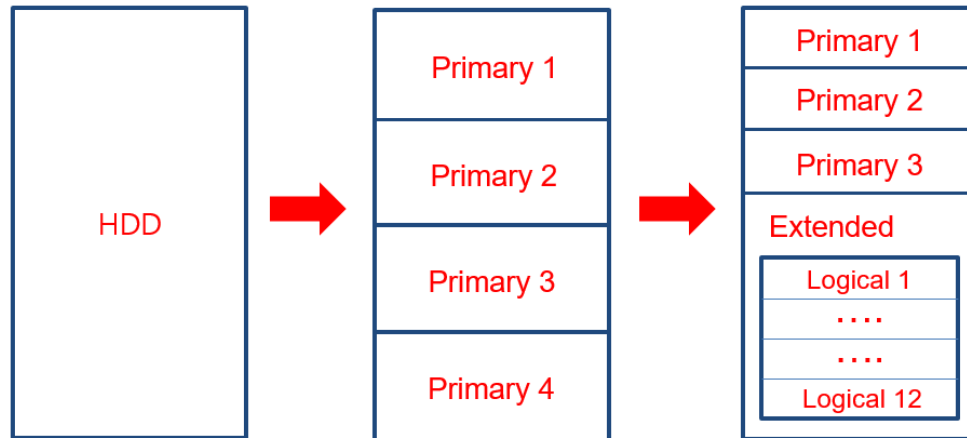


### 3. 파티션 [예상문제]

5. 다음 중 논리 파티션에 부여될 수 있는 최소 파티션 번호 값은?

- ① 5
- ② 6
- ③ 7
- ④ 8

파티션 번호는 1번에서 4번까지는 주 파티션 또는 확장 파티션에서 사용하며 5번부터 논리 파티션에서 사용한다.



### 3. 파티션 [예상문제]

#### 6. 다음 중 파티션에 대한 설명으로 틀린 것은?

- ① 주 파티션의 번호는 1번부터 4번이 할당된다.
- ② 주 파티션을 2개만 분할해서도 사용 가능하다.
- ③ 확장 파티션은 하나의 물리적 디스크에 1개만 선언 가능하다.
- ④ 확장 파티션을 선언해야 주 파티션을 5개 이상 사용 가능하다.

하나의 하드디스크에 최대 4개의 주 파티션 분할이 가능하다. 4개 이상의 파티션을 사용해야 할 때 하나의 확장 파티션에 여러 개의 논리 파티션을 분할하여 데이터를 저장한다.

## 4. 사용자 생성 및 계정관리

### ■ 명령어 **which**

- 명령어의 위치(경로)를 찾아주고 별칭이 있는 경우 alias를 보여줌

```
[linux@localhost ~]$ which ls
alias ls='ls --color=auto'
/usr/bin/ls
```

### ■ 명령어 **alias**

- 명령어를 간소화하여 다른 이름으로 사용할 수 있게 해주는 셸 내부 명령어
- 명령어 형식은 '**alias** **별명=명령어**'

```
[linux@localhost ~]$ alias c=clear
```

- **unalias**는 alias 기능을 해제하는 명령어
- unalias의 형식은 '**unalias** **별명**'

```
[linux@localhost ~]$ unalias c
```

## 4. 사용자 생성 및 계정관리

### ■ 파일 /etc/passwd

- 계정자의 정보를 가지고 있는 파일로 리눅스에 로그인할 때 사용
- 리눅스 계정 정보를 담은 텍스트 파일로 패스워드 자리는 X로 표시

```
[root@localhost linux]# tail -l /etc/passwd
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
gnome-initial-setup:x:981:980::/run/gnome-initial-setup:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/usr/share/empty.sshd:/sbin/nologin
chrony:x:980:979:chrony system user:/var/lib/chrony:/sbin/nologin
dnsmasq:x:979:978:Dnsmasq DHCP and DNS server:/var/lib/dnsmasq:/usr/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
linux:x:1000:1000:서버 123:/home/linux:/bin/bash
jjpark:x:1001:1001:Jinju Park:/home/jjpark:/bin/bash
apache:x:48:48:Apache:/usr/share/httpd:/sbin/nologin
inhatec:x:1002:1002::/home/inhatec:/bin/bash
```

username:password:uid:gid:comment:home directory:shell

## 4. 사용자 생성 및 계정관리

### ■ 파일 /etc/shadow

- 계정자의 패스워드 정보가 암호화 되어 있는 파일로 암호화 패스워드 및 계정의 유효기간 등을 기록하고 있는 파일

• **kdhong:\$6\$w8j< 중간생략>:19585:0:100:7:10:19722:**

중간고사에서는 안 씀

- 사용자 계정:암호화된비밀번호:최종비밀번호변경일:
- **0**은 비밀번호 변경 후 바꿀 수 없는 기간(**minimum password age**)
- **100**은 비밀번호 변경 후 다시 변경하지 않고 사용할 수 있는 기간  
최대 기간 (**maximum password age**)은 **99999**
- **7**은 비밀번호 만료일 전에 경고를 보내는 날짜 수(**password warning period**)
- **10**은 비밀번호의 만료 후 로그인 가능한 날짜 수(**password inactivity period**)
- 사용자 계정의 만료일(**account expiration date**) 로 빈 값은 계정이 만료되지 않는다는 것
- 예약 필드

```
[root@localhost linux]# tail -l /etc/shadow
gdm:!!:19791::::::
gnome-initial-setup:!!:19791::::::
sshd:!!:19791::::::
chrony:!!:19791::::::
dnsmasq:!!:19791::::::
tcpdump:!!:19791::::::
linux:$6$aQ2.2qZxEVufSjq8$KnsghWcMxRJc5siv6x/ra7IzH4Hn0G.Y7f6voynyTIER5SiiGf.6FHY3Dlj0yqFM/mLuzp
jjpark:$6$09x7TfhAHirW3ebb$jmxDWLShLKZqi/8PBaqjEu2.mFE9Bd6BkiNJEryTw1SsbA1SkjgxThN0yHrBpkiWxDKlP
apache:!!:20054::::::
inhatc:$6$qwj3qIvYQIH05kCh$MwrN0fWxyL3JsUqZV8L082ljyQg7FBBHX3lIdQxzZwGltay7EF1j3aN6n.8aPmzG.4kpI
```

## 4. 사용자 생성 및 계정관리

### ■ 명령어 useradd

이 명령어만 adduser 처럼 순서를 바꿔서 쓸 수 있음 다른 것들은 안 됨  
시험에 안 나올 듯

- 새로운 사용자 계정을 추가하는 명령어 (adduser도 사용 가능)
- 형식 : **useradd** [옵션] 계정명
- [옵션] -f : 패스워드가 만기된 후 계정이 영구히 말소될 때까지의 기간 지정  
예) **useradd -f 5 inhatac1** : 패스워드 만기 후 5일 동안 사용
- e : 사용자 계정의 유효기간을 설정 예) **useradd -e 2025-06-30 inhatac2**
- u : UID 지정 예) **useradd -u 1050 inhatac3**
- g : GID 지정 예) **useradd -g 1050 inhatac4**

etc/shadow

```
inhatac1:!!!:20176:0:99999:7:5::  
inhatac2:!!!:20176:0:99999:7::20269:  
inhatac3:!!!:20176:0:99999:7:::  
inhatac4:!!!:20176:0:99999:7:::
```

etc/passwd

```
inhatac1:x:1111:1111::/home/inhatac1:/bin/bash  
inhatac2:x:1112:1112::/home/inhatac2:/bin/bash  
inhatac3:x:1050:1050::/home/inhatac3:/bin/bash  
inhatac4:x:1113:1050::/home/inhatac4:/bin/bash
```

```
[root@localhost linux]# useradd kdh
```

```
[root@localhost linux]# passwd kdh
```

kdh 사용자의 비밀번호 변경 중

※ 명령어 passwd

사용자 계정의 패스워드 입력 및 변경

새 암호 :

새 암호 재입력 :

passwd: 모든 인증 토큰이 성공적으로 업데이트 되었습니다 .



## 4. 사용자 생성 및 계정관리

- 파일 `/etc/default/useradd`

- 명령어 `useradd`로 사용자 계정을 추가할 때 사용되는 정보를 읽어오는 파일

```
[root@localhost linux]# cat /etc/default/useradd
# useradd defaults file
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

## 4. 사용자 생성 및 계정관리

### ■ 명령어 usermod

- 계정 설정을 변경할 때 사용하는 명령어
- 형식 : **usermod** [옵션] 계정명
- [옵션]
  - u : 새로운 UID를 지정
  - g : 새로운 GID를 지정
  - L : 로그인을 못하게 함
  - I : 로그인 ID를 바꾸는 옵션으로 새로운 계정명으로 변경

예) 명령어 usermod를 이용한 계정 변경 방법

**'usermod -I 변경ID 기존ID'**

```
[root@localhost linux]# usermod -l kdh myj
```

```
[root@localhost linux]# usermod -L jjpark
```

로그인 안됨

```
linux:$6$aQ2.  
jjpark:$6$o9x
```



```
linux:$6$aQ2.  
jjpark:!$6$o9
```

## 4. 사용자 생성 및 계정관리

### ■ 명령어 userdel

- 계정을 삭제하는 명령어
- 형식 : **userdel** [옵션] 계정명
- [옵션] -r : 계정명의 /var/spool/mail의 메일 파일과 홈 디렉터리의 내용 모두를 삭제
- f : 로그인 중이거나 다른 사용자가 사용 중이어도 삭제

