

Chapter 03

리눅스 기본 사용법



학습목표

- 우분투 리눅스를 시작하고 종료하는 방법에 대해 설명할 수 있다.
- 우분투 리눅스의 데스크톱 환경을 설명할 수 있다.
- 명령어 창인 터미널의 사용법을 설명할 수 있다.
- 셸이 무엇인지와 그 사용법에 대해 설명할 수 있다.



학습내용

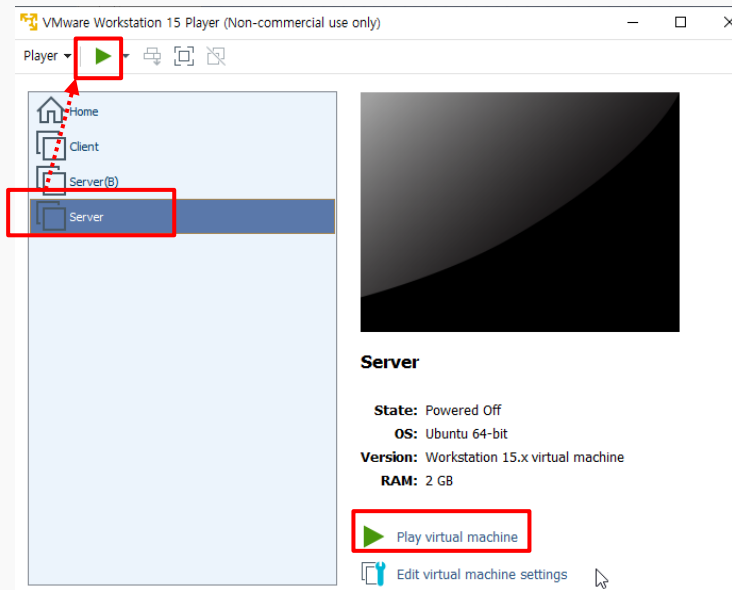
- ❖ 우분투 리눅스 기본 사용법
 - 시작과 종료
 - 우분투 데스크톱 환경
- ❖ 터미널과 셸
 - 터미널 사용법
 - 셸



시작과 종료

시작

- VMware Workstation Player의 3개의 가상머신 중 선택하여 실행
- 상단의 ► 표시나 하단의 Play virtual machine 선택
- 여기서는 Server 가상머신으로 시작

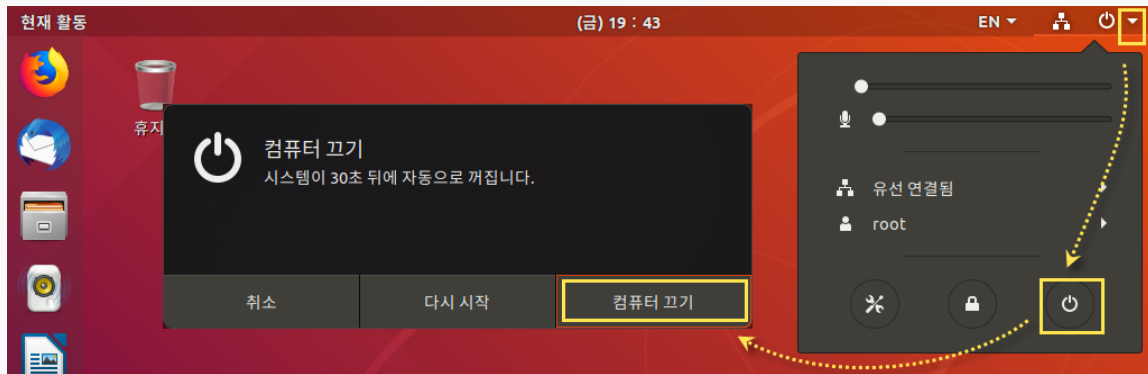




시작과 종료

종료

- 바탕화면 오른쪽 위의 ▼ 아이콘, 컴퓨터 끄기 아이콘, <컴퓨터 끄기>를 이어서 클릭

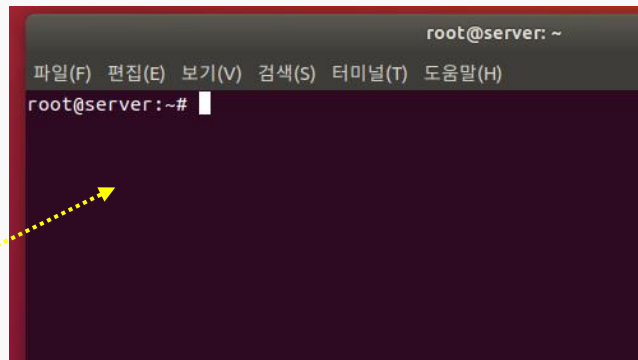
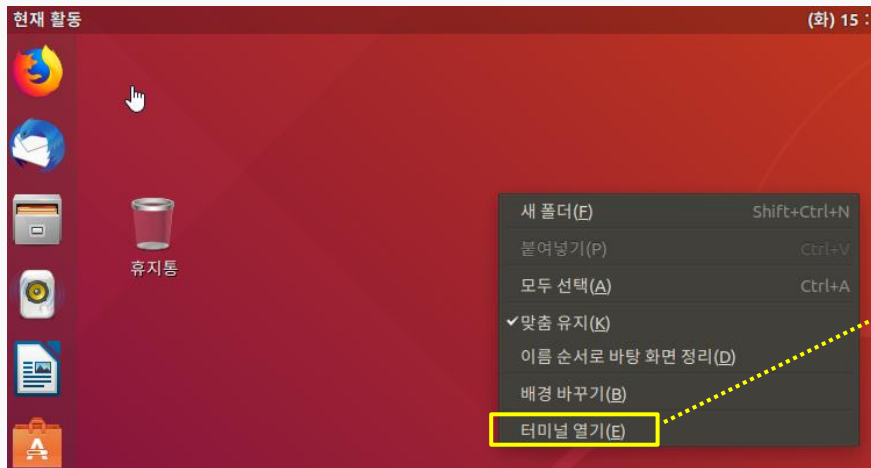




