



# *LINUX*

YBIGTA 2021-1 교육 세션

17기 이다현

21.01.12

# *CONTENTS*

1. CLI vs GUI
2. LINUX
3. BASIC COMMANDS
4. Vim editor
5. Exercise ^^)

The background features several thin, purple lines that intersect to form a complex, abstract geometric pattern. These lines create a series of overlapping triangles and quadrilaterals across the entire frame.

# *1. CLI vs GUI*

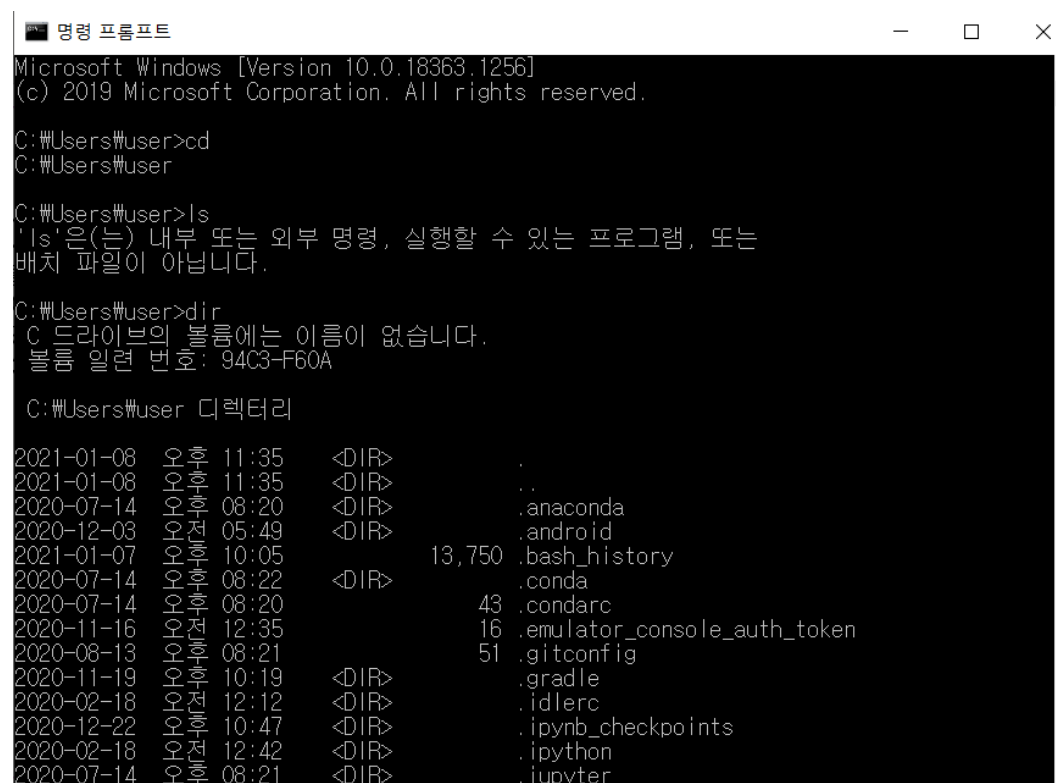
# CLI (Command Line Interface): 명령줄 인터페이스

명령어를 입력하여 컴퓨터를 제어 (Low-level programming).

효율성이 좋다.

서버 환경 등을 구축할 때 파이프라인화 시키는 과정에서 필수적이다.

DOS, CMD, Unix shell (bash)...



```
명령 프롬프트
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\User>cd
C:\Users\User

C:\Users\User>ls
'ls'은(는) 내부 또는 외부 명령, 실행할 수 있는 프로그램, 또는
배치 파일이 아닙니다.

C:\Users\User>dir
C 드라이브의 볼륨에는 이름이 없습니다.
볼륨 일련 번호: 94C3-F60A

C:\Users\User 디렉터리

2021-01-08 오후 11:35 <DIR> .
2021-01-08 오후 11:35 <DIR> ..
2020-07-14 오후 08:20 <DIR> .anaconda
2020-12-03 오전 05:49 <DIR> .android
2021-01-07 오후 10:05 13,750 .bash_history
2020-07-14 오후 08:22 <DIR> .conda
2020-07-14 오후 08:20 43 .condarc
2020-11-16 오전 12:35 16 .emulator_console_auth_token
2020-08-13 오후 08:21 51 .gitconfig
2020-11-19 오후 10:19 <DIR> .gradle
2020-02-18 오전 12:12 <DIR> .idlerc
2020-12-22 오후 10:47 <DIR> .ipynb_checkpoints
2020-02-18 오전 12:42 <DIR> .ipython
2020-07-14 오후 08:21 <DIR> .jupyter
```