```
[root@localhost linux]# userdel kdh
```

### ④ 사용자 계정 삭제 시 고려사항

- ➔ 홈 디렉터리도 삭제할 것인가 삭제하면 홈 디렉터리에 중요한 데이터가 있을 수도 있으므로 삭제 안하면 새로운 사용자가 기존 사용자의 UID를 할당 받으면 기존 사용자의 홈 디렉터리 정보를 볼 수 있는 문제가 발생함
- ➔ 계정을 삭제하지 말고 잠금 할 것인가 잠금만 하면 UID는 삭제하지 않음
- ➔ 삭제될 계정이 소유하는 파일이 또 있는가
  - 삭제 전에 find / -user username -ls로 확인해야 함 해당 계정이 소유하는 파일을 모두 찾을
  - 삭제 후라면 find / -uid UID -ls 또는 find / -nouser -ls로 확인함 삭제되어 존재하지 않는 계정이 소유하는 파일이 있는지 확인

## 4. 사용자 생성 및 계정관리

### ■ 명령어 su

- switch user의 줄임말
- 현재의 사용자 계정에서 로그아웃하지 않고 다른 사용자 계정으로 로그인하여 해당 사용자의 권한을 획득하는 명령

[옵션] -, -, --login : 지정한 사용자의 환경변수를 적용하여 로그인

```
[linux@localhost ~]$ su - root
```

```
[linux@localhost ~]$ su -l root
```

```
[linux@localhost ~]$ su --login root
```

## 4. 사용자 생성 및 계정관리

### ■ 명령어 chage

- chage(change age)는 리눅스 사용자 계정의 비밀번호 만료 정책을 변경
- 사용자의 패스워드에 대한 정보를 출력하고 `/etc/shadow`의 날짜 관련 필드에 모두 설정할 수 있는 명령
- 사용자의 패스워드에 대한 정보를 보여줌

```
[root@localhost ~]# chage linux
linux의 사용기한 정보를 바꿉니다
새로운 값을 넣거나, 기본값을 원하시면 엔터를 치세요

암호의 최소 유효 기간 [20]:
암호의 최대 유효 기간 [8888]:
마지막으로 암호를 바꾼 날 (YYYY-MM-DD) [2024-03-18]:
암호 사용만료 예고 [7]:
암호를 사용할 수 없음 [10]:
계정 만료 날짜 (YYYY-MM-DD) [2065-10-29]:
```

## 4. 사용자 생성 및 계정관리 [예상문제]

1. 다음 (      ) 안에 들어가는 명령으로 알맞은 것은?

```
# (      ) ls  
alias ls='ls --color=auto'  
/usr/ls
```

① which

② find

③ pwd

④ path

명령어 which는 명령어의 위치를 찾거나 alias를 보여준다.

```
[linux@localhost ~]$ which ls  
alias ls='ls --color=auto'  
/usr/bin/ls
```

## 4. 사용자 생성 및 계정관리 [예상문제]

2. 다음 중 명령행에서 c라고 입력하면 clear라는 명령이 실행되도록 설정하려고 할 때 알맞은 것은?

- ① alias -m clear c
- ② alias c=clear
- ③ alias -m clear=c
- ④ alias c clear

명령어 alias는 명령어를 간소화하여 다른 이름으로 사용할 수 있도록 해주는 셸 내부 명령어이다. 명령어 형식은 'alias 별명=명령어'이다.

```
[linux@localhost ~]$ alias c=clear
```

## 4. 사용자 생성 및 계정관리 [예상문제]

3. 명령 F를 입력하면 'ls -alF'가 실행된다. 이 명령의 실행을 중단할 때 가장 알맞은 것은?

- ① alias F 'ls -alF'
- ② alias F= 'ls -alF'
- ③ unalias F
- ④ alias F 'ls -alF'

unalias는 alias 기능을 해제하는 명령어이다. unalias의 형식은 'unalias 별명'이다.

```
[linux@localhost ~]$ unalias c
```



## 4. 사용자 생성 및 계정관리 [예상문제]

### 4. 다음 명령어에 대한 설명으로 맞는 것은?

```
useradd -f 5 ihd
```

- ① 계정 ihd는 비밀번호 만기 후 5일 동안 사용할 수 있다.
- ② 계정 ihd의 우선순위를 5만큼 낮춰 생성한다.
- ③ 계정 ihd의 UID를 5로 지정한다.
- ④ 계정 ihd의 실행 권한 수준을 5로 변경한다.

useradd 옵션 -f는 비밀번호가 만기된 후 계정이 영구히 말소될 때까지의 기간을 지정한다.

## 4. 사용자 생성 및 계정관리 [예상문제]

5. 다음 중 일반적인 사용자 관리 명령어로 틀린 것은?

- ① userdel
- ② useradd
- ③ usercreate
- ④ usermod

- ① userdel : 계정을 삭제하는 명령어이다.
- ② useradd : 새로운 사용자 계정을 추가하는 명령어이다.
- ④ usermod : 계정 설정을 변경할 때 사용하는 명령어이다.

## 4. 사용자 생성 및 계정관리 [예상문제]

6. 다음 중 사용자 생성 및 삭제 명령으로 틀린 것은?

① adduser

② deluser

③ userdel

④ useradd

명령어 adduser와 useradd는 사용자 계정 생성 명령어이다.

명령어 userdel은 사용자 계정 삭제 명령어이다.

명령어 deluser는 우분투에서 사용되는 계정 삭제 명령어이다.

## 4. 사용자 생성 및 계정관리 [예상문제]

7. 사용자 ihd의 비밀번호를 변경하는 데 사용하는 명령어는?

- ① vi /etc/passwd
- ② chage password
- ③ info ihd
- ④ passwd ihd

- ① 파일 /etc/passwd은 리눅스 계정 정보를 담은 텍스트 파일로 패스워드 자리는 X로 표시된다.
- ② 명령어 chage는 패스워드의 만료 정보를 변경한다.
- ③ 명령어 info는 특정한 명령어에 대한 매뉴얼 페이지를 표시한다.

## 4. 사용자 생성 및 계정관리 [예상문제]

8. 다음 중 사용자의 패스워드에 대한 정보를 출력하고 /etc/shadow의 날짜 관련 필드에 모두 설정할 수 있는 명령으로 알맞은 것은?

- ① chage
- ② chpasswd
- ③ /passwd
- ④ usermod

명령어 chage은 사용자의 패스워드에 대한 정보를 보여주거나 제한한다.

```
[root@localhost ~]# chage linux
linux의 사용기한 정보를 바꿉니다
새로운 값을 넣거나, 기본값을 원하시면 엔터를 치세요

암호의 최소 유효 기간 [20]:
암호의 최대 유효 기간 [8888]:
마지막으로 암호를 바꾼 날 (YYYY-MM-DD) [2024-03-18]:
암호 사용만료 예고 [7]:
암호를 사용할 수 없음 [10]:
계정 만료 날짜 (YYYY-MM-DD) [2065-10-29]:
```

## 5. 디렉터리 및 파일

### ■ 명령어 pwd

- 현재 작업 중인 디렉터리의 위치(경로)를 나타내는 명령어

```
[linux@localhost ~]$ pwd  
/home/linux
```

※ pwd = print working directory

### ■ 명령어 cd

- change directory의 약자로 디렉터를 이동할 때 사용하는 명령어

```
[linux@localhost ~]$ cd inhatc  
[linux@localhost inhatc]$
```

- 명령어 'cd ..' 는 현재 디렉터리에서 한 단계 상위 디렉터리로 이동한다.

```
[linux@localhost /]$ cd usr/local/src  
[linux@localhost src]$ cd ..  
[linux@localhost local]$ pwd  
/usr/local
```

## 5. 디렉터리 및 파일

### ■ 명령어 mkdir

- make directory의 약자로 새로운 디렉터를 생성할 때 사용하는 명령어
- (옵션) **-m** : 디렉터를 생성할 때 권한을 생성(디폴트는 755)  
-p : 상위(부모 디렉터리) 경로 생성

```
[linux@localhost ~]$ ls
공개 다운로드 문서 바탕화면 비디오 사진 서식 음악
[linux@localhost ~]$ mkdir dr1 dr2 dr3
[linux@localhost ~]$ ls
dr1 dr2 dr3 공개 다운로드 문서 바탕화면 비디오 사진 서식 음악
```

```
[linux@localhost inha25]$ mkdir -m 777 inhatac25
[linux@localhost inha25]$ mkdir -m 644 inhatac26
[linux@localhost inha25]$ ls -l
합계 0
drwxrwxrwx. 2 linux linux2 6 3월 29 15:16 inhatac25
drw-r--r--. 2 linux linux2 6 3월 29 15:16 inhatac26
```

```
[linux@localhost ~]$ mkdir -p backup/java
[linux@localhost ~]$ ls
backup dr1 dr2 dr3 공개 다운로드 문서 바탕화면
[linux@localhost ~]$ cd backup
[linux@localhost backup]$ ls
java
```

## 5. 디렉터리 및 파일

### ■ 명령어 rmdir

- remove directory의 약자로 디렉터리만 삭제하는 명령어로 디렉터리 안에 파일이 존재할 경우 삭제되지 않음

```
[linux@localhost ~]$ rmdir backup  
rmdir: failed to remove 'backup': 디렉터리가 비어있지 않음
```

- 디렉터리 안에 파일이 있어도 디렉터리를 삭제할 경우  
**rm -r dir 또는 rm -rf dir** 명령어 사용

### ■ 명령어 stty

- 키보드를 이용하여 사용하는 유용한 키(Back Space, Ctrl+C 등)를 설정하는 명령행 편집 기능 명령어 **(프롬프트 설정을 변경하는 명령어)**

### ■ 명령어 sort

- 사용자가 지정한 파일의 내용을 정렬하거나, 정렬된 파일의 내용을 병합



# 5. 디렉터리 및 파일

## ■ 명령어 ls

- 현재 위치한 디렉터리의 파일 목록들을 나타내는 명령어

(옵션) -a : 히든 파일을 포함한 모든 파일과 디렉터리 표시

-l : 크기, 사이즈 등의 자세한 정보 표시

옵션은 시험에 안 나옴

-d : 지정된 디렉터리의 정보 출력

-r : 알파벳 역순으로 정보 출력

-R : 하위 경로와 그 안에 있는 파일 표시

-F : 파일의 특성에 따라 끝에 특수문자를 덧붙여 출력

-i : 각 파일의 인덱스 값 출력

