

- Ch_1 _ 런레벨, 자동완성과 히스토리
- Ch_2 _ gedit, nano, vi 기본 사용법
- Ch_3 _ vi 비정상 종료, CD Rom 마운트
- Ch_4 _ 리눅스 기본 명령어
- Ch_5 _ 사용자와 그룹 및 관련 명령어
- Ch_6 _ 파일의 허가권과 소유권
- Ch_7 _ 명령어:링크
- Ch_8 _ 프로그램 설치를 위한 dpkg

서버 구축 시 필요한 개념과 명령어

- 종료하는 방법

1. 바탕화면의 [▼ 아이콘] > 컴퓨터끄기/로그아웃 > 컴퓨터 끄기
2. 터미널/콘솔에서 시스템 종료 명령 입력

Poweroff, shutdown -p now, halt-p, init 0

- 시스템 재부팅

1. 바탕화면의 [▼ 아이콘] > 컴퓨터끄기/로그아웃 > 다시 시작
2. 터미널/콘솔에서 시스템 재부팅 명령 입력

Shutdown -r now, reboot, init 6

- 로그아웃

1. 바탕화면의 [▼ 아이콘] > 컴퓨터끄기,로그아웃 > 로그아웃
2. 터미널/콘솔에서 시스템 종료 명령 입력

Logout 또는 exit

- 가상 콘솔

- '가상의 모니터'. 우분투는 총 6 개의 가상 콘솔을 제공
- 각각의 가상 콘솔로 이동하는 단축키는 Ctrl + Alt + F2 ~ F7

(Ctrl + Alt + F2 은 X 윈도우 모드)

```
root@server:~# shutdown -h +5
```

➔ 3 번째 가상화면에서 shutdown -h 실행 (+5 는 5 분뒤에 진행함)

- 3 번째 가상화면은 root 로 접속한 상태

```
/etc/gdm3/custom.conf root sudo  
ubuntu@server:~$  
Broadcast message from root@server on tty3 (Tue 2022-03-08 18:24:12 KST):  
The system is going down for poweroff at Tue 2022-03-08 18:29:12 KST!
```

➔ 4 번째 가상화면에서 종료 메시지 확인

- 4 번째 가상화면은 일반사용자(ubuntu)로 접속한 상태. 3 번째 가상화면에서 진행된 shutdown -h 메시지 확인 가능

- 런 레벨 (Runlevel)

Init 명령어 뒤에 붙은 숫자를 런 레벨(Runlevel) 이라고 부른다.

- 런 레벨 0: Power Off (종료 모드)
- 런 레벨 1: Rescue (시스템 복구 모드) // 단일 사용자 모드
- 런 레벨 2: Multi-User // 사용하지 않음
- 런 레벨 3: Multi-User (텍스트 모드의 다중 사용자 모드) [현 server (b)]
- 런 레벨 4: Multi-User // 사용하지 않음

- 런 레벨 5: Graphical (그래픽 모드의 다중 사용자 모드) // 사용하지 않음 [현 server]
- 런 레벨 6: Reboot

런 레벨 모드를 확인하려면 /lib/systemd/system 디렉터리의 runlevel?.target 파일을 확인한다.

런레벨 3 로 변경

```
root@server:/lib/systemd/system# ls -l runlevel?.target
lrwxrwxrwx 1 root root 15 3월 7 18:47 runlevel0.target -> poweroff.target
lrwxrwxrwx 1 root root 13 3월 7 18:47 runlevel1.target -> rescue.target
lrwxrwxrwx 1 root root 17 3월 7 18:47 runlevel2.target -> multi-user.target
lrwxrwxrwx 1 root root 17 3월 7 18:47 runlevel3.target -> multi-user.target
lrwxrwxrwx 1 root root 17 3월 7 18:47 runlevel4.target -> multi-user.target
lrwxrwxrwx 1 root root 16 3월 7 18:47 runlevel5.target -> graphical.target
lrwxrwxrwx 1 root root 13 3월 7 18:47 runlevel6.target -> reboot.target
root@server:/lib/systemd/system#
```

➔ 런레벨 관련 파일 (바로가기)

```
lrwxrwxrwx 1 root root 16 3월 7 18:47 runlevel5.target -> graphical.target
lrwxrwxrwx 1 root root 13 3월 7 18:47 runlevel6.target -> reboot.target
root@server:/lib/systemd/system# pwd
/lib/systemd/system
root@server:/lib/systemd/system# ls -l default.target
lrwxrwxrwx 1 root root 16 3월 7 18:47 default.target -> graphical.target
root@server:/lib/systemd/system#
```

➔ 현재 server 는 graphical 그래픽 모드로 부팅되어있는 것을 확인

```
root@server:/lib/systemd/system# ln -sf /lib/systemd/system/multi-user.target /l
ib/systemd/system/default.target
root@server:/lib/systemd/system# ls -l default.target
lrwxrwxrwx 1 root root 37 3월 11 10:11 default.target -> /lib/systemd/system/mu
lti-user.target
root@server:/lib/systemd/system#
```

➔ Server 를 그래픽 모드에서 텍스트 모드로 변환

- Default.target 이 graphical.target 에서 multi-user.target 으로 변경
- Ls -l default.target 으로 확인가능
- Ln -sf /lib/systemd/system/multi-user.target /lib/systemd.system/default.target 으로 변경

```

Ubuntu 20.04 LTS server tty1

server login: root
Password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.13.0-30-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 updates can be installed immediately.
0 of these updates are security updates.

Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Tue Mar  8 18:22:54 KST 2022 on tty3
root@server:~# _

```

➔ 재부팅 시 텍스트 모드로 (터미널모드) 로 부팅

➤ graphical 그래픽모드로 부팅 원할 시 Startx 입력

```

root@server:~# ln -sf /lib/systemd/system/graphical.target /lib/systemd/system/default.target
root@server:~#

```

➔ server 를 텍스트모드에서 그래픽모드로 다시 변환 후 리부팅

자동 완성과 히스토리

➤ 자동 완성이란 파일명의 일부만 입력한 후 tab 키를 눌러 나머지 파일명을 자동으로 완성하는 기능

Ex. Cd /etc/NetworkManager/dispatcher.d/ 를 입력하려면

Cd /et[tab]ne[tab]di[tab]

