

- Ch\_1 \_ 런레벨, 자동완성과 히스토리
- Ch\_2 \_ gedit, nano, vi 기본 사용법
- Ch\_3 \_ vi 비정상 종료, CD Rom 마운트

## 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 3로 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 /t
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 /t
target
54 ls -l default.target
55 reboot
56 startx
57 ln -sf /lib/systemd/system/graphical.target /t
target
58 reboot
59 reboot
60 history
root@server:~/바탕화면#
```

➔ History 로 지금까지 사용한 명령어들 확인가능

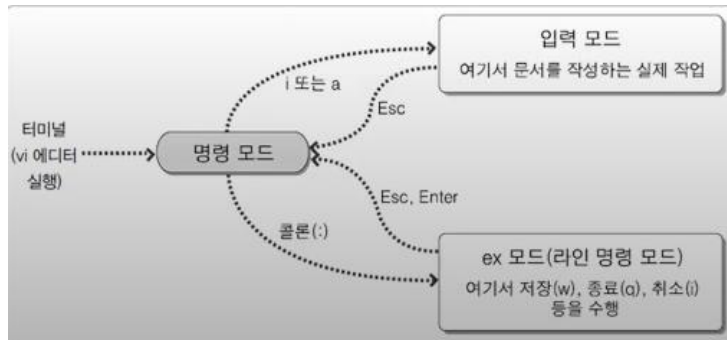
- 보안상 등 이유로 불필요할시 history -c 로 클리어 가능

```
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



### → 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

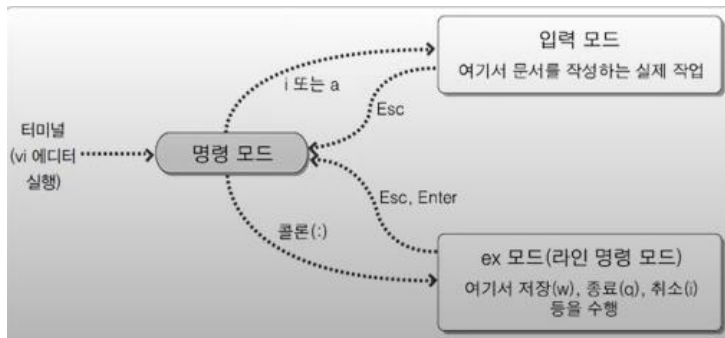
- Gedit 에디터와는 다르게 xwindow 가 없어도 사용 가능 (gedit 에디터는 xwindow 에서만 사용가능)
- Nano 에디터에서 Ctrl + c 로 행 번호 파악 가능 (nano -c 로 에디터를 오픈하면 계속 행번호가 나타남)

```
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)

```
root@server: ~/바탕화면
dasdasffff
에디터연습
:
^L
|
~
~
~
~
~
~
~
~
~
~
:w myfile3.txt
```

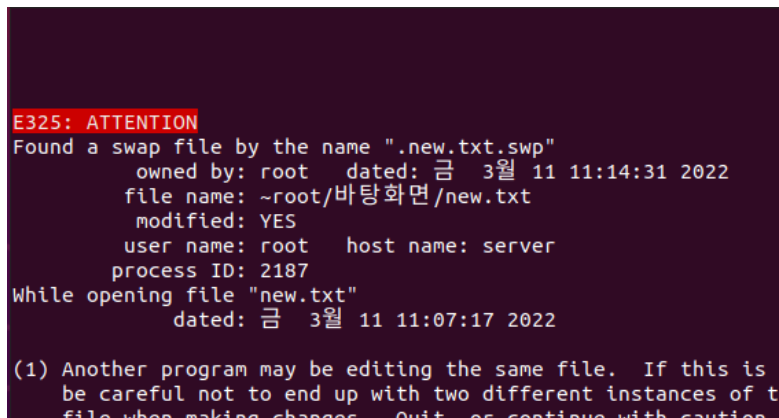
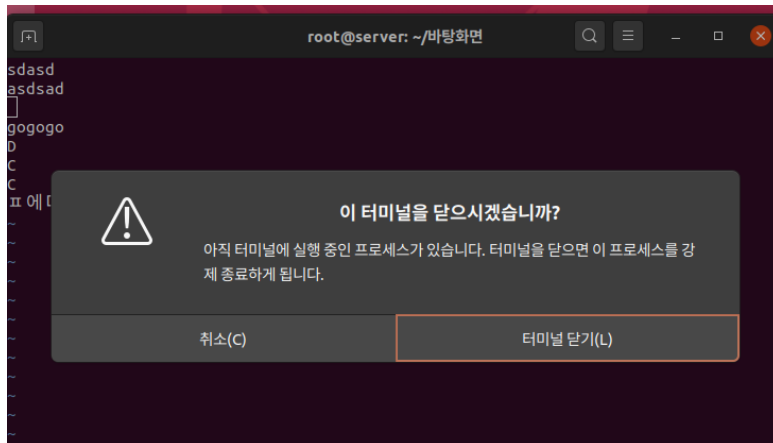
➔ Vi 에디터 사용 화면



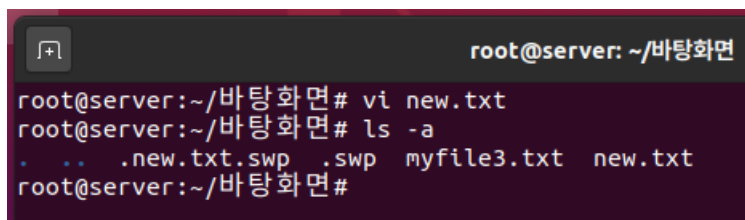
- Vi 에디터도 나노 에디터와 동일하게 xwindow 가 없어도 사용 가능
- vi 에디터 실행 뒤 명령모드에서 i 또는 a 명령어로 입력모드로 전환 후 사용가능
- 문서 작성 후 esc 로 명령모드 이후 shift + ; ex 모드 (라인 명령 모드) ex 모드에서 저장(w) 종료 (q) 취소(c) 가능
- 작성 중 저장하지 않고 종료 (q!)



## Vi 비정상 종료



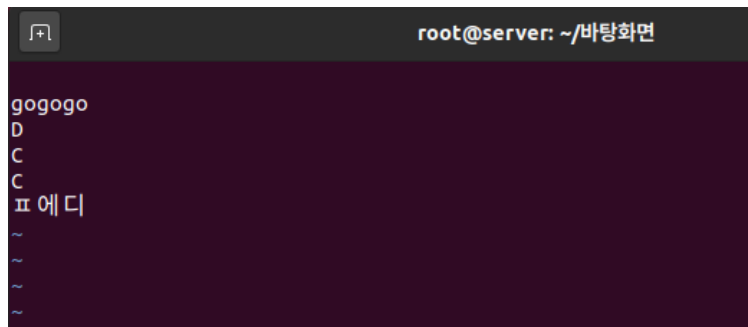
➔ Vi 에디터를 사용 중 터미널을 닫았을 때 나오는 오류



➔ 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
```



