

운영체제 개념

운영체제

- **하드웨어와 소프트웨어를 관리하는 시스템**

- 사용자가 편하게 컴퓨터를 사용할 수 있는 환경을 제공하는 프로그램
- 컴퓨터 시스템의 자원을 효율적으로 할당, 관리, 보호
- 시스템 오류나 잘못된 자원 사용을 감시하고 입출력 장치에 대한 연산/제어를 관리하는 제어 프로그램
- 종류 : Windows, Linux, Unix, MAC OS X 등



운영체제

• 운영 체제 구성 요소

• User Interface

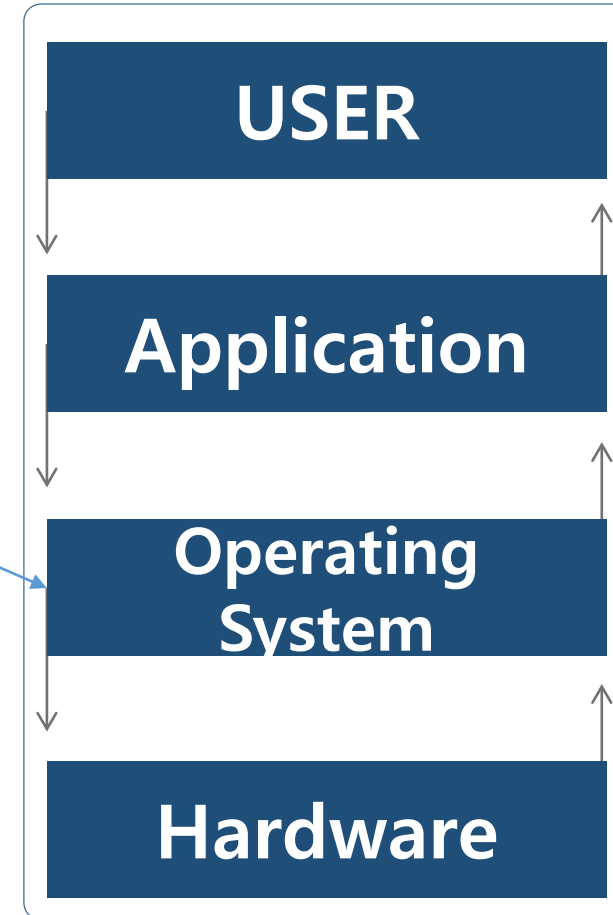
- 사용자가 시스템을 조작(입력), 동작결과 확인(출력)등을 조작하게 도와줌

• Shell(=명령 해석기)

- 사용자가 user interface로 전달한 명령을 시스템 언어에 맞게 해석

• Kernel

- 하드웨어, 프로세스 등을 제어하는 자원 관리자





운영체제

- **용도별 컴퓨터 분류**

- 서비스를 제공 받는 컴퓨터 = Personal Computer = Client Computer
- 서비스를 제공 해주는 컴퓨터 = Server Computer

- 윈도우는 Client 용 OS와 Server용 OS를 따로 제공
- Linux/Unix는 기본 Server용 OS로 제공
- MAC OS X 는 기본 Client용 OS로 제공

운영체제

- **Server Computer**

- 네트워크를 통해 클라이언트에게 서비스를 제공하는 컴퓨터
- 사용 목적 및 하드웨어 구성 등이 Personal computer와 다름(일반적으로 rack에 배치)
- 안정성을 위해 이중화 구성을 함
- 데이터의 안정성(가용성)을 위해 RAID구성



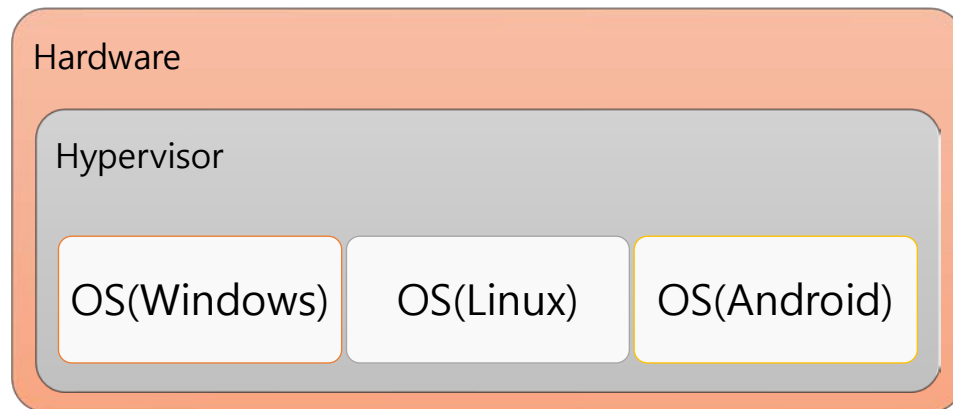
운영체제 - 가상화

- 가상화

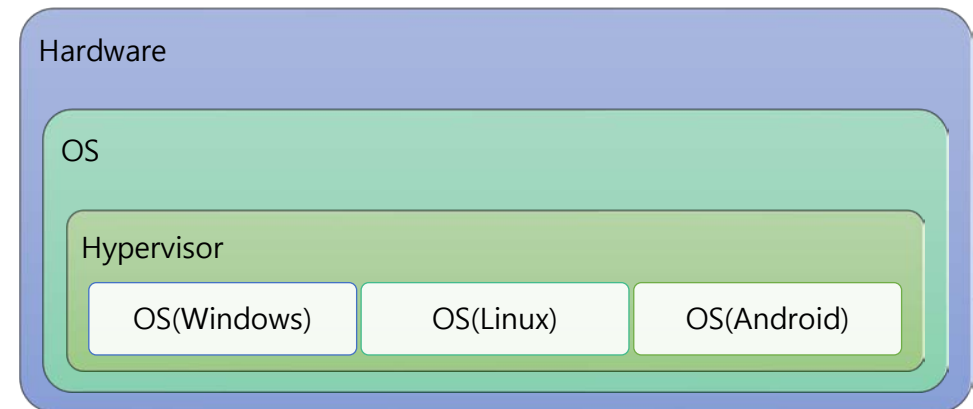
- 하나의 물리적인 컴퓨터를 논리적으로 여러 개의 컴퓨터로 분리하거나 여러 개의 물리적인 컴퓨터를 논리적으로 하나의 컴퓨터로 통합하여 사용하는 환경

- 종류

- Hardware 가상화 : hardware 상단에서 바로 실행
 - VMware ESX, Xen Server, Hyper-V, etc..
- Software 가상화 : OS위에서 실행
 - VirtualBox, VMware Server, KVM, etc...



Hardware 가상화



Software 가상화



Cloud

- **Cloud**

- 인터넷 기반의 컴퓨팅
- 네트워크 상의 Database 서버에 프로그램을 필요 시 마다 호출하여 사용하는 서비스
- 사용자가 원할 때 내부 구조에 대해 이해할 필요 없이 어디에서나 원하는 작업을 지원

- **구축 환경에 따른 분류**

- Public Cloud
 - 인터넷에 접속 가능한 모든 사용자를 위한 cloud 서비스
- Private Cloud
 - 특정 기업이나 조직 만을 대상으로 하는 cloud 서비스
- Community cloud
 - 공통 목적을 가진 특정 기업 또는 단체를 위한 cloud 서비스
- Hybrid Cloud
 - Public cloud, private cloud, community cloud의 혼합형

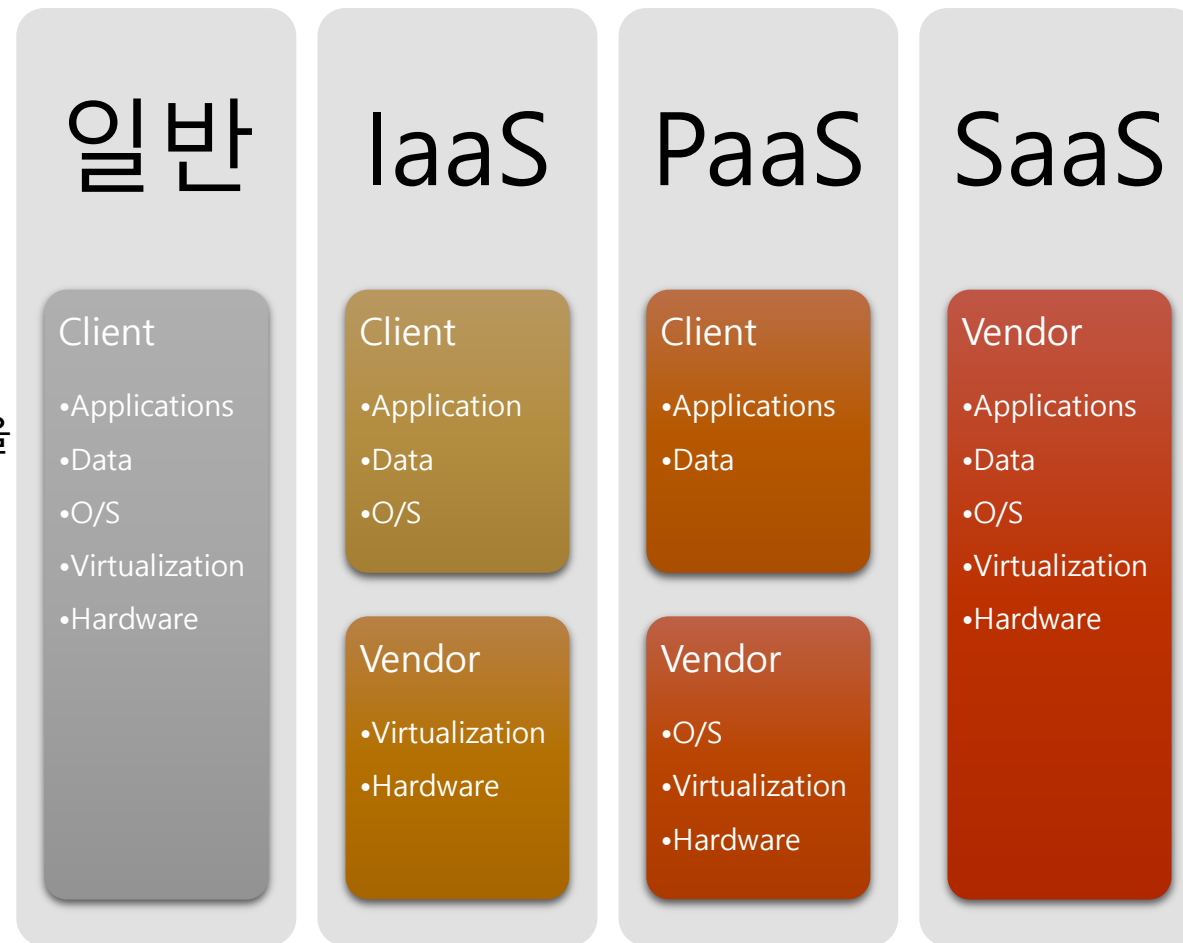
- **가상화 제공 프로그램**

- VMware, VirtualBox, Hyper-V, Xen, etc...

Cloud

- 서비스 형태에 따른 분류

- IaaS(Infrastructure as a Service)
 - 시스템에 필요한 자원을 서비스 형태로 제공
 - 사용자가 자원을 할당 받아 원하는 시스템 구축 가능
- PaaS(Platform as a Service)
 - 프로그램 및 서비스가 실행될 수 있는 환경을 서비스 형태로 제공
 - Infra 자원에 대해서는 알지 못하며 제어권이 없음
- SaaS(Software as a Service)
 - 필요한 소프트웨어를 설치하지 않고 이용 가능하도록 서비스 형태로 제공



Prompt

- Prompt

- 컴퓨터가 입력을 기다리고 있음을 가리키기 위해 화면에 나타나는 표시

- Windows(DOS)

C:\W>	
C:	드라이브 명
W	현재 작업 디렉터리 위치
>	Prompt와 커서를 구분하는 구분자

- Unix(ksh : 콘셸)

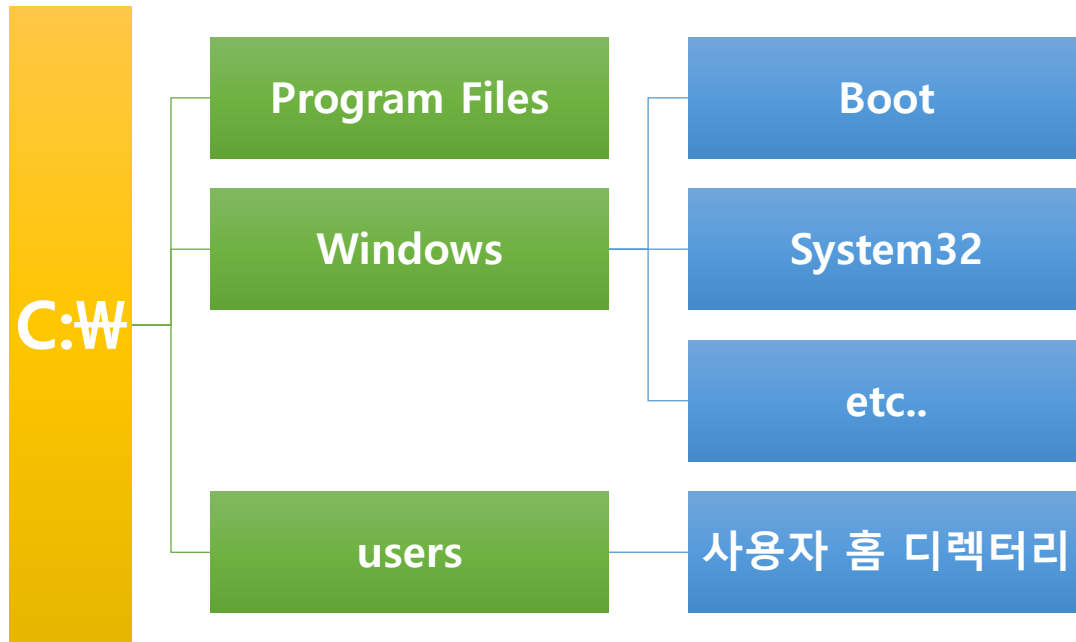
#	
#	관리자 계정
\$	사용자 계정

- Linux(bash shell)

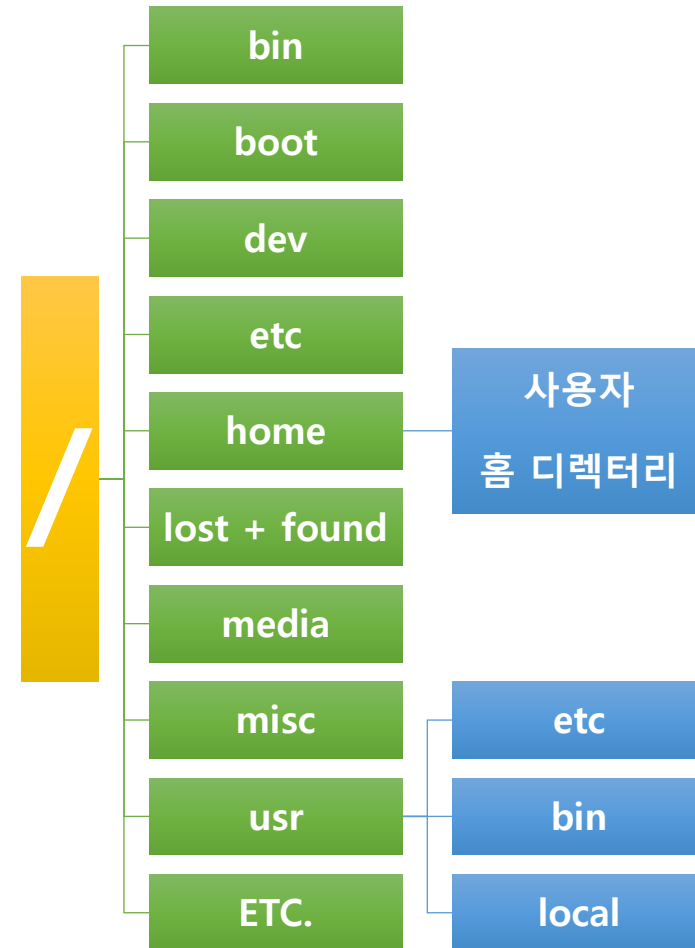
[root@localhost ~]#	
root	로그인 한 사용자 계정 명
@	로그인 계정과 서버이름을 구분하는 구분자
Localhost	리눅스 시스템의 호스트 명(= 서버이름)
~	현재 작업 디렉터리 위치
[]	계정상태와 로그인 정보를 구분하는 구분자
#	로그인 정보 # : 관리자 계정 \$: 사용자 계정

디렉터리 구조

• Windows



• Linux



경로

- 절대 경로

- 최 상위 디렉터리를 기준으로 경로 정보 표현
- 사용자의 현재 위치와는 관계 없음

- 상대 경로

- 현재 위치 기준으로 경로 정보 표현

.	현재 위치 표현
..	상위 디렉터리 표현
~	홈 디렉터리 표현(Linux)

- 경로 표현

- Windows
 - 디렉터리와 디렉터리 사이는 \로 구분
- Linux
 - 디렉터리와 디렉터리 사이는 /로 구분

운영체제 설치(Linux)



Linux

- **Linux**

- Open Source
- Linus Benedict Torvalds(리누스 베네딕트 토르발스)가 개발 함
- 기존 Unix 커널을 경량화 하고 GNU project의 C 라이브러리를 포함한 운영체제
- 기본적인 Linux 커널을 바탕으로 여러 종류의 배포판이 만들어져 있음
 - Linux = Kernel
 - Linux OS = Kernel + GNU
- 버전 표시
 - <Major>.<Minor>.<Patch>
 - Major : 커널의 큰 구조적 변화
 - Minor : 내부적 구현방식의 변화
 - Patch : 같은 버전에서 업데이트