- 자동 완성기능은 빠른 입력효과도 있지만 파일명이나 디렉터리가 틀리지 않고 정확하게 입력되는 효과도 있으므로 자주 사용됨
- 도스 키란 이전에 입력한 명령어를 상/하 화살표 키를 이용하여 다시 나타내는 기능

```

43 ls -l runlevel?.target
44 pwd
45 ls -l default.target
46 ln -sf /lib/systemd/system/multi-user.target /l
target
47 ls -l default.target
48 ln 9sf lib/systemd/system/multi-user.target/li
rget
49 ln -sf /lib/systemd/system/multi-user.target/li
target
50 clear
51 cd /lib/systemd/system
52 ls -l default.target
53 ln -sf /lib/systemd/system/multi-user.target /l
target
54 ls -l default.target
55 reboot
56 startx
57 ln -sf /lib/systemd/system/graphical.target /l
target
58 reboot
59 reboot
60 history
root@server:~/바탕화면#

```

➔ History 로 지금까지 사용한 명령어들 확인가능

➤ 보안상 등 이유로 불필요할시 history -c 로 클리어 가능

```

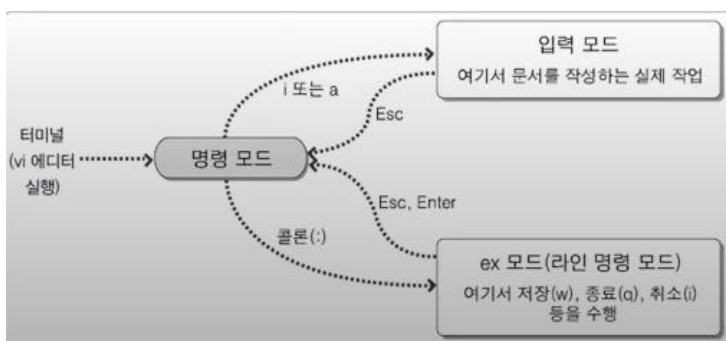
root@server:~# cd /etc/systemd/network

```

➔ 자동완성 기능으로만 완성된 명령어

➤ Cd /e[tab]/sysy[tab]/ne[tab]

에디터 사용 (gedit, nano, vi)



➔ Vi 에디터 사용법 개요도

```
열기(O) *myfile1.txt
1 asdasd
2
3 asdadf
4
5 qgwgqgwgqg
```

➔ Gedit 에디터 화면

- Terminal 에서 gedit 명령어 입력으로 사용 가능
- 메모장과 같은 기능을 가지고 있음

```
root@server:~# gedit
root@server:~# ls -l
합계 40
-rw-r--r-- 1 root root 21 3월 11 10:49 myfile1
drwx----- 3 root root 4096 3월 8 14:41 snap
drwxr-xr-x 2 root root 4096 3월 8 11:08 공개
drwxr-xr-x 2 root root 4096 3월 8 11:08 다운로드
drwxr-xr-x 2 root root 4096 3월 8 11:08 문서
drwxr-xr-x 2 root root 4096 3월 8 11:08 바탕화면
drwxr-xr-x 2 root root 4096 3월 8 11:08 비디오
drwxr-xr-x 2 root root 4096 3월 8 11:08 사진
drwxr-xr-x 2 root root 4096 3월 8 11:08 음악
drwxr-xr-x 2 root root 4096 3월 8 11:08 템플릿
root@server:~# gedit myfile1.txt
root@server:~#
```

➔ Gedit 에디터로 만들어진 txt 파일

```
root@server: ~
GNU nano 4.8 myfile2.txt
나노 에디터 연습
nano~!
```

➔ Nano 나노 에디터 화면

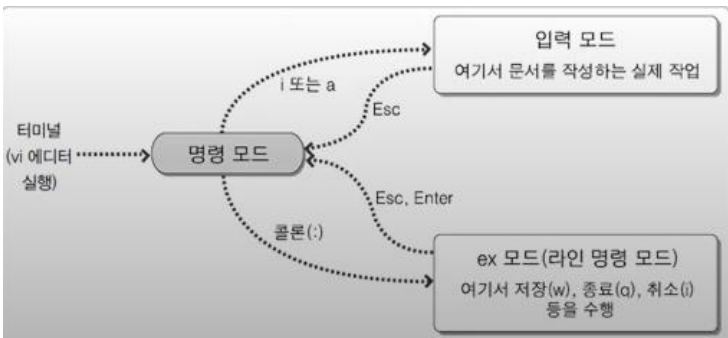
- Gedit 에디터와는 다르게 xwindow 가 없어도 사용 가능
(gedit 에디터는 xwindow 에서만 사용가능)
- Nano 에디터에서 Ctrl + c 로 행 번호 파악 가능 (nano -c 로 에디터를 오픈하면 계속
행번호가 나타남)

```
root@server:~# nano
root@server:~# ls -l
합계 44
-rw-r--r-- 1 root root 21 3월 11 10:49 myfile1
-rw-r--r-- 1 root root 33 3월 11 10:58 myfile2.txt
drwx----- 3 root root 4096 3월 8 14:41 snap
drwxr-xr-x 2 root root 4096 3월 8 11:08 공개
drwxr-xr-x 2 root root 4096 3월 8 11:08 다운로드
drwxr-xr-x 2 root root 4096 3월 8 11:08 문서
drwxr-xr-x 2 root root 4096 3월 8 11:08 바탕화면
drwxr-xr-x 2 root root 4096 3월 8 11:08 비디오
drwxr-xr-x 2 root root 4096 3월 8 11:08 사진
drwxr-xr-x 2 root root 4096 3월 8 11:08 음악
drwxr-xr-x 2 root root 4096 3월 8 11:08 템플릿
root@server:~#
```

➔ Nano 에디터로 만들어진 txt 파일 (myfile2.txt)

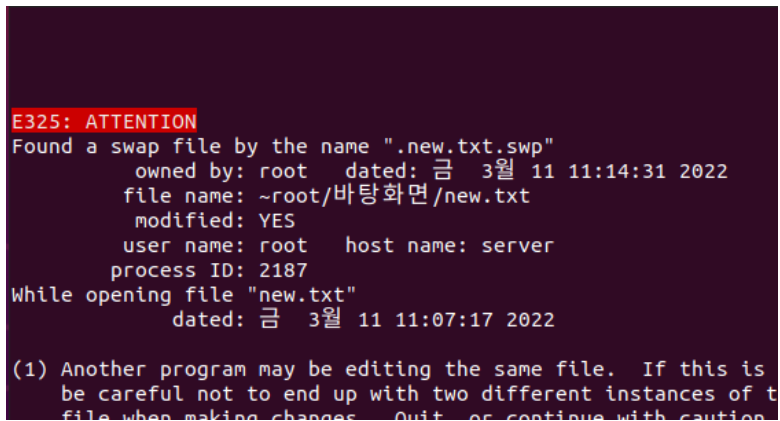
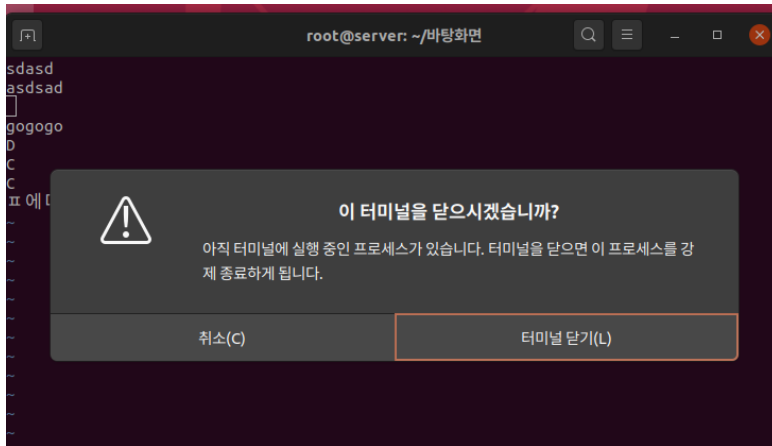
[illegible]

