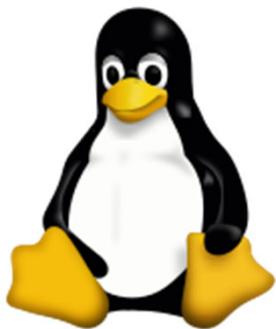


GNU/Linux

리눅스 시스템 관리 (Linux System Management)



Seo, Doo-Ok

Clickseo.com

clickseo@gmail.com



목 차



- 사용자 관리
- 파일 시스템
- 리눅스 부팅 과정



사용자 관리



- 사용자 관리
 - 리눅스 사용자
 - 사용자 및 그룹 관리
 - 사용자 정보 관리
- 파일 시스템
- 리눅스 부팅 과정



리눅스 사용자 (1/6)

● 리눅스 사용자

“리눅스에서 모든 파일과 프로그램은 어떤 사용자에게 의해 소유되어야 한다.”

○ 파일이나 프로그램에 대한 접근 가능성

- **UID**(User ID)와 **GID**(Group ID)를 바탕으로 검사
 - 수행중인 프로그램은 그것을 실행한 사용자의 권한과 허가권을 상속받는다.

○ 패스워드(Password)

- AT&T가 개발한 **DES**(Data Encryption Standard) 알고리즘
 - 초기 : 패스워드를 암호화하고 그 내용은 아무나 볼 수 있게 했다.
 - 해결방안 : **쉐도우 패스워드** 기법, **MD5** 기법 채용(FreeBSD)

○ 홈 디렉터리(Home Directory)

- 시스템에 로그인하는 모든 사용자가 자신의 환경 설정 파일을 저장하는 장소
 - 대부분 **/home** 디렉터리에 각 계정의 홈 디렉터리 생성
 - 단, root 사용자는 별도의 디렉터리 생성 : **/root** -- 유닉스에서는 /
- 시작 스크립트
 - **.bashrc** : bash 가 사용하는 시작 스크립트 파일의 이름
 - 환경 설정 파일은 쉘 스크립트 형태로 존재한다.

리눅스 사용자 (2/6)

- 사용자 계정 관련 파일 : /etc/passwd

- /etc/passwd 파일

```
clickseo:x:1000:1000:Clickseo:/home/clickseo:/bin/bash
```

- 리눅스 사용자에게 대한 정보 : 7개의 필드로 구성
 1. 계정 이름
 2. 암호화된 패스워드 : /etc/shadow 파일에 별도 보관
 3. UID(User ID)
 4. GID(Group ID)
 5. 설명
 6. 홈 디렉터리
 7. 로그인 셸 : 우분투에서는 /bin/bash 를 기본 셸로 사용

```
clickseo@clickseo-VirtualBox: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
clickseo@clickseo-VirtualBox:~$ grep clickseo /etc/passwd
clickseo:x:1000:1000:Clickseo,,,:/home/clickseo:/bin/bash
clickseo@clickseo-VirtualBox:~$
```

리눅스 사용자 (3/6)

- 사용자 계정 관련 파일 : /etc/shadow

- /etc/shadow 파일

- 암호화된 비밀번호 필드 외에도 비밀번호 유효기한 및 계정 사용 여부에 관한 정보

```
clickseo:$6$2w/3k4gsQrRrbV9:17989:0:99999:7:::
```

1. 계정 이름
2. 암호화된 비밀번호
3. 최종 변경일 : 1970년 1월 1일부터 마지막으로 비밀번호가 변경된 날짜 지정
4. 최소 기간(MIN) : 비밀번호를 변경한 후에 다시 변경이 가능한 날짜 남은 날짜 수
5. 최대 기간(MAX) : 비밀번호를 변경해야만 하는 날짜 남은 날짜 수
6. 경고 시간(Warning) : 비밀번호가 만료되기 전에 사용자에게 미리 경고하는 날짜 수
7. 비활성화(Inactive) : 암호가 만료된 후에도 지정한 날짜 동안은 로그인 가능
8. 만료일(Expire) : 1970년 1월 1일 이후로 사용자 계정이 만료되는 날짜 수
9. 예약 항목(Flag) : 향후 사용할 목적으로 비워 둔 항목

```
clickseo@clickseo-VirtualBox: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
clickseo@clickseo-VirtualBox:~$ sudo grep clickseo /etc/shadow
clickseo:$6$J4/6PSMd$Pblb0CtSMKvNZ.FIF3wlf00tr/8Xe1z.Rq0wK9vCp0BnKenIu8JAMxxya
faDr2kc0/ZraFYFJ0wLmpgj/Dxb0:17989:0:99999:7:::
clickseo@clickseo-VirtualBox:~$
```

리눅스 사용자 (4/6)

- 사용자 계정 관련 파일 : /etc/login.defs

- **/etc/login.defs** 파일

- 사용자 계정 설정과 관련된 기본 값을 정의

```
(생략)
UMASK                022
# Password aging controls :
PASS_MAX_DAYS        99999
PASS_MIN_DAYS        0
PASS_WARN_DAYS       7

# useradd : 사용자 계정의 UID 범위
UID_MIN              1000
UID_MAX              60000
#SYS_UID_MIN         100
#SYS_UID_MAX         999

# groupadd : 사용자 계정의 GID 범위
GID_MIN              1000
GID_MAX              60000
#SYS_GID_MIN         100
#SYS_GID_MAX         999
```

리눅스 사용자 (5/6)

- 사용자 계정 관련 파일 : `/etc/group`

- `/etc/group` 파일

- 각 사용자는 적어도 하나의 그룹에 속하고, 필요에 따라 여러 그룹에 동시에 소속

```
clickseo:x:1000:
```

- 그룹 이름
- 암호화된 그룹 비밀번호 : `/etc/gshadow` 파일에 그룹 암호를 저장
- Group ID
- 그룹 멤버 : 쉘표로 사용자 구분

```
clickseo@clickseo-VirtualBox: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
clickseo@clickseo-VirtualBox:~$ grep clickseo /etc/group
adm:x:4:syslog,clickseo
cdrom:x:24:clickseo
sudo:x:27:clickseo
dip:x:30:clickseo
plugdev:x:46:clickseo
lpadmin:x:116:clickseo
clickseo:x:1000:
sambashare:x:126:clickseo
clickseo@clickseo-VirtualBox:~$
```

리눅스 사용자 (6/6)

● 사용자 계정 전환 : su

- 임의의 사용자가 다른 사용자 계정으로 전환하도록 하는 명령
- **su** 와 **su -** 의 차이점
 - **su** : 현재 사용자의 환경을 그대로 물려 받고 root 계정으로 전환
 - PATH의 설정도 원래의 사용자와 동일
 - 관리자 명령어가 있는 `/sbin` 이나 `/usr/sbin` 등은 PATH에 포함되지 않는다.
 - **su -** : root 사용자로 로그인 했을 때와 동일한 환경으로 초기화

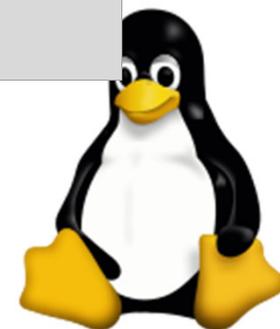
```
clickseo@clickseo-VirtualBox: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
clickseo@clickseo-VirtualBox:~$ su
암호:
root@clickseo-VirtualBox:/home/clickseo# exit
exit
clickseo@clickseo-VirtualBox:~$ su -
암호:
root@clickseo-VirtualBox:~# exit
로그아웃
clickseo@clickseo-VirtualBox:~$
```