```
[linux@localhost etc]$ ls -a
```

.	cni	firewalld	krb5.conf
..	cockpit	flatpak	krb5.conf.d
.pwd.lock	containers	fonts	ld.so.cache
.updated	cron.d	foomatic	ld.so.conf

```
[linux@localhost local]$ ls -i
```

67342923	bin	33771973	include	33771974	libexec	67342931	src
100758658	etc	67342924	lib	67342925	sbin		
647223	games	100758659	lib64	100758660	share		

# 5. 디렉터리 및 파일

## ■ 명령어 ls

```
[linux@localhost sbin]$ ls -F
ModemManager*          foomatic-nonnumericalids*    lvm_import_vdo*
NetworkManager*        foomatic-preferred-driver*    lvmconfig@
accessdb*               foomatic-printermap-to-gutenprint-xml*  lvmdevices@
accton*                 foomatic-replaceoldprinterids*  lvmdiskscan@
adcli*                  fsadm*                         lvmddump*
addgnupghome*           fsck*                          lvmpolld*
addpart*                fsck.cramfs*                  lvmsadc@
adduser@                fsck.ext2*                    lvmsar@
```

```
[linux@localhost etc]$ ls -F
DIR_COLORS              cron.d/                        fonts/                        krb5.conf.d/
DIR_COLORS.lightbgcolor cron.daily/                    foomatic/                    ld.so.cache
GREP_COLORS             cron.deny                      fprintd.conf                 ld.so.conf
NetworkManager/         cron.hourly/                  fstab                         ld.so.conf.d/
```

일반파일(정규파일) : \*

디렉터리 : /

심볼릭링크 : @

## 5. 디렉터리 및 파일

### ■ 명령어 cp

#### - 파일 또는 디렉터를 복사하는 명령어

(옵션) -b : 복사 대상이 존재할 경우 백업 파일을 만들

-f : 복사 대상이 존재할 경우 복사 대상 파일을 지우고 복사

-i : 복사 대상이 존재할 경우 사용자에게 덮어씌울 것인지 유무 확인

-r : 디렉터를 복사할 경우 하위 디렉터리와 파일을 모두 복사

```
[linux@localhost java]$ ls
file1  file2
[linux@localhost java]$ cp file1 file3
[linux@localhost java]$ ls
file1  file2  file3
```

```
[linux@localhost backup]$ ls
cat1.txt  cat2.txt  dir1  dir2  total.txt
[linux@localhost backup]$ cp -r dir2 dir1
[linux@localhost backup]$ cd dir1
[linux@localhost dir1]$ ls
cat1.txt  dir2  file1
```

## 5. 디렉터리 및 파일

### ■ 명령어 rm

#### - 파일 또는 디렉터를 삭제하는 명령어

[옵션] **-f**: 질의 메시지 없이 강제로 파일이나 디렉터리 삭제

**-i**: 삭제 시 사용자에게 질의

**-r**: 삭제 디렉터리가 포함하고 있는 하위 디렉터를 포함하여 모든 파일 삭제

```
[linux@localhost dir1]$ ls
file1  file2
[linux@localhost dir1]$ rm file2
[linux@localhost dir1]$ ls
file1
```

```
[linux@localhost ~]$ rm -i *.txt
rm: remove 일반 파일 'aaa.txt'? y
rm: remove 일반 파일 'bbb.txt'? y
rm: remove 일반 파일 'inhate.txt'? n
```

## 5. 디렉터리 및 파일

### ■ 명령어 mv

원본이 없어지고 새로운 이름으로 생성

- 파일 또는 디렉터를 이동하거나 파일명을 변경할 때 사용하는 명령어  
[옵션] -b : 대상 파일이 이미 있어 지워지는 것을 대비해 백업 파일 생성  
-f : 대상 파일이 이미 있어도 사용자에게 어떻게 처리할지 묻지 않음  
-v : 파일이 옮겨지는 과정이 상세히 표시

```
[linux@localhost java]$ ls
dir1  file1
[linux@localhost java]$ mv file1 file2
[linux@localhost java]$ ls
dir1  file2
[linux@localhost java]$ mv dir1 dir2
[linux@localhost java]$ ls
dir2  file2
```

```
[linux@localhost inhate]$ mv -v bbb.txt ccc.txt
renamed 'bbb.txt' -> 'ccc.txt'
```

## 5. 디렉터리 및 파일

### ■ 명령어 touch

- 0바이트 파일을 생성하거나 파일의 시간을 변경하는 명령어

(옵션) -a : 접근시간(access time)을 변경

-m : 수정시간(modify time)을 변경

-c : 시스템 현재 시간으로 파일 시간 정보를 수정

※ 변경시간(change time) : 파일 내용이 수정되었을 때 기록되는 시간으로 변경이 불가능

```
[linux@localhost inhate]$ touch kkk.txt
[linux@localhost inhate]$ ls -l
합계 0
-rw-r--r--. 1 linux linux2 0  3월  26 00:55 aaa.txt
-rw-r--r--. 1 linux linux2 0  3월  26 00:55 ccc.txt
-rw-r--r--. 1 linux linux2 0  3월  27 06:07 kkk.txt
[linux@localhost inhate]$ touch -c aaa.txt
[linux@localhost inhate]$ ls -l
합계 0
-rw-r--r--. 1 linux linux2 0  3월  27 06:08 aaa.txt
-rw-r--r--. 1 linux linux2 0  3월  26 00:55 ccc.txt
-rw-r--r--. 1 linux linux2 0  3월  27 06:07 kkk.txt
```

# 5. 디렉터리 및 파일

## ■ 명령어 find

- 현재 디렉터리에서부터 하위 디렉터리까지 주어진 조건의 파일을 찾아 해당 경로를 표시함

[옵션] -name : 이름을 기준으로 파일 검색

-user : user의 파일이나 디렉터리 검색

-type : 파일 유형에 따라 파일 검색

-user uname : 지정한 uname의 소유로 된 모든 파일(UID로 지정 가능) 검색

-uid n : 지정된 n값의 UID를 갖는 파일 검색

-perm : 지정된 권한의 파일 검색

[예시] find /etc -name '\*.conf' : /etc에 확장자 conf 파일 검색

find -perm 777 : 허가권이 777인 파일 검색

```
[linux@localhost ~]$ find -perm 777
./.mozilla/firefox/as7m95ic.default-default/lock
./test/sk
```

find /bin -size +10k -size -100k : /bin에서 파일 크기가 10~100k를 찾음

## 5. 디렉터리 및 파일

### ■ 명령어 locate

- 파일의 위치를 찾는 명령어

```
[linux@localhost ~]$ locate *.bak  
/home/linux/list.bak  
/home/linux/list.bak2
```

### ■ 명령어 cat

- 파일의 내용을 출력하는 명령어

[옵션] -n : 각 문장 앞에 번호 표시, 비어있는 행 포함  
-b : 각 문장 앞에 번호 표시, 비어있는 행 제외

```
[linux@localhost ~]$ cat -n inhatac.txt  
1 inhatac 1958  
2 enterprise server management  
3 linux
```



## 5. 디렉터리 및 파일

### ■ 명령어 head

– 지정된 파일의 맨 앞부분부터 내용을 출력하는 명령어

- `head -n 5 / etc/passwd` passwd 파일을 5줄씩 보여줌
- `head -3 /etc/*.conf` 확장자가 conf 인 파일을 3줄씩 보여줌

```
[linux@localhost ~]$ head -n 5 /etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
```

```
bin:x:1:1:bin:/bin:/sbin/nologin
```

```
daemon:x:2:2:daemon:/sbin:/sbin/nologin
```

```
adm:x:3:4:adm:/var/adm:/sbin/nologin
```

```
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

```
[linux@localhost ~]$ head -3 /etc/*.conf
```

```
==> /etc/appstream.conf <==
```

```
#
```

```
# This is the configuration file for AppStream.
```

```
# If data for your distribution is missing, you can submit
```

```
==> /etc/asound.conf <==
```

```
#
```

```
# Place your global alsa-lib configuration here...
```

```
#
```

# 5. 디렉터리 및 파일

## ■ 명령어 tail

### - 지정된 파일의 마지막 부분을 출력하는 명령어

[옵션] -숫자 : 출력을 원하는 줄 수 지정 (지정하지 않으면 10개씩 출력)