➔ Vi 에디터 사용 화면

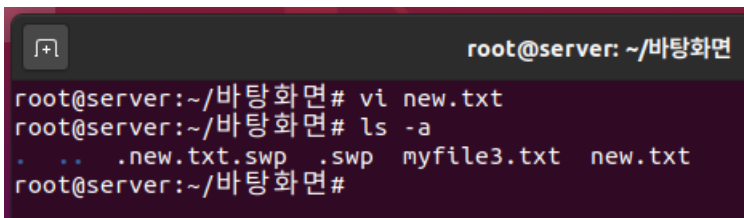


- Vi 에디터도 나노 에디터와 동일하게 xwindow 가 없어도 사용 가능
- vi 에디터 실행 뒤 명령모드에서 | 또는 a 명령어로 입력모드로 전환 후 사용가능
- 문서 작성 후 esc 로 명령모드 이후 shift + ; ex 모드 (라인 명령 모드)
ex 모드에서 저장(w) 종료 (q) 취소(c) 가능
- 작성 중 저장하지 않고 종료 (q!)

Vi 비정상 종료



➔ Vi 에디터를 사용 중 터미널을 닫았을 때 나오는 오류



➔ Vi 에디터로 생성한 new.txt 파일 이외에 .new.txt.swp 라는 임시파일 발견

➤ 정상 종료 시 임시파일은 삭제되는데 비 정상 종료로 인해 임시파일이 삭제 x

```
root@server:~/바탕화면# rm -f .new.txt.swp
```



```
root@server: ~/바탕화면
gogogo
D
C
C
H 에디
~
~
~
~
```

➔ 임시파일 삭제 후 vi 에디터 정상 작동 확인

Vi 기능 요약

- 명령 모드 > 입력 모드

i | 현재 커서의 위치부터 입력 (i)

I | 현재 커서 줄의 맨 앞에서부터 입력 (Shift + I)

a | 현재 커서의 위치 다음 칸부터 입력 (A)

A | 현재 커서 줄의 맨 마지막부터 입력 (Shift + A)

- 명령 모드에서 커서를 이동

H | 커서를 왼쪽으로 한 칸 이동 (←와 같은 의미, H)

J | 커서를 아래로 한 칸 이동 (↓와 같은 의미, J)

K | 커서를 위로 한 칸 이동 (↑와 같은 의미, K)

I | 커서를 오른쪽으로 한 칸 이동 (→와 같은 의미, I)

- 명령 모드에서 삭제, 복사, 붙여넣기

x | 현재 커서가 위치한 글자 삭제 (del 과 같은 의미, X)

X | 현재 커서가 위치한 앞 글자 삭제 (backspace 와 같은 의미, shift+X)

dd | 현재 커서의 행 삭제 (D 연속 두 번 입력)

숫자 DD | 현재 커서부터 숫자만큼의 행 삭제 (숫자 다음 D 연속 두 번 입력)

yy | 현재 커서가 있는 행을 복사 (Y 연속 두 번 입력)

숫자 yy | 현재 커서부터 숫자만큼의 행을 복사 (숫자 다음 yy 연속 두 번 입력)

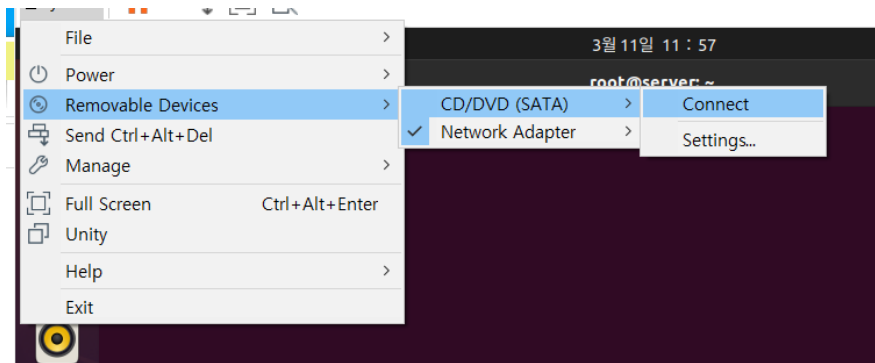
- 물리적인 장치를 특정한 위치 (대개는 디렉터리) 에 연결시켜 주는 과정

```
root@server:~# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,n
proc on /proc type proc (rw,nosuid,nodev,no
udev on /dev type devtmpfs (rw,nosuid,noexe
devpts on /dev/pts type devpts (rw,nosuid,n
tmpfs on /run type tmpfs (rw,nosuid,nodev,n
/dev/sda2 on / type ext4 (rw,relatime,error
securityfs on /sys/kernel/security type sec
tmpfs on /dev/shm type tmpfs (rw,nosuid,nod
```

➔ mount 명령어로 장치에 연결되어 있는 하드웨어장치들 위치 확인

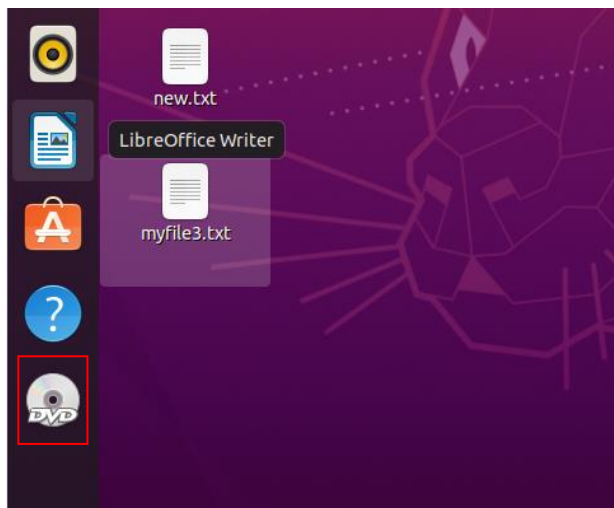
- 현 장치에 연결되어 있는 하드웨어장치들이 어느 장소에 연결이 되어있는지 나타내는 명령어