- **GNU(Gnu Not Unix) 선언문**

- Software의 자유 배포를 위한 약속



Linux

- 배포판 종류

데비안	데비안 GNU/리눅스, 우분투, 리눅스 민트, 칼리 리눅스, 크런치뱅, 하모니카, Damn Small Linux, elementary OS, 스팀 OS
레드햇	레드햇 엔터프라이즈 리눅스, 페도라, 미고, 붉은별, Cent OS, 아시아눅스, 타이젠
맨드리바	Mandriva, Open Mandriva Lx, 마제야, PCLinuxOS, no.1Linux
슬랙웨어	슬랙웨어 리눅스, 오픈수세, Slax
안드로이드	안드로이드, 크롬 OS, Remix OS, Polaris OS
기타	아치 리눅스, Manjaro Linux , 젠투 리눅스

Linux 기본 명령어



Linux 기본 명령어

- **Linux CMD(명령어) 구조**

- #Command
- #Command + Option
- #Command + Argument
- #Command + Option + Argument

Linux 기본 명령어

- **man**

- 매뉴얼 확인(상세 도움말)
- 기본적으로 첫 번째 발견된 매뉴얼 페이지만 출력
- 사용법
 - [root@KH-Linux~]# man [option] <command>

OPTION	
-a	모든 매뉴얼 페이지 출력
-f	간단한 설명만 출력

- man 페이지 종료
 - q 키 클릭

Linux 기본 명령어

- **date**

- 시스템의 시간과 날짜 확인
- 권한이 필요 함

- 사용법

- 확인

- > [root@KH-Linux~]# date

```
[root@KH-Linux ~]# date
2015. 09. 07. ( 월 ) 02:08:51 KST
```

- 수정

- > [root@KH-Linux~]# date <MMDDhhmm[[CC]YY][.ss]>

```
[root@KH-Linux ~]# date
2015. 09. 07. ( 월 ) 02:08:51 KST
```

MM	월
DD	일
hh	시간
mm	분
CC	연도 앞
YY	연도 뒤
ss	초



Linux 기본 명령어

- **rdate**

- 원격지로부터 시간 동기화
- 사용법
 - [root@KH-Linux~]# rdate [option] <NTPserver address>

OPTION	
-p	NTP server 시간 출력
-s	NTP server와 시스템 시간 동기화
NTP Server List	
LG U+	time.bora.net 또는 gps.bora.net
KRISS	time.kriss.re.kr 또는 203.254.163.74



Linux 기본 명령어

- **uname**
 - 시스템 정보 확인
 - 사용법
 - [root@KH-Linux~]# uname [option]

OPTION	
-a	모든 정보 출력
-m	하드웨어 타입(CPU 정보) 출력
-n	hostname 출력
-r	kernel 릴리즈 번호 출력
-s	Kernel 이름 출력
-v	Kernel 버전 출력



Linux 기본 명령어

- **clear**

- 화면 지우기
- 내용을 삭제하는 것이 아님
- 마지막 출력 라인을 화면 제일 위로 올려 줌

- 사용법
 - [root@KH-Linux~]# clear

Linux 기본 명령어

- **echo**

- 표준 출력
- 입력한 문자열을 그대로 출력
- redirection을 이용하여 파일 생성(덮어쓰기/추가쓰기)
- 사용법
 - [root@KH-Linux~]# echo [option] <string>
- Redirection
 - 입/출력 방향 전환

Redirection	
>	왼쪽 결과를 오른쪽 출력(덮어 쓰기)
>>	왼쪽 결과를 오른쪽 출력(추가 쓰기)



Linux 기본 명령어

- **history**

- 사용한 명령어 목록 확인
- 사용법
 - [root@KH-Linux~]# history [num]
 - [num]을 지정하면 해당 숫자만큼 최근 사용한 명령어 출력



Linux 기본 명령어

- **alias**

- 별칭 확인 및 설정
- 사용법
 - [root@KH-Linux~]# alias [별칭='command']
 - alias 설정을 무시하고 실행하고 싶을 때는 "₩별칭" 으로 입력

- **unalias**

- 별칭 해제
- 사용법
 - [root@KH-Linux~]# unalias <별칭>



Linux 기본 명령어

- **poweroff / halt**

- 시스템 즉시 종료
- 사용법
 - [root@KH-Linux~]# poweroff [option]
 - [root@KH-Linux~]# halt [option]
- 차이점
 - 하드디스크 동기화 유무
 - poweroff는 동기화 시키지 않고 종료 : sync로 동기화 하고 사용해야 함



Linux 기본 명령어

- **reboot**
 - 시스템 재시작
 - 사용법
 - `[root@KH-Linux~]# reboot [option]`

Linux 기본 명령어

- **shutdown**

- 시스템 종료 및 재부팅

- 사용법

- [root@KH-Linux~]# shutdown [option] [time] [message]

OPTION	
-k(knock)	종료를 알리는 경고 메시지 전송(실제 종료 되지 않음)
-h	Shutdown 후 종료(halt) 실행
-r	Shutdown 후 재부팅(reboot) 실행
-f(fast reboot)	재부팅 시 fsck(fast system check)를 수행하지 않음(빠른 재부팅)
-c	예약되어 있는 shutdown을 취소 (시간 지정은 안 되지만 사용자에게 메시지 전송은 가능 함)

Linux 기본 명령어

- **init**

- run level 선택
 - run level : 시스템의 동작 방식을 정의해 놓은 것
- 사용법
 - [root@KH-Linux~]# init [option] <run level>

Run Level		
0	종료 모드	하드디스크 동기화 후 종료
1	단일 사용자 모드(single mode)	시스템 복구
2	다중 사용자 CLI	NFS/X-window를 제외한 부팅
3	다중 사용자 CLI	X-window를 제외한 부팅
4	X	사용하지 않음(사용자 정의 가능)
5	다중 사용자 GUI	X-window환경으로 모든 서비스 포함
6	재부팅 모드	시스템 재시작



Linux 디렉터리

- **Linux 디렉터리 구조**

- FHS(Filesystems Hierarchy System) : Tree 형태의 계층적인 구조
- **최상위 디렉터리 : /**

- **Linux 파일 및 디렉터리 특징**

- 대/소문자 구분
- 파일과 디렉터리 이름 중복 불가

- **절대경로**

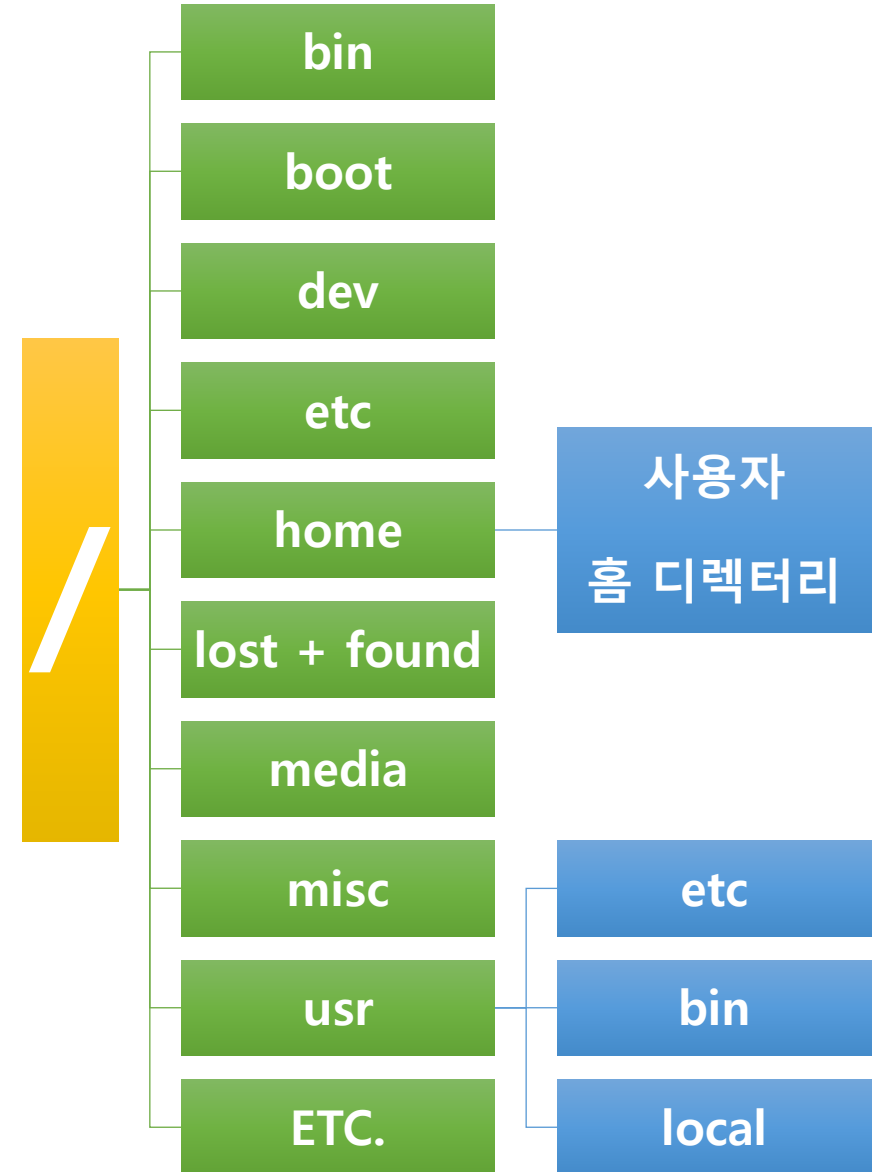
- 최상위 경로를 기준으로 현재 파일이나 디렉터리의 위치를 나타내는 경로
- 사용자의 현재 위치와는 관계 없음

- **상대 경로**

- 현재 위치를 기준으로 파일이나 디렉터리의 위치를 나타내는 경로
- 사용자의 현재 위치에 따라 경로가 달라 짐

Linux 디렉터리

- **/**
 - 최상위 경로
- **bin**
 - 사용자를 위한 명령어 저장
 - ls, ps, mv, cat, chmod, date, mount ...
- **boot**
 - 부트로더 및 부팅을 위한 파일 저장
- **dev**
 - 시스템의 장치(devices) 파일 저장
 - 시스템의 모든 장치가 파일로 표현
 - udev daemon에 의해 관리 됨





Linux 디렉터리

- **etc**
 - 시스템 및 프로그램의 환경설정 파일
 - 계정 파일, 네트워크 설정 파일 등 시스템의 주요 관리 파일
 - 백업 필요
- **home**
 - 사용자 계정의 홈 디렉터리 기본 경로
 - 계정명으로 홈 디렉터리가 생성 됨
- **lib**
 - 시스템 프로그램 실행에 필요한 공유 라이브러리
 - 변경 및 삭제하지 않는 것이 좋음
- **media**
 - 플로피, CD-ROM, DVD 등 이동식 디스크가 mount 됨



Linux 디렉터리

- **mnt**
 - 파일 시스템 임시 mount 경로
- **opt**
 - 추가 S/W 설치 경로
- **proc**
 - 실행 중인 프로세스, CPU, memory 등의 시스템 정보가 가상 파일로 저장
 - 대부분의 파일이 읽기 전용
 - 쓰기 가능한 파일의 설정의 변경하여 커널의 기능을 변경할 수 있음(Kernel Parameter)
- **root**
 - root 계정(관리자)의 홈 디렉터리
 - root 계정으로만 접근이 가능 함



Linux 디렉터리

- **sbin**

- system 관리용 명령 저장
- 시스템 수정, 복구 작업에 사용되는 명령어
- 일반 사용자들의 접근을 제한해야 함
- ifconfig, reboot, shutdown, mount, fsck, mkfs ...

- **srv**

- Service 디렉터리
- 주로 FTP, HTTP, CVS 등의 데이터가 저장 됨

- **sys**

- system 동작에 필요한 파일 저장
- 변경 및 삭제하지 않는 것이 좋음

- **usr**

- 일반 사용자들을 위한 S/W 설치 위치



Linux 디렉터리

- **var**

- 수시로 업데이트 되는 파일들이 위치 함
- log파일
- http, mail, dns 등이 동작할 때 필요한 데이터가 저장되기도 함

- **lost + found**

- 부팅 시 파일 시스템에 문제 발생 시 fsck(file system check) 명령으로 복구할 때 사용 됨
- 한 파일 시스템당 하나만 존재 함

Linux 파일

- 파일 종류

-	일반 파일
b	블록형 특수 장치파일(주로 디스크, /dev/sda 등)
C	문자형 특수 파일(주로 입/출력에 사용, /dev/console 등)
d(/)	디렉터리 파일
P(=)	파이프 파일
S	소켓 파일
l(a)	심볼릭 링크파일

- 확장자의 의미

- 윈도우와 다르게 특별한 의미가 없음
- 보통 프로그램 개발 언어 등을 표시



Linux 파일

- **일반 파일**

- 기본적인 대부분의 파일
- 원하는 형식의 데이터 저장이 가능하며 각 사용자들이 소유 권한을 가짐
- 데이터베이스, 동영상, 음악, 실행 파일 등

- **디렉터리 파일**

- 특수 파일
- 연관 있는 파일들을 하나의 그룹으로 만들어 저장하도록 구분되어 있는 공간
- 하나의 inode를 할당 받으며 파일 이름과 inumber를 매칭하는 항목의 리스트로 구성되어 있음
 - 현재 디렉터리 : .
 - 상위 디렉터리 : ..

Linux 파일

- 디바이스 파일

- 하드웨어를 하나의 파일로 인식하여 관리 함
 - 블록 디바이스 파일 : 디스크, 테이프 등의 저장 장치
 - 문자 디바이스 파일 : 터미널, 프린터 등의 출력 장치
- 키보드 디바이스 파일에서 데이터를 읽음, 사운드 디바이스 파일에 소리파일을 보내면 스피커로 출력 됨

- 링크 파일

- 파일과 디렉터리의 관리를 편리하게 하고자 하는 목적으로 사용 됨
- 상위 디렉터리가 많은 경우 파일명만 이용하여 접근
- 긴 파일명의 파일을 자주 참조할 경우 다른 디렉터리에 연결
- 해킹에 이용될 수 있으므로 주기적으로 확인하여 불필요한 링크 파일은 삭제하는 것이 좋음
- 심볼릭 링크 방식을 더 많이 사용 함

Linux 파일

- **심볼릭(소프트) 링크**

- 원본 파일의 위치정보를 가지는 단축 아이콘
- 윈도우의 바로가기와 비슷한 개념으로 특정 파일을 가리키고 있는 파일
- 심볼릭 링크 파일을 변경해도 원본 파일에는 변화가 없음
- 다른 파일 시스템에서도 링크 가능
- 별도의 inode를 가지며 링크 파일이 가지고 있는 용량만큼 디스크의 공간을 차지 함
- 원본의 링크 카운트가 증가하지 않음

- **하드 링크**

- 원본 파일과 동일한 inode에 연결되는 링크
- 하나의 파일에 여러 개의 이름을 지정해서 사용하는 개념
- 다른 파일 시스템에서 링크 불가능
- 원본과 동일한 inode를 사용 함
- 원본의 링크 카운트가 증가 함

Linux 파일 및 디렉터리 명령어

- **ls**

- 지정한 디렉터리의 파일 목록 출력
- 사용법
 - [root@KH-Linux~]# ls [option] [디렉터리의 상대경로 또는 절대경로]

OPTIONS	
-a/-A	숨김 파일을 포함하여 출력/ . , .. 을 제외하고 출력
-i	파일의 inumber 출력
-l/-n	파일의 inode를 포함한 상세 내용 출력/UID, GID로 출력
-k	파일 크기를 KB 단위로 출력
-m	가로 목록으로 출력
-F	파일 이름에 파일 종류를 구분하는 기호를 함께 출력
-R	하위 디렉터리의 모든 파일 출력
-1	한 줄에 한 파일씩 출력

Linux 파일 및 디렉터리 명령어

- **pwd**

- 현재 디렉터리 위치 확인
- 사용법
 - [root@KH-Linux~]# pwd

- **cd (change directory)**

- 디렉터리 이동
- 사용법
 - [root@KH-Linux~]# cd <이동할 디렉터리의 상대경로 또는 절대경로>

.	현재 위치한 디렉터리
..	현재 위치의 상위 디렉터리
~	계정의 홈 디렉터리
~[계정명]	지정한 계정의 홈 디렉터리
/	최상위 디렉터리



Linux 파일 및 디렉터리 명령어

- **mkdir**

- 빈 디렉터리 생성
- 사용법
 - [root@KH-Linux~]# mkdir [option] <디렉터리 이름>

OPTIONS	
-m	생성할 디렉터리의 권한 설정 (default : 755)
-p	상위 디렉터리 함께 생성



Linux 파일 및 디렉터리 명령어

- **rm**

- 빈 디렉터리 삭제
- 사용법
 - [root@KH-Linux~]# rm [option] <디렉터리 이름>

OPTIONS	
-p	상위 경로의 디렉터리까지 삭제

- 주의 사항
 - 무조건 빈 디렉터리만 삭제 됨



Linux 파일 및 디렉터리 명령어

- touch
 - 파일이 존재 할 때 : 지정된 파일의 시간 정보를 갱신 함
 - 파일이 존재하지 않을 때 : 0byte의 빈 파일 생성
 - 사용법
 - [root@KH-Linux~]# touch [option] <파일 이름>

OPTIONS	
-a	파일의 최근 사용 시간만 변경
-c	파일이 존재하지 않을 때 파일을 생성하지 않음
-m	파일의 최근 변경 시간만 변경
-t	파일의 날짜를 임의의 날짜로 지정하여 변경

Linux 파일 및 디렉터리 명령어

- **cat**

- 지정된 파일 내용을 표준 출력으로 확인
- 사용법
 - [root@KH-Linux~]# cat [option] <파일 이름>

OPTIONS	
-n	라인 번호를 포함하여 출력
-b	라인 번호를 포함하여 출력하지만 공백라인은 제외하고 출력

- redirection을 이용한 파일 생성
 - [root@KH-Linux~]# cat > <파일이름>
 - [내용 입력]
 - [ctrl + d : 저장 후 종료]