-f : 로그 파일처럼 실시간으로 바뀌는 정보를 확인

- **tail -f /var/log/messages** 로그 메시지와 같이 수시로 변하는 파일을 계속 감시 (관리자 권한 명령)

```
[linux@localhost log]$ su
암호 :
[root@localhost log]# tail -f /var/log/messages
Mar 24 16:47:43 localhost NetworkManager[956]: <info> [1711266463.9924] dhcp4 (ens160): state changed new lease, address=192.168.168.128
Mar 24 16:47:44 localhost systemd[1]: Starting Network Manager Script Dispatcher Service...
Mar 24 16:47:44 localhost systemd[1]: Started Network Manager Script Dispatcher Service.
Mar 24 16:47:54 localhost systemd[1]: NetworkManager-dispatcher.service: Deactivated successfully.
Mar 24 16:51:43 localhost gsd-color[2270]: unable to get EDID for xrandr-Virtual-1: unable to get EDID for output
Mar 24 16:51:43 localhost gsd-color[2270]: unable to get EDID for xrandr-Virtual-1: unable to get EDID for output
Mar 24 16:51:51 localhost gnome-shell[2056]: Source ID 10460 was not found when attempting to remove it
Mar 24 16:57:10 localhost systemd[1]: Starting Fingerprint Authentication Daemon...
Mar 24 16:57:10 localhost systemd[1]: Started Fingerprint Authentication Daemon.
Mar 24 16:57:14 localhost su[13123]: (to root) linux on pts/0
Mar 24 16:57:40 localhost systemd[1]: fprintd.service: Deactivated successfully.
Mar 24 17:02:44 localhost NetworkManager[956]: <info> [1711267364.0198] dhcp4 (ens160): state changed new lease, address=192.168.168.128
Mar 24 17:02:44 localhost systemd[1]: Starting Network Manager Script Dispatcher Service...
Mar 24 17:02:44 localhost systemd[1]: Started Network Manager Script Dispatcher Service.
Mar 24 17:02:54 localhost systemd[1]: NetworkManager-dispatcher.service: Deactivated successfully.
```

로그 메시지 보는 것을 종료하려면 **Ctrl + C**를 누름

## 5. 디렉터리 및 파일

### ■ 명령어 more

more 명령의 단점은 뒤로만 가고 다시 앞으로 back 할 수 없음

- 파일의 내용을 화면 단위로 출력 (한 화면을 보여주고 일단 멈춤)
- 한 화면을 모두 보여줄 수 없을 때 '--More--(69%)' 와 같은 메시지가 보임

Space Bar 를 누르면 다음 화면으로 이동

Enter ↵ 를 누르면 한 라인씩 이동

```
[linux@localhost ~]$ ls -al | more
```

'ls -al' 결과를 표준 출력 장치인 화면에 출력 시 화면 단위로 끊어서 표시

```
[linux@localhost ~]$ more -10 +1 /etc/profile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

--More--(20%)
```

10라인 단위로 보여주며, 1번째 라인부터 시작함

## 5. 디렉터리 및 파일

### ■ 명령어 less      작은(less) 것이 아름답다는 의미를 가지고 있음

- more 명령의 개선된 버전
- 위로도 이동 가능하며, 화살표키로 상하좌우로 이동



```
[linux@localhost ~]$ less /etc/services
```

```
msg-auth      31/udp      # MSG Authentication
dsp           33/tcp      # Display Support Protocol
dsp           33/udp      # Display Support Protocol
graphics      41/tcp      # Graphics
graphics      41/udp      # Graphics
mpm-flags     44/tcp      # MPM FLAGS Protocol
mpm-flags     44/udp      # MPM FLAGS Protocol
mpm           45/tcp      # Message Processing Module [recv]
mpm           45/udp      # Message Processing Module [recv]
mpm-snd       46/tcp      # MPM [default send]
:
```

## 5. 디렉터리 및 파일 [예상문제]

1. 파일 또는 디렉터리와 관련된 명령으로 거리가 먼 것은?

- ① ps
- ② rm
- ③ ls
- ④ cp

명령어 ps는 현재 실행되고 있는 프로세스들의 상태 정보를 표시한다.

# 5. 디렉터리 및 파일 [예상문제]

## 2. pwd 명령에 대한 설명으로 알맞은 것은?

- ① 사용자의 홈 디렉터리를 알기 위한 명령이다.
- ② 디렉터리에 존재하는 파일의 개수를 알기 위한 명령이다.
- ③ 사용자의 password를 수정하기 위한 명령이다.
- ④ **작업중인 디렉터리의 경로를 알기 위한 명령이다.**

- ① `cat /etc/passwd | grep 사용자명` : 사용자의 홈 디렉터리를 확인하기 위한 명령
- ② `ls -R 디렉터리명 | wc -l` : 디렉터리에 존재하는 파일의 개수를 알기 위한 명령
- ③ `passwd` : 사용자의 password를 수정하기 위한 명령

```
[root@localhost ~]# cat /etc/passwd | grep jjpark
jjpark:x:1001:1001:Jinju Park:/home/jjpark:/bin/bash
[root@localhost ~]# ls -R usr/bin | wc -l
1517
[root@localhost ~]# passwd
root 사용자의 비밀번호 변경 중
새 암호 :
```

```
[linux@localhost home]$ pwd
/home
```

## 5. 디렉터리 및 파일 [예상문제]

3. 다음 중 파일이나 디렉터리의 이름을 변경하는 명령으로 알맞은 것은?

① touch

② cat

③ mv

④ file

① touch : 0바이트 파일을 생성하거나 파일의 시간을 변경하는 명령어

② cat : 파일의 내용을 확인할 때 사용하는 명령어

④ file : 파일의 종류를 확인할 때 사용하는 명령어

```
[linux@localhost java]$ ls
dir1  file1
[linux@localhost java]$ mv file1 file2
[linux@localhost java]$ ls
dir1  file2
[linux@localhost java]$ mv dir1 dir2
[linux@localhost java]$ ls
dir2  file2
```

## 6. 권한 및 그룹 설정

### ■ 명령어 **chown** (change owner)

- root 사용자가 파일이나 디렉터리의 소유자 또는 그룹을 변경하는 명령  
: 소유자는 파일을 생성한 사용자이고, 그룹은 소유자가 속한 그룹으로 기본 설정
- 형식 : **chown** [옵션] newowner 파일명  
[옵션] -R : 해당 디렉터리 아래의 모든 파일과 서브 디렉터리에 대하여 적용됨

**newowner**

→ 소유자만 변경

**:newowner**

→ 그룹만 변경

**newowner:**

→ 모두 변경

```
[root@localhost dir1]# ls -l
합계 0
-rwxrwxr-x. 1 root root 0  3월  24 12:24 file1
-r---w----. 1 root root 0  3월  24 12:49 file2
-rw-r--r--. 1 root root 0  3월  24 13:43 file3
[root@localhost dir1]# chown linux file1  소유자
[root@localhost dir1]# chown linux: file2  모두
[root@localhost dir1]# chown :linux file3  그룹
[root@localhost dir1]# ls -l
합계 0
-rwxrwxr-x. 1 linux root 0  3월  24 12:24 file1
-r---w----. 1 linux linux 0  3월  24 12:49 file2
-rw-r--r--. 1 root linux 0  3월  24 13:43 file3
```



## 6. 권한 및 그룹 설정

- 명령어 **chgrp** (change group)

- root 사용자가 파일이나 디렉터리의 그룹을 변경하는 명령

- : 그룹은 소유자가 속한 그룹으로 기본 설정

- 형식 : chgrp [옵션] newgroup 파일명

- [옵션] -R : 해당 디렉터리 아래의 모든 파일과 서브 디렉터리에 대하여 적용됨

- 'chown [옵션] :newgroup 파일명'과 동일함

```
[root@localhost esm]# ls -l
합 계 0
-rw-r--r--. 1 linux linux2 0  4월  3 03:41 aaa.txt
-rw-r--r--. 1 linux linux2 0  4월  3 03:43 bbb.txt
[root@localhost esm]# chown :root aaa.txt
[root@localhost esm]# chgrp root bbb.txt
[root@localhost esm]# ls -l
합 계 0
-rw-r--r--. 1 linux root 0  4월  3 03:41 aaa.txt
-rw-r--r--. 1 linux root 0  4월  3 03:43 bbb.txt
```

## 6. 권한 및 그룹 설정

### ■ 파일의 종류

```
$ ls -l
합계 56
-rw-rw-r--. 1 kdhong kdhong 786  12월  2 18:43 cat2.txt
... ..
```

파일의 종류 - : 일반파일, d : 디렉터리, l : 심볼릭링크, b : 블록디바이스 파일, c : 문자디바이스 파일, s : 소켓 파일

```
[linux@localhost run]$ cd /var/run
[linux@localhost run]$ ls -l mc*
srwxr-xr-x. 1 root root 0  4월  3 03:24 mcelog-client
-rw-r--r--. 1 root root 3  4월  3 03:24 mcelog.pid
```

- \* 심볼릭링크 : 원본 파일을 가리키는 링크 (윈도우의 바로가기와 비슷)
- \* 디바이스 파일 : 하드디스크, 주변장치를 파일로 다루기 위함
- \* 소켓 파일 : 프로세스 간 통신(IPC)이나 네트워크 통신을 위해 사용

## 6. 권한 및 그룹 설정

### ■ 접근권한

– 파일이나 디렉터리에 권한이 있는 사용자만 접근하기 위함

: 소유자, 그룹, 기타사용자의 부류에 따라 읽기(r), 쓰기(w), 실행(x) 권한을 설정함

#### ① 파일

읽기(r) : 파일의 내용 보기

쓰기(w) : 파일의 내용 수정과 삭제 또는 이름 바꾸기

실행(x) : 파일을 실행하기

#### ② 디렉터리

읽기(r) : 디렉터리가 가진 파일과 서브 디렉터리의 목록 보기

쓰기(w) : 디렉터리를 만들거나 삭제하기

실행(x) : 디렉터리 내부로 접근하거나 디렉터리에서 프로그램을 실행하기

```
$ ls -ld /home ~jjpark
```

```
drwxr-xr-x. 8 root root 20 12월 3 16:27 /home
```

```
drwx-----. 26 jjpark jjpark 4096 12월 3 16:27 /home/jjpark
```

소유자 그룹 기타

## 6. 권한 및 그룹 설정

### ■ 명령어 **chmod** (change mode)

- 파일의 소유자가 파일의 접근권한(허가권=Permission)을 변경하는 명령  
: 접근 권한에는 읽기(r), 쓰기(w), 실행(x) 이 있음