시작과 종료

터미널을 통한 종료

- 먼저 터미널을 실행하려면 바탕화면에 마우스 커서가 있는 상태에서 마우스 오른쪽 버튼 선택
- 메뉴 중 터미널 열기 선택





시작과 종료

터미널을 통한 종료

- 터미널에서 시스템 종료 명령 실행
- 다음의 4가지 명령 모두 시스템 종료 명령

```
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T)  
root@server:~# poweroff
```

```
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T)  
root@server:~# halt -p
```

```
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T)  
root@server:~# shutdown -P now
```

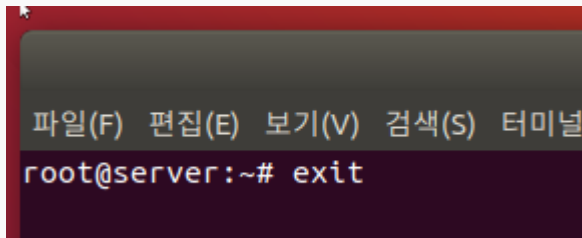
```
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T)  
root@server:~# init 0
```



시작과 종료

터미널 닫기

- 터미널에서 exit입력



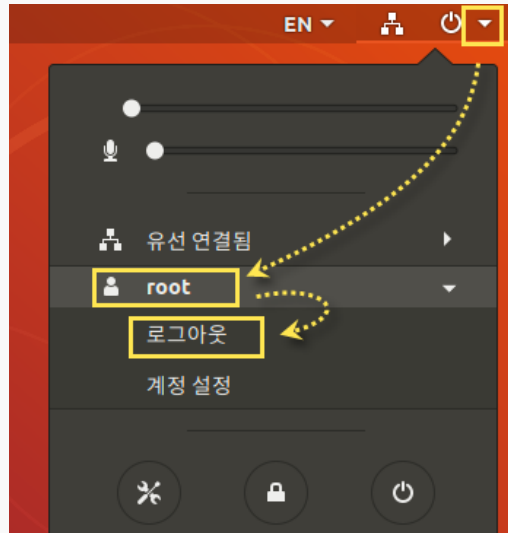
```
파일(F) 편집(E) 보기(V) 검색(S) 터미널
root@server:~# exit
```



시작과 종료

로그 아웃

- 리눅스는 여러 사용자가 동시에 접속하는 다중 사용자(multiuser) 시스템
- 자신만 접속을 끝내는 로그아웃이 필요
- 만약 관리자가 자기 업무가 끝났다고 시스템을 종료해 버리면, 시스템에 접속된 많은 사용자의 컴퓨터도 함께 종료됨
- X 윈도우에서 로그아웃 : 바탕화면 오른쪽 위 ▼ 아이콘, [root], <로그아웃>을 이어서 클릭
- Server(B)와 같은 텍스트 모드에서 로그아웃 : logout 또는 exit 명령 실행