CD 마운트



➔ 우분투 설치 dvd 삽입

- Player > Removable Devices > CD/DVD > Connect



➔ 자동으로 연결된 상태

- 그래픽 모드에서만 자동으로 연결되고, 오류도 많을 수 있어서 수동 연결 권장

```
root@server:~/바탕화면# umount /dev/cdrom
root@server:~/바탕화면# █
```

➔ 수동 연결을 위해 기존 연결 끊기

- Umount /dev/cdrom 명령어

```
root@server:~# cd
root@server:~# mkdir cdrom
mkdir: `cdrom' 디렉토리를 만들 수 없습니다: 파일이 있습니다
root@server:~# ls
cdrom  myfile2.txt  snap  다운로드  바탕화면  사진  템플릿
myfile1  new.txt      공개  문서      비디오    음악
root@server:~# ls /media/
root  ubuntu
root@server:~# mkdir /media/cdrom
root@server:~# ls /media/
cdrom  root  ubuntu
root@server:~# mount /dev/cdrom /media/cdrom
mount: /media/cdrom: WARNING: device write-protected, mounted read-only.
root@server:~# █
```

➔ 수동으로 dvd 연결 (읽기 전용으로 마운트됨)

- ~~➤ mkdir cdrom 명령어를 사용하여 'cdrom' 디렉토리 생성~~
- media 폴더에서 cdrom 파일 생성 (mkdir /media/cdrom 명령어 사용)
- mount /dev/cdrom /media/cdrom 명령어를 사용하여 마운트

dev 는 device 의 약자로 설정해 놓은 iso 파일을 가리킴

```
root@server:~# cd /media/cdrom/
root@server:/media/cdrom# pwd
/media/cdrom
root@server:/media/cdrom# ls
EFI          boot  dists  isolinux  pics  preseed
README.diskdefines  casper  install  md5sum.txt  pool  ubuntu
root@server:/media/cdrom# █
```

➔ dvd 에 생성된 파일들 확인 가능

```
root@server:/media/cdrom# umount /dev/cdrom
umount: /media/cdrom: target is busy.
root@server:/media/cdrom#
```

➔ 기존 연결 Umount 시 오류 발생

- Media/cdrom 에 연결된 상태에서 umount 를 시도하여 오류가 발생
- 다른 디렉토리로 이동하여 오류 해결 가능

```
root@server:/media/cdrom# cd
root@server:~# umount /dev/cdrom
root@server:~#
```

➔ Cd 명령어로 디렉토리 변경 후 umount 명령어 실행

- Cd(change directory) 사용하여 오류 해결
- 이후 player > Removable devices > ...에서 iso 파일 제거하면 물리적으로도 제거완료

● 리눅스 기본 명령어 (1)

➤ ls

Windows 의 'dir' 와 같은 역할로, 해당 디렉토리 안에 있는 파일의 목록을 나열

ex # ls /etc/systemd | [자세히 보기 ls -l] [숨김 파일도 보기 ls -a]

합쳐서 사용 가능 <ls -al> 확장명으로 찾기 <ls *.txt> <txt 파일만

➤ cd

디렉토리를 이동

➤ pwd

현재 디렉토리의 전체 경로를 출력

➤ rm

파일이나 디렉토리를 삭제

```

root@server:~# ls -l
합계 48
drwxr-xr-x 2 root root 4096 3월 11 13:06 cdrom
-rw-r--r-- 1 root root 21 3월 11 10:49 myfile1
-rw-r--r-- 1 root root 33 3월 11 10:58 myfile2.txt
-rw-r--r-- 1 root root 0 3월 11 11:05 new.txt
drwx----- 3 root root 4096 3월 8 14:41 snap
drwxr-xr-x 2 root root 4096 3월 8 11:08 공개
drwxr-xr-x 2 root root 4096 3월 8 11:08 다운로드
drwxr-xr-x 2 root root 4096 3월 8 11:08 문서
drwxr-xr-x 2 root root 4096 3월 11 11:36 바탕화면
drwxr-xr-x 2 root root 4096 3월 8 11:08 비디오
drwxr-xr-x 2 root root 4096 3월 8 11:08 사진
drwxr-xr-x 2 root root 4096 3월 8 11:08 음악
drwxr-xr-x 2 root root 4096 3월 8 11:08 템플릿
root@server:~# rm new.txt
root@server:~# ls
cdrom myfile1 myfile2.txt snap 공개 다운로드 문서
root@server:~#

```

➔ Root 폴더 new.txt 파일을 삭제

➤ mount /dev/cdrom /media/cdrom 명령어를 사용하여 마운트

dev 는 device 의 약자로 설정해 놓은 iso 파일을 가리킴

```

root@server:~# rm cdrom
rm: 'cdrom'를 지울 수 없음: 디렉터리입니다
root@server:~# rm -r cdrom

```

➔ cdrom 폴더를 rm 명령어를 사용하여 삭제 진행 시 오류 발생

➤ 폴더는 rm 명령어로 삭제 불가능

➤ Rm -r 명령어로 삭제 가능

● 리눅스 기본 명령어 (2)

➤ cp

파일이나 디렉토리를 복사

예) #cp abc.txt abb.txt

```

root@server:~#
root@server:~# ls
cdrom myfile1 myfile2.txt snap 공
root@server:~# cp myfile1 file1.txt
root@server:~# ls
cdrom myfile1 myfile2.txt snap 다운로드
file1.txt myfile2.txt 공개 문서
root@server:~#

```

➔ myfile1 파일을 file1.txt 이름으로 복사

➤ touch

크기가 0 인 새 파일을 생성, 이미 존재하는 파일의 생성 날짜를 최신화

➤ Mv

파일과 디렉토리의 이름을 변경하거나 이동

예) mv abc.txt www.txt

```
cdrom myfile1 snap 다운로드
file1.txt myfile2.txt 공개 문서
root@server:~# mv myfile2.txt file2.txt
root@server:~# ls
cdrom file1.txt file2.txt myfile1 sn
root@server:~#
```

➔ Myfile2.txt 파일을 file2.txt 파일로 이름 변경

➤ mkdir

새로운 디렉토리를 생성

예) #mkdir abc

● 리눅스 기본 명령어 (3)

➤ rmdir

디렉토리를 삭제. (단, 비어 있어야 함)