Linux 파일 및 디렉터리 명령어

- **head**
 - 파일의 처음부터 지정된 줄까지 출력
 - default : 처음부터 10행
 - 사용법
 - [root@KH-Linux~]# head [option] <파일 이름>

OPTIONS	
-n NUM	지정한 줄만큼 출력
-c SIZE	지정한 크기만큼 출력(단위 : byte)



Linux 파일 및 디렉터리 명령어

- **tail**
 - 파일의 마지막부터 지정된 줄까지 출력
 - default : 마지막부터 10행
 - 사용법
 - [root@KH-Linux~]# tail [option] <파일 이름>

OPTIONS	
-n NUM	지정한 줄만큼 출력
-c SIZE	지정한 크기만큼 출력(단위 : byte)
-f	실시간 추가되는 내용 출력(주로 log파일을 확인할 때 자주 사용 됨)

Linux 파일 및 디렉터리 명령어

- **more**

- 많은 출력 결과를 화면 사이즈에 맞게 한 페이지씩 출력
- | (파이프)를 이용하여 다른 명령의 결과를 처리할 수 있음(예 : `ls -l /etc | more`)
- 사용법
 - `[root@KH-Linux~]# more [+number] <파일 이름>`

KEY	
Space	한 페이지씩 출력
Enter	한 행씩 출력
b	이전 페이지 출력
d	다음 페이지 반만 출력
/문자열	지정된 문자열 검색
n	"/문자열"로 검색 후 다음 찾기

KEY	
h	키 도움말 출력
=	현재 위치의 행 번호 출력
v	vi(텍스트 편집기)모드로 이동
q	more 동작 종료
!셸명령	지정된 셸명령 실행

Linux 파일 및 디렉터리 명령어

- **grep**

- 파일 또는 명령어의 결과에서 특정 패턴의 문자열 검색
- | (파이프)를 이용하여 다른 명령의 결과를 처리할 수 있음 (예 : ls -l /etc | grep [패턴])
- 사용법
 - [root@KH-Linux~]# grep [option] <패턴> <파일 이름>

OPTIONS	
-c	검색된 문자를 출력하는 대신 검색된 행의 총 수를 출력
-h	파일 이름을 출력하지 않음
-i	대/소문자를 구분하지 않고 검색
-n	검색된 문자의 행의 번호를 함께 출력 함
-s	에러 메시지만 출력
-v	지정된 표현식에 매치되지 않은 행만 출력
-w	표현식을 단어 단위로 검색

Linux 파일 및 디렉터리 명령어

- **cp**

- 원본 파일을 대상 파일로 복사

- 사용법

- [root@KH-Linux~]# cp [option] <원본 파일> <대상 파일>

OPTIONS	
-f	복사할 파일 이름이 겹칠 때 강제로 복사
-i	복사할 파일 이름이 겹칠 때 확인하고 복사
-l	하드 링크 생성
-p	소유주, 소속 그룹, 퍼미션, 시간 등을 동일하게 복사
-U	복사할 대상이 원본과 변경 날짜가 같거나 더욱 최신 파일이면 복사하지 않음
-v	복사하기 전에 복사할 파일 출력
-R / -r	하위 디렉터리까지 복사

Linux 파일 및 디렉터리 명령어

- **mv**

- 파일의 이름변경
- 파일을 다른 경로로 이동

- 사용법

- [root@KH-Linux~]# mv [option] <원본 파일> <대상 파일>

OPTIONS	
-f	이동할 위치에 이동할 파일 이름이 겹치면 확인 하지 않고 강제로 이동 함
-i	이동할 위치에 이동할 파일 이름이 겹치면 확인 함

Linux 파일 및 디렉터리 명령어

- **rm**

- 파일 및 디렉터리 삭제
- 사용법
 - [root@KH-Linux~]# rm [option] <파일 이름>

OPTIONS	
-i	파일 및 디렉터리를 삭제할 때 확인 함 (y=삭제, n=취소)
-f	파일 및 디렉터리를 삭제할 때 확인하지 않고 강제 삭제 함
-r	해당 디렉터리의 하위 디렉터리까지 모두 삭제

Linux 파일 및 디렉터리 명령어

- **ln**

- 링크 파일 생성
- 심볼릭 링크 사용법
 - `[root@KH-Linux~]# ln -s [option] <원본 파일> <심볼릭 링크 파일>`
- 하드 링크 사용법
 - `[root@KH-Linux~]# ln [option] <원본 파일> <하드 링크 파일>`

OPTIONS	
-b	대상파일이 존재할 때 백업 파일을 생성한 후 링크파일 생성
-d	디렉터리 하드링크(시스템 권한으로 제한되는 경우가 있음)
-f	대상파일이 존재할 때 기존의 파일을 강제로 삭제하고 링크파일 생성
-i	대상파일이 존재할 때 기존의 파일을 삭제할 것인지 확인 후 생성
-s	심볼릭 링크파일 생성
-S	백업파일을 생성할 때 원하는 접미사 지정



Linux 파일 및 디렉터리 명령어

- **file**

- 파일의 종류를 자동으로 판단하여 출력
- 사용법
 - [root@KH-Linux~]# file [option] <파일 이름>

OPTIONS	
-f	리스트 파일을 이용하여 파일의 종류 확인
-m	매직 파일을 선택해서 파일의 종류 확인

Linux 파일 및 디렉터리 명령어

- 실습

1. 최상위 디렉터리 하위에 test 이름의 디렉터리 생성
2. 생성한 test 디렉터리로 이동
3. test 디렉터리 하위에 file1의 이름을 갖는 빈 파일 생성
4. file2 파일을 생성하면서 hello라는 데이터를 새로 쓰기(단, 명령어 한 줄로 처리)
5. file2 데이터에 byebye 데이터 이어 쓰기(단, cat 명령어를 이용해서 추가)
6. file2의 데이터 내용 확인
7. 생성한 test 디렉터리 하위에 dir1 디렉터리 생성
8. test 디렉터리 하위에 있는 file2 파일을 이전에 생성한 dir1 디렉터리로 복사
9. test 디렉터리 하위에 있는 file1 파일의 이름을 file1-mod로 변경
10. test 디렉터리 하위에 있는 dir1 디렉터리 하위에 있는 file2 파일의 하드링크 파일을 /root/file2-hard 라는 파일 이름으로 생성
11. test 디렉터리 하위에 있는 dir1 디렉터리 하위에 있는 file2 파일의 심볼릭 링크 파일을 /var/file2-soft 라는 파일 이름으로 생성(단, 원본 파일(file2)을 상대경로로 지정)
12. 최상위 밑에 존재하는 test 디렉터리를 한번에 삭제



Linux 파일 및 디렉터리 명령어

- **find**
 - 지정된 디렉터리 하위의 파일 검색
 - 사용법
 - [root@KH-Linux~]# find [디렉터리] [option] [action]

OPTIONS	
-empty	비어있는 파일 찾기
-uid [UID]	지정된 UID 값으로 찾기
-gid [GID]	지정된 GID 값으로 찾기
-user [username]	소유자로 찾기
-group [groupname]	소속그룹 명으로 찾기
-nouser	소유주가 없는 파일 찾기
-nogroup	소속그룹 없는 파일 찾기
-name [문자열]	파일 이름으로 찾기
-perm [permission]	지정된 권한을 갖는 파일 찾기



Linux 파일 및 디렉터리 명령어

- find

OPTIONS		
-type [f/d/b/...]	지정된 파일 타입을 이용하여 파일 찾기	
-atime [+n/n/-n]	Access 시간을 이용하여 찾기	
-mtime [+n/n/-n]	파일 내용을 변경한 시간을 이용하여 찾기	
-ctime [+n/n/-n]	파일 정보를 변경한 시간을 이용하여 찾기	
	n	정확하게 n일 전에 변경된 파일 찾기
	+n	현재 ~ n일 동안 변경되지 않고, n일 이전에 변경된 파일 찾기
	-n	현재 ~ n일 동안 변경된 파일 찾기
-size [+n/n/-n]	파일의 크기를 이용하여 찾기	
	n	정확하게 n크기의 파일 찾기
	+n	n크기 이상의 파일 찾기
	-n	n크기 이하의 파일 찾기



Linux 파일 및 디렉터리 명령어

- **find**

ACTION		
-print	리스트 출력	
-exec command {} \;	Find의 결과에 command 적용	
-ok command {} \;	Find의 결과에 command를 적용할 것인지 확인 후 적용	
	{}	Find 명령의 결과 집합
	\;	명령의 끝

Linux 파일 및 디렉터리 명령어

- 실습

- /usr 디렉터리에서 rdate 파일 찾기
- /usr 디렉터리에서 rd로 시작하는 파일 찾기
- /usr 디렉터리에서 첫 번째 글자가 r 또는 a 로 시작, 두 번째 글자가 c 또는 d로 시작하는 모든 파일 찾기
- /usr 디렉터리에서 첫 번째 글자가 a 또는 b로 시작하는 4개의 문자로 구성된 파일 찾기
- 사용자 계정(ktest) 홈 디렉터리에서 소유자가 (ls -l 의 3번째 필드) 자신의 계정(ktest)인 것을 찾기
- /root 디렉터리에서 일반 파일만 찾기
- 최상위 디렉터리에서 심볼릭 링크 찾기

Linux 기본 편집기(vi editor)

- **vi editor**

- 문서 편집기
- 한 행씩 편집하는 것이 아니라 한 화면당 편집하는 기능을 지원하는 프로그램
- 사용법
 - [root@KH-Linux~]# vi [option] <파일 이름>

OPTIONS		
-R	읽기 전용으로 열기	
+n	n번째 행에서 열기	
+/문자열	지정된 문자열이 위치한 행에서 열기	
-r	문서 편집 중 비정상 종료가 발생했을 때 편집 중이던 문서로 복구	
[파일1] [파일2] ...	다중 파일 열기	
	[명령모드] :n	다중 파일 열기 상태에서 다음 파일로 이동
	[명령모드] :N	다중 파일 열기 상태에서 이전 파일로 이동
	[명령모드] :args	현재 파일의 이름 확인



Linux 기본 편집기(vi editor)

- **명령 모드**

- vi editor의 기본 모드이며 다른 모드로 전환을 수행할 수 있는 모드
- 간단한 검색, 커서 이동, 복사/붙여넣기 등의 편집을 수행 함

- **편집(입력) 모드**

- 문서의 텍스트를 입력, 수정, 삭제 할 수 있는 모드
- 내용을 작성할 때 사용 됨

- **실행(ex) 모드**

- vi editor를 제어하는 명령어를 실행하는 모드
- 각종 치환, 저장, 파일읽기, vi editor의 환경 설정 등을 수행 함

Linux 기본 편집기(vi editor)

- **모드 이동**

- 명령모드 → 입력모드
- 명령모드 → 실행모드
- 입력/실행모드 → 명령모드



Linux 기본 편집기(vi editor)

- 명령 모드 → 입력 모드

i	현재 커서의 위치부터 입력
I	현재 커서의 행의 처음으로 이동하여 입력
a	현재 커서의 다음부터 입력(오른쪽 한 칸 이동)
A	현재 커서의 행의 마지막으로 이동하여 입력
o	현재 커서의 다음 행에 빈 줄을 생성한 후 입력
O	현재 커서의 이전 행에 빈 줄을 생성한 후 입력
s	현재 커서의 한 문자를 삭제한 후 입력
S	현재 커서의 한 행을 삭제한 후 입력
r	현재 커서의 한 문자를 수정
R	현재 커서의 위치부터 수정 입력 모드(replace)로 변경

- 명령 모드 → 실행 모드

:	명령어 실행 모드
/	검색 모드(아래로 검색)
?	검색 모드(위로 검색)

Linux 기본 편집기(vi editor)

- 명령 모드 명령어

u	되돌리기(실행 취소)
ctrl + r	되돌리기 취소
커서 이동	
h	왼쪽
l	오른쪽
k	위
j	아래
단어 단위로 이동	
w	현재 커서를 기준으로 다음 단어로 이동
[n]w	현재 커서를 기준으로 n개 만큼 다음 단어로 이동
b	현재 커서를 기준으로 이전 단어로 이동
[n]b	현재 커서를 기준으로 n개 만큼 이전 단어로 이동

Linux 기본 편집기(vi editor)

- 명령 모드 명령어

행 단위로 이동	
0(숫자)	한 행의 처음으로 이동(home)
\$	한 행의 마지막으로 이동(end)
G	한 문서의 마지막 행으로 이동
gg	한 문서의 처음 행으로 이동
[n]G	n번째 행으로 이동
대/소문자 변환	
~	현재 커서의 문자를 대/소문자로 변환
[n]~	현재 커서를 기준으로 오른쪽으로 n개 만큼의 문자를 대/소문자로 변환
.	마지막 명령 재실행
복사	
yw	현재 커서 위치의 한 단어를 복사(버퍼에 임시저장)
[n]yw	현재 커서를 기준으로 오른쪽으로 n개의 단어를 복사(버퍼에 임시저장)
yy	현재 커서가 위치한 한 행을 복사(버퍼에 임시저장)
[n]yy	현재 커서를 기준으로 아래의 n개의 행을 복사(버퍼에 임시저장)

Linux 기본 편집기(vi editor)

- 명령 모드 명령어

붙여 넣기	
p	현재 커서를 기준으로 아래쪽으로 행을 붙여 넣기
P	현재 커서를 기준으로 위쪽으로 행을 붙여 넣기
삭제	
x	현재 커서가 위치의 한 문자 삭제
[n]x	현재 커서를 기준으로 오른쪽으로 n개의 문자를 삭제
d←	현재 커서를 기준으로 왼쪽으로 한 문자 잘라내기
d→	현재 커서를 기준으로 오른쪽으로 한 문자 잘라내기
d↓	현재 커서가 위치한 행 + 다음 행 잘라내기
d↑	현재 커서가 위치한 행 + 이전 행 잘라내기
[n]d[←,→]	현재 커서를 기준으로 방향키 방향으로 n개의 문자 잘라내기
[n]d[↑,↓]	현재 커서를 기준으로 방향키 방향으로 n개의 행 잘라내기
dd	현재 커서가 위치한 한 행을 잘라내기
[n]dd	현재 커서를 기준으로 아래로 n개의 행 잘라내기

Linux 기본 편집기(vi editor)

- 실행 모드 명령어

문자열 검색	
:set ic	검색에서 대/소문자 구분 안 함
:set noic	검색에서 대/소문자를 구분 함
/문자열	커서의 위치에서 아래로 검색
?문자열	커서의 위치에서 위로 검색
n	(검색 상태에서) 다음 문자열 검색
N	(검색 상태에서) 이전 문자열 검색
:%s /문자열1/문자열2 /g	"문자열1"을 "문자열2"로 치환
시스템 명령	
!:bash	Shell환경으로 전환(vi editor로 재 전환 : exit, ctrl + D)
!:명령어	Shell환경에서 명령어를 실행한 후 vi editor로 재 전환
!:명령어	Shell환경에서 실행된 명령의 결과를 vi editor로 출력

Linux 기본 편집기(vi editor)

- 실행 모드 명령어(사용자 정의 옵션)

:n split [파일 이름]	현재 문서에서 n행 만큼 화면을 자르고 다른파일 열기
:set nu	행 번호 표시
:set nonu	행 번호 숨기기(default)
:set ts=n	Tab문자의 칸 수(간격) 지정(default : 8)
:set list	Tab문자 또는 개행 문자 표시
:set nolist	Tab문자 또는 개행 문자 숨기기
:ab	정의된 별칭(문자) 확인
:ab [별칭 이름] [문자열]	별칭 정의(입력 모드에서 사용 함)
:una [별칭 이름]	별칭 해제
:map [매크로] [명령어]	매크로 정의(명령 모드에서 사용 함)
:unmap [매크로]	매크로 해제
:set (종료 : q)	현재 환경설정 내용 확인
:set all (종료 : q)	모든 환경설정 내용 확인

Linux 기본 편집기(vi editor)

- 실행 모드 명령어

종료	저장(w), 종료(q), 강제(!)
w	문서 저장
w!	강제 문서 저장
q	저장하지 않고 종료
q!	저장하지 않고 강제 종료
wq(=x)	저장 후 종료
wq!	저장 후 강제 종료
wq(=x) [파일 이름]	다른 이름으로 저장
e!	현재 문서를 저장하지 않고 최종 저장된 상태로 다시 열기

Linux 기본 편집기(vi editor)

- **vi 환경설정**

- vi editor에서의 환경설정은 일시적인 설정
- 환경설정 파일을 수정해야지 고정적인 설정이 가능 함
- 계정으로 환경설정이 가능하며 각 계정의 홈 디렉터리에서 설정
 - ~/.exrc

- **vim 환경설정**

- vi editor의 환경설정 방식과 동일 하지만 설정 파일의 위치가 다름
- 공통
 - /etc/vimrc
- 계정별 설정
 - 예제파일 복사 : # /usr/share/vim/vim74/vimrc_example.vim ~/.vimrc
 - 환경 설정 내용 수정

Linux 기본 편집기(vi editor)

- 실습

1. vi로 /root/install.log 파일을 열기
2. 프롬프트를 200번 라인으로 이동
3. 200번 라인부터 5라인을 복사한 후 문서의 제일 마지막 라인에 붙여넣기
4. 현재 문서에서 라인 번호를 표시
5. 300번 라인으로 이동
6. 수정모드로 edit 문자열 입력
7. 6번에서 작업했던 내용 취소
8. 100번 라인으로 이동
9. 현재프롬프트를 기준으로 오른쪽으로 5개의 문자 삭제
10. 현재 프롬프트를 기준으로 위쪽으로 mailcap 문자열 검색
11. 현재문서에서 el6로 되어있는 모든 문자열을 centos6로 변경
12. 문서의 마지막으로 이동
13. find / -name install.log 명령의 출력 결과를 문서로 불러들이기
14. /root/anaconda-ks.cfg 파일의 내용을 현재 문서로 불러들이기
15. 500번 라인부터 50라인을 잘라내서 600번 라인 밑으로 붙여넣기
16. 저장하고 종료