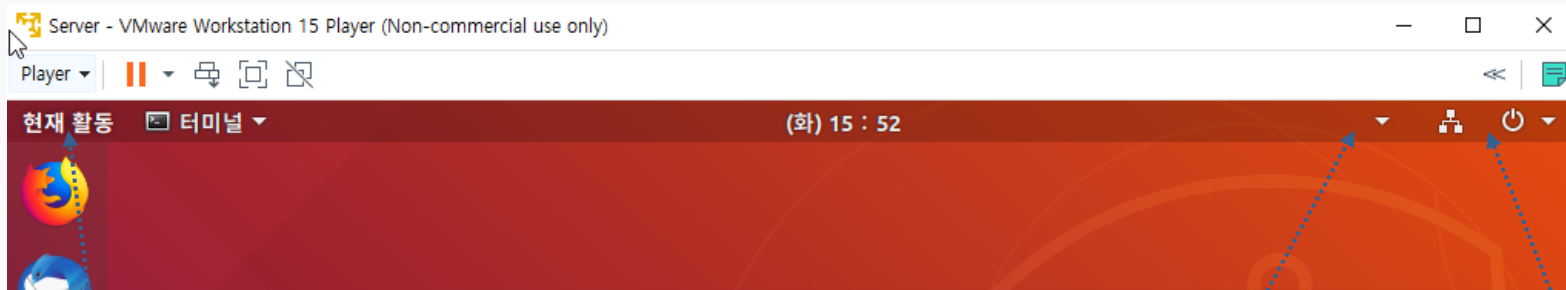


우분투 리눅스 기본 사용법



우분투 데스크톱 환경

우분투 데스크톱 상단의 구성요소



현재활동
자주 사용하는
프로그램의 아이콘과
현재 실행중인
프로그램 스냅샷 출력

한글 입력 방식

기타 설정
소리 조절, 네트워크
설정, 사용자 정보,
시스템 설정, 리눅스
종료

우분투 리눅스 기본 사용법



우분투 데스크톱 환경

시작 아이콘들

- 왼쪽에는 프로그램을 시작할 수 있는 아이콘들 배열
- 파이어 폭스는 웹 브라우저
- 파일은 윈도우 탐색기에 해당
- 오피스는 LibreOffice로 마이크로 소프트의 오피스에 해당
- 우분투 소프트웨어는 우분투 리눅스에 설치할 수 있는 소프트웨어를 보여주고 설치하거나 제거할 수 있음
- 프로그램 표시는 우분투에 현재 설치되어 있는 소프트웨어들을 보여주고 시작하는 메뉴