예) # rmdir abc

➤ Cat

텍스트로 작성된 파일을 화면에 출력

예) cat a.txt b.txt

➤ Head, tail

텍스트로 작성된 파일의 앞 10 행 또는 마지막 10 행 출력

예) #head /etc/systemd/bootchart.conf

➤ More

텍스트로 작성된 파일을 화면에 페이지 단위로 출력

예) # more /etc/systemd/system.conf

● 리눅스 기본 명령어 (4)

➤ Less

More 와 용도가 비슷하지만 더 확장됨

예) # less /etc/systemd/system.conf

➤ File

File 이 어떤 종류의 파일인지를 표시

예) # file /etc/systemd/system.conf

➤ clear

명령창을 깨끗하게 지움

예) clear

● 사용자와 그룹 (1)

➤ 리눅스는 다중 사용자 시스템(multi-user System) 임

➤ 기본적으로 root 라는 이름을 가진 슈퍼유저(superuser)가 있으며, 모든 작업을 할 수 있는 권한이 있음

➤ 모든 사용자는 하나 이상의 그룹에 소속되어 있음

➤ 사용자는 /etc/passwd 파일에 정의되어 있음

```
passwd
/etc
1 root:x:0:0:root:/root:/bin/bash
2 daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
3 bin:x:2:2:bin:/bin:/usr/sbin/nologin
4 sys:x:3:3:sys:/dev:/usr/sbin/nologin
5 sync:x:4:65534:sync:/bin:/bin/sync
6 ntp:x:5:60:games:/usr/games:/usr/sbin/nologin
39 whoopie:x:man:/var/noneexistent:/usr/sbin/nologin
40 colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
41 geoclue:x:122:127:./var/lib/geoclue:/usr/sbin/nologin
42 pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
43 gnome-initial-setup:x:124:65534:./run/gnome-initial-setup:/bin/false
44 gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
45 ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
46 systemd-coredump:x:999:999:systemd Core Dumper:/usr/sbin/nologin
```


➤ 각 행의 의미는 다음과 같음

사용자 이름:암호:사용자 ID:사용자가 소속된 그룹 ID:추가정보:홈 디렉터리:기본 셸

➤ 암호 x 는 따로 관리하고 있음 (암호가 없다는 것이 아님)

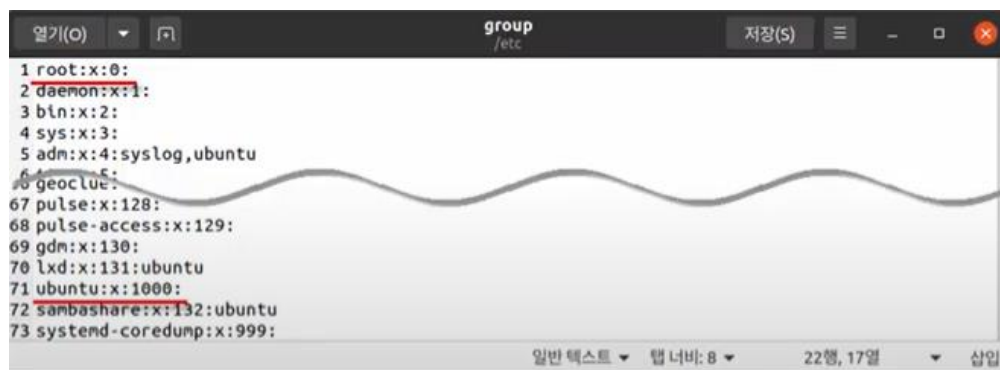
➤ Root 사용자의 홈 디렉터리는 /root

일반 사용자의 홈 디렉터리는 /home/사용자이름

● 사용자와 그룹 (2)

➤ 사용자의 비밀번호는 /etc/shadow 파일에 정의되어 있음

➤ 그룹은 /etc/group 파일에 정의되어 있음



➤ 각 행의 의미는 다음과 같음

그룹명:비밀번호:그룹 id:보조 그룹 사용자 (보조 그룹 사용자는 참고사항)

➤ 보조 그룹 사용자 (root 기준 0) 은 passwd 파일 사용자가 소속된 그룹 ID(root 기준 0) 과 같은 의미



● 사용자와 그룹 관련 명령어 (1)



➤ adduser

새로운 사용자를 추가

예) # adduser newuser1

➤ passwd

사용자의 비밀번호를 지정하거나 변경

예) # passwd newuser1

➤ usermod

사용자의 속성을 변경

예) # usermod -groups ubuntu newuser1

➤ userdel

사용자를 삭제

예) # userdel newuser2

➤ chage

사용자의 암호를 주기적으로 변경하도록 설정

예) # chage -m 2 newuser1

➤ groups

현재 사용자가 속한 그룹을 보여줌

예) # groups

➤ groupadd

새로운 그룹을 생성

예) # groupadd newgroup1

➤ groupmod

그룹의 속성을 변경

예) #groupmod - new-name mygroup1 newgroup2

● 사용자와 그룹 관련 명령어 (3)

➤ groupdel

그룹을 삭제

예) # groupdel newgroup2

➤ gpasswd

그룹의 암호를 설정하거나, 그룹의 관리를 수행

예) # gpasswd mygroup1

● 실습

```
root@server:~/바탕화면# adduser user1
'user1' 사용자를 추가 중...
새 그룹 'user1' (1001) 추가 ...
새 사용자 'user1' (1001) 을(를) 그룹 'user1' (으)로 추가 ...
'/home/user1' 홈 디렉터리를 생성하는 중...
'/etc/skel'에서 파일들을 복사하는 중...
새 암호:
새 암호 재입력:
passwd: 암호를 성공적으로 업데이트했습니다
user1의 사용자의 정보를 바꿉니다
새로운 값을 넣거나, 기본값을 원하시면 엔터를 치세요
이름 []:
방 번호 []:
직장 전화번호 []:
집 전화번호 []:
기타 []:
정보가 올바릅니까? [Y/n]
root@server:~/바탕화면# tail /etc/passwd
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:120:125:/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/bin/false
geoclue:x:122:127:/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534:/run/gnome-initial-setup:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/usr/sbin/nologin
user1:x:1001:1001:,,,:/home/user1:/bin/bash
root@server:~/바탕화면#
```

➔ user1 사용자를 추가

➤ adduser 명령어 입력 (adduser user1)

- tail 명령어 사용하여 passwd 파일 맨 마지막 생성된 user1 사용자 정보 확인

```
root@server:~/바탕화면# tail -5 /etc/group
lxd:x:131:ubuntu
ubuntu:x:1000:
sambashare:x:132:ubuntu
systemd-coredump:x:999:
user1:x:1001:
root@server:~/바탕화면#
```

➔ 그룹을 따로 설정하지 않은 user1 의 그룹 확인

- 그룹을 지정하지 않고 생성하게 되면 자동으로 그룹도 생성 (사용자 이름과 같음)
일반적인 상황에서는 그룹부터 생성하고 유저를 생성

```
root@server:~/바탕화면# userdel -r user1
userdel: user1 mail spool (/var/mail/user1) not found
root@server:~/바탕화면#
```

➔ 사용자를 삭제.