- 형식 : **chmod** [옵션] mode 파일명

[옵션] -R : 해당 디렉터리 아래의 모든 파일과 서브 디렉터리까지 권한을 변경

#### ① 8진수 모드

읽기(r)는 4, 쓰기(w)는 2, 실행(x)은 1

예) **rwxr-xr-x**는 3개씩 구분하면 **rwX r-x r-x**이며

하이픈(-)만 0이고 **rwX**는 1로 변경하면 **111 101 101** 이고 8진수는 **755**

• **chmod -R 755 dir1**

#### ② 기호 모드

형식 : **[ugoa][+--=][rwx]**    +와 =의 차이점 알기

**ugoa**는 순서대로 **user, group, other, all**을 의미하며 생략하면 all

**+--=**는 순서대로 권한의 **추가, 제거, 지정**을 의미함

• **chmod u+x file1**

• **chmod u=rwx,g=rx,o=x file2**

## 6. 권한 및 그룹 설정

- **chmod** 사용 예시

```
[linux@localhost dir1]$ ls -l
합계 0
-rw-r--r--. 1 linux linux 0  3월  24 08:14 file1
-rwxr-xr--. 1 linux linux 0  3월  24 08:14 file2
[linux@localhost dir1]$ cd ..
[linux@localhost backup]$ chmod -R 755 dir1
[linux@localhost backup]$ cd dir1
[linux@localhost dir1]$ ls -l
합계 0
-rwxr-xr-x. 1 linux linux 0  3월  24 08:14 file1
-rwxr-xr-x. 1 linux linux 0  3월  24 08:14 file2
```

## 6. 권한 및 그룹 설정

- **chmod** 사용 예시

```
[linux@localhost dir1]$ ls -l
```

합계 0

```
-rw-rw-rw-. 1 linux linux 0 3월 24 08:14 file1
```

```
-rw-rw-rw-. 1 linux linux 0 3월 24 08:14 file2
```

```
[linux@localhost dir1]$ chmod u+x file1
```

```
[linux@localhost dir1]$ chmod u=rwx,g=rx,o=x file2
```

```
[linux@localhost dir1]$ ls -l
```

합계 0

```
-rwxrw-rw-. 1 linux linux 0 3월 24 08:14 file1
```

```
-rwxr-x--x. 1 linux linux 0 3월 24 08:14 file2
```

## 6. 권한 및 그룹 설정

### ■ 명령어 **umask** (User File Creation Mask)

- 파일 또는 디렉터리가 생성될 때 기본적으로 파일 권한을 설정하는 값  
: 보통 **/etc/bashrc**에 설정되어 있음

```
[linux@localhost run]$ cat -n /etc/bashrc
```

```
79      [ `umask` -eq 0 ] && umask 022
```

권한  
파일은 666  
디렉터리는 777

#### ➔ **umask** [-S] [*mask*]

- 단순히 **umask** (또는 **umask -S**) 명령을 실행하면 현재 기본값을 알 수 있음
- **-S** 옵션은 기호 모드로 결과를 보여줌
- **umask 002**
  - 마스크 값이 **002**라는 것은 ----- **w-**,  
즉 기타 사용자에게 쓰기 권한을 부여하지 않겠다는 것
  - 결국 마스크 값이 **002**라는 것은 디렉터리에 **775**라는 접근권한을 주는 것
  - 파일의 경우 실행 권한은 부여되지 않으며 접근권한은 **664**가 됨
- 참고로 **touch file** 명령은 파일의 접근/수정 시간을 현재 시간으로 변경하며  
파일이 존재하지 않으면 파일을 생성함

## 6. 권한 및 그룹 설정

- **umask** 사용 예시

```
[linux@localhost etc]$ umask
0022
[linux@localhost etc]$ umask -S
u=rwx,g=rx,o=rx
```

```
[linux@localhost esm]$ touch file_test.txt
[linux@localhost esm]$ mkdir directory_test
[linux@localhost esm]$ ls -l
합계 0
drwxr-xr-x. 2 linux linux2 6  4월  3 05:37 directory_test
-rw-r--r--. 1 linux linux2 0  4월  3 05:36 file_test.txt
```



## 6. 권한 및 그룹 설정

### ■ 특수권한 SetUID

- 일반 사용자가 **passwd** 명령으로 비밀번호를 수정하면, 결국 **/etc/passwd** 또는 **/etc/shadow** 파일이 수정되어야 함(권한 상승이 필요함)
- 실행 파일에 **SetUID**가 설정되어 있으면, 해당 파일이 실행될 때 '실행시킨 사용자'가 아닌 '파일의 소유자' 권한으로 실행됨
  - 프로세스의 **EUID**는 파일의 소유자가 됨.  
Effective User ID : 일시적으로 파일 소유자의 권한을 얻게 되는 것
- 파일의 접근권한에서 소유자의 실행 권한에 **x**가 아닌 **s**가 표시됨

```
$ ls -l /etc/passwd
```

```
-rw-r--r--. 1 root root 2161  8월 11 01:51 /etc/passwd
```

```
$ ls -l /usr/bin/passwd
```

 사용자가 로그인 할 때 참조하는 파일로 Set-UID가 부여됨

```
-rwsr-xr-x. 1 root root 32656  5월 15  2022 /usr/bin/passwd
```

소유자 권한으로 프로세스 수행

- 설정 방법은 **chmod 4755 file** 또는 **chmod u+s file**
- 해제 방법은 **chmod 0755 file** 또는 **chmod u-s file**

## 6. 권한 및 그룹 설정

### ■ 특수권한 SetGID

- 파일의 접근권한에서 **그룹의 실행 권한에 s가 표시됨**
- **SetGID**가 설정된 파일을 실행하면 '실행자의 그룹 권한'이 아닌 '파일 소유자의 그룹 권한'으로 실행됨     **/bin/su** : 사용자 전환에 사용되며 **Set-UID** 또는 **Set-GID** 가 부여됨
- 디렉터리에 설정된 경우, 그 디렉터리에 파일을 만들 때 '파일의 소유 그룹'이 '디렉터리의 소유 그룹'으로 지정됨. 그룹 구성원의 공유 공간으로 사용될 수 있음
- 설정 방법은 **chmod 2775 file** 또는 **chmod g+s file**

```
[root@localhost esm]# ls -l /etc/passwd
-rw-r--r--. 1 root root 3089  3월  29 14:12 /etc/passwd
[root@localhost esm]# chmod u+s /etc/passwd
[root@localhost esm]# ls -l /etc/passwd
-rwSr--r--. 1 root root 3089  3월  29 14:12 /etc/passwd
[root@localhost esm]# chmod g+s /etc/passwd
[root@localhost esm]# ls -l /etc/passwd
-rwSr-Sr--. 1 root root 3089  3월  29 14:12 /etc/passwd
```

## 6. 권한 및 그룹 설정

### ■ 특수권한 StickyBit

- 파일의 접근권한에서 기타 사용자의 실행 권한에 t가 표시됨
- 주로 공유 디렉터리에 설정되며, 누구나 그 디렉터리에 파일을 생성할 수 있지만  
파일의 소유자만 삭제할 수 있음 본인 소유의 파일만 삭제 가능
- 설정 방법은 **chmod 1777 file** 또는 **chmod o+t file**

```
$ ls -ld /tmp
```

```
drwxrwxrwt. 18 root root 4096  9월 11 11:57 /tmp
```

 임시 디렉터리로 Sticky Bit가 부여

```
[root@localhost linux]# ls -ld /tmp
```

```
drwxrwxrwt. 24 root root 4096  5월  6 00:41 /tmp
```

## 6. 권한 및 그룹 설정 [예상문제]

1. 다음 파일의 소유 그룹을 ihd로 변경하고자 할 때 (        ) 안에 들어갈 명령으로 알맞은 것은?

```
[root@www~]# (        ) :ihd test.txt
```

- ① chown
- ② chmod
- ③ chgrp
- ④ csh

그룹을 변경하고자 할 때 형식은 'chown [옵션][:그룹명]파일명'이다.  
그룹명 앞에 콜론(:)을 붙여 사용자 소유권과 그룹 소유권을 구분한다.

```
# chown linux file1   소유자
# chown linux: file2  모두
# chown :linux file3  그룹
```

## 6. 권한 및 그룹 설정 [예상문제]

2. 명령어 'ls -l'을 실행한 결과에 대한 설명으로 알맞은 것은?

```
srw-r--r-- 1 root root 0 Aug 8 13:41 control
```

- ① 소켓 파일을 의미한다.
- ② 입출력에 사용되는 특수 파일을 의미한다.
- ③ 심볼릭 링크 파일을 의미한다.
- ④ 블록 구조의 특수파일을 의미한다.

-는 일반파일, d는 디렉터리, l는 심볼릭링크, s는 소켓 파일, b는 블록디바이스 파일, c는 문자디바이스 파일이다.

## 6. 권한 및 그룹 설정 [예상문제]

3. 다음 중 허가권(Permission)에 대한 설명으로 틀린 것은?

- ① 파일의 내용을 볼 수 있는 권한 표시는 r을 사용한다.
- ② 실행 파일을 실행시킬 수 있는 권한 표시는 x를 사용한다.
- ③ 디렉터리 안에 파일을 생성 또는 삭제할 수 없는 권한 표시는 w를 사용한다.
- ④ 디렉터리 내부로 접근할 수 있는 권한 표시는 x를 사용한다.

파일을 생성 또는 삭제할 수 없는 권한 표시는 하이픈(-)을 사용한다.

## 6. 권한 및 그룹 설정 [예상문제]

4. 다음 중 보안상 가장 안전한 umask 설정 값으로 알맞은 것은?