프로그램 선택

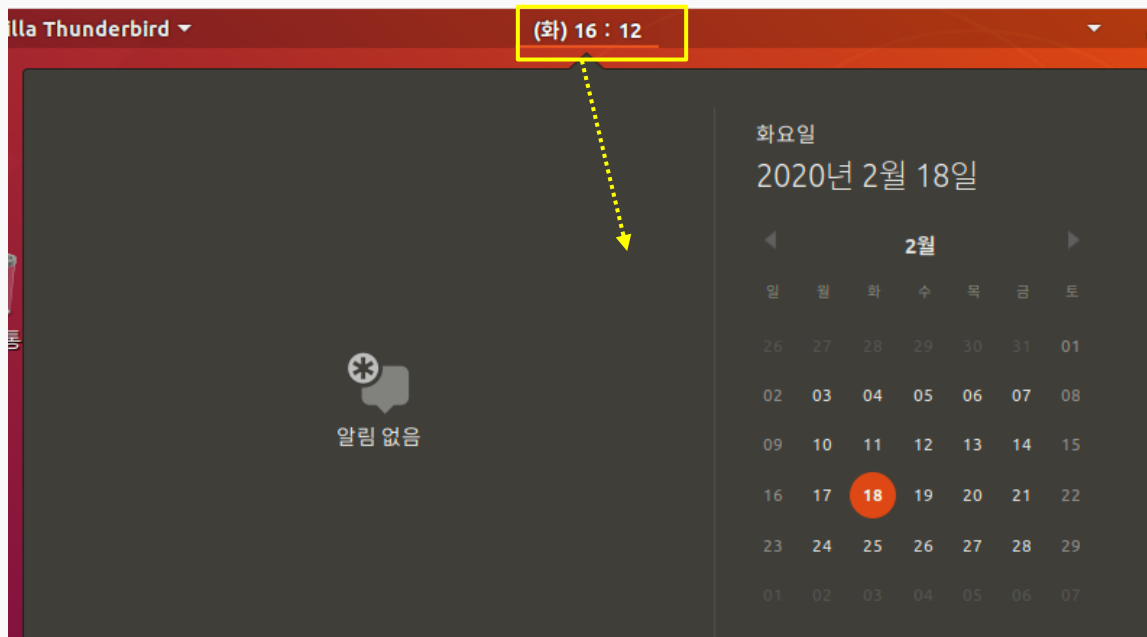
-



우분투 데스크톱 환경

날짜와 시간 및 알림 관리

- 바탕화면 상단의 시간을 선택하면 날짜와 시간을 설정하고 알림을 확인

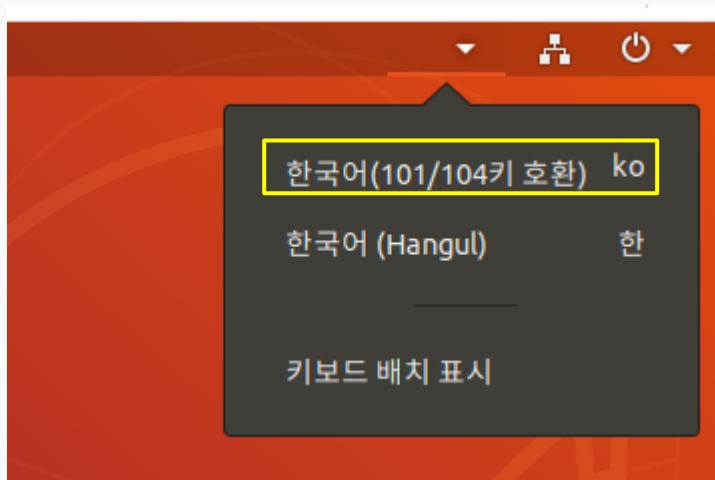




우분투 데스크톱 환경

한국어 입력 방식 선택

- 바탕화면의 오른쪽에 있는 ko를 선택하면 한국어 입력 방식 선택 가능

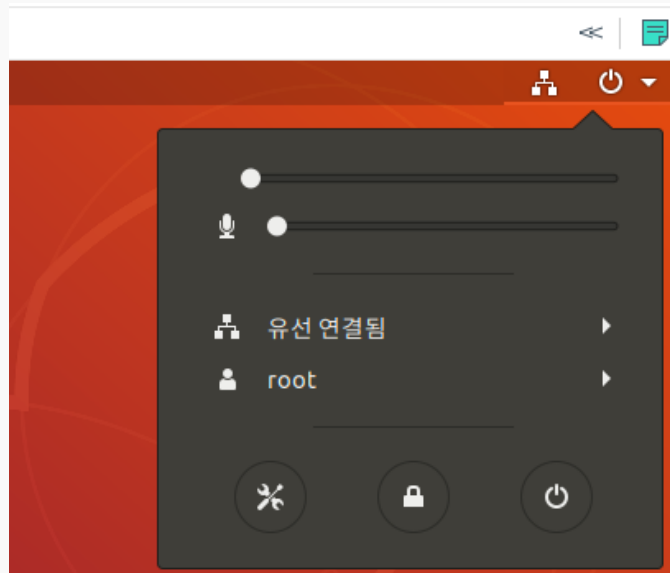




우분투 데스크톱 환경

기타 설정

- 바탕화면의 상단 오른쪽에 있는 아이콘을 클릭
- 소리 설정, 네트워크 설정, 블루투스 설정, 사용자 설정, 시스템 설정, 시스템 종료 등의 메뉴