- **sudo** : 특정 작업만 수행할 수 있는 권한만 부여



사용자 관리

사용자 및 그룹 관리



사용자 및 그룹 관리 (1/9)

● 사용자 추가 : useradd

일반 형식	useradd [option] 계정명	
주요 옵션	-u UID	직접 UID 값 지정
	-g GID	사용자의 기본 그룹 지정(기본 그룹)
	-G group[, ...]	사용자가 기본 그룹 외에 추가적으로 속할 그룹 지정(다중 그룹)
	-c COMMENT	사용자 이름 또는 부가적인 설명을 지정 (값에 공백이 있을 때는 작은 따옴표를 사용해야 한다.)
	-d HOME_DIR	사용자의 홈 디렉토리를 직접 지정
	-s SHELL	사용자의 로그인 셸을 직접 지정
	-m DIR	사용자 홈 디렉토리를 생성

- **passwd** : 생성된 계정의 암호를 설정하여야 한다.

사용자 및 그룹 관리 (2/9)

- 사용자 추가 : 사용자 계정 관련 정보 및 파일

- **useradd -D** : 기본 설정 값 확인

- 기본 설정 항목들은 **/etc/default/useradd** 파일에 저장

```
clickseo@clickseo-VirtualBox: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
clickseo@clickseo-VirtualBox:~$ useradd -D
GROUP=100
HOME=/home
INACTIVE=-1
EXPIRE=
SHELL=/bin/sh
SKEL=/etc/skel
CREATE_MAIL_SPOOL=no
clickseo@clickseo-VirtualBox:~$
```

- **GROUP** : 기본 등록 그룹의 GID로 100은 users 그룹이다.
- **HOME** : 홈 디렉터리의 생성 위치
- **INACTIVE** : -1 이면 INACTIVE 기능이 비활성화된다(0 이면 암호가 만료되자마자 바로 계정이 잠긴다).
- **EXPIRE** : 계정 종료일을 지정한다.
- **SHELL** : 기본 로그인 셸을 지정한다.
- **SKEL** : 홈 디렉터리에 복사할 기본 환경 파일의 위치 -- **/etc/skel/** 디렉터리
- **CREATE_MAIL_SPOOL** : 메일 디렉터리의 생성 여부를 지정한다.

사용자 및 그룹 관리 (3/9)

- 사용자 추가 : 사용자 계정 관련 디렉터리

- **/etc/skel/** 디렉터리

- 사용자 계정의 홈 디렉터리에 공통으로 배포할 파일들이 존재한다.
 - 사용자 계정 생성 시에 자동으로 복사한다.

```
clickseo@clickseo-VirtualBox: /etc/skel
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
clickseo@clickseo-VirtualBox:/etc/skel$ pwd
/etc/skel
clickseo@clickseo-VirtualBox:/etc/skel$ ls -la
합계 40
drwxr-xr-x  2 root root  4096 2월 10 09:15 .
drwxr-xr-x 123 root root 12288 5월 22 01:03 ..
-rw-r--r--  1 root root   220 4월 5 2018 .bash_logout
-rw-r--r--  1 root root  3771 4월 5 2018 .bashrc
-rw-r--r--  1 root root   807 4월 5 2018 .profile
-rw-r--r--  1 root root  8980 4월 16 2018 examples.desktop
clickseo@clickseo-VirtualBox:/etc/skel$
```

사용자 및 그룹 관리 (4/9)

- 사용자 추가 : `adduser`

일반 형식	<code>adduser</code> [option] 계정명	
주요 옵션	<code>--uid</code> UID	직접 UID 값을 지정
	<code>--gid</code> GID	사용자의 기본 그룹 지정(기본 그룹)
	<code>--home</code> DIR	사용자의 홈 디렉토리를 직접 지정
	<code>--shell</code> SHELL	사용자의 로그인 셸을 직접 지정
	<code>--gecos</code> GECOS	사용자 이름 또는 부가적인 설명을 지정 (값에 공백이 있을 때는 작은 따옴표를 사용해야 한다.)

사용자 및 그룹 관리 (5/9)

● 사용자 추가

○ **useradd** test

```
clickseo@clickseo-VirtualBox: /home
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
clickseo@clickseo-VirtualBox:~$ sudo useradd test
clickseo@clickseo-VirtualBox:~$ grep test /etc/passwd
test:x:1001:1001::/home/test:/bin/sh
clickseo@clickseo-VirtualBox:~$ cd /home
clickseo@clickseo-VirtualBox:/home$ ls
clickseo
clickseo@clickseo-VirtualBox:/home$
```

useradd

○ **adduser** test

```
clickseo@clickseo-VirtualBox: /home
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
clickseo@clickseo-VirtualBox:~$ sudo adduser test
'test' 사용자를 추가 중...
새 그룹 'test' (1001) 추가 ...
새 사용자 'test' (1001) 을(를) 그룹 'test' (으)로 추가 ...
'/home/test' 홈 디렉터리를 생성하는 중...
'/etc/skel'에서 파일들을 복사하는 중...
새 UNIX 암호 입력:
새 UNIX 암호 재입력:
passwd: 암호를 성공적으로 업데이트했습니다
test의 사용자의 정보를 바꿉니다
새로운 값을 넣거나, 기본값을 원하시면 엔터를 치세요
이름 []:
방 번호 []:
직장 전화번호 []:
집 전화번호 []:
기타 []:
정보가 올바릅니까? [Y/n] y
clickseo@clickseo-VirtualBox:~$ grep test /etc/passwd
test:x:1001:1001:,,,:/home/test:/bin/bash
clickseo@clickseo-VirtualBox:~$ cd /home
clickseo@clickseo-VirtualBox:/home$ ls
clickseo test
clickseo@clickseo-VirtualBox:/home$
```

adduser

사용자 및 그룹 관리 (6/9)

- 사용자 계정 정보 변경 : `usermod`

일반 형식	<code>usermod</code> [option] 계정명	
주요 옵션	<code>-u</code> UID	사용자의 UID 값을 변경
	<code>-g</code> GID	사용자의 그룹을 변경
	<code>-c</code> COMMENT	사용자 설명 필드의 내용을 변경
	<code>-d</code> HOME_DIR	사용자의 홈 디렉터리를 변경
	<code>-s</code> SHELL	사용자의 로그인 셸을 변경
	<code>-l</code> USER_NAME	사용자의 계정명을 변경

- (주의) 사용자의 로그인 이름이나 UID를 변경하기 전에 사용자가 현재 시스템에 로그인하고 있거나 프로세스를 수행시키고 있는지를 확인해야 한다.

사용자 및 그룹 관리 (7/9)

- 사용자 계정 삭제 : `userdel`

일반 형식	<code>userdel [option] 계정명</code>	
주요 옵션	<code>-r</code>	사용자의 홈 디렉터리와 메일 디렉터리를 삭제한다.
	<code>-f</code>	사용자가 로그인 중에도 강제로 삭제한다.