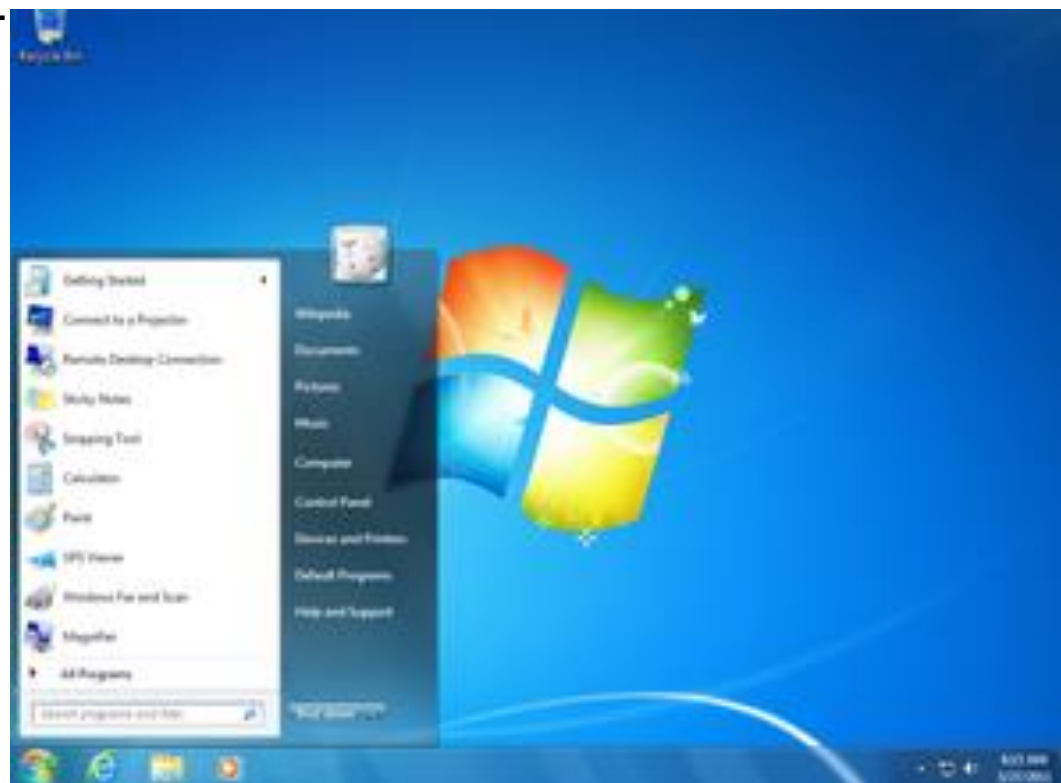
# GUI (Graphical User Interface): 그래픽 인터페이스

Graphic Tools(마우스, 모니터...)를 이용하여 제어

=이미 구축된 그래픽 환경에서 클릭...으로 사용

CLI보다 직관적이지만 한계가 있다.

Windows, MacOS, Ubuntu GUI Desktop...





The background features several thin, purple lines that intersect to form a complex geometric pattern. These lines create a series of overlapping triangles and polygons across the entire frame, with some lines extending towards the edges.

## *2. LINUX*

# LINUX: Linux Is Not UniX

-Unix 기반으로 만들어진 GNU+Linux 커널 = Linux

-오픈소스 운영체제

-데스크톱/랩톱 용도 뿐만 아니라 웹 서버, 클라우드, 안드로이드 및 포터블 게이밍 콘솔 등의 모바일 기기, 각종 임베디드 기기 (가전용, 상업용, 산업용, 차량용 기기)등을 구동하는 운영 체제

-CLI로 작업

-Linux 커널 사용은 shell로! (ex. bash)

-Linux의 종류: Debian 계열(Debian, Ubuntu…), Red Hat 계열(CentOS, Fedora…)

apt-get, apt, dpkg

yum, rpm

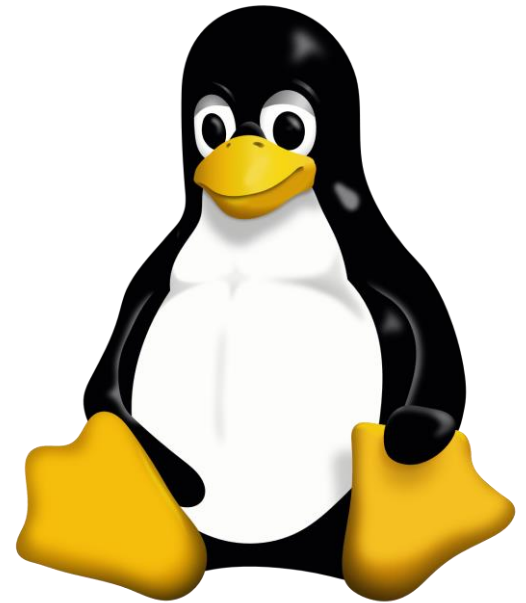


## 리눅스의 장점

- I. 리눅스는 유닉스와 완벽하게 호환가능
- II. 리눅스는 공개 운영체제
- III. 리눅스는 PC용 OS보다 안정적
- IV. 리눅스는 무료
- V. 리눅스는 하드웨어의 기능을 알차게 사용
- VI. 리눅스는 강력한 네트워크를 구축
- VII. 리눅스는 강력한 보안 기능
- VIII. 리눅스는 인터넷의 모든 기능을 지원
- IX. 리눅스는 개발 환경이 풍부