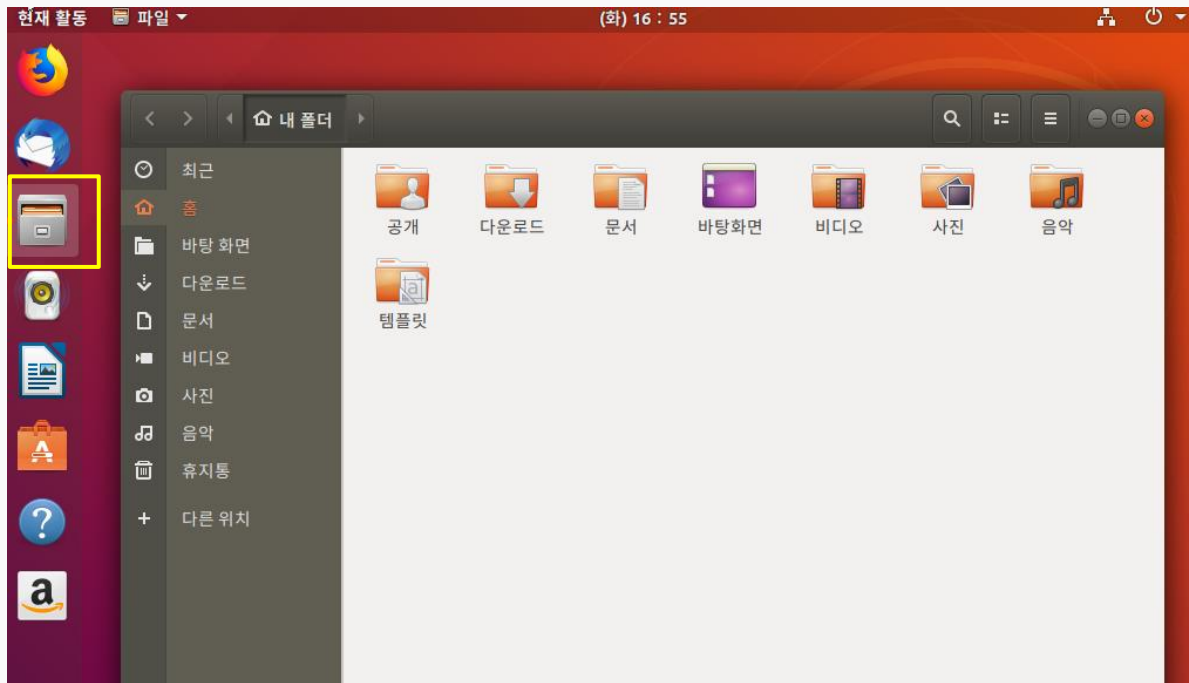
우분투 리눅스 기본 사용법



우분투 데스크톱 환경

파일

- 파일을 관리하는 프로그램, 윈도우 탐색기와 같음



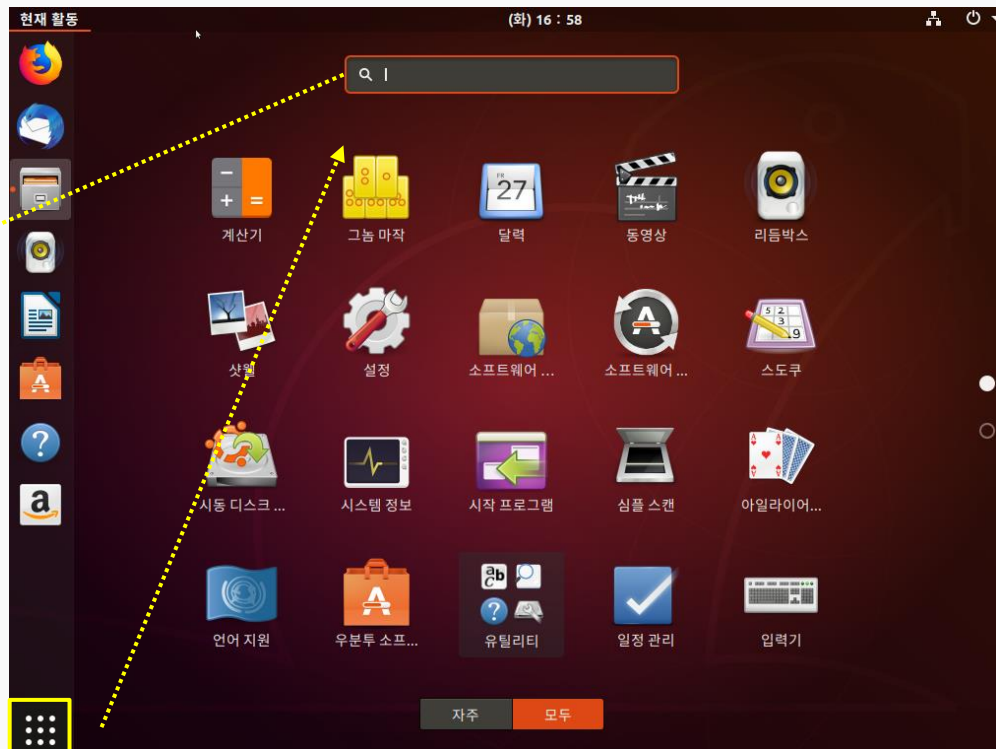
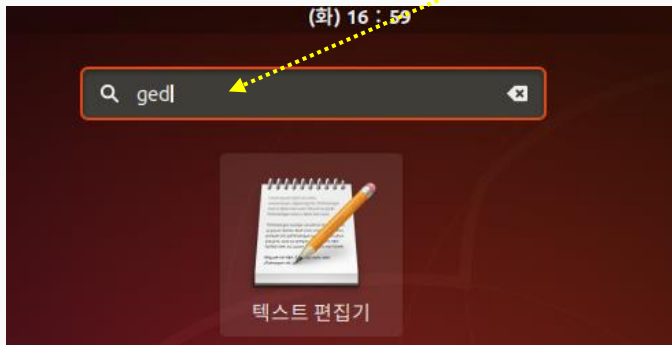
우분투 리눅스 기본 사용법



우분투 데스크톱 환경

텍스트 편집기 찾기

- 프로그램 표시를 선택하고
검색창에 gedit라고 입력하면
텍스트 편집기를 찾을 수 있음.



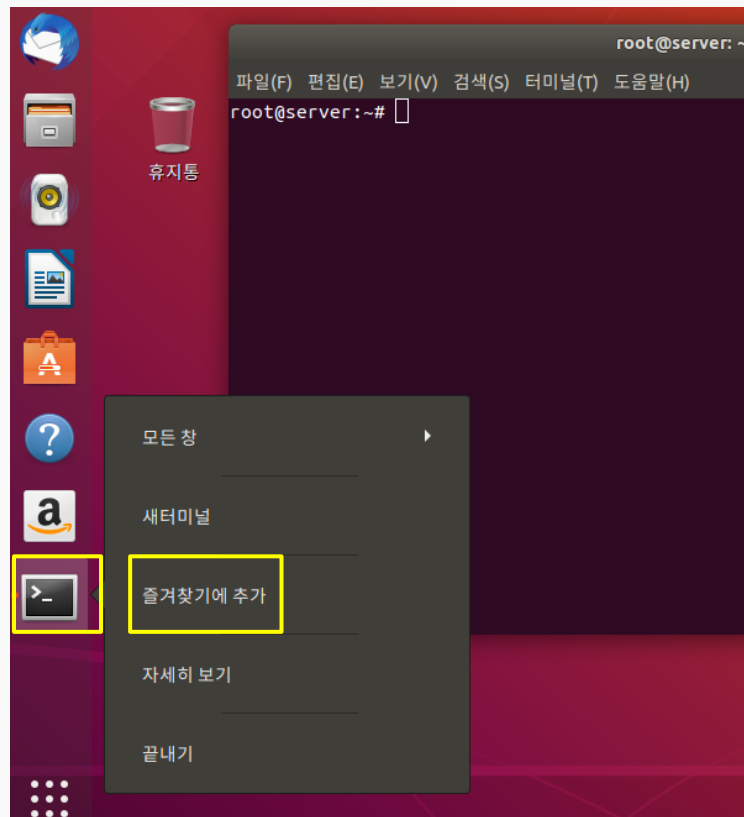
우분투 리눅스 기본 사용법



우분투 데스크톱 환경

자주 사용하는 프로그램 아이콘 고정

- 자주 사용하는 프로그램의 아이콘을 시작 아이콘에 추가가능
- 프로그램 실행후 해당 아이콘을 그림과 같은 메뉴에서 “즐거찾기에 추가” 하면 시작 아이콘에 고정됨.
- 터미널은 리눅스에서 자주 사용되며 즐겨찾기에 추가해두면 편리함.