사용자 및 그룹 관리 (8/9)

- **그룹 추가** : **groupadd**

- **/etc/group** 파일에 그룹 추가

일반 형식	groupadd [option] 그룹명	
주요 옵션	-g GID	새로운 그룹의 GID 지정
	-r	추가하려는 그룹이 시스템 그룹일 때 주는 옵션

- **addgroup** [option] 그룹명

- --gid GID : 새로운 그룹의 GID 지정
- 별도의 옵션을 지정하지 않으면 **/etc/adduser.conf** 파일에 지정된 GID를 기준으로 가장 마지막 번호의 다음 번호로 자동 설정된다.

- **그룹 삭제** : **groupdel**

사용자 및 그룹 관리 (9/9)

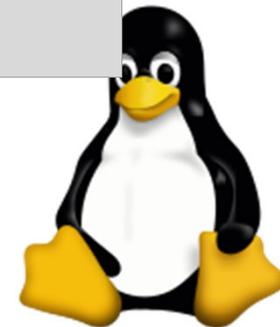
- **그룹 등록 정보 변경** : `groupmod`

일반 형식	<code>groupmod</code> [option] 그룹명	
주요 옵션	<code>-g</code> GID	변경할 그룹의 GID를 변경한다.
	<code>-n</code> newGroupName	그룹에 부여할 새로운 이름을 지정할 때 사용한다.



사용자 관리

사용자 정보 관리



사용자 정보 관리 (1/3)

- 사용자 확인 : id, groups

- id

- 자신의 사용자 번호(UID)와 그룹번호, 그리고 자신이 속한 그룹 명을 보여준다.

```
[clickseo@localhost clickseo]$ id
uid=1000(clickseo) gid=1000(clickseo)
groups=1000(clickseo),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),116(lpadmin),126(sambashare)
```

- groups

- 자신이 속한 그룹 및 다른 계정의 그룹을 볼 수 있다.

```
[clickseo@localhost clickseo]$ groups
clickseo adm cdrom sudo dip plugdev lpadmin sambashare
```

사용자 정보 관리 (2/3)

● 사용자 정보 확인

○ **finger** : 사용자 정보 구하기

일반 형식	finger [option]	
주요 옵션	default	자신의 시스템에 로그인한 사용자 정보를 알고 싶을 때 이용한다.
	user_name	특정 계정의 정보를 알고자 할 때 사용한다.

```
clickseo@clickseo-VirtualBox: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
clickseo@clickseo-VirtualBox:~$ users
clickseo
clickseo@clickseo-VirtualBox:~$ finger
Login      Name      Tty      Idle      Login Time  Office      Office Phone
clickseo   Clickseo  *:0      Apr 18 15:50 (:0)
clickseo@clickseo-VirtualBox:~$ finger clickseo
Login: clickseo      Name: Clickseo
Directory: /home/clickseo      Shell: /bin/bash
On since Thu Apr 18 15:50 (KST) on :0 from :0 (messages off)
No mail.
No Plan.
clickseo@clickseo-VirtualBox:~$
```

○ **users** : 현재 시스템에 로그인한 사용자 계정 조회

사용자 정보 관리 (3/3)

● 사용자 확인

사용자 확인	설 명
who	현재 시스템을 이용하고 있는 사용자 확인
whoami	현재 자신이 어떤 계정으로 로그인하여 시스템을 사용하고 있는지 확인
who am i	사용자 계정, 접속 단말, 접속시간, 호스트 이름 확인

```
[clickseo@localhost clickseo]$ who
clickseo :0      2019-12-31 23:59 (:0)
[clickseo@localhost clickseo]$ whoami
clickseo
[clickseo@localhost clickseo]$ who am i
```

tty (teletypewriter) : 가상 콘솔 또는 터미널

pts (Unix98 pseudo-terminal) : 가상 터미널(X-window에서 사용)

파일 시스템



- 사용자 관리
- **파일 시스템**
 - 리눅스 디렉터리 구조
 - 리눅스 파일 시스템
 - 디스크 관리
- 리눅스 부팅 과정



리눅스 디렉터리 구조 (1/5)

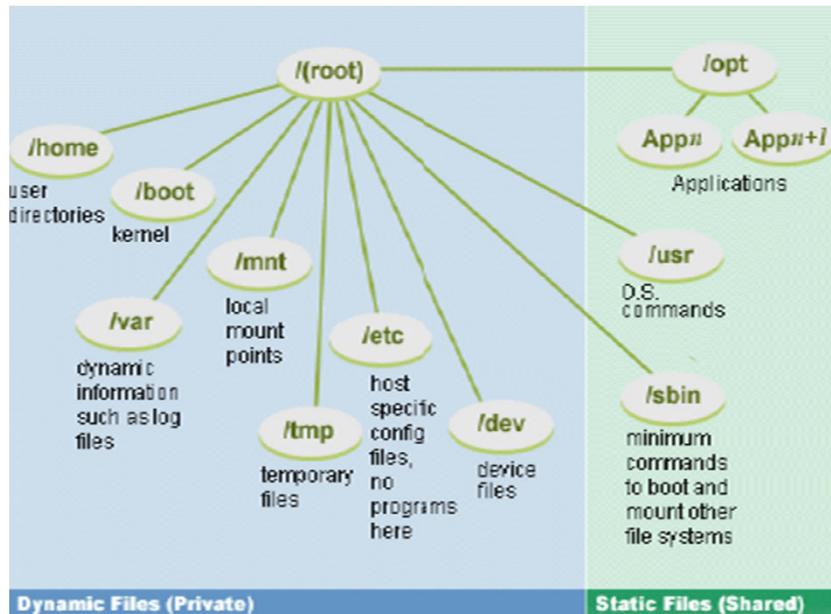
● 리눅스 디렉터리 구조

○ 고정적인 파일(Static Files)

- 공유 가능한(Shared) 카테고리 : /usr, /sbin, /opt 등

○ 유동적인 파일(Dynamic Files)

- 환경 파일, 디바이스 파일, 커널 파일, 로그 파일, 임시 파일, 사용자 홈 디렉터리 등
- /boot, /home, /tmp, /dev, /etc, /var 등



```
clickseo@clickseo-VirtualBox: /
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
clickseo@clickseo-VirtualBox:/$ ls
bin  etc      lib      mnt  run  swapfile  var
boot home    lib64    opt  sbin sys       vmlinuz
cdrom initrd.img lost+found proc snap tmp       vmlinuz.old
dev  initrd.img.old media    root  srv  usr
clickseo@clickseo-VirtualBox:/$
```

리눅스 디렉터리 구조 (2/5)

● 디렉터리 구성과 용도

디렉터리	설 명
/	마운트 되는 리눅스 파일 시스템이 있는 최상위 디렉터리
/root	(ROOT) 시스템 관리자인 root 사용자의 홈 디렉터리
/home	(HOME) 사용자 홈 디렉터리
/bin	(BINaries) 리눅스의 기본 명령어(binary) 들이 존재하는 디렉터리
/sbin	(System BINaries) 관리자가 사용하는 시스템 운영에 필요한 명령어들이 존재
/etc	(ETCetera) 환경 설정 파일 들이 존재하는 디렉터리
/proc	(PROCeSS) 프로세스와 시스템 정보를 제공하기 위한 목적으로 설계된 가상 파일 시스템을 사용하는 디렉터리
/dev	(DEVIce) 장치 파일 들이 존재하는 디렉터리 문자 장치(character device)와 블록 장치(block device)
/tmp	(TeMPorary) 프로세스 생성 과정에서 발생하는 임시 파일을 저장하는 디렉터리 시스템 재 구동을 할 경우 이 디렉터리에 존재하는 파일은 모두 삭제됨으로 파일을 저장하는 용도로 사용할 수 없다.