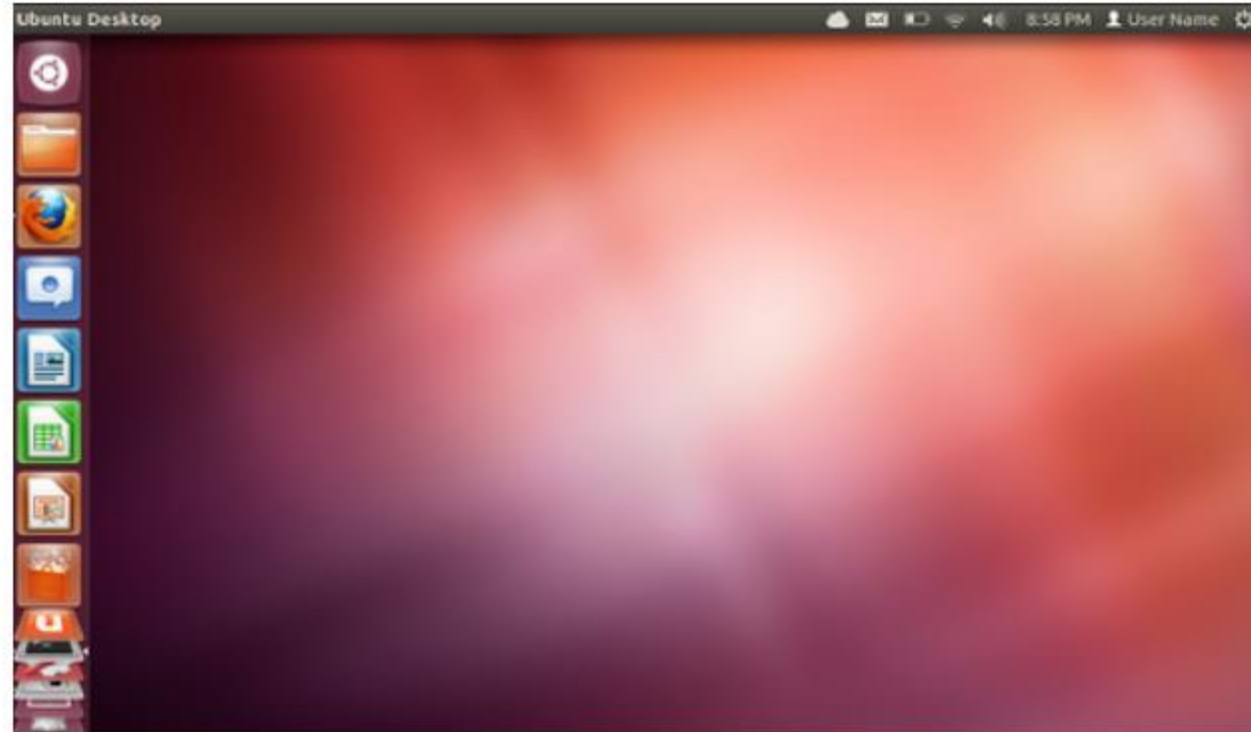
## 리눅스의 단점

- I. 공개운영체제이기 때문에 문제점 발생시 보상받을 수 없음
- II. 한글입출력을 하는데 어려움이 있음
- III. 기술지원의 부족
- IV. 특정 하드웨어에 대한 자원이 부족
- V. 사용자의 숙련된 기술이 요구





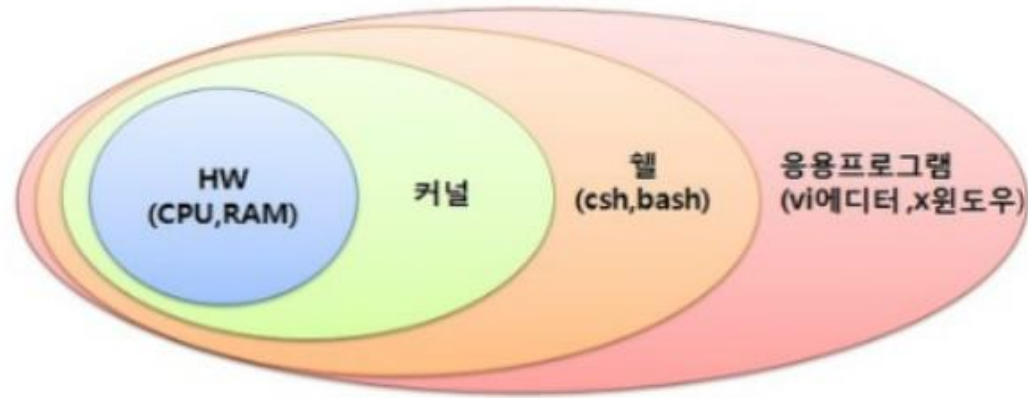
## Ubuntu



<https://www.ubuntu.com>

데스크탑에서 리눅스를 쉽게 사용할 수 있도록 만들어진 리눅스 배포판입니다.

우분투는 데비안/GNU/리눅스에 기초한 리눅스로 고유의 데스크탑 환경인 유니티를 사용합니다. 영국에 기반을 둔 캐노니컬이라는 회사의 지원을 받으며 6개월에 한번씩 새판이 나오고, 사용자 편의성에 많은 초점을 맞추고 있습니다. 2012년 기준 PC에서 가장 인기있는 리눅스 배포판입니다.



### 1) 커널 - Kernel

프로그램의 실행과정에서 가장 핵심적인 연산이 이루어지는 부분으로 코어, 리눅스라고 부른다. 하드웨어를 직접 제어하고, 프로세스 관리, 메모리 관리, 파일시스템 관리 등을 수행하는 운영체제의 핵심으로, 사용자가 실행시키는 응용프로그램 (Application)과 하드웨어 사이의 관리자 역할을 수행하며 셸과 연관되어 셸에서 명령하는 작업을 수행하고 그 결과를 셸로 보내는 일을 한다.

### 2) 셸 - Shell

셸은 운영체제에서 사용자가 입력하는 명령을 대신 해석해 커널에게 전달, 실행해주는 프로그램이다. 셸은 사용자가 입력한 문자열을 해석하고 해당하는 명령어를 찾아서 커널에 작업을 요청하게 됩니다. 그리고 커널에서 작업을 수행한 결과를 다시 셸로 보내면 셸은 그 결과를 유저에게 알려주는 형식입니다.

셸의 종류에는 MacOS의 bash, C Programming 스타일의 csh 등이 있습니다.