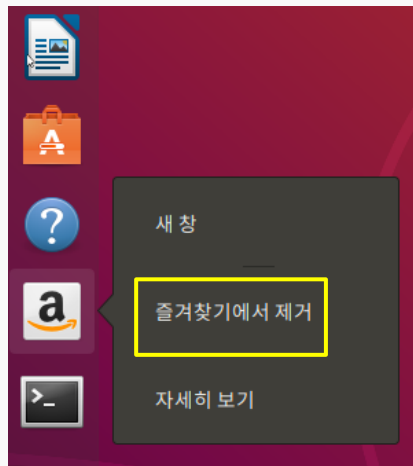




우분투 데스크톱 환경

자주 사용하지 않는 프로그램 아이콘 제거


- 반대로 자주 사용하지 않는 아이콘은 그림과 같이 “즐거찾기에서 제거”를 선택하면 아이콘을 지울 수 있음.





터미널 사용법

자동 완성 기능

- 리눅스는 터미널창에 길고 어려운 파일명이나 디렉토리명을 입력해야하는 경우가 많음.
- 하지만 자동 완성 기능을 이용하면 보다 쉽게 파일명이나 디렉토리 명을 입력할 수 있음.
- 파일 이름의 일부만 입력하고 Tab을 눌러 나머지 파일 또는 디렉터리(폴더)의 이름을 자동으로 완성하는 기능
- 예를 들어 “/lib/systemd”라는 디렉토리로 찾아가는 명령은 `cd /lib/systemd`이다.
- 터미널 창에서 `cd /lib/sys`  을 입력하면 자동으로 /lib/systemd가 완성됨



터미널 사용법

자동 완성 기능

- 예를 들어 “/lib/systemd/system” 이라는 디렉토리를 찾아간다면
- 터미널 창에 #cd /lib/system/ 입력후 Tab키를 빠르게 두번 입력하면 선택 가능한 모든 디렉토리와 파일명이 출력되고
- 터미널 창에 #cd /lib/system/sys 를 입력 후 Tab키를 입력하면 자동으로 #cd /lib/system/system 까지 완성됨.

```
root@server: ~  
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)  
libsystemd-shared-237.so      systemd-logind  
network/                     systemd-makefs  
resolv.conf                   systemd-modules-load  
set-cpufreq                   systemd-networkd  
system/                       systemd-networkd-wait-online  
system-generators/           systemd-quotacheck  
system-preset/                systemd-random-seed  
system-shutdown/              systemd-remount-fs  
system-sleep/                 systemd-reply-password  
systemd                       systemd-resolved  
systemd-ac-power              systemd-rfkill  
systemd-backlight             systemd-shutdown  
systemd-binfmt                systemd-sleep  
systemd-cgroups-agent         systemd-socket-proxyd  
systemd-cryptsetup            systemd-sulogin-shell  
systemd-dissect                systemd-sysctl  
systemd-fsck                   systemd-sysv-install  
systemd-fsckd                  systemd-timedated  
systemd-growfs                 systemd-timesyncd  
systemd-hibernate-resume      systemd-udev  
systemd-hostnamed              systemd-update-utmp  
systemd-initctl                 systemd-user-sessions  
systemd-journald                systemd-veritysetup  
--More--
```

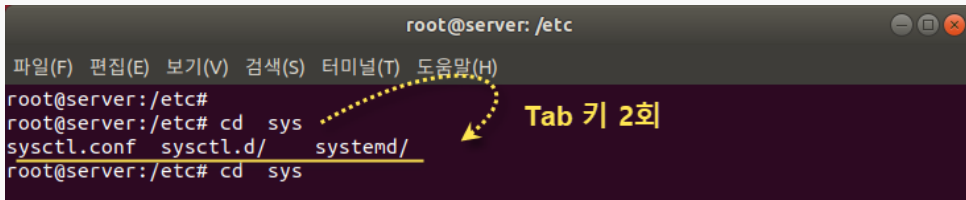


터미널 사용법

자동 완성 기능

- 비슷한 이름이 여러 개 있을 경우
- 예) /etc/systemd 로 이동하려고 하는데 /etc에는 sys로 시작하는 디렉토리가 3개 있다. 이중 systemd를 선택할 때 자동완성 기능으로 쉽게 이동할 수 있다.

- ① 먼저 cd /etc 명령을 입력, /etc 디렉터리로 이동
- ② 이때 cd sys입력후 **Tab** 입력하면 아무런 반응이 없음
- ③ 다시 **Tab** 을 입력하면 3개의 이름 후보가 나옴



```
root@server: /etc
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:/etc#
root@server:/etc# cd sys
sysctl.conf sysctl.d/ systemd/
root@server:/etc# cd sys
```

The screenshot shows a terminal window with the title 'root@server: /etc'. The command history shows 'cd sys' being entered, followed by a list of suggestions: 'sysctl.conf', 'sysctl.d/', and 'systemd/'. A yellow dotted arrow points from the text 'Tab 키 2회' to the second 'Tab' key press in the command sequence.