Linux Network 설정

- Network interface 설정

- 인터페이스의 정보 확인 및 설정 명령어
- # ip addr

```
[root@tomcat ~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:40:6a:ac brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    inet 192.168.255.135/24 brd 192.168.255.255 scope global dynamic noprefixroute ens160
        valid_lft 1753sec preferred_lft 1753sec
    inet6 fe80::20c:29ff:fe40:6aac/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

종류	파일
인터페이스 동작 유무 설정	# ip link [NIC 이름] up/down
인터페이스 정보 변경(일시적)	# ip addr add [ip주소]/[prefix] dev [NIC 이름] # ip route add default via [게이트웨이주소] dev [출구NIC 이름]
인터페이스 정보 확인	# ip addr
	# ip link show # ip route (routing table확인)

Linux Network 설정

- Network interface 설정 파일

- # vim /etc/sysconfig/network-scripts/ifcfg-eth[N]

종류	명령어
DEVICE=<name>	인터페이스 별칭
HWADDR=<mac address>	NIC의 MAC 주소
TYPE=<name>	네트워크 환경 종류
UUID=<id>	장치 ID (시스템이 NIC장치를 식별하기 위해 부여한 값)
ONBOOT=<yes/no>	부팅 시 활성화 여부
NM_CONTROLLED=<yes/no>	NetworkManager 프로그램 허용 여부
BOOTPROTO=<none/dhcp/static>	IP 할당 방식
IPADDR=<ip address>	IP 주소
NETMASK=<subnet mask>	서브넷 마스크
GATEWAY=<default gateway address>	기본 Next Hop주소 (Default route로만 설정 됨)
DNS1=<주 DNS Server address>	이름 해석을 위해 접근할 주 DNS 서버
DNS2=<보조 DNS Server address>	이름 해석을 위해 접근할 보조 DNS 서버
설정 후 네트워크 서비스 재 시작	# service network restart 또는 # /etc/rc.d/init.d/network restart



Linux Network 설정

- **NetworkManager를 이용한 IP 설정 (영구적인 설정)**

- # nmcli con show
- # nmcli con add type ethernet ₩
con-name [connection id] ₩
ipv4.method manual ₩
ipv4.addresses 192.168.255.100/24 ₩
ipv4.gateway 192.168.255.2 ₩
ipv4.dns "8.8.8.8 168.126.63.1" ₩
autoconnect yes ₩
ifname [device 이름]
- # nmcli con up [connection id]
- # nmcli con reload

- **connection id(profile)변경**

- # nmcli con mod "[변경전 connection-id]" connection.id "[변경할 connection-id]" ₩
connection.interface-name [NIC 이름]

계정 및 그룹



Linux 계정 및 그룹 관리

- 계정

- 사용자를 식별하고 인증을 받거나 로컬/네트워크 자원에 접근하는 것을 가능하게 해주는 객체

- 그룹

- 비슷한 객체의 **논리적인 집합**
- **권한을 편리하게 할당하기 위하여 사용**
- 하나의 그룹에 소속된 계정은 그룹의 권한을 그대로 상속 받음

- 사용 목적

- 사용자 식별 → 인증/접근
- 권한 할당 → 보안성 향상
- 사용자 별 환경 제공
 - 인식할 수 있는 하드웨어 성능, 관리기능, 제공 서비스 등이 다름

Linux 계정 관리

- 계정 관리 파일

/etc/passwd	사용자 계정의 기본 정보
/etc/shadow	암호화된 패스워드 및 정책 설정 정보
/etc/login.defs	전체 사용자 계정의 기본 정책이 설정된 파일
/etc/default/useradd	사용자 계정 생성 시 적용되는 기본 설정
/etc/skel	홈 디렉터리에 기본적으로 제공되는 파일의 저장 위치

- 계정 관리 명령어

useradd	사용자 계정 생성
adduser	사용자 계정 생성
passwd	사용자 패스워드 설정
usermod	사용자 계정의 설정 변경
chage	사용자 패스워드 관리
userdel	사용자 계정 삭제

Linux 계정 관리 파일

- **/etc/passwd 파일**

- 사용자 계정의 기본 정보가 저장된 파일
- [root@KH-Linux~]# cat /etc/passwd

①	②	③	④	⑤	⑥	⑦
root	:x	:0	:0	:root	:/root	:/bin/bash

① root	사용자 계정 이름
② x	패스워드가 암호화되어 /etc/shadow파일에 저장되어 있음
③ 0	uid (관리자 : 0, 사용자 : 500 이상)
④ 0	gid (관리자 : 0, 사용자 : 500 이상)
⑤ root	계정 정보(comment)
⑥ /root	사용자의 홈 디렉터리 관리자 : /root , 사용자 : /home/계정이름
⑦ /bin/bash	사용자의 shell 정의 (default : bash)

Linux 계정 관리 파일

- **/etc/shadow 파일**

- 사용자 계정의 암호화된 패스워드 및 정책 설정 정보가 저장된 파일
- [root@KH-Linux~]# cat /etc/shadow

①	②	③	④	⑤	⑥	⑦	⑧	⑨
root	:\$6\$NAzqsXeFGHz3Hlp3\$kS2BgO...	:16692	:0	:99999	:7	:	:	:

① root	사용자 계정 이름
② :\$6...3H...2BgO...	암호화된 패스워드 정보 (형식 : \$hash_id\$salt\$hash_value)
③ 16692	패스워드의 마지막 변경일(1970년 1월 1일부터 경과된 일 수로 표시)
④ 0	마지막 변경일로부터 패스워드를 최소 사용해야 하는 일 수
⑤ 99999	마지막 변경일로부터 패스워드를 최대 사용할 수 있는 일 수(99999 : 무한대)
⑥ 7	패스워드 만료를 경고하는 일(만료일 이전 n일부터 경고)
⑦	패스워드 만료 후 사용 유예기간
⑧	계정 사용 만료일(월/일/연도)
⑨	예약된 필드(사용하지 않음)

Linux 계정 관리 파일

- **/etc/login.defs 파일**

- 시스템에서 생성되는 전체 사용자 계정의 기본 정책이 설정된 파일
- [root@KH-Linux~]# cat /etc/login.defs

① MAIL_DIR	메일 파일의 저장 경로
② PASS_MAX_DAYS	패스워드 최대 사용 일 수 (지정된 기간 이후 재설정)
③ PASS_MIN_DAYS	패스워드 최소 사용 일 수 (지정된 기간 중 변경 불가능)
④ PASS_MIN_LEN	패스워드 최소 길이
⑤ PASS_WARN_AGE	패스워드 만료 경고 일
⑥ UID_MIN	사용자 계정의 최소 uid

⑦ UID_MAX	사용자 계정의 최대 uid
⑧ GID_MIN	사용자 계정의 최소 gid
⑨ GID_MAX	사용자 계정의 최대 gid
⑩ USERDEL_CMD	계정 삭제 명령(userdel)이 실행 되기 전 실행할 프로그램 설정
⑪ CREATE_HOME	사용자 홈 디렉터리 생성 여부
⑫ UMASK	사용자 계정 홈 디렉터리의 umask 설정

Linux 계정 관리 파일

- **/etc/default/useradd 파일**

- 사용자 계정 생성 시 적용되는 기본 정책이 설정된 파일
- [root@KH-Linux~]# cat /etc/default/useradd

```
# useradd defaults file
```

- ① GROUP=100
- ② HOME=/home
- ③ INACTIVE=-1
- ④ EXPIRE=
- ⑤ SHELL=/bin/bash
- ⑥ SKEL=/etc/skel
- ⑦ CREATE_MAIL_SPOOL=yes

① GROUP	기본 그룹 ID
② HOME	사용자 홈 디렉터리 생성 위치
③ INACTIVE	패스워드 만료 후 유예기간
④ EXPIRE	패스워드 만료일 (YYYY-MM-DD)
⑤ SHELL	사용자의 기본 shell
⑥ SKEL	사용자 별 기본 설정파일의 위치
⑦ CREATE_MAIL_SPOOL	메일 저장 파일 생성 여부



Linux 계정 관리 파일

- **/etc/skel 디렉터리**

- 사용자 계정의 홈 디렉터리에 제공 할 기본 환경설정 파일이 저장된 디렉터리
- 사용자 홈 디렉터리의 초기 환경 제공을 위해 사용 됨
- 사용자 계정을 생성 했을 때 skel 디렉터리에 있는 파일들이 자동으로 사용자 홈 디렉터리로 복사됨
- [root@KH-Linux~]# ls -a /etc/skel

Linux 계정 관리 명령어

- **useradd 명령어 (=adduser)**

- 사용자 계정 생성

- 사용법

- [root@KH-Linux~]# useradd [옵션] [계정 이름]

- 옵션을 설정하지 않으면 "/etc/default/useradd"에 지정된 기본 값으로 생성 됨

OPTIONS	
-u	사용자 계정의 uid 설정
-g	사용자 계정의 gid 설정
-G	사용자 계정의 2차 그룹의 gid 설정
-d	사용자의 홈 디렉터리 설정
-e	사용자 계정의 만료일 설정

OPTIONS	
-f	사용자 계정의 만료 유예일 설정
-s	사용자 계정의 로그인 기본 shell 설정
-M	사용자의 홈 디렉터리를 생성하지 않음
-c	/etc/passwd의 설명(comment) 설정
-D	/etc/default/useradd 파일의 기본설정 변경

Linux 계정 관리 명령어

- **usermod 명령어**

- 사용자 계정의 설정 변경
- 사용법
 - [root@KH-Linux~]# usermod [옵션] [계정명]

OPTIONS	
-u	사용자 계정의 uid 변경
-g	사용자 계정의 gid 변경
-G	사용자 계정의 2차 그룹의 gid 변경
-c	사용자 계정의 설명(comment) 변경

OPTIONS	
-d	사용자 계정의 홈 디렉터리 변경
-e	사용자 계정의 만료일 변경
-f	사용자 계정의 만료 유예일 변경
-s	사용자 계정의 로그인 기본 shell 변경

Linux 계정 관리 명령어

- **passwd 명령어**

- 사용자 계정의 패스워드 생성 및 설정 변경
- 관리자는 모든 계정의 패스워드 변경 가능
- 사용자 계정은 자신의 패스워드만 변경 가능

- 사용법
 - [root@KH-Linux~]# passwd [옵션] [계정명]

OPTIONS	
-d	패스워드를 null값으로 변경 (패스워드 삭제)
-l	패스워드 잠금(Lock)
-u	패스워드 잠금 해제
-S	사용자 계정의 패스워드 설정 내용 확인

Linux 계정 관리 명령어

- **chage 명령어**

- 사용자 계정의 패스워드 설정 변경
- 사용법
 - [root@KH-Linux~]# chage [옵션] [계정명]

OPTIONS	
-l(소문자 l)	chage 설정 내용 확인
-M	패스워드의 사용 최대기간(재설정) 변경
-m	패스워드의 최소 유지 기간 변경
-I(대문자 i)	패스워드의 사용기간 만료 후 사용자 계정 만료 유예기간 변경 (0 : 즉시 계정 만료)
-E	사용자 계정의 만료 기간 변경 (0 : 즉시, -1 : 비활성, YYYY-MM-DD : 직접 지정)
-d	패스워드의 최종 변경일 수정
-W	패스워드 만료에 대한 경고일 변경



Linux 계정 관리 명령어

- **userdel 명령어**
 - 사용자 계정 삭제
 - 사용법
 - [root@KH-Linux~]# userdel [옵션] [계정명]

OPTIONS	
-r	사용자의 홈 디렉터리까지 함께 삭제



Linux 그룹 관리

- 그룹 관리 파일

/etc/group	사용자 그룹의 기본 정보
/etc/gshadow	사용자 그룹의 암호화된 패스워드 정보

- 그룹 관리 명령어

groups	사용자 그룹 정보 확인
groupadd	사용자 그룹 생성
groupmod	사용자 그룹 설정 변경
gpasswd	사용자 그룹 패스워드 설정, 그룹관리자 및 구성원 설정
groupdel	사용자 그룹 삭제

Linux 그룹 관리 파일

- **/etc/group 파일**

- 사용자 그룹의 기본 정보가 저장된 파일
- [root@KH-Linux~]# cat /etc/group

①	②	③	④
root	:x	:0	:

① root	그룹 이름
② x	그룹의 패스워드 x : /etc/gshadow 파일에 암호화된 패스워드 저장
③ 0	gid (관리자 : 0, 사용자 : 500 이상)
④	그룹에 소속된 사용자 목록(참조)

Linux 그룹 관리 파일

- **/etc/gshadow 파일**

- 사용자 그룹의 암호화된 비밀번호 정보, 그룹 관리자 및 구성원 목록이 저장된 파일
- [root@KH-Linux~]# cat /etc/gshadow

①	②	③	④
root	:	:	:

① root	그룹 이름
②	암호화된 그룹의 비밀번호 정보
③	그룹의 관리자 목록
④	그룹의 구성원 목록



Linux 그룹 관리 명령어

- **groups 명령어**

- 사용자 그룹의 정보 확인

- 사용법

- [root@KH-Linux~]# groups [계정명]

- 계정명을 지정하지 않으면 현재 계정의 사용자 그룹 정보를 확인 함

- 계정명을 지정하면 지정된 계정의 사용자 그룹 정보를 확인 함



Linux 그룹 관리 명령어

- **groupadd 명령어**
 - 사용자 그룹 생성
 - 사용법
 - [root@KH-Linux~]# groupadd [옵션] [계정명]

OPTIONS	
-g	그룹 gid 설정
-o	중복 gid 생성 허용
-r	0~499 범위에서 자동으로 gid 값을 지정 함 (사용 가능한 큰 값부터 순차적으로 할당 됨)



Linux 그룹 관리 명령어

- **groupmod 명령어**
 - 사용자 그룹의 설정 변경
 - 사용법
 - [root@KH-Linux~]# groupmod [옵션] [계정명]

OPTIONS	
-g	그룹 gid 변경
-o	중복 gid 변경 허용
-n	새로운 그룹명으로 변경



Linux 그룹 관리 명령어

- **gpasswd 명령어**

- 사용자 그룹의 패스워드 설정
- 사용자 그룹의 관리자 지정 및 추가
- 사용자 그룹의 멤버 추가 및 삭제

- 사용법
 - [root@KH-Linux~]# gpasswd [옵션] [계정 이름] [그룹 이름]

OPTIONS	
-A	그룹의 관리자 추가
-a	그룹의 멤버 추가
-d	그룹의 멤버 삭제
-M	그룹의 멤버 수정



Linux 그룹 관리 명령어

- **groupdel 명령어**

- 사용자 그룹 삭제

- 사용법

- [root@KH-Linux~]# groupdel [그룹 이름]

Linux 계정 및 그룹 관리 실습

- 실습

- 사용자 계정 생성
- 모든 작업이 완료된 후 검사 받고 작업한 내용 삭제할 것

User : testuser

UID : 1200

GID : tg

Shell : sh

HOME : /hosting

Skel : /skelsam

Auto Creating Directory : public_html

Auto Creating File : .sam

권한[소유권, 허가권]

Linux Ownership(소유권)

- 소유권(Ownership)

- 개체에 대한 소유자와 소유그룹
- Linux 시스템의 모든 개체(파일 및 디렉터리)는 소유자와 소속그룹이 지정되어 있음
- 파일 생성 시 파일을 생성한 계정이 파일의 소유자가 됨
- 파일 생성 시 파일을 생성한 계정의 기본 그룹이 파일의 소유 그룹이 됨
- 명령어를 이용해 소유자와 소유그룹 변경 가능
- 확인
 - [root@KH-Linux~]# ls -l [파일 이름]

```
- rw- r- - r- - . 1 root root 46422 2016-01-30 15:11 install.log
```

소유자 소유그룹

- 소유권(Ownership) 관리 명령어

chown	파일의 소유권 변경
chgrp	파일의 소속그룹 소유권 변경

Linux Ownership(소유권)

- **chown 명령어**

- 개체(파일 및 디렉터리)의 소유자 및 소속그룹 변경
- 계정이름/그룹 이름 대신 UID/GID 로 사용 가능
- 사용법
 - [root@KH-Linux~]# chown [옵션] <계정 이름> <파일 이름>
 - [root@KH-Linux~]# chown [옵션] <:그룹 이름> <파일 이름>
 - [root@KH-Linux~]# chown [옵션] <계정 이름:그룹 이름> <파일 이름>

OPTIONS	
-R	하위 디렉터리 및 파일 전체의 소유권 변경



Linux Ownership(소유권)

- **chgrp 명령어**

- 개체(파일 및 디렉터리)의 소속그룹 변경
- 사용법
 - `[root@KH-Linux~]# chgrp [옵션] <gid 또는 그룹이름> <파일 이름>`

OPTIONS	
-R	하위 디렉터리 및 파일 전체의 소유권 변경

Linux Permission(허가권)

- 허가권(Permission)

- 멀티 유저 환경에서 **개체(파일, 디렉터리 등)에 접근할 수 있는 자격**
- 개체를 읽거나(read), 쓰거나(write), 실행(execute)하는 행위
- 사용자(user), 사용그룹(group), 나머지 사용자(other) 단위로 권한을 부여 함
- 확인

- [root@KH-Linux~]# ls -l [파일 이름]