리눅스 디렉터리 구조 (3/5)

● 디렉터리 구성과 용도

디렉터리	설 명
/usr	(USeR) 프로그램을 설치할 때 패키지에 포함된 대부분의 파일이 위치한 곳
	/usr/bin/ : 압축 파일과 네트워크 실행 파일, 자료 전송 파일 등 실행 파일들이 존재 /usr/sbin/ : 주로 네트워크 관련된 실행 명령어와 실행 데몬들 /usr/include/ : 기본 C 라이브러리
/opt	(OPeRaTion) Add-on 패키지가 설치되는 디렉터리 표준 레드햇 리눅스 배포판에 존재하지 않는 패키지가 설치될 경우 이 디렉터리에 설치되기도 한다(거의 사용하지 않는다).
/misc	(MISCellaneous) 시스템 아키텍처에 독립적인 자료들이 저장되는 디렉터리 (리눅스 계열에서는 거의 사용하지 않는다.)

리눅스 디렉터리 구조 (4/5)

● 디렉터리 구성과 용도

디렉터리	설 명
/boot	(BOOT) 부팅에 필요한 커널과 부팅 할 때 필요한 핵심 파일이 위치하는 디렉터리
/lib	(LIB raries) 시스템 운영 및 프로그램 구동 할 때 필요한 공유 라이브러리(shared library)와 부팅 할 때 사용되는 커널 모듈(kernel module)이 포함되어 있다.
/lost+found	(LOST+FOUND) 시스템을 부팅할 때 파일시스템의 이상 유무를 진단하여 문제가 있으면 구동되는 fsck (File System Check) 명령어가 사용되는 디렉터리로 개별 파일시스템의 최상위 디렉터리마다 존재한다. 만약 손상된 데이터를 복구하지 못했다면, 문제가 발생한 파일시스템에 존재하는 lost+found 디렉터리에 복구하지 못한 데이터 파일이 복사된다.
/mnt	(MouNT) CD-ROM과 플로피 장치에 대한 기본 마운트 포인트를 제공 - CD-ROM : /mnt/cdrom - Floppy : /mnt/floppy

리눅스 디렉터리 구조 (5/5)

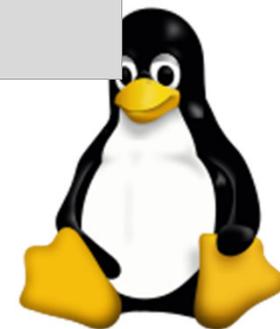
● 디렉터리 구성과 용도

디렉터리	설 명
/var/	<p>(VARiable data) 내용이 자주 변경되는 가변(variable) 자료가 저장되는 디렉터리</p> <ul style="list-style-type: none">- 시스템 운영할 때 발생하는 로그(log)- 메일 송/수신할 때 임시로 저장되는 스푼(spool)- 프린터로 전송하기 위해 임시로 저장하는 프린트 스푼(spool) 데이터- 레드햇 배포판에서 패키지로 제공하는 Apache 웹 서버 관련 자료(/var/www)- MySQL의 데이터 베이스- FTP의 최상위 디렉터리



파일 시스템

리눅스 파일 시스템



리눅스 파일 시스템 (1/10)

● **/dev** 디렉터리

- 문자(Character) 장치
- 블록(Block) 장치

- IDE 하드 디스크 : `/dev/hda`, `/dev/hdb`
- SCSI 하드 디스크 : `/dev/sda`, `/dev/sdb`
- CD-ROM : `/dev/cdrom`
- SCSI CD-ROM : `/dev/scd0`
- 프린터 포트 : `/dev/lp0`, `/dev/lp1`

“리눅스는 유닉스처럼 모든 것을 파일로 간주한다.”

리눅스 파일 시스템 (2/10)

● 파일 시스템(File System)

○ 운영체제가 파일을 시스템의 디스크상에 구성하는 방식

- 파일과 디렉터리의 집합을 구조적으로 관리하는 체계
 - 파일의 실제 데이터와 메타-데이터(파일의 위치, 크기, 소유자, 허가권 등 ls -l로 볼 수 있는 파일 정보)를 유지/저장하는 체계
- 비동기식 파일 시스템
 - 재 부팅 시 무결성 검사(fsck)를 위해 파일시스템의 크기에 따라 상당한 시간을 소모
 - 예 : 리눅스 **ext2** 파일 시스템
- **Journaling File Systems**
 - 무결성, 속도, 쉬운 변환
 - 저널링 파일 시스템 종류
 - » IBM사의 독자적인 파일시스템인 **JFS**(Journaling File System)
 - » SGI사의 **XFS**(eXtended File System)
 - » 독일의 한스 라이저(Hans Reiser)가 개발한 **ReiserFS**(Reiser File System)
 - » **ext3**(Extension3)

리눅스 파일 시스템 (3/10)

- 리눅스 파일 시스템 : ext1, ext2
 - Minix : 과거 Minix에서 사용되었던 파일 시스템
 - 파일 이름은 30문자로 제한, 파일 시스템마다 최대 64MB 성능 제한
 - **Xiafs** : Minix 파일 시스템의 수정 버전
 - 파일 이름과 파일 시스템에 대한 제한을 보완한 버전
 - **ext(ext1) : Extended File System**, 리눅스 초기에 사용되던 파일 시스템
 - 1992년 4월, 리눅스 **0.96c** 에 포함되어 발표
 - 리눅스 초기에 사용되던 파일 시스템
 - 호환성이 없던 ext2의 구 버전
 - 파일 시스템의 최대 크기는 2GB, 파일 이름의 길이는 255바이트까지 지원
 - **ext2**
 - 1993년 01월, **ext** 파일 시스템의 문제 해결
 - 뛰어난 안정성과 속도로 가장 유명한 파일 시스템
 - 이론적으로 파일 시스템의 크기는 32TB까지 가능

리눅스 파일 시스템 (4/10)

- 리눅스 파일 시스템 : ext3, ext4

- ext3

- 2001년 11월, **ext2** 기반으로 개발되어 호환이 가능
 - **Journaling 기능을 도입 복구기능 강화**
 - 파일 시스템의 최대 크기는 블록의 크기에 따라 2~32TB까지 지원

- ext4

- 2008년 12월, ext3 와 호환성 유지
- 1EB(엑사바이트, 1EB=1,024×1,024TB) 이상의 볼륨과 16TB 이상의 파일을 지원

- XFS(eXtended File System)

- 1993년, 실리콘 그래픽스가 개발한 고성능 저널링 파일 시스템
- 2000년 5월, **GNU GPL**로 공개
 - 2001년, 리눅스에 이식되었고 현재 대부분의 리눅스 배포판에서 지원
- XFS는 64bit 파일 시스템으로 최대 16EB까지 지원

리눅스 파일 시스템 (5/10)