- ④ 여기서 t를 하나 더 입력해서 syst가 되면 syst으로 시작하는 디렉토리가 systemd 하나 이므로 뒤에 emd가 자동 완성으로 붙음.



터미널 사용법

명령어 관리

- 터미널 창에서 여러 개의 명령을 계속 입력하게 되는데 앞에 입력했던 명령어를 다시 입력하려면 방향 키를 이용하면 편리함.
- 그동안 입력했던 명령어를 방향키 ↑와 방향키 ↓를 사용하면 명령을 입력했던 순서대로 보여주며 선택하려면 Enter를 누르면 됨.
- 이전에 사용했던 명령을 모두 보려면 history 명령을 입력
- 이들중 다시 사용하려면 !와 앞의 번호를 입력하면 됨.

```
root@server: /etc
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:/etc/systemd# history
 1 passwd
 2 whoami
 3 exit
 4 gedit /etc/gdm3/custom.conf
 5 gedit /etc/pam.d/gdm-password
 6 gedit /etc/pam.d/gdm-autologin
 7 gedit /root/.profile
 8 reboot
 9 exit
10 cd /lib/systemd/system/
11 pwd
12 ls
13 clear
14 cd
15 clear
16 cd /lib/systemd/
17 ls
18 clear
19 cd /etc/systemd/
20 clear
21 history
root@server:/etc/systemd#
```



터미널 사용법

명령어 관리

- 예를 들어 앞의 예에서 17번 ls명령을 다시 수행하려면
- # !17 입력

```
21 history
root@server:/etc/systemd# !17
ls
journald.conf  network      system      timesyncd.conf  user.conf
logind.conf    resolved.conf system.conf  user
```

- 그림과 같이 ls명령이 수행되어 결과를 출력함.
- 복잡하고 긴 길이의 명령을 입력해야 하는 경우에 유용하게 사용됨.



터미널 사용법

터미널 화면 청소

- 터미널 창에 #clear를 입력하면 기존 화면 내용을 제거

```
root@server: /lib/systemd
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:~# cd /lib/systemd/
root@server:/lib/systemd# ls
libsystemd-shared-237.so  systemd-fsck
network                  systemd-fsckd
resolv.conf              systemd-growfs
set-cpufreq              systemd-hibernate-resume
system                   systemd-hostnamed
system-generators        systemd-initctl
system-preset            systemd-journald
system-shutdown          systemd-locale
system-sleep             systemd-logind
systemd                  systemd-makefs
systemd-ac-power         systemd-modules-load
systemd-backlight        systemd-networkd
systemd-binfmt           systemd-networkd-wait-online
systemd-cgroups-agent    systemd-quotacheck
systemd-cryptsetup       systemd-random-seed
systemd-dissect          systemd-repart
root@server:/lib/systemd# clear
```

```
root@server: /lib/systemd
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:/lib/systemd#
```




셸

셸(shell)이란?

- 앞서 리눅스 배포판의 구성에서 보았듯이 셸은 커널과 응용 프로그램 사이에 있는 레이어
- 셸은 명령어 해석기로 명령과 프로그램을 실행할 때 사용하는 인터페이스
- 사용자가 입력한 명령을 해석하여 커널에 전달하거나 커널의 처리 결과를 사용자에게 전달하는 역할





셸



셸의 종류

- 유닉스 운영체제에서 사용하는 셸의 종류는 bourne shell(본셸), C shell(씨셸), Korn shell(콘셸), t shell(티셸), bash(bourne-again) shell(배쉬셸), ash shell(애쉬셸) 등이 있음
- 리눅스는 이중에서 bash shell(배쉬셸)을 기본적으로 사용함
- 셸의 종류에 따라 프롬프트의 모양이 다름.
 - ✓ 배쉬셸은 \$
 - ✓ C 셸은 %
 - ✓ T셸은 >

```
root@server:/etc
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)
root@server:/etc/systemd#
```

- 하지만 모든 셸에서 Super user는 프롬프트 모양이 #



Bash 셸의 특징

- 에일리어스(alias, 명령 단축) 기능
- 히스토리 기능(↑ 또는 ↓)
- 연산 기능
- Job Control 기능
- 자동 이름 완성 기능(Tab)
- 프롬프트 제어 기능
- 명령 편집 기능



셸



셸 명령문의 형식

- (프롬프트) 명령 [옵션...] [인자...]
- 명령문의 예

```
# ls -l
# rm -rf /mydir
# find . / -name "*.conf"
```



환경 변수

- 윈도우 OS의 시스템등록정보의 환경변수와 같은 개념
- 환경 변수 사용 목적
 - ✓ PATH와 같은 경로를 미리 지정하여 실행파일등을 쉽게 실행
 - ✓ 반복적인 일을 할 때 매번 입력하지않고 환경 변수에 미리 지정하면 편리
- 셸에서는 여러 가지 환경 변수 값을 불러올 수 있음
- 이미 설정된 환경 변수는 `echo $환경변수` 명령으로 확인
- 호스트 이름을 출력하려면 `echo $HOSTNAME` 명령을 실행
- `export 환경변수=값` 명령을 실행하면 환경 변수 값을 변경(확인할 때는 `printenv` 명령 실행)



셸



환경 변수

- 다음 표는 주요 환경 변수들
- 이들 중 PATH는 꼭 기억해두자.