```
-rw- r- - r- - . 1 root root 46422 2016-01-30 15:11 install.log
```

- 파일 형식을 제외한 10자리의 문자

- 허가권(Permission) 관리 명령어

chmod	파일의 허가권 변경
umask	개체가 생성될 때 부여되는 기본 permission 설정

Linux Permission(허가권)

• 허가권(Permission) 구분

- 앞에서부터 3자리씩 소유자(user), 소속그룹(group), 나머지 사용자(other)로 구분 함
- 마지막(10번째 문자) . 은 확장권한(권한 ACL)을 나타냄

①User	②Group	③Other	④DACL
rw-	r--	r--	.

①소유자(user)	파일의 소유자일 경우 적용되는 권한
②소속그룹(group)	파일의 소유 그룹 멤버 일 경우 적용되는 권한
③나머지 사용자(other)	소유자 또는 소속 그룹의 멤버가 아닌 나머지 사용자일 경우 적용되는 권한

• 허가권(Permission) 종류

문자	허가권	파일	디렉터리
r	읽기	파일 내용을 읽을 수 있는 권한	디렉터리 목록 확인 권한
w	쓰기	파일 내용을 쓰거나 삭제할 수 있는 권한	디렉터리 안에서 파일을 생성, 삭제 할 수 있는 권한
x	실행	실행파일(바이너리 파일)에 대한 실행 권한	디렉터리의 경로 변경, 디렉터리 안에 있는 항목을 읽을 수 있는 권한

Linux Permission(허가권)

- **허가권(Permission) 인증 절차**

- ① 사용자가 xxx.txt파일을 수정하기 위하여 쓰기 권한을 요청
- ② xxx.txt파일의 uid와 권한을 요청한 사용자의 uid를 비교
- ③ 일치하면 요청한 사용자를 파일의 소유주로 인정하고 소유주(user)의 permission 을 부여
- ④ 일치하지 않으면 gid를 비교
- ⑤ gid가 일치하면 사용자를 파일의 소속그룹멤버로 인정하고 소속그룹(group)에 할당된 permission을 부여
- ⑥ gid도 일치하지 않으면 나머지 사용자(other)에 해당하는 permission을 부여



Linux Permission(허가권)

- **chmod**

- 개체(파일 및 디렉터리)의 허가권 변경
- 상대모드 또는 절대모드로 허가권을 지정 함

- 사용법

- `[root@KH-Linux~]# chmod <상대모드 또는 절대모드 허가권> <파일 이름>`

Linux Permission(허가권)

• 허가권(Permission) 상대 모드

- <소유권> <연산자> <허가권>

소유권	
u	user
g	group
o	other
a	all

연산자	
+	권한 추가
-	권한 삭제
=	권한 지정

허가권	
r	읽기
w	쓰기
x	실행
s	소유자, 소속그룹만 실행

- 예제

현재 permission			실행	결과	설명
user	group	other			
rwX	rwX	rwX	# chmod a-x file	rw-rw-rw-	실행 권한 모두 제거
rwX	rw-	r--	# chmod u-x,o+w file	rw-rw-rw-	user(실행 제거), other(쓰기추가)
rwX	rw-	r--	# chmod u-x,g-w file	rw-r--r--	user(실행제거), group(쓰기제거)
rw-	rw-	rw-	# chmod a+x,g-w,o-w file	rwxr-xr-x	All(실행추가), group(쓰기제거), other(쓰기제거)

- **허가권(Permission) 절대 모드**

- 권한을 숫자 형태로 표현

소유자(user)			소속그룹(group)			나머지 사용자(other)		
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1
2^2	2^1	2^0	2^2	2^1	2^0	2^2	2^1	2^0

- 예제 (rw-r-xr-x)

소유자(user)			소속그룹(group)			나머지 사용자(other)		
r	w	x	r	w	x	r	w	x
4	2	1	4	2	1	4	2	1
4(r)	2(w)	-	4(r)	-	1(x)	4(r)	-	1(x)
4+2 = 6			4 + 1 = 5			4 + 1 = 5		
655								

Linux Permission(허가권)

- **umask**

- 리눅스 시스템에서 자원을 생성할 때 설정되는 기본 권한을 통합 관리하기 위해 사용
- **디렉터리는 777 권한, 파일은 666 권한**에서 **umask 값을 뺀 값**이 기본 권한으로 설정 됨
 - 파일은 기본 실행권한을 빼고 설정 됨
- 예제

디렉터리 최고 권한(777)	r	w	x	r	w	x	r	w	x
umask 값(022)	-	-	-	-	w	-	-	w	-
디렉터리 기본권한	r	w	x	r	-	x	r	-	x
	7			5			5		

파일 최고 권한(666)	r	w	-	r	w	-	r	w	-
umask 값(022)	-	-	-	-	w	-	-	w	-
파일 기본권한	r	w	-	r	-	-	r	-	-
	6			4			4		



Linux Permission(허가권)

- **umask**

- umask 값 확인
 - `[root@KH-Linux~]#umask`
- umask 값 변경
 - `[root@KH-Linux~]#umask [값]`
 - ex) `#umask 137`
- umask 값 영구 변경
 - `/etc/profile` 파일에서 설정

Linux 특수 권한 (Special Permission)

- 특수 권한(Special Permission)

- 시스템 관리의 효율성을 높이기 위해 사용하는 특별한 권한
- 보안의 취약점이 될 수 있으므로 꼭 필요한 경우가 아니면 사용하지 않는 것이 좋음

특수 권한	절대 모드	상대 모드
Setuid	4000	u+s
Setgid	2000	g+s
Sticky bit	1000	o+t

Linux 특수 권한 (special permission)

- Setuid (4000 = u+s)

- 파일을 실행하는 동안 해당 파일의 소유주의 권한으로 실행
- 일반적인 실행은 파일을 이용하려는 사용자의 소유권에 의해 실행권한이 부여되지만, setuid 권한이 적용되어 있는 파일은 실행되는 동안은 사용자가 아닌 파일의 소유주의 권한을 할당 받음
- 실행 파일에만 설정이 가능 함
- 소유자의 허가권에서 실행 권한이 x가 아닌 s로 표시 됨(예 : rwSr--r--)
- 예 (passwd)
 - 패스워드를 변경하면 /etc/shadow 파일의 읽기 및 쓰기 권한을 가지고 있어야지 패스워드를 기록할 수 있음
 - passwd 실행 파일에 setuid 권한이 부여 되어 있으므로 passwd 명령이 실행되는 동안은 소유주의 권한(root)으로 동작하게 됨

Linux 특수 권한 (special permission)

- Setgid (2000 = g+s)

- 파일을 실행하는 동안 해당 파일의 소속 그룹 권한으로 실행
- 파일이 실행되는 동안은 사용자가 아닌 파일의 소속 그룹 권한을 할당 받음
- 파일 및 디렉터리에서 설정 가능
- 소속 그룹의 허가권에서 실행 권한이 x가 아닌 s로 표시 됨(예 : rwxr-sr--)
- 디렉터리 적용
 - setgid가 적용된 디렉터리의 하위에 생성되는 디렉터리도 동일하게 setgid가 설정 됨
 - setgid가 적용된 디렉터리 안에서 새로운 디렉터를 생성하면 setgid에 명시된 소속그룹으로 설정 됨

Linux 특수 권한 (special permission)

- Sticky bit (1000 = o+t)

- 모든 소유권에 모든 허가권을 부여하지만 삭제는 소유주만 가능 함
- 디렉터리에 설정 함
- 나머지 사용자(other)의 허가권에서 실행 권한이 x가 아닌 t로 표시 됨(예 : rwxr-xr-t)
- /tmp, /var/tmp에 적용되어 있음
- 기본 소유권과 허가권을 먼저 확인하기 때문에 기본 허가권에 삭제권한이 있으면 삭제가 가능 함

파일시스템의 파일 속성

- 파일 속성(File Attribution)

- 특정 파일 시스템에서 지원하는 파일의 속성
- 파일의 손상을 방지하여 보호하기 위한 목적으로 사용 됨
- 허가되지 않은 사용자가 파일을 변경하지 못하게 하여 파일 보안을 제공 함

속성 종류	
a	파일에 추가로 쓰기만 가능(덮어쓰기 불가능)
c	디스크에 압축상태로 저장 함(자동 압축/해제)
d	Dump 명령에 의해 백업되지 않음(덤프 방지)
i	변경, 삭제, 이름변경, 추가쓰기, 링크 등이 모두 불가능 함
s	파일을 삭제할 때 블록이 모두 0으로 덮어써지고 다시 저장됨(안전한 삭제)
S	파일의 수정된 내용이 디스크상에 바로 동기화
u	삭제된 파일의 내용을 저장함(데이터 복구 가능)



파일 속성 관리 명령어

- 파일 속성 관리 명령어

lsattr	지정된 파일의 속성 확인
chattr	지정된 파일의 속성 변경



파일 속성 관리 명령어

- **lsattr**

- 파일에 설정된 속성 확인

- **사용법**

- `[root@KH-Linux~]# lsattr [옵션] [파일 이름]`

OPTIONS	
-R	지정된 디렉터리 하위의 모든 파일의 속성 정보 확인
-a	숨김 파일을 포함한 지정된 디렉터리 내부에 모든 파일의 속성 정보 확인
-d	디렉터리 구조를 무시하고 모두 일반파일 형식으로 속성 정보 확인
-v	파일의 버전 정보 확인

파일 속성 관리 명령어

- **chattr**

- 파일에 설정된 속성 확인

- **사용법**

- [root@KH-Linux~]# chattr [옵션] [mode] [파일 이름]

OPTIONS	
-R	지정된 디렉터리 하위의 모든 파일의 속성 정보 변경
-V	변경된 속성 정보에 대해 상세한 정보 출력
-f	오류 메시지 출력
-v version	파일의 버전 정보를 설정

속성 연산자(mode)	
+	속성을 추가 함
-	속성을 제거 함
=	정확히 지정된 속성만 설정 함

소프트웨어 관리



Linux 압축

- 용도

- 배포의 편리성 : 데이터의 빠르고 편리하게 송/수신하기 위해 사용 함
- 파일보관의 편리성 : 대용량의 데이터를 보관하기 위해 사용 함

- 압축 명령어 종류

- gzip(GNU zip)
- bzip2
- tar(tape archive)



Linux 압축

- **gzip**

- 리눅스에서 범용적으로 사용되는 압축 프로그램
- 일반적으로 tar로 묶여진(archive) 파일의 용량을 줄이기 위한 목적으로 사용 됨
- 압축 파일을 생성할 때 기본적으로 ".gz" 확장자를 붙여 줌
- 사용법
 - [root@KH-Linux~]# gzip [옵션] <파일 이름>

Linux 압축

- gzip 옵션

OPTIONS		
-n [1~9]	압축 수행 시 처리 속도 조정(1 : 가장 빠른 압축, 9 : 가장 느린 압축)	
-c	압축의 결과를 출력하고 원본파일을 보존 함	
-d	압축 해제(=gunzip)	
-f	시스템이 사용중인 파일을 압축하는 등의 강제 압축	
-l (소문자 L)	압축 파일의 정보 확인	
	compressed	압축 된 후의 용량
	uncompressed	압축 되기 전의 용량
	ratio	압축 효율
	uncompressed_name	압축 되지 전의 파일 이름
-r	한번에 여러 개의 파일의 압축(디렉터리에 포함된 모든 파일 압축)	
-t	압축 파일 테스트 (-v 옵션과 함께 사용해야 함)	
-v	자세한 정보 출력	

Linux 압축

- **bzip2**

- 압축률은 gzip보다 좋지만 처리속도가 느림
- 압축에 사용되는 알고리즘(BWT)의 특허 때문에 라이선스를 잘 확인하고 이용해야 함
- 압축 파일을 생성할 때 기본적으로 ".bz2" 확장자를 붙여 줌
- 사용법
 - [root@KH-Linux~]# bzip2 [옵션] <파일 이름>

OPTIONS	
-t	압축 파일 테스트
-d	압축 해제
-f	강제 압축
-v	자세한 정보 출력



Linux 압축

- **tar**

- 초기에 Tape Backup을 목적으로 사용 되었음
- 많은 양의 파일을 하나의 큰 파일로 묶어주기 위해 사용 됨
- 파일 배포 시 많이 사용 됨
- 내부 옵션을 통해 gzip 또는 bzip2 압축을 지원 함

- 사용법(압축 생성)
 - [root@KH-Linux~]# tar [옵션] <생성할 tar파일 이름> <묶을 파일 이름>

- 사용법(압축 해제)
 - [root@KH-Linux~]# tar [옵션] <tar파일 이름> -C <해제할 경로>



Linux 압축

- tar 옵션

OPTIONS	
-c	정의한 파일 이름으로 파일을 묶음
-f	옵션 뒤에 tar 파일을 명시하고 있음
-v	자세한 내용 출력
-t	tar 파일의 내용 출력
-x	tar 파일에서 정의된 파일을 추출
-u	이미 tar 파일에 포함된 파일보다 더 최근의 파일로 변경
-r	생성될 tar 파일의 제일 끝 부분에 tar 확장자 명시
-p	원본 파일의 permission 유지
-C	tar 파일이 해제될 위치 지정
-b	블록 단위로 작업
-g	증분 백업
-z	tar로 묶여진 파일을 gzip으로 압축하여 최종 파일을 생성
-j	tar로 묶여진 파일을 bzip2로 압축하여 최종 파일을 생성

Linux 압축

- tar 사용 예

- 지정한 파일 묶기
 - `[root@KH-Linux~]# tar -cvf <생성할 tar파일 이름> <묶을 파일 이름 또는 디렉터리>`
- 지정한 파일 묶기 + gzip 압축
 - `[root@KH-Linux~]# tar -zcvf <생성할 tar파일 이름> <묶을 파일 이름 또는 디렉터리>`
- 지정한 파일 묶기 + bzip2 압축
 - `[root@KH-Linux~]# tar -jcvf <생성할 tar파일 이름> <묶을 파일 이름 또는 디렉터리>`
- 지정한 파일 묶기 해제
 - `[root@KH-Linux~]# tar -xvf <해제할 tar파일 이름>`
- 지정한 파일 묶기 해제 + gzip 압축 해제
 - `[root@KH-Linux~]# tar -zxvf <해제할 tar파일 이름>`
- 지정한 파일 묶기 해제 + bzip2 압축 해제
 - `[root@KH-Linux~]# tar -jxvf <해제할 tar파일 이름>`



Linux Software 관리

- **Linux Software 관리**

- 운영체제를 설치할 때 초기에 설치되는 기본프로그램 외 관리자나 사용자가 추가로 필요한 소프트웨어를 설치할 필요가 있음
- 필요에 따라 **소프트웨어 설치 관리 및 제거**
- 사용자가 소프트웨어를 설치하고 관리할 수 있게 도와주는 프로그램 및 설치 방법이 다양함

- **CentOS**

- RPM(Redhat Package Manager)
- yum
- source code compile



Linux Software 관리 - CentOS

- rpm

- 패키지 배포 및 관리 프로그램
- 사용자가 미리 확보한 패키지 파일(.rpm)을 이용하여 원하는 프로그램을 설치하는 방식
- 장점
 - 패키지의 설치 및 제거가 편리함
 - 패키지의 업그레이드가 편리함
 - source code compile 방법 보다 설치 속도가 빠름
 - 설치할 패키지를 검증 할 수 있음
 - 질의를 통해 패키지의 자세한 정보 확인이 가능 함
- 단점
 - 사용자가 직접 필요한 패키지를 확보해야 함
 - > rpm 서버(phone.net), 설치 CD등을 통해 확보 함
 - 사용자가 직접 의존성을 해결해야 함

Linux Software 관리 - CentOS

- rpm 사용법(패키지 정보 확인)

- [root@KH-Linux~]# rpm -q[옵션]

OPTIONS	
-a	모든 패키지에 대해 질의
-f	파일에 대한 패키지 질의
-p	설치되거나 설치되지 않은 패키지 파일에 대한 질의
-i	패키지의 이름, 버전, 설명 등의 정보 확인
-R	대상 패키지와 의존성이 있는 패키지 목록 확인
--provider	패키지가 제공하는 기능 확인
-l(소문자 L)	패키지에 포함되어 있는 파일 정보 확인
-s	패키지에 포함되어 있는 파일의 상태 확인(normal : 정상, not installed : 설치되지 않음, replaced : 다른 파일로 교체)
-d	문서 파일만 확인
-c	설정 파일만 확인
--scripts	설치와 제거 과정에서 사용되는 shell script가 존재하면 내용 출력
--dump	수정일, MD5 체크섬, 모드, 소유자/소속그룹, 설정파일 여부, 문서파일 여부, 심볼릭 링크 여부 등 (-, -c, -d 옵션 중 하나는 반드시 함께 사용해야 함)

Linux Software 관리 - CentOS

- rpm 사용법(패키지 설치)

- [root@KH-Linux~]# rpm -i[옵션] <rpm 패키지 이름>

OPTIONS	
-v, --vv	설치 패키지의 메시지를 출력(더 자세히)
-h	패키지를 설치할 때 #(해쉬 마크) 출력
--replacepkgs	패키지 교체(이전 버전이 설치되어 있어도 새로 설치)
--replacefiles	동일한 파일명이 있으면 교체
--oldpackage	예전 패키지로 교체
--force	--replacepkgs, --replacefiles, --oldpackage 옵션을 모두 사용 함
--precent	패키지의 파일을 설치할 때 진행률을 %로 표시
--includedocs	문서 파일을 포함하여 설치
--excludedocs	문서 파일을 제외하고 설치
--nodeps	패키지의 의존성을 무시하고 설치
--aid	패키지의 의존성을 검사하여 의존성이 필요한 패키지를 자동으로 설치
--test	패키지를 실제 설치하지 않고 충돌사항만 점검하여 확인

Linux Software 관리 - CentOS

- rpm 사용법(패키지 업데이트)

- [root@KH-Linux~]# rpm -U[옵션] <rpm 패키지 이름>

OPTIONS	
-v, --vv	설치 패키지의 메시지를 출력(더 자세히)
-h	패키지를 설치할 때 #(해쉬 마크) 출력
--replacepkgs	패키지 교체(이전 버전이 설치되어 있어도 새로 설치)
--replacefiles	동일한 파일명이 있으면 교체
--oldpackage	예전 패키지로 교체
--force	--replacepkgs, --replacefiles, --oldpackage 옵션을 모두 사용 함
--precent	패키지의 파일을 설치할 때 진행률을 %로 표시
--includedocs	문서 파일을 포함하여 설치
--excludedocs	문서 파일을 제외하고 설치
--nodeps	패키지의 의존성을 무시하고 설치
--aid	패키지의 의존성을 검사하여 의존성이 필요한 패키지를 자동으로 설치
--test	패키지를 실제 설치하지 않고 충돌사항만 점검하여 확인



Linux Software 관리 – CentOS

- rpm 사용법(패키지 삭제)

- [root@KH-Linux~]# rpm -e[옵션] <rpm 패키지 이름>

OPTIONS	
--nodeps	의존성 여부를 무시하고 삭제
--test	충동 사항을 점검



Linux Software 관리 - CentOS

- yum

- 패키지 배포 및 관리 프로그램
- 네트워크에 존재하는 패키지 서버(repository)로부터 필요한 패키지를 다운하여 설치까지 완료 함
- 장점
 - rpm의 의존성 문제를 해결 함
 - 패키지를 별도로 확보할 필요가 없음
 - 자동화 되어 있는 설치 및 삭제를 지원
 - update된 패키지를 검사하여 최신 패키지를 설치 함
- 단점
 - 네트워크 연결이 반드시 필요 함
 - repository에서 서비스를 중단하면 서비스를 받을 수 없음
 - 불필요한 패키지가 설치될 가능성이 있음

Linux Software 관리 - CentOS

- yum 사용법

- [root@KH-Linux~]# yum [옵션] <mode> <패키지 이름>

OPTIONS	
-y	설치 여부를 확인하지 않고 바로 설치
MODE	
install	패키지 설치
check-update	설치된 패키지 중 업데이트가 가능한 항목을 확인 함
update	패키지 업데이트
remove	패키지 삭제
info	패키지 정보 확인
localinstall	미리 확보한 rpm 패키지 설치



Linux Software 관리 - CentOS

- **yum repository(저장소) 관리**

- 환경설정 파일 경로
 - /etc/yum.repos.d/CentOS-Base.rep
- 관리 항목
 - mirrolist : 패키지가 설치된 사이트들의 주소 리스트가 있는 사이트 주소
 - #baseurl : 실제 패키지가 설치된 사이트 주소
- mirror list 정보 확인
 - <http://www.centos.org/download/mirrors/>

Linux Software 관리 - CentOS

- yum repository(저장소) 관리

- 분류

[base]	yum의 기본 repository 경로 설정
[update]	업데이트된 패키지를 위한 경로 설정
[extras]	사용중인 패키지와 관련된 추가 패키지의 경로 설정
[centosplus]	존재하는 패키지들의 기능적으로 확장된 추가 패키지의 경로 설정
[contrib]	사용자에 의해 제작된 패키지 경로 설정

- 세부 항목

name	Repository의 이름
mirrorlist	mirrorlist에 등록된 주소로 접근하여 해당 repository가 제공하는 패키지를 자동으로 설치 및 업데이트 함
baseurl	직접 repository의 주소 지정
enable	지정된 repository 사용 여부
gpgcheck	gpg(패키지 변조 확인) 사용 여부

Linux Software 관리 - CentOS

- **source code compile**

- rpm으로 제공되지 않는 프로그램을 설치해야 할 때 사용 함
- 일반적으로 소스코드 내부에 주석으로 설치방법이 명시되어 있거나, readme 또는 install 이라는 파일을 통해 설치 방법을 제공 함
- 컴파일러가 반드시 필요 함
 - C 컴파일러(gcc 패키지), C++ 컴파일러(gcc=c++ 패키지)
- 장점
 - 설치가 진행될 때 전체 과정을 제어하며 세부 환경설정 및 설치 경로 지정이 가능 함
- 단점
 - 컴파일 환경을 미리 준비해야 함
 - 정확한 방법을 숙지하지 않으면 설치 시 에러가 많이 발생 함
 - 설치 및 관리 시간이 오래 걸림
 - 검증되지 않은 불법 소프트웨어가 설치 될 가능성이 있음



Linux Software 관리 - CentOS

- **source code compile 설치**

1. 준비 단계

- 컴파일러 설치 : gcc, gcc-c++, glibc(또는 glibc*)

2. 설치를 원하는 소프트웨어의 source code 확보

- 직접 다운로드 또는 wget을 이용한 다운로드
- wget 사용법
 - > `[root@KH-Linux~]# wget <소프트웨어의 다운로드 웹 주소>`



Linux Software 관리 - CentOS

- **source code compile 설치**

- 3. 설치

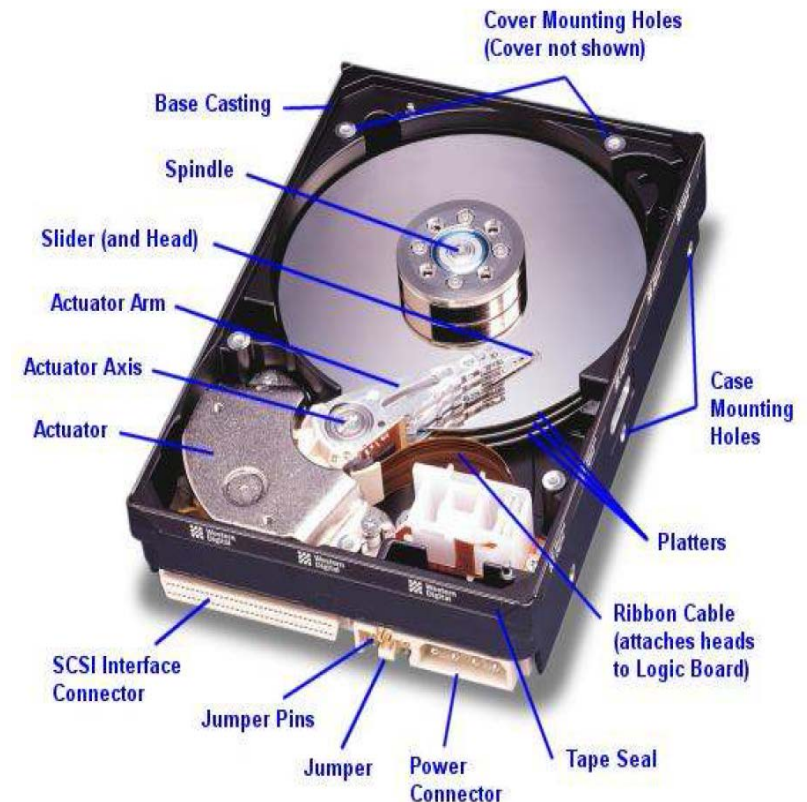
- 컴파일 환경설정
 - > **[root@KH-Linux~]# ./configure**
 - > 소프트웨어 설치를 위한 시스템의 환경정보 설정 단계
 - > 도움말 확인 : `configure --help`
 - 컴파일
 - > **[root@KH-Linux~]# make**
 - > 정상적으로 컴파일을 완료한 후에 설치파일을 실행 해야 함
 - > 오류가 발생했을 때 실패한 컴파일 삭제 : `make clean`
 - 컴파일 된 소프트웨어 설치
 - > **[root@KH-Linux~]# make install**

디스크 관리

Linux 디스크 관리

• HDD(Hard Disk Drive) 물리적 구조

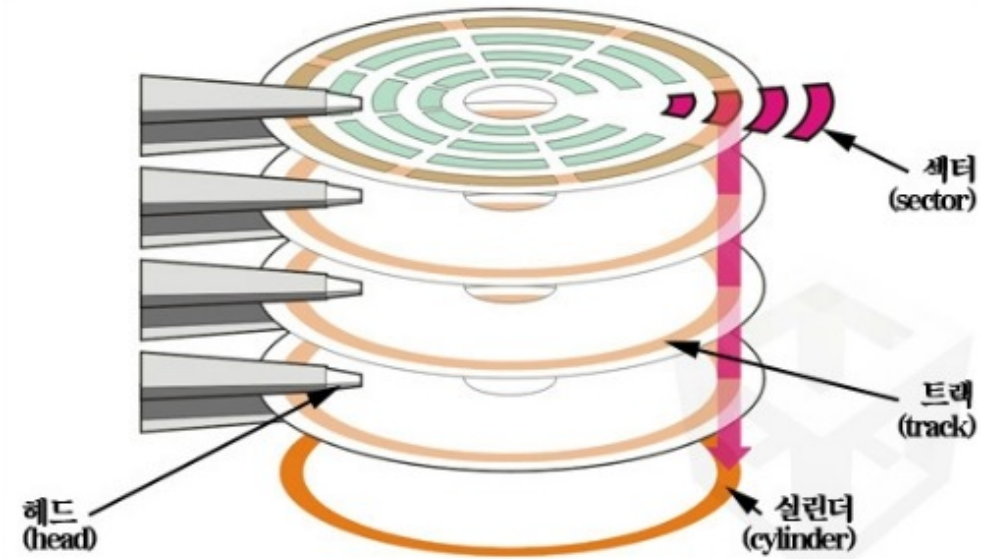
- 플래터(Platter)
 - 원형 판 모양의 데이터를 저장하는 위치
- 스피들(Spindle)
 - 플래터가 회전할 수 있도록 모터와 직접 연결된 축
- 헤드(Head)
 - 플래터 표면에 자성을 이용하여 데이터를 저장/삭제/읽어오는 장치
 - 플래터가 양면으로 코팅되기 때문에 헤드의 수는 플래터의 2배
- 액츄에이터(Actuator)
 - 헤드를 움직이는 장치
 - 액츄에이터 암을 움직여서 헤드가 지정된 위치로 이동하게 함
- 카트리지(Cartridge)
 - 플래터, 스피들, 헤드, 액츄에이터를 밀봉하는 알루미늄 케이스
- 회로기판(PCB)
 - 하드디스크 하부에 위치



Linux 디스크 관리

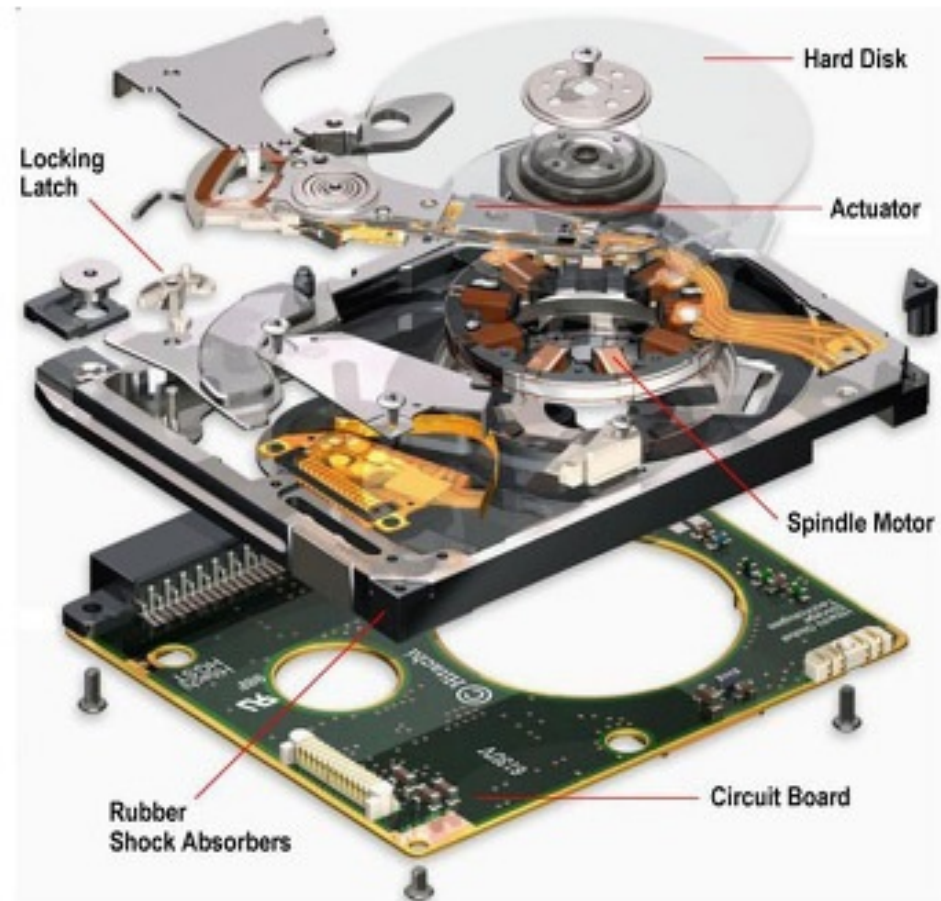
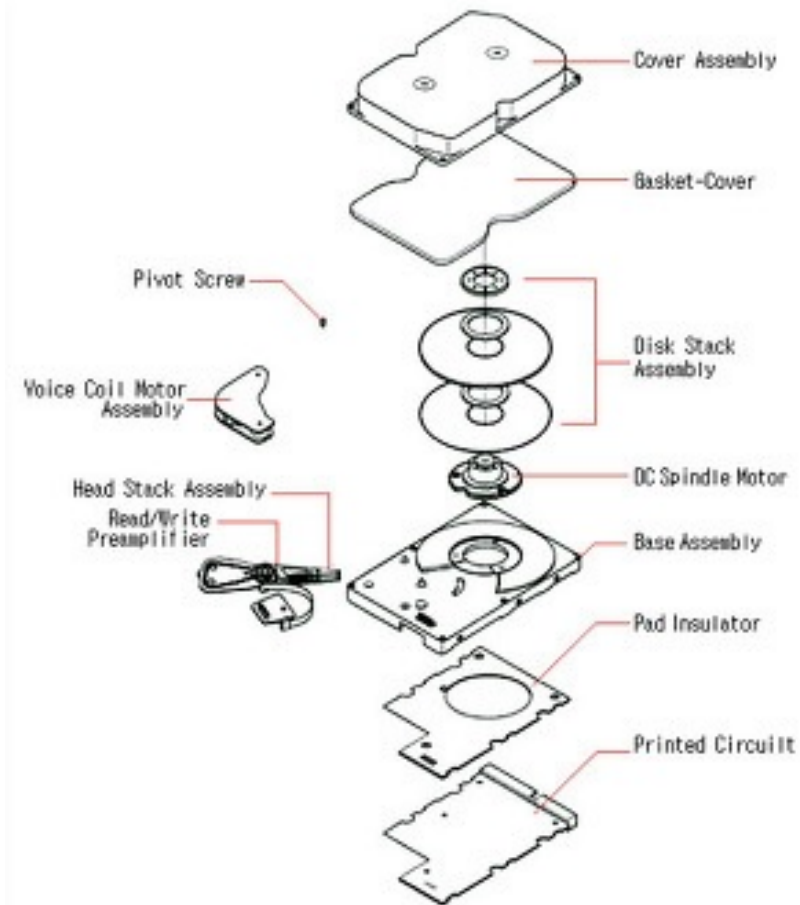
- HDD(Hard Disk Drive) 논리적 구조

- 트랙(Track)
 - 플래터의 동심원
- 실린더(Cylinder)
 - 여러 개의 플래터의 트랙을 수직으로 관통하는 영역
- 섹터(Sector)
 - 트랙에서 하나의 실린더 영역
 - 하드 디스크의 최소 단위(=512bytes)
- 클러스터(Cluster)/블록(Block)
 - 섹터의 논리적인 영역(=4096bytes)
 - 하드 디스크가 읽고 쓰는 "논리적인" 최소 단위
 - Windows = Cluster / Linux = Block
 - 디스크 포맷 시 지정 가능



Linux 디스크 관리

- HDD(Hard Disk Drive) 구조



Linux 디스크 관리

- **df**

- disk free
- 파일 시스템에 할당된 전체 용량, 사용량, 사용 가능 용량, 사용률, 마운트 정보 등
- 일반 사용자로도 사용 가능

- **사용법**

- [root@KH-Linux~]# df [옵션]

OPTIONS	
-a	모든 파일 시스템의 정보 확인
-i	Size 대신 inode 사용 정보 확인
-k	단위를 KB로 확인(default)
-h	단위를 MB로 확인
-t	지정한 종류의 파일 시스템 확인
-x	지정한 종류의 파일 시스템을 제외한 정보 확인

Linux 디스크 관리

- df(디스크 용량을 MB(Mega Byte) 단위로 표시)