### 3) 응용프로그램 - Application

리눅스는 각종 프로그래밍 개발도구, 문서 편집도구, 네트워크 관련 도구 등 매우 다양한 응용프로그램을 제공한다. (Editor, 통신프로그램, X윈도우, 웹서버, DB서버 등)

**리눅스 커널**: 리눅스 공동체에 의해서 완전히 새로이 개발된 독창적인 소프트웨어

리눅스 시스템: 리눅스 커널 + 다른 소프트웨어(자체적, 협동 모두) 포함

리눅스 배포본: 간편한 설치를 위해 표준화된+미리 컴파일된 패키지의 집합

커널: 모든 시스템 자원들을 다루고 하드웨어와 직접 소통하는 core, privileged executive(관리자)

커널: 가상 메모리와 프로세스 등을 포함하는 운영체제의 핵심을 다루는 부분

시스템 라이브러리: 응용 프로그램이 커널과 소통하는 함수들을 정의, 커널 코드의 특권이 필요하지 않은 다른 기능들을 구현한 함수들도 정의.

(ex. **C library**)

시스템 유틸리티: 개별적이고 특수한 관리 기능을 수행하는 프로그램

(초기화, 시스템 구성, **daemon**(시스템 내에서 계속 돌아가면서  
네트워크 연결 요청에 대한 응답, 터미널로부터의 로그인 요청 처리,  
로그 파일 업데이트...등 수행)

모든 커널 코드는 privileged mode로 수행 (컴퓨터의 모든 자원들에 대한 일체의 접근 권한을 가짐)

=커널 모드

⇔ 사용자 모드 (시스템 자원 중 통제된 일부분만 접근 가능)

리눅스 시스템이 제공하는 ‘사용자 모드’의 프로그램:

시스템 utility: 시스템을 초기화하고 관리하는 데 필요한 모든 프로그램  
(ex. 네트워크 인터페이스의 설정, 시스템에 사용자를 추가하거나 제거...)  
사용자 utility: 시스템의 기본 작동을 위해 필요하지만 상향된 특권을 필요로 하지는 않음  
(ex. 파일 복사, 디렉토리 생성, 텍스트 파일 편집...파일 관리 유틸리티...)  
(ex. **shell**: UNIX 시스템의 표준 명령-라인 인터페이스.  
bash, bourne-Again shell...)

### Access Control

리눅스는 사용자에게 따라 읽기,쓰기,실행 권한을 protection mask형태로 부여한다.  
(어떤 사용자는 읽기만 가능, 어떤 사용자는 모두 가능...)

**root**(관리자)계정은 접근 권한 검사를 거치지 않고 시스템 내의 모든 객체에 접근이 가능!

The background features several thin, purple lines that intersect to form a complex geometric pattern of triangles and polygons. The lines are light purple and extend across the entire frame.

# *3. BASIC COMMANDS*

# 명령어 사용 예시(bash)

command [options] [Arguments] 의 형태

동사

형용사/부사

목적어

대응 이런 느낌 →

ls : 현재 디렉토리 파일 및 폴더 출력

cd : 경로 이동

mkdir : 폴더 생성

rmdir : 폴더 삭제

rm : 파일 삭제

echo : 문자열 출력

```
MINGW64/c/Users/User/Desktop/Minseok/hello/folder1
hacha_es01@hacha_es-N01 MINGW64 ~/Desktop/Minseok/hello (master)
$ ls
folder1/

hacha_es01@hacha_es-N01 MINGW64 ~/Desktop/Minseok/hello (master)
$ ls -a
./ ../ .t/ folder1/

hacha_es01@hacha_es-N01 MINGW64 ~/Desktop/Minseok/hello (master)
$ cd folder1

hacha_es01@hacha_es-N01 MINGW64 ~/Desktop/Minseok/hello/folder1 (master)
$ echo hahaha
hahaha

hacha_es01@hacha_es-N01 MINGW64 ~/Desktop/Minseok/hello/folder1 (master)
$ echo hahaha > hahaha.txt

hacha_es01@hacha_es-N01 MINGW64 ~/Desktop/Minseok/hello/folder1 (master)
$ ls
hahaha.txt

hacha_es01@hacha_es-N01 MINGW64 ~/Desktop/Minseok/hello/folder1 (master)
$
```