- 리눅스 파일 시스템 : /proc/filesystems

- /proc/filesystems

- 현재 시스템이 지원하는 파일 시스템의 종류 확인 가능

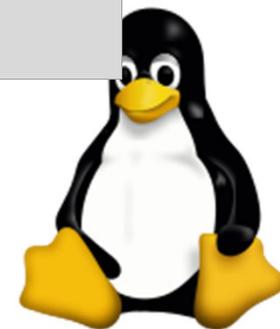
nodev : 해당 파일 시스템이 블록 장치와 연결되어 있지 않다는 것(가상 파일 시스템)

```
clickseo@clickseo-VirtualBox:/proc$ cat /proc/filesystems | more
nodev sysfs
nodev rootfs
nodev ramfs
nodev bdev
nodev proc
nodev cpuset
nodev cgroup
nodev cgroup2
nodev tmpfs
nodev devtmpfs
nodev configfs
nodev debugfs
nodev tracefs
nodev securityfs
nodev sockfs
nodev dax
nodev bpf
nodev pipefs
nodev hugetlbfs
nodev devpts
nodev ext3
nodev ext2
nodev ext4
nodev squashfs
nodev vfat
nodev ecryptfs
nodev fuseblk
nodev ecryptfs
nodev fuseblk
nodev fuse
nodev fusectl
nodev pstore
nodev mqueue
nodev autofs
nodev xfs
nodev jfs
nodev msdos
nodev ntfs
nodev minix
nodev hfs
nodev hfsplus
nodev qnx4
nodev ufs
nodev btrfs
clickseo@clickseo-VirtualBox:/proc$
```



파일 시스템

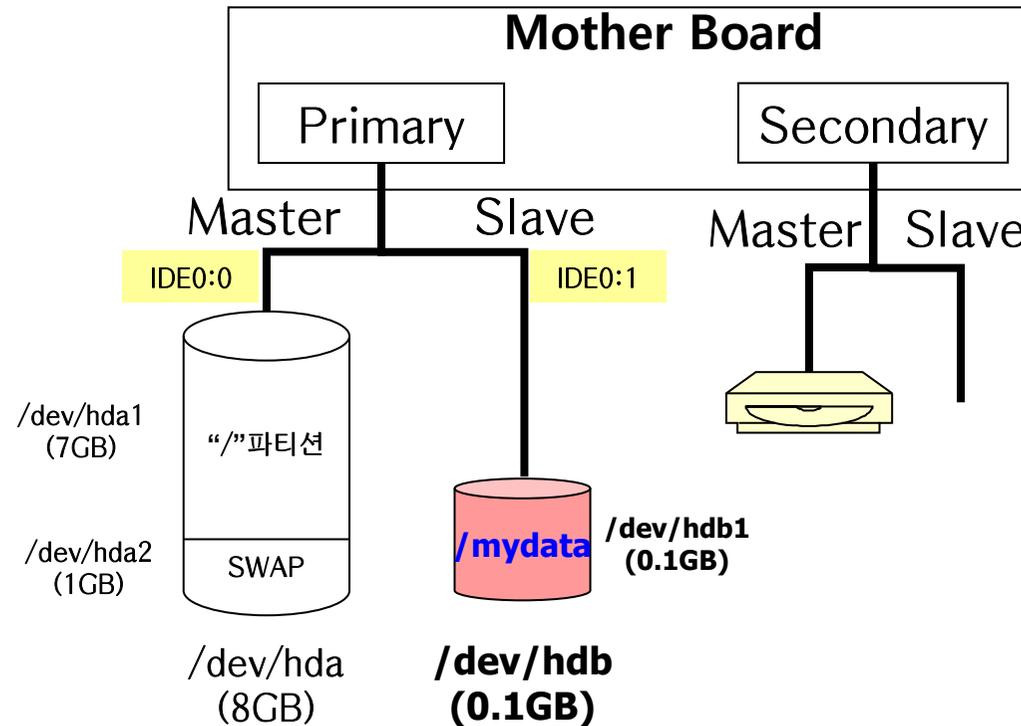
디스크 관리



디스크 관리 (1/6)

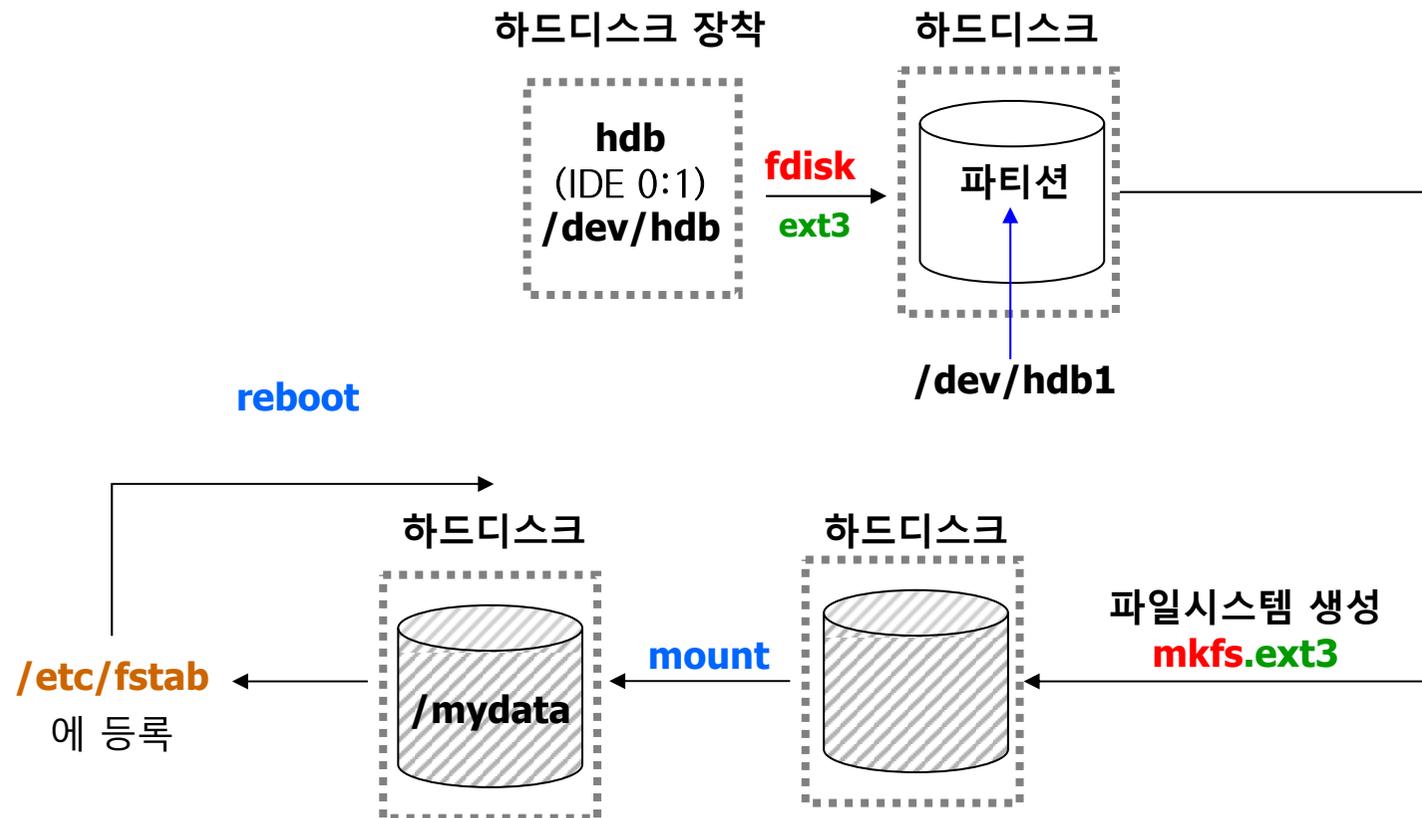
● 한 개의 하드디스크 추가

- 작은 하드디스크(0.1GB)를 하나 장착하고,
- `/mydata` 라는 디렉터리에 마운트 해 보도록 하자.