- -r 명령어를 추가하면 사용자가 사용하던 디렉터리까지 삭제

```
root@server:~/바탕화면# groupadd ubuntuGroup
root@server:~/바탕화면#
root@server:~/바탕화면# tail -5 /etc/group
lxd:x:131:ubuntu
ubuntu:x:1000:
sambashare:x:132:ubuntu
systemd-coredump:x:999:
ubuntuGroup:x:1001:
root@server:~/바탕화면# adduser --gid 1001 user1
'user1' 사용자를 추가 중...
```

➔ 그룹 생성 뒤 해당 그룹 지정하여 사용자 생성

- groupadd 명령어를 사용하여 ubuntuGroup 그룹 생성 후 해당 그룹 번호 확인
- adduser 명령어로 사용자 생성. --gid 명령어 사용하여 그룹 지정
(adduser --gid 1001 user1) adduser -gid (그룹 번호) (사용자 name)

```
root@server:~/바탕화면# tail -5 /etc/passwd
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
ubuntu:x:1000:1000:ubuntu,,,:/home/ubuntu:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
user1:x:1001:1001:,,,:/home/user1:/bin/bash
user2:x:1002:1001:,,,:/home/user2:/bin/bash
```

➔ 같은 방법으로 사용자 2(user2) 생성 뒤 두 사용자의 정보 확인

➤ user1, user2 모두 같은 그룹 (1001) 인 모습을 확인 가능

```
root@server:~/바탕화면# tail -5 /etc/shadow
gdm:*:18375:0:99999:7:::
ubuntu:$6$P4hzxyu791fERP5$PX0JfHngUfyH9SfqJKR
hz4hYbGh/:19058:0:99999:7:::
systemd-coredump:!!:19059:::::::
user1:$6$STK0NdzedfYGa9RB$zHdP2Bwhr/sNsY3SFL6d
LILPBRr1:19062:0:99999:7:::
user2:$6$uB5qys5w0ek56Tik$0FcKfAqGZhGBMqUuyh7h
ZJQqtMq/:19062:0:99999:7:::
root@server:~/바탕화면#
```

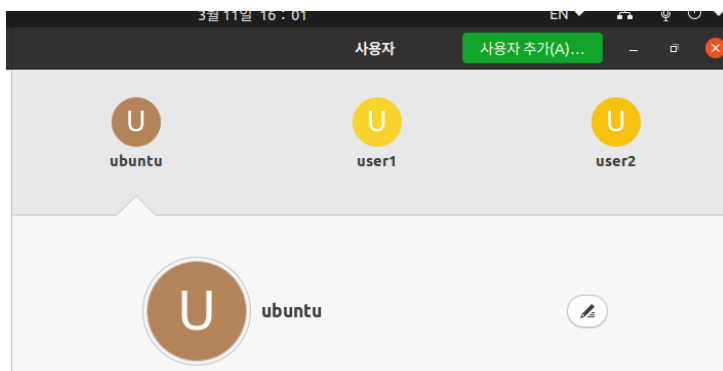
➔ shadow 파일을 통해 user 의 비밀번호 정보 확인

➤ 비교를 위해 user1 과 user2 의 비밀번호를 동일하게 설정('1234') 하였으나
암호화된 비밀번호는 전혀 다른 패턴 (비밀번호 파악 유추 불가)

```
root@server:~/바탕화면# ls -l /home/
합계 12
drwxr-xr-x 14 ubuntu ubuntu    4096 3월  8 10:42 ubuntu
drwxr-xr-x  2 user1  ubuntuGroup 4096 3월 11 15:20 user1
drwxr-xr-x  2 user2  ubuntuGroup 4096 3월 11 15:24 user2
root@server:~/바탕화면# ls -a /home/user1
. . . .bash_logout .bashrc .profile
root@server:~/바탕화면# ls -a /etc/skel/
. . . .bash_logout .bashrc .profile
root@server:~/바탕화면#
```

➔ 생성된 사용자들의 디렉터리 확인

➤ /home/user1 디렉터리의 파일들은 /etc/skel/ 디렉터리의 파일들을 복사해서 넣는 개념
(사용자를 생성할 때 추가할 파일들은 skel 디렉터리에 추가하면 됨)



취소(C) 사용자 추가 추가(A)

계정 종류(T) 표준 관리자

전체 이름(F) user3 ✓

사용자 이름(U) user3 ✓

홈 폴더의 이름입니다. 바꿀 수 없습니다.

암호

☐ 다음 로그인에 암호를 설정하도록 허용(L)

☒ 지금 암호 설정(N)

암호(P) [masked]

일반적인 단어를 피해 보십시오.

확인(C) []

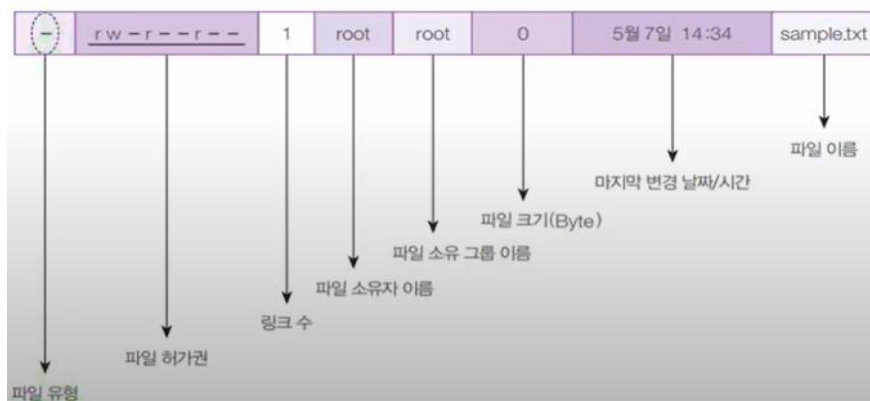
➔ xwindow 모드로도 관리 가능

➤ 암호 설정 시 8 글자 이상 , 문자+숫자 필수

● 파일과 디렉터리의 소유와 허가권 (1)

➤ 파일의 리스트와 파일 속성

```
root@server: ~/바탕화면
root@server:~/바탕화면# touch sample.txt
root@server:~/바탕화면# ls -l sample.txt
-rw-r--r-- 1 root root 0 5월 7 14:34 sample.txt
root@server:~/바탕화면#
```



● 파일과 디렉터리의 소유와 허가권 (2)