- ① umask 000
- ② umask 066
- ③ umask 077
- ④ umask 022

umask 077로 설정하면 디렉터리 권한이 700(777-077)이 되어 소유자만 모든 권한을 갖게되고, 그룹과 기타 사용자들에게는 어떠한 권한도 주지 않게 된다.

## 6. 권한 및 그룹 설정 [예상문제]

5. 다음 중 Sticky-Bit가 설정된 디렉터리로 알맞은 것은?

- ① /root
- ② /tmp
- ③ /usr
- ④ /etc

디렉터리 tmp는 모든 사용자들이 공동으로 사용하는 디렉터리 특성을 갖고 있기 때문에 sticky-bit의 공용 디렉터리 설정 시 사용한다.



## 6. 권한 및 그룹 설정 (예상문제)

6. 다음 중 data 디렉터리를 공유 모드로 설정할 때 명령으로 알맞은 것은?

- ① `chmod o+s data/`
- ② `chmod u+s data/`
- ③ `chmod o+t data/`
- ④ `chmod g+s data/`

권한 기호 't'는 sticky-bit 설정을 의미한다. 따라서 해당 디렉터리 data는 공유 디렉터리 설정을 의미한다.

## 6. 권한 및 그룹 설정 [예상문제]

7. /project 디렉터리에 Set-GID를 설정 시 (        ) 안에 들어갈 옵션으로 알맞은 것은?

```
# chmod (        ) /project
```

- ① g+s
- ② a+r
- ③ g+t
- ④ o+t

Set-GID는 프로그램을 실행하는 동안 프로세스는 파일의 그룹과 같은 권한으로 실행된다.  
Set-GID의 설정 코드는 대문자 S 또는 소문자 s이다.

# 7. 파일시스템 관리

## ■ 파일시스템

- 저장장치를 디렉터리 구조로 조직화하고 파일에 이름을 부여하는 등의 파일의 저장과 검색을 위한 체계
  - : 파일에 파일명과 경로를 부여하여 저장이나 검색을 위해 파일을 구분하고 식별하기 위한 방법을 제공
- 파일시스템의 종류에 따라 파일시스템의 크기, 파일 이름의 길이, 파일의 크기, 파일의 총 개수, 파일의 복구, 성능/보안/유연성 등에 차이가 남

# 7. 파일시스템 관리

## ■ 파일시스템의 구조

용량이 크거나 지속적인 데이터 증가가 예상되는 테이블에서 데이터를 여러 개의 작은 단위로 나누므로 성능 저하를 방지하고 관리를 편이하게 하는 방법



해당 파일 시스템 관련 정보(블록의 크기, 블록의 개수, 블록 그룹의 개수, inode의 개수)를 저장



파일 하나의 정보가 레코드로 구성되어 있으며 파일 이름, 소유자, 권한, 시간, 디스크에서의 위치 등에 대한 정보를 담고 있음

디렉터리별로 디렉터리 엔트리와 실제 파일에 대한 데이터가 저장

# 7. 파일시스템 관리

## ■ 파일시스템의 종류

### ① ext1

- 레미 카드가 개발한 것으로, 1992년 4월 리눅스 커널 0.96c에 포함되어 발표
- ext는 'Extended File System'의 약자로 extfs라고도 하며, MFS의 기능을 확장했다는 의미가 있음
- MFS의 경우 16비트 주소 체계를 사용하여 파일 시스템의 최대 크기가 64MB였지만 ext는 2GB로 늘어났고, 파일 이름의 길이도 MFS는 14B인데 비해 ext는 유닉스처럼 255B까지 지원했음
- ext 파일 시스템에는 inode의 수정과 데이터의 수정 시간 지원이 안 되는 문제, 링크드 리스트로 파일 시스템을 구성하여 파일 시스템이 복잡해지고 파편화되는 문제가 있었음
- 현재 리눅스에서는 ext 파일 시스템을 사용하지 않음

# 7. 파일시스템 관리

## ■ 파일시스템의 종류

### ② ext2

- ext2 파일 시스템의 공식 명칭은 ‘Second Extended File System(Ext2 fs)’
- ext 파일 시스템의 문제를 해결하기 위해 내놓은 것으로, 역시 레미 카드가 개발하여 1993년 1월에 공개했음
- ext2 파일 시스템은 ext 파일 시스템을 기반으로 확장성도 고려하여 만들었음
- 리눅스 커널 2.6.17 이전 버전에서는 ext2 파일 시스템의 최대 크기가 2TB로 제한되었으나 현재는 이론적으로 32TB까지 지원됨
- ext2는 ext3 파일 시스템이 도입되기 전까지 사실상 리눅스의 표준 파일 시스템으로 사용되었음
- 지금도 부팅 가능한 USB 플래시 드라이브와 기타 SSD 장치에 사용되고 있음

# 7. 파일시스템 관리

## ■ 파일시스템의 종류

### ③ ext3 [저널링 기술 적용]

- ext3 파일 시스템은 스티븐 트위디가 개발하여 2001년 11월 리눅스 커널 2.4.15에 추가되었음
- ext3는 ext2를 기반으로 개발되어 호환이 가능. ext2 파일 시스템의 파일을 별도의 변경 없이 ext3 파일 시스템에 바로 이식할 수 있음
- 가장 큰 장점은 저널링 기능을 도입했다는 것. 저널링은 디스크에 기록되는 데이터의 복구 기능을 강화한 것. 데이터를 디스크에 기록하기 전에 먼저 저널에 수정 사항을 기록함. 저널은 로그 기록 같은 것
- 디스크에 데이터를 기록하기 전에 갑자기 전원이 나가거나 시스템에 충돌이 발생할 경우, 이 저널의 기록을 보고 빠르게 복구할 수 있음
- ACL(Access Control List)를 통한 접근 제어를 지원
- 단점은 inode의 동적 할당이나 다양한 블록 크기와 같은 최신 파일 시스템 기능이 부족하고, 온라인 조각 모음 기능이 없다는 것. 파일 시스템의 최대 크기는 블록의 크기에 따라 2~32TB까지 지원됨

# 7. 파일시스템 관리

## ■ 파일시스템의 종류

### ④ ext4 [저널링 기술 적용]

- ext4 파일 시스템 안정판은 2008년 12월 25 일 리눅스 커널 2.6.28에 포함되어 공개되었음
- 1EB[엑사바이트,  $1\text{EB}=1,024 \times 1,024\text{TB}$ ] 이상의 볼륨과 16TB 이상의 파일을 지원하며, ext2 및 ext3와 호환성을 유지하고 있음
- ext3에서는 서브 디렉터리의 수가 32,000개로 제한되었으나 ext4에서는 64,000개로 늘어났으며, 온라인 조각 모음 기능도 지원함

### ⑤ xfs

- 1993년 실리콘 그래픽스가 개발한 고성능 저널링 파일 시스템으로, 2000년 5월 GNU GPL로 공개됨
- 2001년 리눅스에 이식되었고 현재 대부분의 리눅스 배포판에서 지원하고 있음
- 64비트 파일 시스템으로 최대 16EB까지 지원
- 로키 리눅스는 XFS 파일 시스템을 기본으로 사용하고 있음



# 7. 파일시스템 관리

## ■ 저널링 기술의 특징

- ① 데이터를 디스크에 쓰기 전에 별도의 로그에 데이터를 남겨 놓는 기술
- ② 저널링 기술이 적용된 파일시스템은 ext3, ext4, XFS, JFS, ResierFS 등이 있음
  - . XFS(eXetended File Syatem) : 고성능 저널링. SGI에서 개발. 신속한 복구 기능
  - . JFS(Journaling File Syatem) : IBM사에서 개발한 저널링 파일시스템
  - . ReiserFS : 독일의 한스라이저가 개발. 리눅스용 파일시스템 중에 가장 안정적
- ③ 전원공급 문제나 시스템 오류와 같은 상황에 복구가 가능함
- ④ fsck(파일시스템 무결성 검사도구) 로 복구하는 것 보다 속도가 빠르고,  
복구의 안정성도 뛰어남 ※ fsck : 파일 시스템을 검사하고 수리

저널링 기술은 기존의 fsck에 걸리는 시간을 단축하기 위해 데이터를 디스크에 쓰기 전에 로그(log)에 데이터를 남겨 시스템의 비정상적인 종료에도 로그를 사용해 빠르고 안정적인 복구 기능을 제공하는 기술

## 7. 파일시스템 관리

### ■ 명령어 mount

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

- 특정 디바이스를 특정 디렉터리처럼 사용하기 위해 장치와 디렉터리를 저장

[형식] mount [옵션] [디바이스명] [디렉터리명]

[옵션] -a : /etc/fstab 파일에 정의된 모든 시스템을 마운트

-n : /etc/mtab 파일에 정보를 저장하지 않고 마운트

-f : 실제 마운트를 하는 것이 아니라 마운트가 가능한지를 테스트

- 파일 /etc/mtab은 현재 마운트된 블록 시스템 정보를 표시

### ■ 명령어 unmount

```
# unmount /mnt
```

- 마운트를 해제함

[형식] unmount [옵션] [디바이스명] [디렉터리명]

[옵션] -a : /etc/mtab 파일에 정의된 모든 시스템을 마운트 해제

-n : /etc/mtab 파일을 갱신하지 않고 마운트 해제

-t : unmount할 파일시스템을 지정

# 7. 파일시스템 관리

```
[linux@localhost ~]$ cat /etc/fstab
```

```
#
# /etc/fstab
# Created by anaconda on Sat Mar  9 20:56:49 2024
#
# 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.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rl-root      /                    xfs      defaults        0 0
UUID=5e9655ad-8432-4623-9f74-0fb8906882a2 /boot                xfs      defaults        0 0
/dev/mapper/rl-swap      none                 swap     defaults        0 0
```

```
[linux@localhost ~]$ cat /etc/mtab
```

```
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
sysfs /sys sysfs rw,seclabel,nosuid,nodev,noexec,relatime 0 0
devtmpfs /dev devtmpfs rw,seclabel,nosuid,size=4096k,nr_inodes=456259,mode=755,inode64 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
tmpfs /dev/shm tmpfs rw,seclabel,nosuid,nodev,inode64 0 0
devpts /dev/pts devpts rw,seclabel,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000 0 0
tmpfs /run tmpfs rw,seclabel,nosuid,nodev,size=742620k,nr_inodes=819200,mode=755,inode64 0 0
cgroup2 /sys/fs/cgroup cgroup2 rw,seclabel,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot 0 0
```