디스크 관리 (2/6)

- 한 개의 하드디스크 추가



디스크 관리 (3/6)

- 한 개의 하드디스크 추가 : fdisk

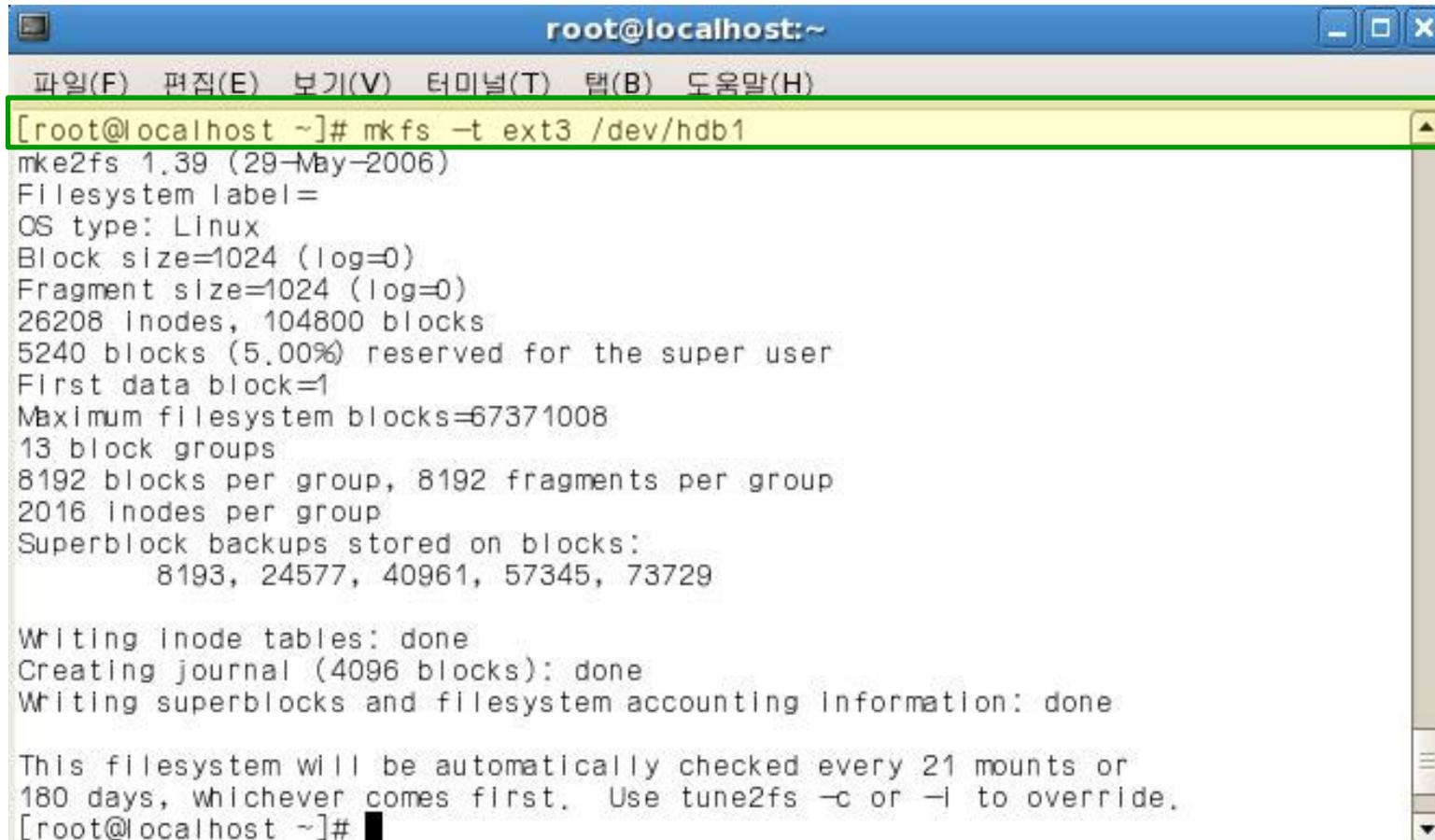
```
root@localhost:~  
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)  
[root@localhost ~]# fdisk /dev/hdb  
Command (m for help): n  
Command action  
  e   extended  
  p   primary partition (1-4)  
p  
Partition number (1-4): 1  
First cylinder (1-208, default 1):  
Using default value 1  
Last cylinder or +size or +sizeM or +sizeK (1-208, default 208):  
Using default value 208  
Command (m for help): p  
Disk /dev/hdb: 107 MB, 107374080 bytes  
16 heads, 63 sectors/track, 208 cylinders  
Units = cylinders of 1008 * 512 = 516096 bytes  


| Device    | Boot | Start | End | Blocks  | Id | System |
|-----------|------|-------|-----|---------|----|--------|
| /dev/hdb1 |      | 1     | 208 | 104800+ | 83 | Linux  |

  
Command (m for help): w  
The partition table has been altered!  
Calling ioctl() to re-read partition table.  
Syncing disks.  
[root@localhost ~]# █
```

디스크 관리 (4/6)

- 한 개의 하드디스크 추가 : mkfs



```
root@localhost:~
파일(F) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
[root@localhost ~]# mkfs -t ext3 /dev/hdb1
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
26208 inodes, 104800 blocks
5240 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67371008
13 block groups
8192 blocks per group, 8192 fragments per group
2016 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Writing inode tables: done
Creating journal (4096 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.
[root@localhost ~]#
```

디스크 관리 (5/6)

- 한 개의 하드디스크 추가
 - 생성된 파티션 마운트

```
root@localhost:/
파일(E) 편집(E) 보기(V) 터미널(T) 탭(B) 도움말(H)
[root@localhost ~]# cd /
[root@localhost /]# ls
bin  dev  home  lost+found  misc  net  proc  sbin  srv  tmp  var
boot  etc  lib  media      mnt  opt  root  selinux  sys  usr
[root@localhost /]# mkdir mydata
[root@localhost /]# ls
bin  dev  home  lost+found  misc  mydata  opt  root  selinux  sys  usr
boot  etc  lib  media      mnt  net     proc  sbin  srv      tmp  var
[root@localhost /]# mount /dev/hdb1 /mydata
[root@localhost /]# mount
/dev/hda1 on / type ext3 (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)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
/dev/hdb1 on /mydata type ext3 (rw)
[root@localhost /]#
```

디스크 관리 (6/6)

● mkfs

○ 파일 시스템 만들기

일반 형식	mkfs [옵션] <파일 시스템 유형> <파티션> mke2fs <파티션>	
주요 옵션	-t	만들자고 하는 파티션 작성
	-c	파일 시스템을 만들면서 Bad Block이 있는지 여부를 설정한다.
	-m	루트를 위한 예비 공간 조절

- 파일 시스템 수리 : **fsck, e2fsck**

```
[root@clickseo /]# fsck -t ext3 /dev/sda9
```

```
[root@clickseo /]# e2fsck /dev/sda9
```