```
[root@localhost ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       32G   3.6G   27G   12% /
tmpfs           495M  224K  495M    1% /dev/shm
/dev/sda2       2.0G   3.1M   1.9G    1% /home
/dev/sda3       2.0G  283M   1.6G   16% /tmp
/dev/sda5       2.0G  375M   1.5G   21% /var
```



Linux 디스크 관리

- **du**
 - disk usage
 - 파일 및 디렉터리의 용량 확인
- **사용법**
 - [root@KH-Linux~]# du [옵션] [파일 또는 디렉터리 이름]

OPTIONS	
-a	파일까지 확인(default : 디렉터리)
-s	전체 용량의 합계를 확인
-b	단위를 byte로 확인
-k	단위를 KB로 확인
-l	하드 링크된 파일까지 확인
-h	용량 단위 표시

Linux 디스크 관리

- du(디렉터리 전체 사용량 확인)

```
[root@localhost ~]# du -sh /var
372M    /var
```

- du(개별 디렉터리 사용량 확인)

```
[root@localhost ~]# du -sh /var/*
4.0K    /var/account
284M    /var/cache
4.0K    /var/crash
4.0K    /var/cvs
8.0K    /var/db
8.0K    /var/empty
4.0K    /var/games
4.0K    /var/gdm
71M     /var/lib
4.0K    /var/local
```




Linux 디스크 관리

- **fdisk**
 - file system disk
 - 파티션 테이블을 생성, 수정, 삭제
- **사용법(현재 파일 시스템의 파티션 정보 확인)**
 - [root@KH-Linux~]# fdisk -l
- **사용법(파티션 생성)**
 - [root@KH-Linux~]# fdisk [디스크 장치이름]
 - Windows의 diskpart처럼 디스크 관리 모드가 지원 됨

Linux 디스크 관리

- fdisk(현재 시스템의 파티션 테이블 정보 확인)

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

```
Disk /dev/sda: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0007303a
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	4178	33553408	83	Linux
/dev/sda2		4178	4439	2097152	83	Linux
/dev/sda3		4439	4700	2097152	83	Linux
/dev/sda4		4700	5222	4194304	5	Extended
/dev/sda5		4700	4961	2097152	83	Linux
/dev/sda6		4961	5222	2095104	82	Linux swap / Solaris