option

argument



# 명령어 사용 예시(bash)

command [options] [Arguments] 의 형태

동사

형용사/부사

목적어

대응이런느낌→

ls : 현재 디렉토리 파일 및 폴더 출력

cd : 경로 이동

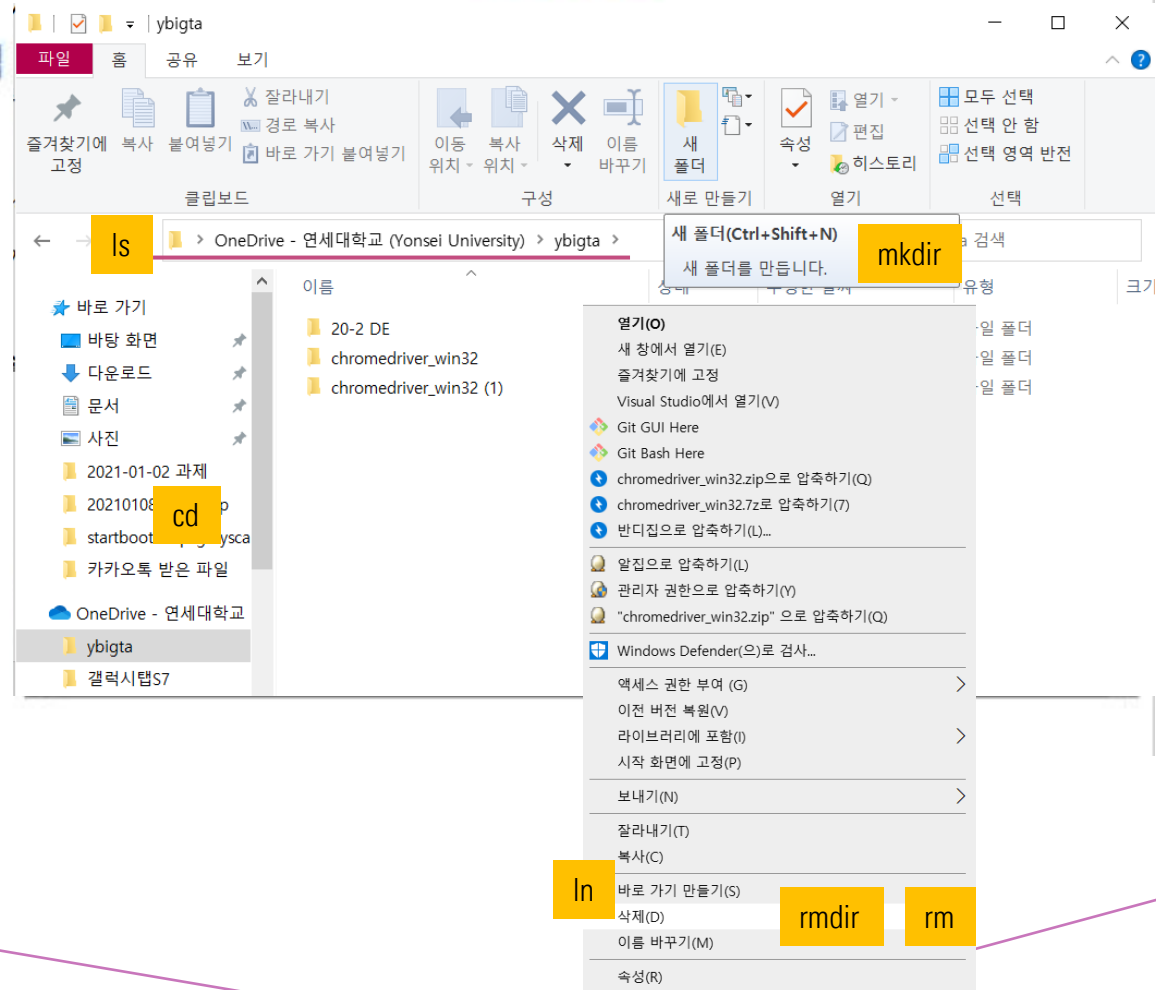
mkdir : 폴더 생성

rmdir : 폴더 삭제

rm : 파일 삭제

echo : 문자열 출력

ln : 바로가기 생성



# 명령어 사용 예시(bash)

command [options] [Arguments] 의 형태

동사

형용사/부사

목적어

대응 이런 느낌 →

ls : 현재 디렉토리 파일 및 폴더 출력

cd

<https://github.com/LeCoupa/awesome-cheatsheets/blob/master/languages/bash.sh>  
(명령어를 굳이 외울 필요는 없습니다! 너무 많아요)

mk

<https://dora-guide.com/linux-commands/>

<https://linuxjourney.com/> (강추!)

rm

<https://edu.goorm.io/lecture/12984/생활코딩-리눅스-linux>

rm : 파일 삭제

echo : 문자열 출력

```
MINGW64/c/Users/User/Desktop/Minseok/hello/folder1
hacha_es01@hacha_es-N01 MINGW64 ~/Desktop/Minseok/hello/folder1 (master)
$ echo hahaha > hahaha.txt
hacha_es01@hacha_es-N01 MINGW64 ~/Desktop/Minseok/hello/folder1 (master)
$ ls
hahaha.txt
hacha_es01@hacha_es-N01 MINGW64 ~/Desktop/Minseok/hello/folder1 (master)
$
```

앞으로 자주 마주치실 에러와 관련된 명령어들을 조금만 더 소개해볼게요:

apt, apt-get? = 뭐 설치할 때 자주 쓸 명령어입니다. 뭔가 설치하기 전에 apt 업데이트부터 해줍니다.

#### **Install a package from a repository**

<https://linuxjourney.com/lesson/package-management-systems>

```
Debian: $ apt install package_name
```

```
RPM: $ yum install package_name
```

#### **Remove a package**

```
Debian: $ apt remove package_name
```

```
RPM: $ yum erase package_name
```

#### **Updating packages for a repository**

It's always best practice to update your package repositories so they are up to date before you install and update a package.

```
Debian: apt update; apt upgrade
```

```
RPM: yum update
```

앞으로 자주 마주치실 에러와 관련된 명령어들을 조금만 더 소개해볼게요:

.tar? .gz? =zip파일이라고 보시면 돼요.

<https://linuxjourney.com/lesson/compressed-archives-tar>

### Compressing files with gzip

gzip is program used to compress files in Linux, they end in a .gz extension.

To compress a file down:

```
$ gzip mycoolfiler
```

To decompress the file:

```
$ gunzip mycoolfiler.gz
```

앞으로 자주 마주치실 에러와 관련된 명령어들을 조금만 더 소개해볼게요:

.tar? .gz? =zip파일이라고 보시면 돼요.