리눅스 부팅 과정

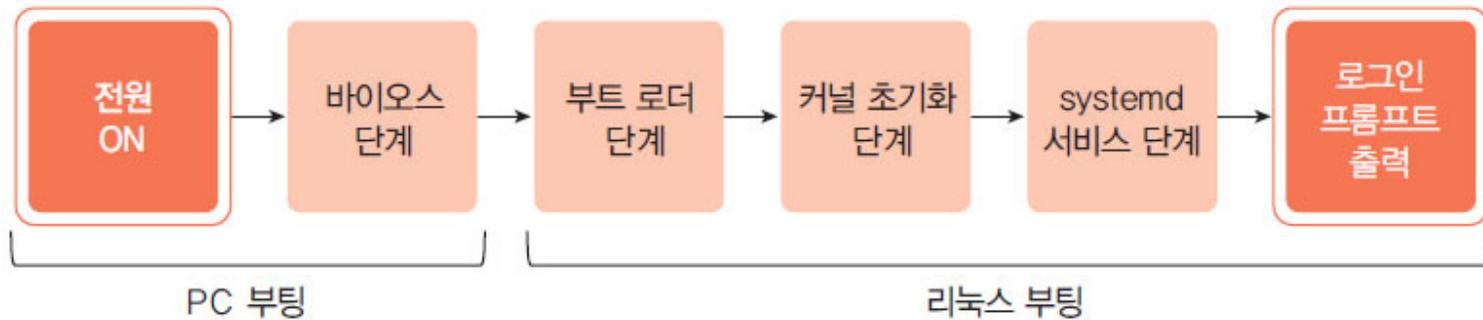


- 사용자 관리
- 파일 시스템
- 리눅스 부팅 과정
 - 부트 로더
 - systemd 서비스



리눅스 부팅 과정 (1/6)

● 리눅스 부팅 과정



○ 로그인 프롬프트

- 그래픽 로그인 시스템 : **GDM**(GNOME Display Manager)



리눅스 부팅 과정 (2/6)

● 부팅 과정 : POST

○ 1단계 : POST 과정

- Power On Self Test
- ROM BIOS(Basic Input Output System)
 - 메모리 용량과 연결된 하드디스크 개수, 키보드 및 마우스 연결 여부 등을 자동으로 점검
 - 어느 장치로 부팅을 할 것인지 확인하여 설정된 장치(일반적으로 하드디스크)의 부팅에 필요한 블록 정보를 읽어온다.



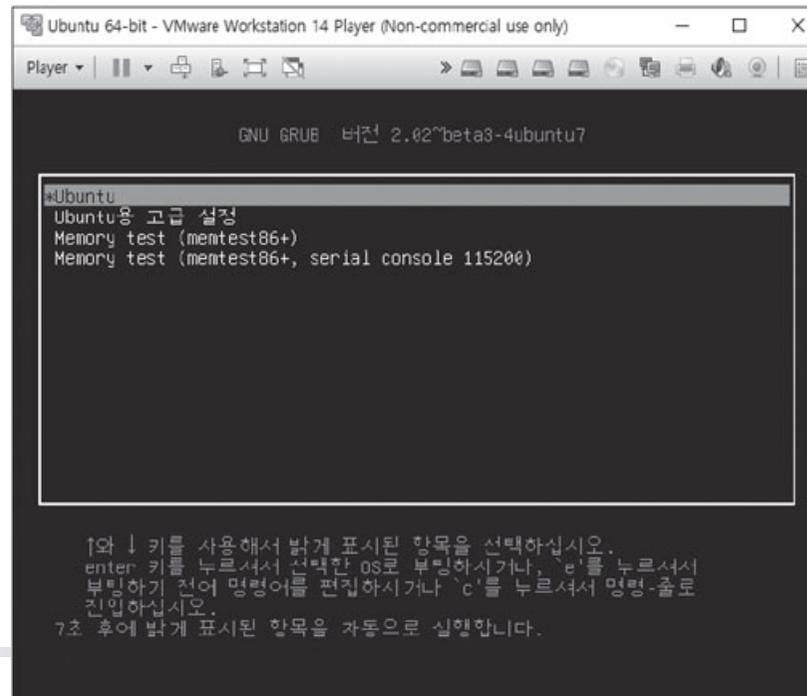
- BOIS 과정 이후 BIOS에서 설정된 부팅할 때 사용할 장치의 **0번 트랙, 0번 실린더, 0번 헤드**에 위치하는 **Boot Block**으로 제어권이 넘어간다.
- **MBR**(Master Boot Record) : 2차 부팅 프로그램(Boot Loader)의 위치 저장

리눅스 부팅 과정 (3/6)

● 부팅 과정 : Boot Loader

○ 2단계 : Boot Loader

- 리눅스에서는 대표적인 Boot Loader는 **GRUB**와 **LILO** 이다.
- 여러 운영체제 중에서 부팅할 운영체제를 선택할 수 있도록 메뉴를 제공
 - **Boot Loader는 리눅스 커널을 메모리에 로딩 한다.**
 - 리눅스 커널은 **/boot** 디렉터리 아래에 **vmlinuz-버전명**의 형태로 제공



리눅스 부팅 과정 (4/6)

● 부팅 과정 : Kernel

○ 3단계 : Kernel

- **Boot Loader**(GRUB 또는 LILO)에 의해 메모리에 적재
- **리눅스 커널은 압축 이미지 파일**
 - Kernel은 Boot Loader의 정보를 참조하여 루트 파일시스템 마운트를 시도
 - » 루트 파일 시스템(/) 마운트 실패 --> **kernel panic** 혹은 **시스템 종료(halt)**를 진행
 - » 루트 파일 시스템(/)은 읽기 전용(read-only)으로 마운트
 - 커널은 가장 먼저 시스템에 연결된 메모리, 디스크, 키보드, 마우스 등 장치들을 검사
 - 장치 검사 등 기본적인 초기화 과정이 끝나면, 커널은 **fork**를 사용하지 않고 생성되는 프로세스와 스레드 생성
- **swapper 프로세스 구동 : 프로세스 번호(PID) 0 번**
 - 메모리 관리, 디스크 관리, 프로세스 관리 등을 수행
 - 프로세스 번호(PID) **1** 번인 **init** 라는 프로세스를 실행
- 부팅 시 출력된 메시지 **dmesg** 명령이나 **more /var/log/boot.log** 명령으로 확인이 가능하다.

리눅스 부팅 과정 (5/6)

- **부팅 과정 : init**

- **4단계 : **init** 프로세스**

- 전통적으로 유닉스에서는 init 프로세스가 처음 생성된 프로세스로 PID가 1번이다.
 - init 프로세스는 부팅 과정에서 커널에 의해 실행 된다.
 - init 프로세스는 스크립트를 순차적으로 실행하여 다른 프로세스를 동작 시킨다.
- **Ubuntu에서는 systemd 서비스가 기존의 init 스크립트를 대체한 것**
 - init 와 관련된 스크립트 파일은 `/etc/init.d` 디렉터리에 있으며 아직 일부 서비스의 스크립트 파일이 남아 있음

- **systemd** : System and Service Manager

리눅스 부팅 과정 (6/6)