Linux 디스크 관리

- 리눅스를 사용하는 시스템에 하드 디스크 추가

- 시스템 종료
- VMware Settings를 이용하여 SCSI 방식의 하드디스크 추가(40GB)
- 리눅스 시스템 로그인 후 시스템의 파티션 테이블 확인(fdisk -l)

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

```
Disk /dev/sda: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x0007303a
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	4178	33553408	83	Linux
/dev/sda2		4178	4439	2097152	83	Linux
/dev/sda3		4439	4700	2097152	83	Linux
/dev/sda4		4700	5222	4194304	5	Extended
/dev/sda5		4700	4961	2097152	83	Linux
/dev/sda6		4961	5222	2095104	82	Linux swap / Solaris

```
Disk /dev/sdb: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000
```

Linux 디스크 관리

- **fdisk를 이용하여 /dev/sdb 장치에 파티션 설정**

- fdisk /dev/sdb

```
[root@localhost ~]# fdisk /dev/sdb
```

```
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel  
Building a new DOS disklabel with disk identifier 0x1482e4b3.
```

```
Changes will remain in memory only, until you decide to write them.
```

```
After that, of course, the previous content won't be recoverable.
```

```
Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)
```

```
WARNING: DOS-compatible mode is deprecated. It's strongly recommended to  
switch off the mode (command 'c') and change display units to  
sectors (command 'u').
```

```
Command (m for help): █
```



Linux 디스크 관리

- fdisk 관리 모드 설정 명령

OPTIONS	
a	부팅 파티션을 설정
d	파티션 삭제
l	설정 가능한 파티션 타입 출력
m	파티션 설정 도움말 출력
n	새로운 파티션 생성
p	현재 설정되어 있는 파티션 정보 출력
t	파티션 타입 변경
q	fdisk 메뉴 빠져나가기
w	작업 내용 저장

Linux 디스크 관리

- 1번 파티션에 주파티션 생성(2GB)

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-5221, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-5221, default 5221): +2G
```

- 2번 파티션에 확장 파티션 생성(4GB)

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
e
Partition number (1-4): 2
First cylinder (263-5221, default 263):
Using default value 263
Last cylinder, +cylinders or +size{K,M,G} (263-5221, default 5221): +4G
```

Linux 디스크 관리

- 확장 파티션에 논리 파티션 생성(2GB)

```
Command (m for help): n
Command action
  l   logical (5 or over)
  p   primary partition (1-4)
l
First cylinder (263-785, default 263):
Using default value 263
Last cylinder, +cylinders or +size{K,M,G} (263-785, default 785): +2G
```

- 파티션 정보 확인

```
Command (m for help): p

Disk /dev/sdb: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x1482e4b3
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	262	2104483+	83	Linux
/dev/sdb2		263	785	4200997+	5	Extended
/dev/sdb5		263	524	2104483+	83	Linux

Linux 디스크 관리

- 파티션 타입 변경(5번 파티션을 Swap 타입으로 설정)

```
Command (m for help): t
Partition number (1-5): 5
Hex code (type L to list codes): 82
Changed system type of partition 5 to 82 (Linux swap / Solaris)
```

```
Command (m for help): p
```

```
Disk /dev/sdb: 42.9 GB, 42949672960 bytes
255 heads, 63 sectors/track, 5221 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x1482e4b3
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	262	2104483+	83	Linux
/dev/sdb2		263	785	4200997+	5	Extended
/dev/sdb5		263	524	2104483+	82	Linux swap / Solaris



Linux 디스크 관리

- 설정 정보 저장 후 디스크 관리 모드 종료

```
Command (m for help): wq
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```



Linux 디스크 관리

- **mkfs**
 - 파일 시스템 생성
- **사용법**
 - [root@KH-Linux~]# mkfs [옵션] [장치 이름]

OPTIONS	
-t [종류]	파일 시스템의 종류 선택
-c	파일 시스템을 생성하기 전에 bad block을 검사 함
-v	작업 상태와 결과를 자세히 출력 함

Linux 디스크 관리

- 파일 시스템 생성(/dev/sdb1 장치에 ext4 파일 시스템 생성)

```
[root@localhost ~]# mkfs -t ext4 /dev/sdb1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
131648 inodes, 526120 blocks
26306 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=541065216
17 block groups
32768 blocks per group, 32768 fragments per group
7744 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 29 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Linux 디스크 관리 - 마운트(Mount)

- **마운트(Mount)**

- 운영체제가 물리적인 장치를 이용할 수 있도록 연결 함
- Linux OS는 모든 장치를 파일 단위로 관리하기 때문에 새로 추가된 장치는 임의의 디렉터리(Mount Point)에 연결 시켜서 사용해야 함

- **마운트 포인트**

- 하드 디스크를 운영체제로 연결할 때 사용한 디렉터리
- 사용 중이던 디렉터리를 마운트 포인트로 이용할 경우 존재하던 파일과 디렉터리에 접근할 수 없게 되므로 마운트 포인트는 비어 있는 디렉터를 사용(마운트 포인트 해제 시 기존의 파일 및 디렉터리에 접근 가능)

Linux 디스크 관리 - 마운트(Mount)

- **mount**

- 파일 시스템 마운트 명령

- **사용법**

- [root@KH-Linux~]# mount [-t 파일시스템 유형] [-o 옵션] [장치 이름] [마운트 포인트]

OPTIONS	
async	마운트 된 파일시스템에 비동기 입출력을 사용
auto	/etc/fstab에 지정된 파일시스템에 대해 부팅 시에 자동으로 마운트
defaults	rw, suid, dev, exec, auto, nouser, async를 종합적으로 사용
dev	해당 파일시스템을 문자 디바이스나 블록 디바이스를 이용해 해석
exec	파일 시스템에 포함된 프로그램을 실행할 수 있도록 함
noauto	자동 마운트가 되지 않도록 함
noexec	해당 파일 시스템의 프로그램이 실행되지 않도록 함, 특정 보안 목적을 위해 사용
nosuid	실행 파일에 존재하는 suid, sgid 비트의 기능 제한
nouser	루트(root) 외의 사용자가 파일 시스템을 마운트 하거나 언마운트 하는 것을 제한

Linux 디스크 관리 - 마운트(Mount)

- **mount**

- 파일 시스템 마운트 명령

- **사용법**

- [root@KH-Linux~]# mount [-t 파일시스템 유형] [-o 옵션] [장치 이름] [마운트 포인트]

OPTIONS	
ro	읽기 전용으로 마운트
rw	읽기와 쓰기 가능하도록 마운트
suid	실행 파일에 존재하는 suid, sgid 비트의 기능을 사용
sync	마운트 된 파일시스템에 동기식 입출력을 사용
user	일반 사용자의 파일시스템 마운트 허용
users	모든 일반 사용자가 파일 시스템을 마운트, 언마운트 가능하도록 허용
noatime	access time을 기록하지 않음, 자주 파일에 액세스 할 경우 유용

Linux 디스크 관리 - 마운트(Mount)

- 파일 시스템 마운트

```
[root@localhost ~]# mkdir /data
[root@localhost ~]# mount /dev/sdb1 /data
[root@localhost ~]# mount
/dev/sda1 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sda2 on /home type ext4 (rw)
/dev/sda3 on /tmp type ext4 (rw)
/dev/sda5 on /var type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
vmware-vmblock on /var/run/vmblock-fuse type fuse.vmware-vmblock (rw,nosuid,nodev,default_permissions,allow_other)
gvfs-fuse-daemon on /root/.gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev)
/dev/sdb1 on /data type ext4 (rw)
```



Linux 디스크 관리 - 마운트(Mount)

- **umount**

- 마운트 해제
- 운영체제와 장치의 연결을 해제 함

- **사용법**

- [root@KH-Linux~]# umount [마운트 포인터]

Linux 디스크 관리 - 마운트(Mount)

- 파일 시스템 언마운트

```
[root@localhost ~]# umount /data
[root@localhost ~]# mount
/dev/sda1 on / type ext4 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
tmpfs on /dev/shm type tmpfs (rw)
/dev/sda2 on /home type ext4 (rw)
/dev/sda3 on /tmp type ext4 (rw)
/dev/sda5 on /var type ext4 (rw)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
vmware-vmblock on /var/run/vmblock-fuse type fuse.vmware-vmblock (rw,nosuid,nodev,default_permissions,allow_other)
gvfs-fuse-daemon on /root/.gvfs type fuse.gvfs-fuse-daemon (rw,nosuid,nodev)
```



Linux 디스크 관리 - 마운트(Mount)

- 기타 마운트 명령

- CD-ROM 마운트

- `mount -t iso9660 /dev/cdrom [마운팅 포인트]`
 - `mount -t iso9660 /dev/cdrom /media/cdrom`

- ISO 이미지 마운트

- `mount -t iso9660 [ISO 이미지 파일] [마운팅 포인트]`
 - `mount -o loop /root/CentOS-6.7.iso /media/iso`



Linux 디스크 관리 실습

- 실습

- 20GB 크기를 가진 하드디스크 추가 장착
- 1번 파티션은 기본 파티션으로 8GB 용량으로 분할
- 2번 파티션은 확장 파티션으로 12GB 용량으로 분할 후 4GB를 가지는 논리 파티션 추가
- 각각 두 개의 파티션을 ext4 파일시스템 유형으로 포맷
- 기본 파티션은 /game에 마운트
- 논리 파티션은 /music에 마운트

Linux 디스크 관리 - 마운트(Mount)

- 파일 시스템 마운트 관리 파일(/etc/fstab)

- 리눅스가 부팅되면서 파일 시스템을 어디에 자동으로 마운트하고, 외부 장치들에 대한 마운트를 어떻게 설정하는지, 권한 및 복구 등의 옵션을 어떻게 이용할 지 지정하는 파일
- 시스템 부팅 시 /etc/fstab에 기록되어 있는 순서대로 파티션이 마운트 되어 한 개의 디렉터리 트리가 만들어 짐

```
1
2 #
3 # /etc/fstab
4 # Created by anaconda on Tue Aug 23 02:53:26 2016
5 #
6 # Accessible filesystems, by reference, are maintained under '/dev/disk'
7 # See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
8 #
9 UUID=3949d150-ce85-47e7-a594-763be9bb9c01 / ext4 defaults 1 1
10 UUID=e521f654-01b6-453d-b6aa-e34c0aa12560 /home ext4 defaults 1 2
11 UUID=45ebfe84-72d3-4fb7-a41e-e716b9a26ea4 /tmp ext4 defaults 1 2
12 UUID=bf56e302-cae4-4e8f-87de-e49df185a985 /var ext4 defaults 1 2
13 UUID=4a00b0fb-32e9-4c78-a795-83e961701542 swap swap defaults 0 0
14 tmpfs /dev/shm tmpfs defaults 0 0
15 devpts /dev/pts devpts gid=5,mode=620 0 0
16 sysfs /sys sysfs defaults 0 0
17 proc /proc proc defaults 0 0
18 /dev/sdb1 /data ext4 defaults 1 2
```

Linux 디스크 관리 - 마운트(Mount)

- **fstab 확인**

- [root@KH-Linux~]# cat /etc/fstab
- ①UUID=ec78d766-639a-4c6e-80a5-627449f11768 ②/ ③ext4 ④defaults ⑤1 ⑥1

① UUID=ex ...	장치 명(Label로 지정할 수도 있음)	
② /	마운트 포인터	
③ ext4	장치의 파일 시스템 종류	
④ defaults	마운트 할 때 사용하는 옵션 종류	
⑤ 1	덤프의 필요 여부를 지정 함(하드 디스크 : 1 , 그 외 : 0)	
⑥ 1	fsck 명령(무결성 체크)의 수행 여부	
	0	체크하지 않음
	1이상	체크 함(숫자가 작은 순서대로 체크 함)
	1	최상위 파티션
	2	그 외의 파티션의 번호를 지정하여 순서대로 체크 함

Linux 디스크 관리 - UUID

- **UUID(Universally Unique Identifier)**

- 16Byte(128Bit)로 이루어진 규격화된 숫자
- 네트워크 상에서 서로 모르는 개체들을 식별하고 구별하기 위한 고유한 이름
- 중앙 관리 시스템이 없는 분산 시스템에서 정보를 유일하게 식별하기 위한 값

- **UUID 확인**

- blkid

```
[root@localhost ~]# blkid
/dev/sda1: UUID="3949d150-ce85-47e7-a594-763be9bb9c01" TYPE="ext4"
/dev/sda2: UUID="e521f654-01b6-453d-b6aa-e34c0aa12560" TYPE="ext4"
/dev/sda3: UUID="45ebfe84-72d3-4fb7-a41e-e716b9a26ea4" TYPE="ext4"
/dev/sda5: UUID="bf56e302-cae4-4e8f-87de-e49df185a985" TYPE="ext4"
/dev/sda6: UUID="4a00b0fb-32e9-4c78-a795-83e961701542" TYPE="swap"
/dev/sdb1: UUID="1bf97d09-dc4f-47cc-ae5d-d8d9176d63db" TYPE="ext4"
```

RAID

Linux RAID

- **RAID(Redundant Array of Inexpensive Disks)**

- 장애 발생 요인을 최대한 제거한 고성능 무정지 저장 장치
- 여러 개의 Disk를 하나의 가상 Disk로 구성, 대용량 저장 장치 구축 가능
- 다수의 Disk에 Data를 분할, 병렬 전송함으로 써 Data 전송 속도 향상
- System 가동 중 Disk Fault 시에도 시스템 정지 없이 새로운 Disk로 교체하면 Data 자동 복구
- 하드웨어 RAID와 소프트웨어 RAID 두 가지 방식이 있음

- **RAID 패키지 설치 확인**

- **#rpm -qa mdadm**
- 패키지가 없을 경우 yum으로 설치

```
[root@localhost ~]# rpm -qa mdadm
mdadm-3.3.2-5.el6.x86_64
```


Linux RAID

- 환경 구성

- 기존 추가 하드디스크 제거(주의: Auto Mounting 설정 제거)
- SCSI 방식의 4개의 하드디스크(8G) 추가

Hardware Options	
Device	Summary
Memory	1 GB
Processors	1
Hard Disk (SCSI)	40 GB
New Hard Disk (SCSI)	8 GB
New Hard Disk (SCSI)	8 GB
New Hard Disk (SCSI)	8 GB
New Hard Disk (SCSI)	8 GB
CD/DVD (IDE)	Auto detect
Network Adapter	Custom (VMnet8)
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

하드디스크 구성	
/dev/sdb	8GB
/dev/sdc	8GB
/dev/sdd	8GB
/dev/sde	8GB



Linux RAID

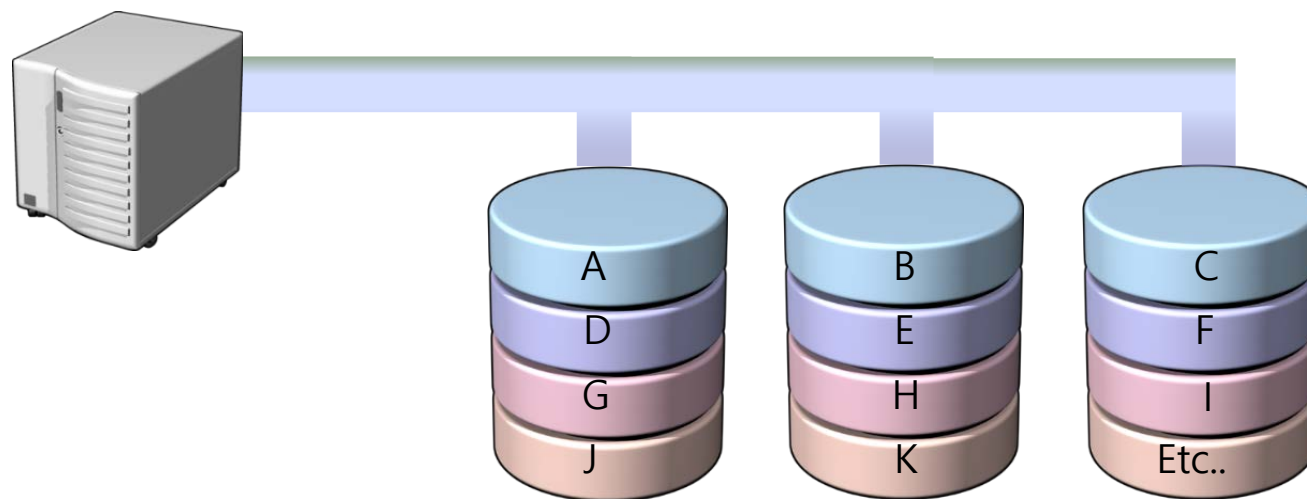
- **mdadm**
 - Software RAID 관리 명령
 - 사용법
 - `[root@KH-Linux~]# mdadm --create [RAID 장치명] --level=[RAID 레벨] --raid-device=[장치 수] [장치명1] [장치명2].....`

OPTION	
--create	Raid 장치 생성
--stop	Raid 중지
--level	Raid 종류 설정
--detail	Raid 장치 상세 정보 출력
--scan	Raid 장치 검색
-a	Raid 장치에 Disk 추가
-r	Raid 장치에 Disk 삭제

Linux RAID (RAID-0)

- **Striping(RAID-0)**

- 디스크 최소 2개 이상 필요
- 모든 볼륨의 최적의 성능(속도)을 제공
- 내결함성이 없음
- 확장 및 축소 불가능



Linux RAID (RAID-0)

- **RAID-0 구성**

- 1) 동일한 용량을 갖는 다수의 파티션 생성(RAID 타입)
- 2) RAID 0 레벨의 RAID 구성
- 3) RAID 장치 포맷
- 4) RAID 장치 마운트

하드디스크 구성	
/dev/sdb1	1GB
/dev/sdc1	1GB

Linux RAID (RAID-0)

• 1) 동일한 용량을 갖는 다수의 파티션 생성(RAID 타입)

- 파티션 생성 -> 리눅스 RAID 자동 탐지 타입으로 변경
- fdisk /dev/sdb -> n -> p -> 1 -> enter -> +1G -> t -> fd -> p -> wq
- fdisk /dev/sdc -> n -> p -> 1 -> enter -> +1G -> t -> fd -> p -> wq

하드디스크 구성	
/dev/sdb1	1GB
/dev/sdc1	1GB

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1044, default 1):
Using default value 1
Last cylinder, +cylinders or +size{K,M,G} (1-1044, default 1044): +1G

Command (m for help): t
Selected partition 1
Hex code (type L to list codes): fd
Changed system type of partition 1 to fd (Linux raid autodetect)

Command (m for help): p

Disk /dev/sdb: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x5ba232a6
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	132	1060258+	fd	Linux raid autodetect

Linux RAID (RAID-0)

• 2) RAID 0 레벨의 RAID 구성

- mdadm --create /dev/md0 --level=[RAID레벨] --raid-devices=[RAID 구성 수] [장치1] [장치2]
- mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb1 /dev/sdc1