<https://linuxjourney.com/lesson/compressed-archives-tar>



### Creating archives with tar

Unfortunately, gzip can't add multiple files into one archive for us. Luckily we have the tar program which does. When you create an archive using tar, it will have a .tar extension.

```
$ tar cvf mytarfile.tar mycoolfiler mycoolfiler2
```

- c - create
- v - tell the program to be verbose and let us see what it's doing
- f - the filename of the tar file has to come after this option, if you are creating a tar file you'll have to come up with a name

### Unpacking archives with tar

To extract the contents of a tar file, use:

```
$ tar xvf mytarfile.tar
```

- x - extract
- v - tell the program to be verbose and let us see what it's doing
- f - the file you want to extract

앞으로 자주 마주치실 에러와 관련된 명령어들을 조금만 더 소개해볼게요:

.tar? .gz? =zip파일이라고 보시면 돼요.

<https://linuxjourney.com/lesson/compressed-archives-tar>

### Compressing/uncompressing archives with tar and gzip

Many times you'll see a tar file that has been compressed such as: mycompressedarchive.tar.gz, all you need to do is work outside in, so first remove the compression with gunzip and then you can unpack the tar file. Or you can alternatively use the **z** option with tar, which just tells it to use the gzip or gunzip utility.

Create a compressed tar file:

```
$ tar czf myfile.tar.gz
```

Uncompress and unpack:

```
$ tar xzf file.tar
```

If you need help remember this: **eXtract** all **Zee Files**!



앞으로 자주 마주치실 에러와 관련된 명령어들을 조금만 더 소개해볼게요:

`./xxxx/xxxx.sh`? `./`는 현재 디렉토리를 의미한다. CLI창에서 `.sh`파일을 실행시킬 수 있다.

<https://linuxjourney.com/lesson/compile-source-code>

```
$ ./configure
```

The `./` allows you to execute a script in the current directory.

```
$ make
```

```
$./xxxx.sh
```

```
$sh xxxx.sh
```

앞으로 자주 마주치실 에러와 관련된 명령어들을 조금만 더 소개해볼게요:  
root 권한으로 실행을 시켜야 하는 경우가 있어요!

<https://linuxjourney.com/lesson/root-user>

### 퀴즈

What command do you use to run as root?

정답 확인하기

### 퀴즈

What command do you use to run as root?

✓ 정답!

sudo

정답 확인하기

앞으로 자주 마주치실 명령어들을 조금만 더 소개해볼게요:

root 권한으로 실행을 시켜야 하는 경우가 있어요!

<https://linuxjourney.com/lesson/root-user>

## 2. root

We've looked at one way to get superuser access using the sudo command. You can also run commands as the superuser with the su command. This command will "substitute users" and open a root shell if no username is specified. You can use this command to substitute to any user as long as you know the password.

```
$ su
```

There are some downsides to using this method: it's much easier to make a critical mistake running everything in root, you won't have records of the commands you use to change system configurations, etc. Basically, if you need to run commands as the superuser, just stick to sudo.

Now that you know what commands to run as the superuser, the question is how do you know who has access to do that? The system doesn't let every single Joe Schmoe run commands as the superuser, so how does it know? There is a file called the `/etc/sudoers` file, this file lists users who can run sudo. You can edit this file with the **visudo** command.

## Adding Users

You can use the `adduser` or the `useradd` command. The `adduser` command contains more helpful features such as making a home directory and more. There are configuration files for adding new users that can be customized depending on what you want to allocate to a default user.

```
$ sudo useradd bob
```

You'll see that the above command creates an entry in `/etc/passwd` for bob, sets up default groups and adds an entry to the `/etc/shadow` file.

## Removing Users

To remove a user, you can use the `userdel` command.

```
$ sudo userdel bob
```

This basically does its best to undo the file changes by `useradd`.

## Changing Passwords

```
$ passwd bob
```

This will allow you to change the password of yourself or another user (if you are root).

앞으로 자주 마주치실 명령어들을 조금만 더 소개해볼게요:

ERROR: operation not permitted

<https://linuxjourney.com/lesson/modifying-permissions>

#### **Adding permission bit on a file**

```
$ chmod u+x myfile
```

The above command reads like this: change permission on myfile by adding executable permission bit on the user set. So now the user has executable permission on this file!

#### **Removing permission bit on a file**

```
$ chmod u-x myfile
```

#### **Adding multiple permission bits on a file**

```
$ chmod ug+w
```

The numerical representations are seen below:

- 4: read permission
- 2: write permission
- 1: execute permission

<https://linuxjourney.com/lesson/modifying-permissions>

Let's look at an example:

```
$ chmod 755 myfile
```

Can you guess what permissions we are giving this file? Let's break this down, so now 755 covers the permissions for all sets. The first number (7) represents user permissions, the second number (5) represents group permissions and the last 5 represents other permissions.

Wait a minute, 7 and 5 weren't listed above, where are we getting these numbers? Remember we are combining all the permissions into one number now, so you'll have to get some math involved.

$7 = 4 + 2 + 1$ , so 7 is the user permissions and it has read, write and execute permissions

$5 = 4 + 1$ , the group has read and execute permissions

$5 = 4 + 1$ , and all other users have read and execute permissions



The background features several thin, purple lines that intersect to form a complex geometric pattern. These lines create a series of overlapping triangles and polygons across the entire frame, with some lines extending towards the edges.

# *3. VIM EDITOR*

사실, 간단하게 텍스트문서를 만드는 건 echo, cat 명령어로도 가능합니다.