● 시스템 종료 : shutdown

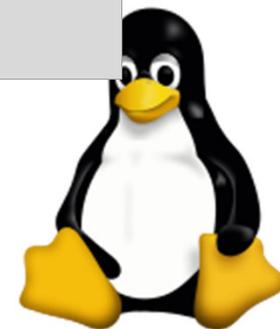
일반 형식	shutdown [옵션] 시간 [경고 메시지]	
주요 옵션	-r	시스템 종료 후 재부팅 한다.
	-h	시스템 종료 후 컴퓨터가 중지 상태에 놓인다.
	-c	실행 중인 shutdown 명령을 취소한다.
	-k	실제로 시스템을 종료하는 것이 아니라, 사용자들에게 메시지만 전달한다.
	명령 매개변수인 시간은 아래와 같이 지정할 수 있다. now 지금 당장 +m 지정한 m분 이후 hh : mm 몇 시(hh), 몇 분(mm)	

- **reboot** : **shutdown -r now**
- **halt** : **shutdown -h now**
 - **reboot, halt, poweroff** 명령은 **/var/log/wtmp** 파일에 시스템 종료 기록을 남기고 시스템을 종료하거나 재시작



리눅스 부팅 과정

부트 로더



부트 로더 (1/2)

- **GNU GRUB**(GNU Grand Unified Bootloader)

- <http://www.gnu.org/software/grub/>

- 1995년, GNU/Hurd 커널의 부팅 작업의 일환으로 개발

- 리눅스의 전통적인 Boot Loader인 **LILO**의 단점을 보완하여 GNU 프로젝트의 일환으로 개발
 - RedHat 7.x 부터 기본 Boot Loader로 제공
 - LILO는 리눅스에서만 사용이 가능하지만 GRUB는 윈도우에서도 사용할 수 있다.
- GRUB의 가장 최신 버전은 **GRUB2**로 우분투에서 기본 Boot Loader로 사용한다.

- 라이선스 : GNU GPL 3.0(GPLv3)

- 프로그래밍 언어 : Assembly, C

GRUB Git : savannah.gnu.org

<https://git.savannah.gnu.org/git/grub.git/>



부트 로더 (2/2)

- **GRUB : 설정 파일**

- **설정 파일 : /etc/default/grub**

- GRUB 메뉴 설정 내용이 저장되어 있다.
- /etc/default/grub 파일 수정 후 **update-grub** 를 실행하면 자동으로 /etc/grub.d 의 파일들이 참조되어 **/boot/grub/grub.cfg** 가 생성된다.

```
clickseo@clickseo-VirtualBox: /etc/default
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
clickseo@clickseo-VirtualBox:/etc/default$ more grub
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
GRUB_TIMEOUT_STYLE=hidden
GRUB_TIMEOUT=0
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""

# Uncomment to enable BadRAM filtering, modify to suit your needs
# This works with Linux (no patch required) and with any kernel that obtains
# the memory map information from GRUB (GNU Mach, kernel of FreeBSD ...)
#GRUB_BADRAM="0x01234567,0xfefefefe,0x89abcdef,0xefefefef"

# Uncomment to disable graphical terminal (grub-pc only)
#GRUB_TERMINAL=console

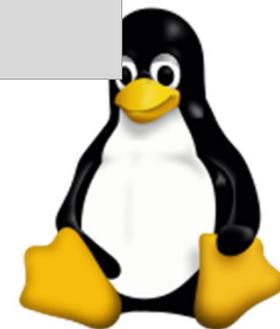
# The resolution used on graphical terminal
# note that you can use only modes which your graphic card supports via VBE
# you can see them in real GRUB with the command `vbeinfo'
#GRUB_GFXMODE=640x480

# Uncomment if you don't want GRUB to pass "root=UUID=xxx" parameter to Linux
#GRUB_DISABLE_LINUX_UUID=true
```



리눅스 부팅 과정

systemd 서비스



systemd 서비스 (1/4)

● Run Level

- **init** 는 시스템의 단계를 7개로 정의하여 구분한다.
 - 각 단계에 따라 셸 스크립트를 실행하는데, 이 단계들을 런 레벨이라고 한다.

RunLevel	의 미	관련 스크립트 위치
0	시스템 종료(halt)	/etc/rc0.d
1, S	단일 사용자 모드(응급 복구 모드)	/etc/rc1.d
	root 패스워드를 잊어버렸거나, 시스템 백업 및 복구 작업 시 런 레벨 1로 진입	/etc/rcS.d
2		/etc/rc2.d
3	다중 사용자 모드	/etc/rc3.d
4		/etc/rc4.d
5	그래픽 다중 사용자 모드	/etc/rc5.d
6	재 시작(reboot)	/etc/rc6.d

- 런 레벨별로 실행 스크립트 파일은 [/etc/init.d](#) 디렉터리에 있는 파일에 대한 심볼릭 링크

systemd 서비스 (2/4)

- **Run Level : runlevel**

- **runlevel** : 이전과 현재 시스템의 런 레벨을 출력하는 명령어
 - **runlevel** 명령어는 이번 레벨과 현재 레벨을 순차적으로 표기

```
clickseo@clickseo-VirtualBox: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
clickseo@clickseo-VirtualBox:~$ runlevel
N 5
clickseo@clickseo-VirtualBox:~$
```

- N 5 : 이전 레벨이 N(**none**)이며, 현재 레벨이 5라는 것을 의미한다.

- 시스템 운영 중 런 레벨 전환 : **init**

Table 1. Mapping between runlevels and systemd targets

Runlevel	Target
0	poweroff.target
1	rescue.target
2, 3, 4	multi-user.target
5	graphical.target
6	reboot.target

systemd 서비스 (3/4)

- **systemd**

- systemd는 init 방식에 비해 다음과 같은 장점을 가지고 있다.
 - 소켓 기반으로 동작하여 **inetd** 와 호환성을 유지한다.
 - 쉘과 독립적으로 부팅이 가능하다.
 - 마운트 제어가 가능하다.
 - **fsck** 제어가 가능하다.
 - 시스템 상태에 대한 스냅샷을 유지한다.
 - **SELinux**와 통합이 가능하다.
 - 서비스에 시그널을 전달할 수 있다.
 - **shutdown** 전에 사용자 세션의 안전한 종료가 가능하다.

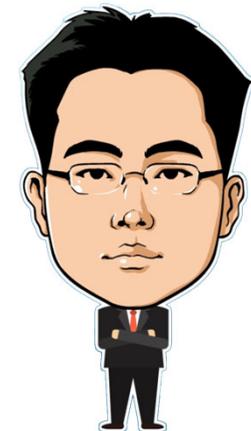
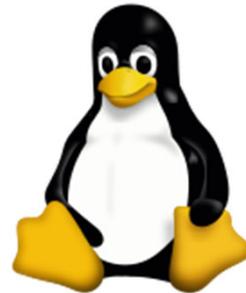
systemd 서비스 (4/4)

- **systemd : unit**

- systemd는 관리 대상의 이름을 '서비스명.유닛 종류'의 형태로 관리
 - 각 유닛은 같은 이름과 종류로 구성된 설정 파일과 동일한 이름을 사용
 - 유닛과 관련한 보다 자세한 내용은 `man systemd.unitName` 으로 확인

참고문헌

- [1] 이종원, "IT CookBook, 우분투 리눅스(개정판) : 시스템 & 네트워크", 한빛아카데미, 2018.
[2] 백창우, "유닉스 리눅스 프로그래밍 필수 유틸리티", 한빛미디어. 2010.



이 강의자료는 저작권법에 따라 보호받는 저작물이므로 무단 전제와 무단 복제를 금지하며, 내용의 전부 또는 일부를 이용하려면 반드시 저작권자의 서면 동의를 받아야 합니다.

Copyright © Clickseo.com. All rights reserved.