# 7. 파일시스템 관리

- **mount의 -t 옵션은 파일시스템의 유형을 결정**
  - . vfat(Virtual File Allocation Table) : 마이크로소프트 파일시스템 FAT32를 마운트
  - . ext2, ext3, ext4 : 리눅스 파일시스템 마운트
  - . iso9660 : CD-ROM이나 DVD 마운트
  - . smbfs : 삼바 파일시스템 마운트
  - . nfs : 네트워크 파일시스템의 공유된 영역을 마운트
- **mount의 -o 옵션은 추가 설정**
  - . ro : 읽기 전용으로 마운트
  - . r/w : 읽기/쓰기 모드로 마운트
  - . loop : Loop 디바이스나 CD-ROM 이미지 파일 iso 마운트
  - . remount : 파티션을 재마운트
  - . noatime : 파일이 변경되기 전까지 access time이 변경되지 않음
  - . sync : 파일시스템에 대한 입출력을 동기화
  - . user : 일반 사용자가 마운트

-t 뒤에 파일시스템 온다는 것만 알아두기  
- noatime은 시험에 나올 수 있음

## 7. 파일시스템 관리

### ▪ 다양한 장치 mount의 예

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

# 7. 파일시스템 관리

## ■ 명령어 fdisk

- 새로운 파티션의 생성, 기존 파티션의 삭제, 파티션의 타입 결정 등의 작업을 수행하며, 한번에 한 디스크에 대해서만 작업을 수행
- 1대의 서버에 여러 개의 하드디스크가 설치될 수 있으므로 어떤 디스크의 파티션을 변경할 것인지 알려 주어야 함

[형식] fdisk [옵션] [장치명]

[옵션] -v : fdisk 버전 정보 표시

-l : 현재 디스크의 파티션 테이블 정보 표시

-s : 지정된 파티션의 크기를 블록 단위로 표시

[주요 명령어] p : 디스크 정보 표시

n : 파티션 생성

t : 파티션 속성(Swap, RAID 등) 지정

d : 파티션 삭제

w : 변경된 파티션 정보 저장

q : 파티션 설정 작업 종료

## 7. 파일시스템 관리

```
[root@localhost linux]# fdisk -v
fdisk from util-linux 2.37.4
```

```
[root@localhost linux]# fdisk -l
```

**Disk /dev/nvme0n1: 40 GiB, 42949672960 bytes, 83886080 sectors**

Disk model: VMware Virtual NVMe Disk

Units: sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

Disklabel type: dos

Disk identifier: 0x4b175097

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/nvme0n1p1	*	2048	2099199	2097152	1G	83	Linux
/dev/nvme0n1p2		2099200	83886079	81786880	39G	8e	Linux LVM

**Disk /dev/mapper/rl-root: 35.09 GiB, 37673238528 bytes, 73580544 sectors**

Units: sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

I/O size (minimum/optimal): 512 bytes / 512 bytes

# 7. 파일시스템 관리

## ■ 명령어 mkfs

- fdisk로 하드디스크 파티션을 나눈 후 해당 파티션에 맞는 파일시스템 생성
- 해당 파티션의 마운트 정보와 파일 시스템 정보를 확인

[형식] mkfs [옵션] 장치명

[옵션] -V : 자세한 정보 보기

-t 파일시스템 : 생성할 파일시스템 타입(ext2, ext3, ext4 등)을 지정

-c : 파일시스템을 생성하기 전에 배드블록(Bad Block)을 검사

-i 파일명 : 지정된 파일명으로부터 배드블록(Bad Block) 목록 읽기

-v : 작업상태와 결과를 자세히 보기

```
[root@localhost linux]# mkfs -V
mkfs from util-linux 2.37.4
[root@localhost linux]# mkfs -c
Usage: mkfs.ext2 [-c|-l filename] [-b block-size] [-C cluster-size]
        [-i bytes-per-inode] [-I inode-size] [-J journal-options]
        [-G flex-group-size] [-N number-of-inodes] [-d root-directory]
        [-m reserved-blocks-percentage] [-o creator-os]
        [-g blocks-per-group] [-L volume-label] [-M last-mounted-directory]
        [-O feature[,...]] [-r fs-revision] [-E extended-option[,...]]
        [-t fs-type] [-T usage-type] [-U UUID] [-e errors_behavior][-z undo_file]
        [-jnvDFSV] device [blocks-count]
```



## 7. 파일시스템 관리

### ■ 명령어 mke2fs

- ext2, ext3, ext4 타입의 리눅스 파일시스템을 생성하는 명령어
- fdisk로 파티션 작업을 한 후에 mke2fs 또는 mkfs 명령어로 파일시스템을 생성해야 함

[형식] mke2fs [옵션] 장치명

[옵션] -t : 파일시스템 타입 지정(ext2, ext3, ext4 등)

-b : 블록 크기를 바이트 수로 지정

-f : 프래그먼트 크기 지정

-i : inode 당 바이트 수를 지정

-j : 파티션을 저널링 파일시스템 ext3으로 지정

-R : RAID4 장치를 포맷할 때 사용하는 특수 옵션

[예시] 파티션 /dev/sda2을 파일 시스템 ext3로 생성

```
# mke2fs -j /dev/sda2
```

```
# mkfs.ext3 /dev/sda2
```

```
# mke2fs -t ext3 /dev/sda2
```

```
# mkfs -t ext3 /dev/sda2
```

## 7. 파일시스템 관리

### ■ Mke2fs 명령 사용 예제

- mke2fs 명령으로 /dev/nvme0n3p2 파티션에 ext4 파일 시스템을 생성
- 블록 크기는 4,096B로 지정함

```
[root@localhost ~]# mke2fs -t ext4 -b 4096 /dev/nvme0n3p2
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 128000 4k blocks and 128000 inodes
Filesystem UUID: d82f3ae6-9b0b-4c8c-a974-5a91234315dd
Superblock backups stored on blocks:
    32768, 98304

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

# 7. 파일시스템 관리

## ■ 명령어 fsck

- 파일시스템의 무결성을 점검하고 대화식으로 복구하는 명령어
- 디렉터리 /lost+found는 fsck에서 사용하는 디렉터리
- fsck 명령은 손상된 디렉터리나 파일을 수정할 때 임시로 /lost+found 디렉터리에 작업을 수행하고 정상적인 복구가 되면 사라짐

[형식] fsck [옵션] 장치명

[옵션] -A : /etc/fstab에서 모든 파일시스템을 점검

-a : 질의 없이 자동 복구 (오류 발견 시 자동으로 복구를 시도)

-r : 질의 후 복구 (복구 시도 전에 확인을 요청)

-s : fsck 동작을 시리얼화, 대화형 모드에서 여러 파일시스템 점검 시 유용

-t 파일시스템 : 점검할 파일시스템 유형 지정

## 7. 파일시스템 관리

### ■ 명령어 du (Disk Usage)

#### – 디렉터리별로 디스크 사용량을 확인

[형식] du [옵션] [파일 및 디렉터리명]

[옵션] -h : 용량 단위(KB, MB, GB)로 표시

-a : 디렉터리가 아닌 모든 파일에 대한 정보 표시

-m : 결과 값을 MB 단위로 표시

-k : 결과 값을 KB 단위로 표시 (기본값)

-s : 사용량의 총 합계만 표시 (파일의 전체 크기를 합한 값으로 표시)

-c : 모든 파일의 디스크 사용 정보를 보여주고 나서 합계를 표시

[예시] du -sh /\* : 디렉터리별 크기를 KB, MB, GB 등의 단위로 출력

```
[linux@localhost ~]$ du -sh /home/linux
58M      /home/linux
```

# 7. 파일시스템 관리

## ■ 명령어 df (Disk Free)

- 시스템에 마운트된 하드 디스크의 용량을 파티션 단위로 확인하는 명령어
- 기본적으로 1KB 블록 단위로 출력하며 옵션을 통해 변경이 가능

[형식] df [옵션] [파일명]

[옵션] -h : 용량 단위(KB, MB, GB)로 표시

-T : 파일시스템의 종류(유형과 파티션 정보)를 출력

-t : 표시되는 파일시스템 유형을 지정

-a : 0 블록의 파일시스템을 포함하여 모든 파일시스템을 출력

-k : --block-size=1K와 같은 의미

-i : inode 사용률 확인, 사용 공간, 사용 퍼센트를 출력

```
[linux@localhost ~]$ df -hT
```

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
devtmpfs	devtmpfs	4.0M	0	4.0M	0%	/dev
tmpfs	tmpfs	1.8G	0	1.8G	0%	/dev/shm
tmpfs	tmpfs	726M	12M	715M	2%	/run
/dev/mapper/rl-root	xfs	36G	5.0G	31G	15%	/
/dev/nvme0n1p1	xfs	960M	409M	552M	43%	/boot
tmpfs	tmpfs	363M	100K	363M	1%	/run/user/1000

# 7. 파일시스템 관리

## ■ 파일 /etc/fstab (file system table)

- 리눅스에서 사용하는 파일시스템 정보를 정적으로 저장하고 있는 파일
- 리눅스 파일시스템 정보와 부팅 시 마운트 정보를 가지고 있음