```
[root@localhost ~]# mdadm --create /dev/md0 --level=0 --raid-devices=2 /dev/sdb1 /dev/sdc1
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
[root@localhost ~]# cat /proc/mdstat
Personalities : [raid0]
md0 : active raid0 sdc1[1] sdb1[0]
      2117632 blocks super 1.2 512k chunks

unused devices: <none>
[root@localhost ~]# mdadm --detail /dev/md0
/dev/md0:
   Version : 1.2
  Creation Time : Fri Nov  4 04:53:23 2016
    Raid Level : raid0
   Array Size : 2117632 (2.02 GiB 2.17 GB)
  Raid Devices : 2
 Total Devices : 2
 Persistence : Superblock is persistent
```

Linux RAID (RAID-0)

- 3) RAID 장치 포맷

- mkfs -t ext4 /dev/md0

```
[root@localhost ~]# mkfs -t ext4 /dev/md0
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=128 blocks, Stripe width=256 blocks
132464 inodes, 529408 blocks
26470 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=545259520
17 block groups
32768 blocks per group, 32768 fragments per group
7792 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 39 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Linux RAID (RAID-0)

- 4) RAID 장치 마운트

- mkdir /raid0
- mount /dev/md0 /raid0

```
[root@localhost ~]# mkdir /raid0
[root@localhost ~]# mount /dev/md0 /raid0
[root@localhost ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	32G	3.6G	27G	12%	/
tmpfs	495M	228K	495M	1%	/dev/shm
/dev/sda2	2.0G	3.1M	1.9G	1%	/home
/dev/sda3	2.0G	283M	1.6G	16%	/tmp
/dev/sda5	2.0G	374M	1.5G	21%	/var
/dev/md0	2.0G	3.1M	1.9G	1%	/raid0

Linux RAID (RAID-0)

- RAID 자동 연결 및 자동 마운트

- 1) RAID 자동 구성

- 구성한 설정은 부팅 시 사라지게 되므로, 부팅 시 RAID 설정이 구성되도록 /etc/mdadm.conf에 RAID 설정 구성
 - mdadm --detail --scan >> /etc/mdadm.conf

```
[root@localhost ~]# mdadm --detail --scan >> /etc/mdadm.conf
[root@localhost ~]# cat /etc/mdadm.conf
ARRAY /dev/md0 metadata=1.2 name=localhost.localdomain:0 UUID=cd654336:e002ed23:8c7f08f7:8c2ef30b
```

- 2) /etc/fstab에 마운트 정보 추가

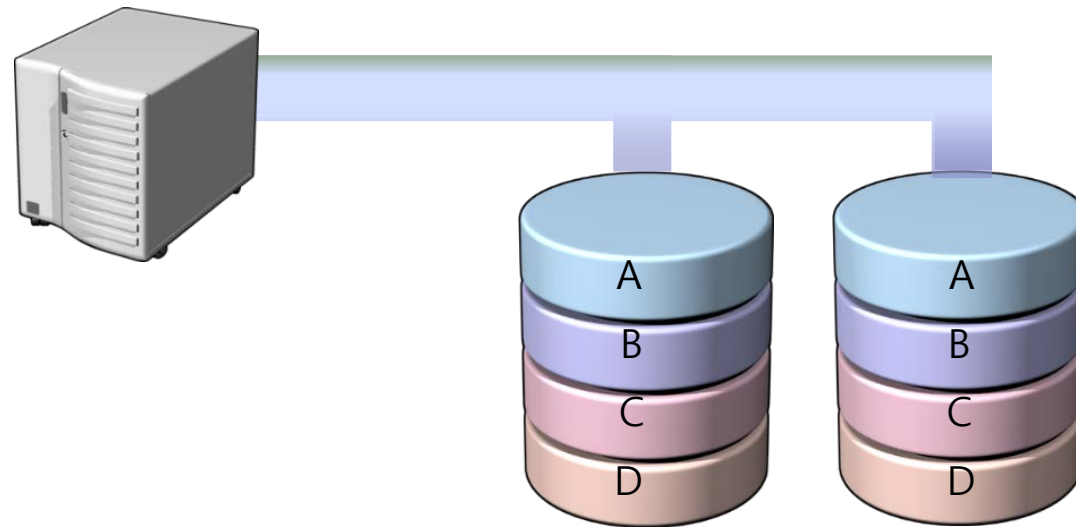
- vim /etc/fstab

18	/dev/md0	/raid0	ext4	defaults	1 2
----	----------	--------	------	----------	-----

Linux RAID (RAID-1)

- **Mirroring(RAID-1)**

- 디스크 2개 필요
- 하나의 파티션을 미러링 함
- 동일한 복사본을 두 개 만들어서 중복성을 제공 함(내결함성)
- 확장 및 축소 불가능



Linux RAID (RAID-1)

- **RAID-1 구성**

- RAID-0 구성 절차와 동일
 - mdadm 명령으로 RAID 구성 시 --level=1로 설정

- **RAID-0 마운트 해제 및 구성 중지**

- umount /raid0
- mdadm --stop /dev/md0
 - 다시 시작할 경우 : mdadm --run /dev/md0

```
[root@localhost ~]# umount /raid0
[root@localhost ~]# mdadm --stop /dev/md0
mdadm: stopped /dev/md0
```

Linux RAID (RAID-1)

- **RAID-1 구성**

- 1) RAID-0 마운트 해제 및 RAID 구성 중지
- 2) RAID-1 레벨의 RAID 구성
- 3) RAID 장치 포맷
- 4) RAID 장치 마운트

하드디스크 구성	
/dev/sdb1	1GB
/dev/sdc1	1GB



Linux RAID (RAID-1)

- **1) RAID-0 마운트 해제 및 레이드 구성 중지**

- umount /raid0
- mdadm --stop /dev/md0

```
[root@localhost ~]# umount /raid0
[root@localhost ~]# mdadm --stop /dev/md0
mdadm: stopped /dev/md0
```

Linux RAID (RAID-1)

- 2) RAID-1 레벨의 RAID 구성

- mdadm --create /dev/md1 --level=1 --raid-devices=2 /dev/sdb1 /dev/sdc1

```
[root@localhost ~]# mdadm --create /dev/md1 --level=1 --raid-device=2 /dev/sdb1 /dev/sdc1
mdadm: /dev/sdb1 appears to be part of a raid array:
        level=raid1 devices=2 ctime=Mon Nov 19 14:55:50 2018
mdadm: Note: this array has metadata at the start and
        may not be suitable as a boot device.  If you plan to
        store '/boot' on this device please ensure that
        your boot-loader understands md/v1.x metadata, or use
        --metadata=0.90
mdadm: /dev/sdc1 appears to be part of a raid array:
        level=raid1 devices=2 ctime=Mon Nov 19 14:55:50 2018
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md1 started.
```

Linux RAID (RAID-1)

- 2) RAID-1 레벨의 RAID 구성

- mdadm --detail /dev/md1

```
[root@localhost ~]# mdadm --detail /dev/md1
/dev/md1:
    Version : 1.2
    Creation Time : Mon Nov 19 14:56:56 2018
    Raid Level : raid1
    Array Size : 1059200 (1034.55 MiB 1084.62 MB)
    Used Dev Size : 1059200 (1034.55 MiB 1084.62 MB)
    Raid Devices : 2
    Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Mon Nov 19 14:57:01 2018
    State : clean
    Active Devices : 2
    Working Devices : 2
    Failed Devices : 0
    Spare Devices : 0


    Name : localhost.localdomain:1 (local to host localhost.localdomain)
    UUID : 34a87af0:b9d70a3b:095b7d3d:d0a55313
    Events : 17


   Number    Major   Minor   RaidDevice State
    -----
      0         8       17         0   active sync    /dev/sdb1
      1         8       33         1   active sync    /dev/sdc1
```

Linux RAID (RAID-1)

- 3) RAID 장치 포맷

- mkfs -t ext4 /dev/md1

```
[root@localhost ~]# mkfs -t ext4 /dev/md1
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
66240 inodes, 264800 blocks
13240 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=272629760
9 block groups
32768 blocks per group, 32768 fragments per group
7360 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376

Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 21 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```


Linux RAID (RAID-1)

- 4) RAID 장치 마운트

- mkdir /raid1
- mount /dev/md1 /raid1

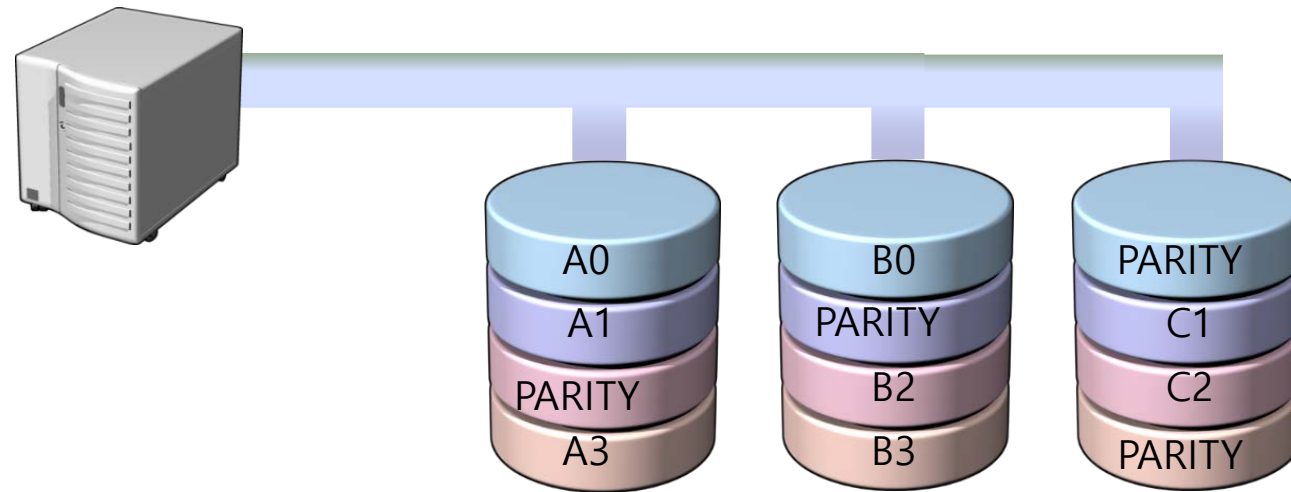
```
[root@localhost ~]# mkdir /raid1
[root@localhost ~]# mount /dev/md1 /raid1
[root@localhost ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	32G	3.6G	27G	12%	/
tmpfs	491M	224K	491M	1%	/dev/shm
/dev/sda2	2.0G	3.2M	1.9G	1%	/home
/dev/sda3	2.0G	69M	1.8G	4%	/tmp
/dev/sda5	2.0G	513M	1.4G	28%	/var
/dev/md1	987M	1.3M	934M	1%	/raid1

Linux RAID (RAID-5)

- **Fault Tolerance using Distributed Parity Bit(RAID-5)**

- 디스크 최소 3개 이상 필요
- 추가 디스크 하나의 비용만으로 내결함성 제공
- 오류 검사를 위해 parity bit 사용
- 확장 및 축소 불가능



Linux RAID (RAID-5)

- **RAID-5 구성**

- 1) RAID-1 마운트 해제 및 RAID 구성 중지
- 2) RAID-5 레벨의 RAID 구성
- 3) RAID 장치 포맷
- 4) RAID 장치 마운트

하드디스크 구성	
/dev/sdb1	2GB
/dev/sdc1	2GB
/dev/sdd1	2GB



Linux RAID (RAID-5)

- 1) RAID-1 마운트 해제 및 레이드 구성 중지

- umount /raid1
- mdadm --stop /dev/md1

```
[root@localhost ~]# umount /raid1  
[root@localhost ~]# mdadm --stop /dev/md1  
mdadm: stopped /dev/md1
```

Linux RAID (RAID-5)

- 2) RAID-5 레벨의 RAID 구성

- mdadm --create /dev/md5 --level=5 --raid-devices=3 /dev/sdb1 /dev/sdc1 /dev/sdc1

```
[root@localhost ~]# mdadm --create /dev/md5 --level=5 --raid-device=3 /dev/sdb1 /dev/sdc1 /dev/sdd1
mdadm: /dev/sdb1 appears to be part of a raid array:
        level=raid5 devices=3 ctime=Mon Nov 19 14:21:33 2018
mdadm: /dev/sdc1 appears to be part of a raid array:
        level=raid5 devices=3 ctime=Mon Nov 19 14:21:33 2018
mdadm: /dev/sdd1 appears to be part of a raid array:
        level=raid5 devices=3 ctime=Mon Nov 19 14:21:33 2018
Continue creating array? y
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md5 started.
```

Linux RAID (RAID-5)

- 2) RAID-5 레벨의 RAID 구성

- mdadm --detail /dev/md5

```
[root@localhost ~]# mdadm --detail /dev/md5
/dev/md5:
    Version : 1.2
  Creation Time : Mon Nov 19 14:35:32 2018
    Raid Level : raid5
    Array Size : 4204544 (4.01 GiB 4.31 GB)
  Used Dev Size : 2102272 (2.00 GiB 2.15 GB)
    Raid Devices : 3
  Total Devices : 3
 Persistence : Superblock is persistent

    Update Time : Mon Nov 19 14:35:38 2018
      State : clean, degraded, recovering
  Active Devices : 2
Working Devices : 3
Failed Devices : 0
Spare Devices : 1

    Layout : left-symmetric
  Chunk Size : 512K

Rebuild Status : 57% complete

    Name : localhost.localdomain:5 (local to host localhost.localdomain)
   UUID : ae31bb5f:fd56588f:848a8bfd:7859890a
  Events : 10

   Number   Major   Minor   RaidDevice State
     0         8       17         0 active sync /dev/sdb1
     1         8       33         1 active sync /dev/sdc1
     3         8       49         2 spare rebuilding /dev/sdd1
```

Linux RAID (RAID-5)

- 3) RAID 장치 포맷

- mkfs -t ext4 /dev/md5

```
[root@localhost ~]# mkfs -t ext4 /dev/md5
mke2fs 1.41.12 (17-May-2010)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=128 blocks, Stripe width=256 blocks
262944 inodes, 1051136 blocks
52556 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=1077936128
33 block groups
32768 blocks per group, 32768 fragments per group
7968 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 26 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

Linux RAID (RAID-5)

- 4) RAID 장치 마운트

- mkdir /raid5
- mount /dev/md5 /raid5

```
[root@localhost ~]# mkdir /raid5
[root@localhost ~]# mount /dev/md5 /raid5
[root@localhost ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	32G	3.6G	27G	12%	/
tmpfs	491M	224K	491M	1%	/dev/shm
/dev/sda2	2.0G	3.2M	1.9G	1%	/home
/dev/sda3	2.0G	69M	1.8G	4%	/tmp
/dev/sda5	2.0G	513M	1.4G	28%	/var
/dev/md5	3.9G	8.1M	3.7G	1%	/raid5

Linux RAID (RAID-5)

- RAID-5 장애 구성

- mdadm /dev/md5 -a /dev/sde1
 - spare 파티션 추가

```
[root@localhost ~]# mdadm /dev/md5 -a /dev/sde1
mdadm: added /dev/sde1
```

- mdadm --detail /dev/md5
 - RAID-5 장치 정보 확인

Number	Major	Minor	RaidDevice	State	
0	8	17	0	active sync	/dev/sdb1
1	8	33	1	active sync	/dev/sdc1
3	8	49	2	active sync	/dev/sdd1
<u>4</u>	<u>8</u>	<u>65</u>	<u>-</u>	<u>spare</u>	<u>/dev/sde1</u>

Linux RAID (RAID-5)

- RAID-5 장애 구성

- mdadm /dev/md5 -f /dev/sdc1
 - 장애 디스크의 디바이스에 결함이 있음을 설정

```
[root@localhost ~]# mdadm /dev/md5 -f /dev/sdc1
mdadm: set /dev/sdc1 faulty in /dev/md5
```

- mdadm --detail /dev/md5
 - RAID-5 장치 정보 확인

Number	Major	Minor	RaidDevice	State	
0	8	17	0	active sync	/dev/sdb1
4	8	65	1	spare rebuilding	/dev/sde1
3	8	49	2	active sync	/dev/sdd1
<u>1</u>	<u>8</u>	<u>33</u>	<u>-</u>	<u>faulty</u>	<u>/dev/sdc1</u>

Linux RAID (RAID-5)

- RAID-5 장애 구성

- mdadm /dev/md5 -r /dev/sdc1

- 장애 디스크 삭제

```
[root@localhost ~]# mdadm /dev/md5 -r /dev/sdc1
mdadm: hot removed /dev/sdc1 from /dev/md5
```

- mdadm --detail /dev/md5

- RAID-5 장치 정보 확인

Number	Major	Minor	RaidDevice	State	
0	8	17	0	active sync	/dev/sdb1
4	8	65	1	active sync	/dev/sde1
3	8	49	2	active sync	/dev/sdd1