---

### Text 3-1

#### *echo* 명령과 텍스트 문서

echo는 나열된 인자를 화면에 출력하는 명령어이다.

```
[학번@linux-pplab ~]$ echo Hello, World  
[학번@linux-pplab ~]$ echo "Hello, World"
```

>, 혹은 >> 를 이용하여 화면에 출력된 내용을 파일을 생성하거나  
파일에 내용을 추가할 수 있다.

```
[학번@linux-pplab ~]$ echo Hello, World > hello.txt  
[학번@linux-pplab ~]$ echo Hello, World > hello.txt  
[학번@linux-pplab ~]$ echo Good, Bye > bye.txt  
[학번@linux-pplab ~]$ echo Good, Bye >> bye.txt
```

cat 명령을 이용하여 생성한 파일의 내용을 확인 해본다.

```
[학번@linux-pplab ~]$ cat hello.txt  
[학번@linux-pplab ~]$ cat bye.txt
```

> 는 파일을 새로 생성한 후, 화면에 출력된 내용을 쓴다.  
>> 는 파일의 내용 끝에, 화면에 출력된 내용을 쓴다.

---

***hello.txt*와 *bye.txt*의 차이점이 무엇인지 간단히 설명하시오.**

hello.txt  
Hello, World

bye.txt  
Good, Bye  
Good, Bye

사실, 간단하게 텍스트문서를 만드는 건 echo, cat 명령어로도 가능합니다.

---

### ***Text 3-2***

#### *cat 명령과 텍스트 문서*

*Text 3-1*의 사용에서 알 수 있듯이 cat 명령은 파일의 내용을 화면에 출력하는 명령이다.

cat은 이외에도 키보드로 입력받은 내용을 화면에 출력하는 기능도 가지고 있다.

```
[학번@linux-pplab ~]$ cat << END
> Hello!
> My job is student.
> Good bye.
> END
```

cat으로 입력받은 내용을 파일에 기록한다.

```
[학번@linux-pplab ~]$ cat << END > file.txt
> Hello!
> My job is student.
> Good bye.
> END
[학번@linux-pplab ~]$ cat file.txt
```

END의 역할은 무엇일까요?

사실, 간단하게 텍스트문서를 만드는 건 echo, cat 명령어로도 가능합니다.

### cat을 이용해 여러 줄 입력하기

- `cat` 다음에 `<<` 를 쓰고 원하는 표시자를 씁니다. 아래의 경우에는 `EOF` 를 사용하였는데 다른 단어를 사용해도 됩니다. 단, 처음에 사용한 단어가 끝에도 같아야 합니다. 여기에 `EOF` 를 사용했기 때문에 마지막에 입력을 끝내려면 똑같이 `EOF` 를 입력하면됩니다.
- 이러한 부분을 `Here-Document` 라고 하는데 자세한 내용은 [http://pubs.opengroup.org/onlinepubs/9699919799/utilities/V3\\_chap02.html#tag\\_18\\_07\\_04](http://pubs.opengroup.org/onlinepubs/9699919799/utilities/V3_chap02.html#tag_18_07_04)에 나와있습니다.

```
$ cat << EOF
> This
> is
> multiline test
> EOF
This
is
multiline test
$
```

# Vim(vi) Editor

리눅스 환경에서 자주 사용할 에디터  
문서편집 효율이 향상되고, unix 계열에서 기본적으로 제공하는 에디터이기  
때문에 추가 설치가 필요 없다.

처음에는 낯설겠지만, 앞으로 익숙해져 보아요... ( ' ; ω ; ' )



# Vim(vi) Editor

<https://linuxjourney.com/lesson/vim-text-editor>

- 명령모드: `esc`가 눌러진 상태
  - `x` : 커서가 위치한 곳에서 한 문자 삭제
  - `(n)dd` : 한 라인 삭제(숫자 `n`을 붙이면 커서 기준으로 아래 `n`줄 삭제)
  - `(n)yy` : 한 라인 복사(숫자 `n`을 붙이면 커서 기준으로 아래 `n`줄 복사)
  - `p` : 커서 오른쪽에 붙여넣기
- 입력모드: 실제로 글을 쓸 수 있는 모드
  - `i`, `a`, `o`, `insert`
- 편집모드: `esc`를 누른 상태에서 파일 자체에 대해 명령
  - `:wq` - 작성 파일을 저장하고 종료
  - `:q!` - 작성 파일을 저장하지 않고 무조건 종료
  - `:set nu` - 줄번호 표시
  - `:set nonu` - 줄번호 표시 해제
- 여기 보이시는 `:`를 꼭 앞에 붙여주셔야합니다. `:`를 누르면 커서가 밑으로 이동합니다. 그럼 거기에다가 입력하고 엔터를 누르시면 입력이 됩니다.



The background features several thin, purple lines that intersect to form a complex geometric pattern of triangles and polygons. The lines are of varying lengths and orientations, creating a dynamic and abstract visual field.

# *4. EXERCISE*

가상머신을 설치하라고 하기엔.. 너무 가혹한 것 같아서..! ｡: (；∩；) ｡:

<https://cocalc.com/>

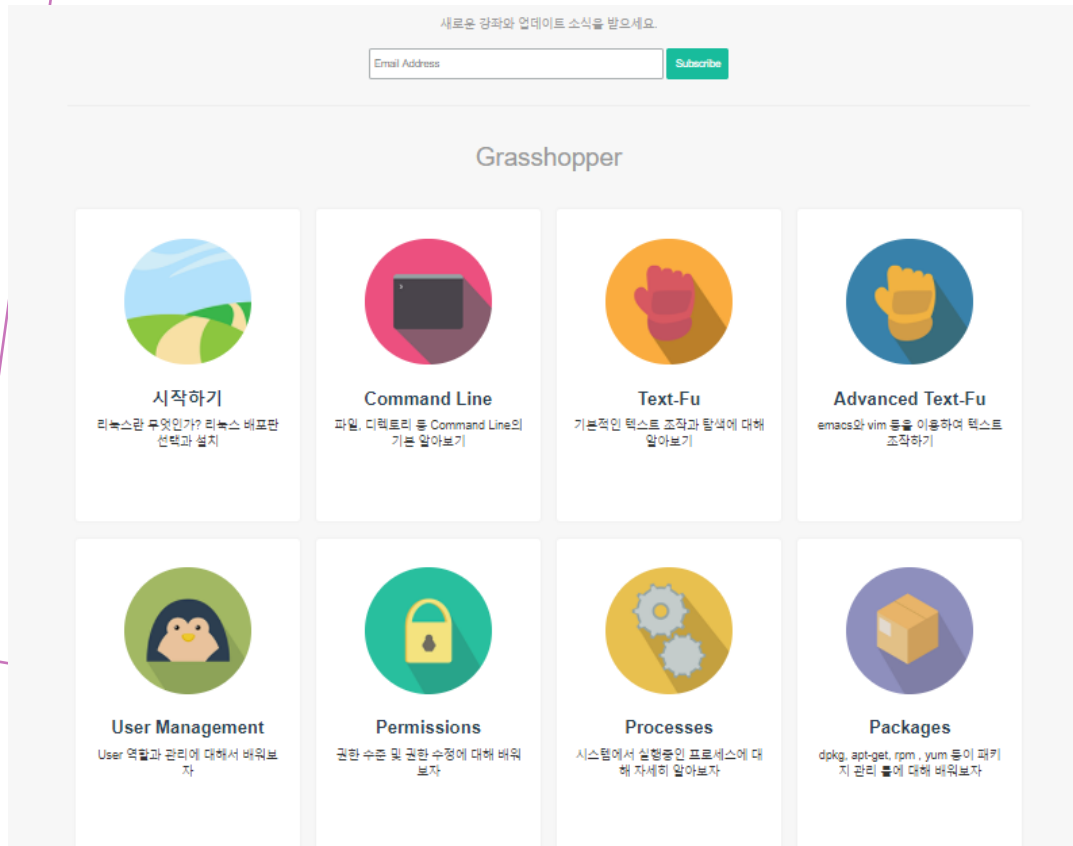
[https://www.tutorialspoint.com/unix\\_terminal\\_online.php](https://www.tutorialspoint.com/unix_terminal_online.php)

온라인 사이트에서 가볍게! 리눅스 명령어 맛보기를 해보아요 ㅎㅎ



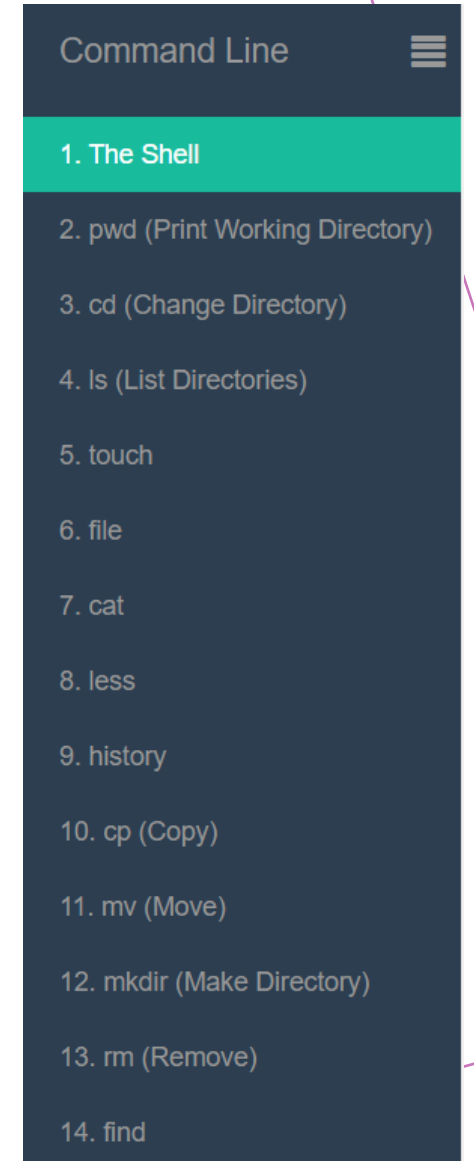
숙제:

<https://linuxjourney.com/> 참고해서 Command Line, Text Fu, Advanced Text-Fu, User Management, Permissions 챕터에 소개된 명령어들을 각 챕터당 2개 이상씩 실행해보고 화면 캡처해서 총 5장 제출하기!



```
~$ cd ybigta
~/ybigta$ pwd
/home/user/ybigta
~/ybigta$ cd home
bash: cd: home: No such file or directory
~/ybigta$ cd $HOME
~$ ls
'Welcome to CoCalc.term' ybigta
```

이런 거 다섯장! 와! 너무 쉽다!



출처:

<https://linuxjourney.com/>

<https://hongci.tistory.com/5>

<https://drive.google.com/drive/u/0/folders/1xUoR7cxdWrswgxq9EIiKR6BdCiMzp5iG>

<https://drive.google.com/drive/u/0/folders/1zsZoWP4GVCCAsyo8lrDmr84zvbECIps8>

<https://dora-guide.com/linux-commands/>

<https://cocalc.com/>

[https://www.tutorialspoint.com/unix\\_terminal\\_online.php](https://www.tutorialspoint.com/unix_terminal_online.php)

고양이 그림

<https://1boon.kakao.com/share/catpc>