```
[linux@localhost etc]$ cat /etc/fstab
```

```
#
# /etc/fstab
# Created by anaconda on Sat Mar  9 20:56:49 2024
#
# 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.
#
# After editing this file, run 'systemctl daemon-reload' to update systemd
# units generated from this file.
#
/dev/mapper/rl-root      /                    xfs      defaults            0 0
UUID=5e9655ad-8432-4623-9f74-0fb8906882a2 /boot                xfs      defaults            0 0
/dev/mapper/rl-swap      none                 swap      defaults            0 0
```

디바이스

마운트지점

파일시스템 유형

옵션

덤프여부

파일시스템 검사여부

# 7. 파일시스템 관리

## ■ 파일 /etc/fstab 라인을 구성하는 항목의 의미

필드	내용
디바이스	장치의 이름으로 디바이스 파일(예를 들어 <u>/dev/sda1</u> ) 또는 <b>UUID(universally unique identifier)</b> 가 사용됨 <span style="color: red;">SCSI 디스크의 첫 번째 파티션</span>
마운트 지점	파일 시스템 트리에서 디바이스가 부착되는 위치(디렉터리)
파일 시스템 유형	리눅스에서 허용하는 파일 시스템의 유형 <span style="color: red;">→ 수동으로 마운트</span>
옵션	마운트 옵션으로 <b>mount</b> 명령에서 <u>-o</u> 옵션을 사용하는 것과 의미가 같음
덤프 여부	백업을 위한 것으로 <b>0</b> 은 덤프를 하지 말라는 것이며, <b>1</b> 은 하라는 것
파일 시스템 검사 여부	파일 시스템 검사를 위한 것으로 <b>0</b> 은 검사를 하지 말라는 것이며, 루트(/) 파일 시스템의 경우는 <b>1</b> 일 때, 나머지 파일 시스템의 경우에 <b>2</b> 일 때 검사를 함

\* UUID : 16진수 32개로 파티션을 식별하는 숫자 (운영체제에서 자동으로 부여)

```
UUID=5e9655ad-8432-4623-9f74-0fb8906882a2 /boot xfs defaults 0 0
```

# 7. 파일시스템 관리

## ■ 파일 /etc/fstab 설정 옵션 참고만 시험 안 나옴

속성	의미
defaults	일반적인 파일 시스템에 지정하는 속성이다. rw, nouser, auto, exec, suid 속성을 모두 포함한다.
auto	부팅 시 자동으로 마운트한다.
exec	실행 파일이 실행되는 것을 허용한다.
suid	setuid, setgid의 사용을 허용한다.
ro	읽기 전용 파일 시스템이다.
rw	읽기, 쓰기가 가능한 파일 시스템이다.
user	일반 사용자도 마운트가 가능하다.
nouser	일반 사용자의 마운트가 불가능하다. root만 마운트할 수 있다.
noauto	부팅 시 자동으로 마운트하지 않는다.
noexec	실행 파일이 실행되는 것을 허용하지 않는다.
nosuid	setuid, setgid의 사용을 금지한다.
usrquota	사용자별로 디스크 쿼터 설정이 가능하다.
grpquota	그룹별로 디스크 쿼터 설정이 가능하다.



## 7. 파일시스템 관리 [예상문제]

1. 리눅스 파일 시스템에서 특별한 종류의 디스크 블록으로 파일 이름, 소유주, 권한, 시간, 디스크에서의 위치 등에 대한 정보를 담고 있는 것으로 알맞은 것은?

① inode

② data block

③ partition table

④ super block

② data block은 디렉터리별로 디렉터리 엔트리와 실제 파일에 대한 데이터가 저장된다.

③ partition table은 용량이 크거나 지속적인 데이터 증가가 예상되는 테이블에서 데이터를 여러 개의 작은 단위로 나누므로 성능 저하를 방지하고 관리를 편이하게 하는 방법이다.

④ super block은 해당 파일 시스템 관련 정보(블록의 크기, 블록의 개수, 블록 그룹의 개수, inode의 개수)를 저장한다.

# 7. 파일시스템 관리 [예상문제]

2. 다음 (        ) 안에 에서 설명하는 파일 시스템의 종류로 알맞은 것은?

리눅스 커널 2.4 버전부터는 (        ) 파일 시스템 기능이 있는 ext3를 사용하였고, 시스템에 충돌이 발생하거나 전원 문제가 발생한 경우에 데이터 복구 확률을 높여준다.

① 저널링(journaling)

② msdos

③ ext

④ ext2

저널링 파일 시스템은 시스템의 비정상적인 종료 시 로그를 이용해 빠르면서도 안정적인 복구가 가능하도록 한다. 대표적인 파일 시스템들로는 ext3, ext4, XFS, JFS, ResierFS 등이 있다.

## 7. 파일시스템 관리 [예상문제]

3. 명령어 mount 옵션 중 access time을 갱신하지 않도록 할 때 설정하는 것은?

① remount

② noatime

③ loop

④ atime

① 명령어 remount는 파티션을 재마운트할 때 사용한다.

③ 명령어 loop는 Loop 디바이스나 CD-ROM의 이미지 파일을 ios로 마운트한다.

④ 명령어 atime은 마지막으로 파일에 접속한 시간을 나타낸다.

## 7. 파일시스템 관리 [예상문제]

4. /dev/sda2 파티션을 ext4 파일 시스템으로 생성 시 (                    ) 안에 들어갈 내용으로 틀린 것은?

```
[root@www~]# (                    ) /dev/sda2
```

- ① mke2fs -j
- ② mke2fs -t ext4
- ③ mkfs -t ext4
- ④ mkfs.ext4

옵션 -j는 파티션 /dev/sda2를 저널링 파일 시스템 ext3로 생성한다.

## 7. 파일시스템 관리 [예상문제]

5. /dev/sda3 파티션을 ext3 파일 시스템으로 생성 시 (        ) 안에 들어갈 내용으로 틀린 것은?

```
[root@www~]# (        ) /dev/sda3
```

- ① msks.ext3
- ② mke2fs -t ext3
- ③ mkfs -c
- ④ mke2fs -j

옵션 -c는 파일 시스템을 생성하기 전에 배드블록(Bad Block)을 검사한다. ext3 파일 시스템으로 생성하고자 한다면 mkfs -c -t ext3와 같이 추가 옵션을 넣어야 한다.

## 7. 파일시스템 관리 [예상문제]

6. 다음 중 생성되는 파일 시스템의 종류가 다른 것은?

- ① `mke2fs -j /dev/sdb1`
- ② `mkfs.ext3 /dev/sdb1`
- ③ `mkfs dev/sdb1`
- ④ `mke2fs -t ext3 /dev/sdb1`

명령어 `mkfs /dev/sdb1`은 해당 파티션의 마운트 정보와 파일 시스템 정보를 확인할 수 있다. 나머지 3개의 명령어들은 파티션 `/dev/sdb1`을 파일 시스템 `ext3`로 생성한다.

## 7. 파일시스템 관리 [예상문제]

### 7. 다음 조건으로 파일 시스템을 생성하려 할 때 알맞은 것은?

- 가. RAID 장치인 /dev/md0를 ext3 파일 시스템으로 생성하고  
블록 사이즈를 8192바이트로 한다.
- 나. stripe당 블록 사이즈를 32바이트로 설정한다.

- ① `mke2fs -j -b 8192 -R stripe=32 /dev/md0`
- ② `mke2fs -j -b 4096 -R stripe=32 /dev/md0`
- ③ `mke2fs -j -b 8192 -R stripe=64 /dev/md0`
- ④ `mke2fs -j -b 4096 -R stripe=64 /dev/md0`

옵션 -j는 파티션을 저널링 파일시스템 ext3으로 지정하고, 옵션 -b는 파일 시스템의 블록 사이트를 정하며, 옵션 -R stripe는 RAID 장치에 적합한 데이터 구조를 만든다.

## 7. 파일시스템 관리 [예상문제]

8. 다음 중 파일 시스템을 검사하고 수리하는 명령으로 알맞은 것은?

① mount

② fsck

③ unmount

④ eject

① mount는 장치와 디렉터리를 연결하는 명령어이다.

③ unmount는 연결된 장치와 디렉터리의 연결을 해제하는 명령어이다.

④ eject는 이동식 보조기억장치를 제거하는 명령어이다.



## 7. 파일시스템 관리 [예상문제]

9. 다음 중 파일이나 디렉터리의 디스크 사용량을 확인할 때 사용하는 명령어는?

① free

② fsck

③ df

④ du

명령어 fsck는 손상된 파일을 수정하고자 할 때 디렉터리 /lost+found에서 작업을 수행한다.

① free : 시스템의 메모리 상태(유휴 메모리 양, Swap 메모리 양 등)를 점검하는 명령어

② fsck : 파일 시스템을 점검하고 복구하는 명령어

③ df : 현재 마운트 된 디스크의 크기, 사용량, 남아있는 용량 등의 정보를 확인하는 명령어

## 7. 파일시스템 관리 [예상문제]

10. 다음 중 fsck가 시스템 부팅 시 파일 시스템 점검을 위해 참조하는 필드 영역으로 알맞은 것은?

```
[root@www~]# tail - /etc/fstab/dev/sda1 /backup_data ext4 defaults 1 2
                                     A          B    C D
```

① A

② B

③ C

④ D

① A : 파일 시스템이 마운트 될 위치를 명시한 것이다.

② B : 파일 시스템의 속성 설정으로 default는 auto, exec, suid, ro, rw 등을 속성으로 가진다.

③ C : dump 명령으로 값 1은 데이터 백업이 가능하다.



Contribution

Marathon

Innovation

# The End !