파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

```
root@server:~# echo $HOME
/root
root@server:~# echo $HOSTNAME
server
root@server:~# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
root@server:~#
```

표 11-1 배시셸의 주요 환경 변수

환경 변수	설명	환경 변수	설명
HOME	현재 사용자의 홈 디렉터리	PATH	실행 파일을 찾는 디렉터리 경로
LANG	기본 자원 언어	PWD	사용자의 현재 작업 디렉터리
TERM	로그인 터미널 유형	SHELL	로그인해서 사용하는 셸
USER	현재 사용자의 이름	DISPLAY	X 디스플레이 이름
COLUMNS	현재 터미널의 칼럼 수	LINES	현재 터미널의 라인 수
PS1	1차 명령 프롬프트 변수	PS2	2차 명령 프롬프트(대개 >)
BASH	배시셸의 경로	BASH_VERSION	배시셸의 버전
HISTFILE	히스토리 파일의 경로	HISTSIZE	히스토리 파일에 저장되는 명령어 개수
HOSTNAME	호스트 이름	USERNAME	현재 사용자의 이름
LOGNAME	로그인 이름	LS_COLORS	ls 명령의 확장자 색상 옵션
MAIL	메일을 보관하는 경로	OSTYPE	운영체제 유형

```
root@server:~# export TEST=test
root@server:~# echo $TEST
test
root@server:~#
```



셸



.bashrc

- 사용자의 홈 디렉토리에는 항상 자신의 .bashrc파일이 있음
- .bashrc 파일은 터미널이 실행되면서 자동으로 잡히는 환경 변수를 선언할 수 파일
- 그래서 매번 필요한 환경 변수는 .bashrc파일에 `export NAME=value` 라는 명령으로 저장해 둘수 있음.

```
root@server:~# ls -alF
합계 84
drwx----- 17 root root 4096 2월 18 16:56 ./
drwxr-xr-x 24 root root 4096 2월 17 14:37 ../
-rw----- 1 root root 942 2월 18 15:41 .ICEauthority
-rw----- 1 root root 203 2월 18 17:16 .bash_history
-rw-r--r-- 1 root root 3106 4월 9 2018 .bashrc
drwx----- 11 root root 4096 2월 18 16:07 .cache/
drwx----- 12 root root 4096 2월 18 17:02 .config/
drwx----- 3 root root 4096 2월 18 16:56 .gnupg/
drwxr-xr-x 3 root root 4096 2월 17 17:35 .local/
drwx----- 4 root root 4096 2월 18 16:07 .mozilla/
```



📁 .bashrc의 사용 예

- PATH 변수에 자신의 홈 디렉토리를 추가하고 ARCH 라는 변수에 arm이라는 값을 추가한 예
- 먼저 gedit로 .bashrc파일을 열기
- .bashrc 파일 끝부분에 그림과 같이 PATH와 ARCH 변수 추가
- 저장하고 나와서 새 터미널을 열어 echo명령으로 추가된 것을 확인

```
root@server:~# gedit .bashrc
```

```
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
#    . /etc/bash_completion
#fi

export PATH=$PATH:/root
export ARCH=arm
```

sh ▼ 탭 너

```
root@server:~# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/root
root@server:~# echo $ARCH
arm
root@server:~#
```




셸

셸 스크립트

- 셸은 명령어 해석 기능 외에 프로그래밍도 가능함.
- 셸 스크립트를 통해 프로그램을 만들 수 있음.
- MS-DOS의 배치(batch) 파일과 유사하나 훨씬 막강한 기능
- 셸 스크립트를 잘 사용하면 새로 프로그램을 만들 필요 없이 간편하게 기능 구현 가능한 경우도 있음.
- 리눅스와 유닉스에서 여러 방식으로 묶여진 명령어 조합이 반복적으로 사용된다면 이를 셸 스크립트를 이용하여 하나의 명령어로 만들어 사용할 수 있음
- 스크립트의 첫줄은 항상 `#!`로 시작
 - ✓ `#!/bin/sh`, `#!/bin/csh`, `#!/bin/bash`, `#!/bin/ksh`, `#!/bin/tcsh`와 같이 셸의 종류에 따라 다름.




셸

셸 스크립트 작성 실습

- 셸 스크립트도 일반적인 프로그래밍 언어와 비슷하게 변수, 반복문, 제어문 등을 사용 가능
- 별도로 컴파일하지 않고 텍스트 파일 형태로 셸에서 바로 실행

- ① Gedit로 test.sh 파일을 열기
- ② 그림과 같이 USER 변수와 HOME변수를 출력
- ③ 셸에서 sh 스크립트 파일을 통해 스크립트 파일 실행

```
root@server:~# gedit test.sh
```

```
열기(O) ▾   
#!/bin/sh  
echo "user name: " $USER  
echo "home directory: " $HOME  
exit 0
```

```
root@server:~# ls  
test.sh  공개  다운로드  문서  바탕화면  
root@server:~# sh test.sh  
user name: root  
home directory: /root  
root@server:~#
```

Q & A

Thank You