➤ 파일 유형

디렉터리일 경우에는 d, 일반적인 파일일 경우에는 -가 표시

➤ 파일 허가권 (Permission)

"rw-", "r--", "r--" 3 개씩 끊어서 읽음 (r 은 read, w 는 write, x 는 execute 의 약자)

첫 번째 "rw-"는 소유자(user)의 파일접근 권한

두 번째의 "r--"는 그룹(Group)의 파일접근 권한

세 번째의 "r--"는 그 외의 사용자(other)의 파일접근 권한

숫자로도 표시 가능 (8 진수)

소유자(User)			그룹(Group)			그 외 사용자(Other)		
r	w	-	r	-	-	r	-	-
4	2	0	4	0	0	4	0	0
6			4			4		

● 파일과 디렉터리의 소유와 허가권 (3)

➤ chmod 명령

파일 허가권 변경 명령어

예) # chmod 777 sample.txt

➤ 파일 소유권(Ownership)

파일을 소유한 사용자와 그룹을 의미

➤ chown/chgrp 명령

파일의 소유권을 바꾸는 명령어 (root 사용자만 사용가능)

예) # chown ubuntu.ubuntu sample.txt 또는

소유자 소유그룹

chown ubuntu sample.txt 및 # chgrp ubuntu sample.txt

```

root@server:~#
root@server:~# ls -l test
-rw-r--r-- 1 root root 44  3월 11 16:29 test
root@server:~# whoami
root
root@server:~# ./test
bash: ./test: 허가 거부
root@server:~#

```

➔ 관리자 권한인 root 사용자로 test 파일 실행 시 허가 거부 발생

- nano 에디터로 test 파일 생성
- root 사용자로 nano 에디터로 생성된 test 파일 실행 > 허가 거부
- root 사용자 권한이 rw(읽기 쓰기) 만 허용, 실행 권한 없는 상태

```

root@server:~# chmod 755 test
root@server:~#
root@server:~# ls -l test
-rwxr-xr-x 1 root root 44  3월 11 16:29 test
root@server:~# ./test
./test: 줄 1: 안녕하세요: 명령어를 찾을 수 없음
backups cache crash lib local lock log mail metrics opt run snap spool tmp
root@server:~#

```

➔ 실행 권한 추가 후 실행 완료

- test 파일의 파일 허가권을 755 로 변경 ('chmod 755 test')

-rw-r--r-- > -rwx-r-x-r-x 로 변경됨

- 변경 후 root 사용자로 실행 완료

```

root@server:~# chown ubuntu test
root@server:~#
root@server:~# ls -l test
-rwxr-xr-x 1 ubuntu root 44  3월 11 16:29 test
root@server:~#
root@server:~# chgrp ubuntu test
root@server:~# ls -l test
-rwxr-xr-x 1 ubuntu ubuntu 44  3월 11 16:29 test
root@server:~#

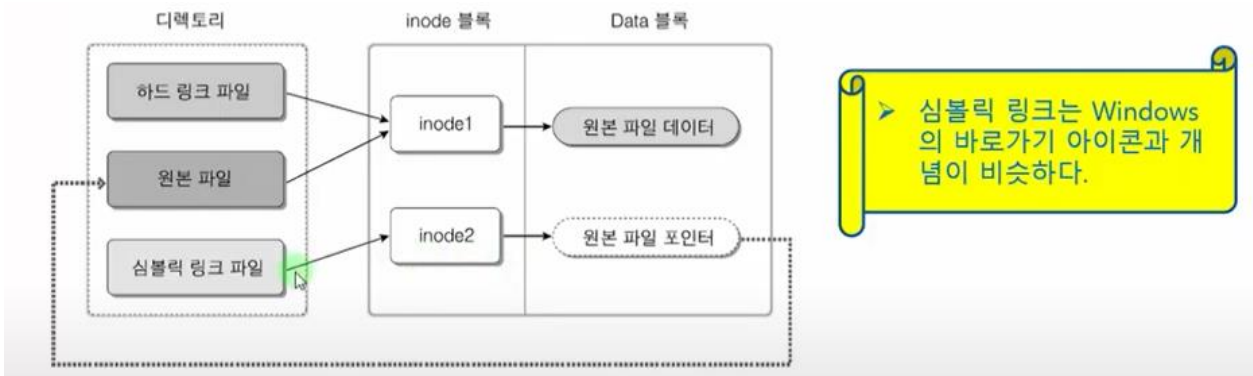
```

➔ ubuntu 사용자로 소유권 변경

- chown 명령어를 사용하여 test 파일의 소유권을 ubuntu 사용자로 변경
- chgrp 명령어를 사용하여 test 파일의 소유권을 ubuntu 그룹으로 변경

- 링크

- 파일의 링크(Link)에는 하드 링크(Hard Link)와 심볼릭 링크(Symbolic Link 또는 Soft Link) 두 가지가 있음



- 하드 링크를 생성하면 "하드 링크파일"만 하나 생성되며 같은 inode1 을 사용 (명령: # ln 링크대상파일이름 링크파일이름)
- 심볼릭 링크를 생성하면 새로운 inode2 를 만들고, 데이터는 원본 파일을 연결하는 효과 (명령 : # ln -s 링크대상파일이름 링크파일이름)

- linktest 디렉터리에 basefile 과 basefile 에 연결되는 hardlink, softlink 생성

- 연결 확인 후 basefile 을 다른 디렉터리로 이동 (mv 명령어)

```
root@server:~/linktest# mv basefile ../
root@server:~/linktest# ls
hardlink  softlink
root@server:~/linktest# cat hardlink
원본 파일입니다...
root@server:~/linktest# cat softlink
cat: softlink: 그런 파일이나 디렉터리가 없습니다
root@server:~/linktest#
```

- ➔ hardlink 는 작동되나 softlink 는 파일을 찾지 못함
- hardlink 는 원본 파일과 같은 inode1 에 연결하여 원본 파일 데이터와 연결된 반면 softlink 는 inode2 에 연결되어 원본 파일 포인터로 원본 파일을 찾아야 하기 때문에 다른 디렉터리로 이동한 원본 파일을 찾지 못함

```

root@server:~/linktest# mv ../basefile .
root@server:~/linktest# cat hardlink
원본 파일입니다...
root@server:~/linktest# cat softlink
원본 파일입니다...
root@server:~/linktest# █

```

➔ 원본 파일이 원래 디렉터리(linktest) 로 돌아오니 softlink 도 정상 작동

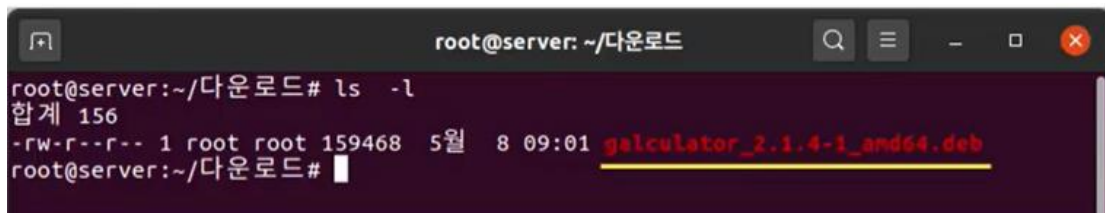
● 프로그램 설치를 위한 dpkg (1)

➤ dpkg (Debian Package)

windows 의 "setup.exe"와 비슷한 설치 파일

확장명은 *.deb 이며, 이를 패키지(package)라고 부름.