➔ 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)

l | 커서를 오른쪽으로 한 칸 이동 (→와 같은 의미, l)

- 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)

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

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

P | 복사한 내용을 현재 행 이전에 붙여 넣기 (shift + p)

[illegible]

➔ 나노 에디터 'set number' 명령어로 행 나타냄

```
LS(1)                                User Commands
NAME
  ls - list directory contents
SYNOPSIS
  ls [OPTION]... [FILE]...
DESCRIPTION
  List information about the FILES (the current dir
  Sort entries alphabetically if none of -cftuvSUX nor
  fied.

  Mandatory arguments to long options are mandator
  too.

  -a, --all
      do not ignore entries starting with .

  -A, --almost-all
      do not list implied . and ..

  --author
Manual page ls(1) line 1 (press h for help or q to quit)
```

➔ Man 명령어를 사용한 도움말 출력

- Man ls 명령어로 ls 에 대한 도움말을 출력한 상태
- 넘기기는 space 뒤로 이동은 b 나가기는 q

마운트와 CD/DVD, USB 의 활용

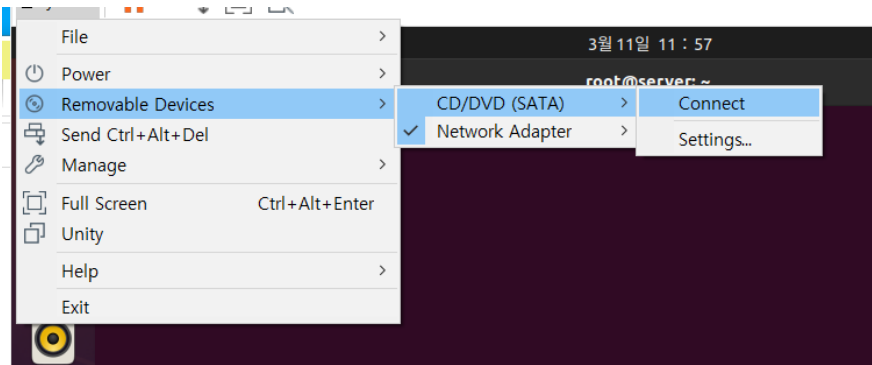
- 물리적인 장치를 특정한 위치 (대개는 디렉터리) 에 연결시켜 주는 과정

```
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

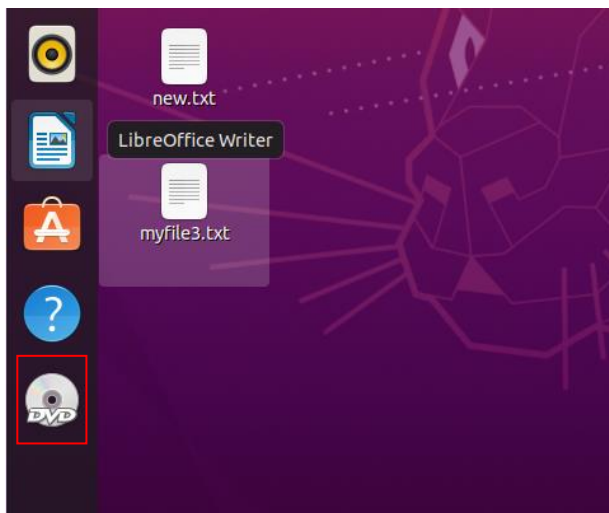
- 현 장치에 연결되어 있는 하드웨어장치들이 어느 장소에 연결이 되어있는 지 나타내는 명령어

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 umonunt 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