➤ 파일의 의미



```

root@server: ~/다운로드
root@server:~/다운로드# ls -l
합계 156
-rw-r--r-- 1 root root 159468  5월  8 09:01 calculator_2.1.4-1_amd64.deb
root@server:~/다운로드# █

```

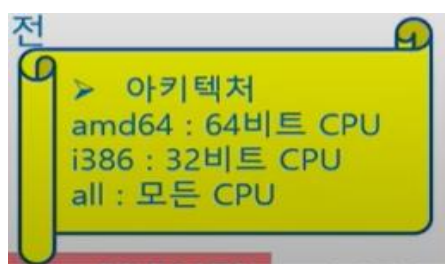
패키지이름_버전-개정번호_아키텍처.deb

패키지이름 : calculator > 패키지(프로그램)의 이름

버전 : 2.1.4 > 대개 3 자리수로 구성. 주 버전,부 버전,패치버전

개정번호 : 1 > 문제점을 개선할 때마다 붙여지는 번호

아키텍처 : amd64 > 64 비트 CPU 를 의미



- 프로그램 설치를 위한 dpkg (2)

- 자주 사용하는 dpkg 명령어 옵션

- ✓ 설치

- `dpkg -i 패키지파일이름.deb` (소문자 아이)

- ✓ 삭제

- `dpkg -r 패키지이름`

- `dpkg -p 패키지이름 > 설정파일까지 삭제`

- ✓ 패키지 조회

- `dpkg -i 패키지이름 > 설치된 패키지에 대한 정보를 보여줌` (소문자 엘)

- `dpkg -L 패키지이름 > 패키지가 설치한 파일 목록을 보여줌`

- ✓ 아직 설치되지 않은 deb 파일 조회

- `dpkg --info 패키지파일이름.deb > 패키지 파일에 대한 정보를 보여줌`

- 프로그램 설치를 위한 dpkg (3)

- dpkg 명령의 단점

- ✓ '의존성' 문제

- A 패키지가 설치되기 위해서 B 패키지가 필요할 경우, dpkg 명령으로는 해결이 까다로움

- ✓ 이름 해결하기 위해 apt 가 등장

```
root@server:~/다운로드# ls -l
합계 220
-rw-r--r-- 1 root root 58668 3월 11 17:42 axel_2.17.5-1ubuntu1_amd64.deb
-rw-r--r-- 1 root root 161800 3월 11 17:41 galculator_2.1.4-1build1_amd64.deb
root@server:~/다운로드#
```

➔ deb 파일 2 개 저장 상태 확인

```

root@server:~/다운로드# dpkg --info axel_2.17.5-1ubuntu1_amd64.deb
new Debian package, version 2.0.
size 58668 bytes: control archive=1452 bytes.
 947 바이트, 22 줄 control
1440 바이트, 21 줄 md5sums
Package: axel
Version: 2.17.5-1ubuntu1
Architecture: amd64
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Installed-Size: 201
Depends: libc6 (>= 2.15), libssl1.1 (>= 1.1.0)
Section: web
Priority: optional
Homepage: https://github.com/axel-download-accelerator/axel
Description: light command line download accelerator
 Axel tries to accelerate the downloading process by using multiple
 connections for one file, similar to DownThemAll and other famous
 programs. It can also use multiple mirrors for one download.
.
Using Axel, you will get files faster from Internet. So, Axel can
 speed up a download up to 60% (approximately, according to some tests).
.
Axel supports HTTP, HTTPS, FTP and FTPS protocols.
.
Axel tries to be as light as possible, so it might be useful as a
 wget clone (and other console based programs) on byte-critical systems.
Original-Maintainer: Joao Eriberto Mota Filho <eriberto@debian.org>
root@server:~/다운로드#

```

➔ "dpkg --info 패키지이름" 명령어로 해당 패키지 내용 확인

```

root@server:~/다운로드# dpkg -i axel_2.17.5-1ubuntu1_amd64.deb
Selecting previously unselected package axel.
(데이터베이스 읽는중 ...현재 181353개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack axel_2.17.5-1ubuntu1_amd64.deb ...
Unpacking axel (2.17.5-1ubuntu1) ...
axel (2.17.5-1ubuntu1) 설정하는 중입니다 ...
Processing triggers for man-db (2.9.1-1) ...
root@server:~/다운로드#

```

➔ "dpkg -i 패키지이름" 명령어로 해당 패키지 다운로드

```

root@server:~/다운로드# dpkg -r axel
(데이터베이스 읽는중 ...현재 181378개의 파일과 디렉터리가 설치되어 있습니다.)
axel (2.17.5-1ubuntu1)를 제거합니다...
Processing triggers for man-db (2.9.1-1) ...
root@server:~/다운로드#

```

➔ "dpkg -r 패키지이름" 명령어로 해당 패키지 삭제

```

root@server:~/다운로드# dpkg -i galculator_2.1.4-1build1_amd64.deb
Selecting previously unselected package galculator.
(데이터베이스 읽는중 ...현재 181353개의 파일과 디렉터리가 설치되어 있습니다.)
Preparing to unpack galculator_2.1.4-1build1_amd64.deb ...
Unpacking galculator (2.1.4-1build1) ...
dpkg: 종속성 문제로 galculator의 구성이 차단되었습니다:
 galculator 패키지는 다음 패키지에 의존: libquadmath0 (>= 4.6): 하지만:
 libquadmath0 패키지는 설치하지 않았습니다.

dpkg: error processing package galculator (--install):
 의존성 문제 - 설정하지 않고 남겨둠
Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...
Processing triggers for desktop-file-utils (0.24-1ubuntu2) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for hicolor-icon-theme (0.17-2) ...
Processing triggers for man-db (2.9.1-1) ...
처리하는데 오류가 발생했습니다:
 galculator
root@server:~/다운로드#

```

➔ 의존성 패키지를 dpkg -i 명령어로 설치 시 오류

➤ 해당 패키지가 의존하는 패키지부터 설치해야 오류 해결