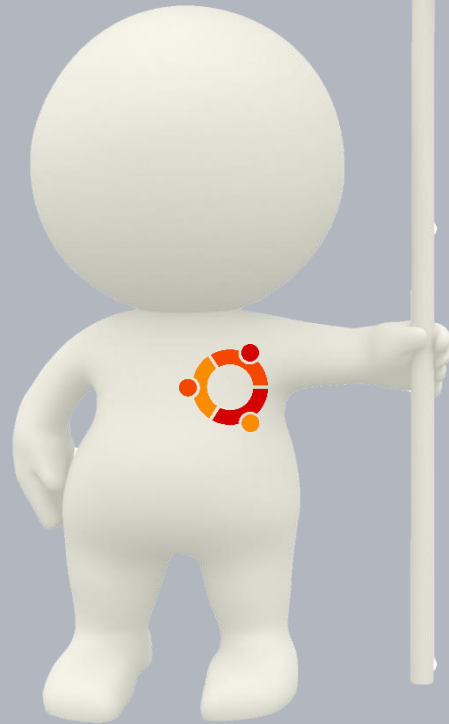


# 리눅스(Linux) 기본 명령어



# 명령어 Help 기능



# Ubuntu Server 도움말(Help)

## 명령어 --help

```
yummy@ubuntu:~$ ls --help
```

Usage: ls [OPTION]... [FILE]...

List information about the FILES (the current directory by default).

Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.

- a, --all do not ignore entries starting with .
- A, --almost-all do not list implied . and ..
- author with -l, print the author of each file
- b, --escape print C-style escapes for nongraphic characters
- block-size=SIZE with -l, scale sizes by SIZE when printing them;  
                        e.g., '--block-size=M'; see SIZE format below
- B, --ignore-backups do not list implied entries ending with ~
- c with -lt: sort by, and show, ctime (time of last  
modification of file status information);  
with -l: show ctime and sort by name;  
otherwise: sort by ctime, newest first

# Ubuntu Server 도움말(man)

## man 명령어

yummy@ubuntu:~\$ man ls

LS(1) User Commands

LS(1)

### NAME

ls - list directory contents

### SYNOPSIS

ls [OPTION]... [FILE]...

### DESCRIPTION

List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.

-a, --all

do not ignore entries starting with .

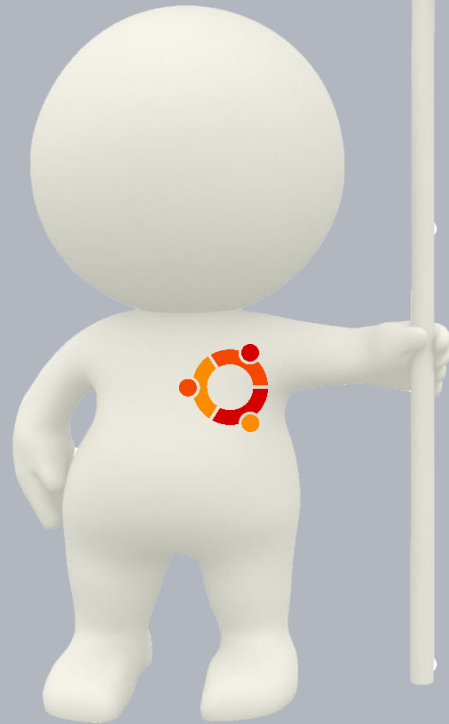
-A, --almost-all

do not list implied . and ..

--author

with -l, print the author of each file

# 기본 명령어



# 기본 명령어

---

## sudo 명령어

- 리눅스는 관리자인 root 사용자를 일반적으로 잘 사용하지 않고, 주로 sudo 명령을 통해 root의 권한으로 작업을 진행한다.

```
yummy@ubuntu:~$ sudo su - root
[sudo] password for yummy:
root@ubuntu:~#
root@ubuntu:~# passwd
New password:
Retype new password:
passwd: password updated successfully
root@ubuntu:~#
```

# 기본 명령어

## ls 명령어

- LiSt의약자로 Windows의 dir명령과같은 역할을 한다. 즉 해당 디렉터리(폴더)에 있는 파일 목록을 나열한다.

옵 션	설 명
<b>-a</b>	디렉터리의 목록(숨김 파일 포함)
<b>-l</b>	디렉터리의 목록을 자세히 보여줌(퍼미션, 포함된 파일수, 소유자, 그룹, 파일크기,수정일자,파일이름)
<b>-S</b>	파일 크기 순으로 정렬하여 출력한다.(대문자 S)
<b>-r</b>	리스트를 거꾸로 출력한다.(ls 명령어의 기본은 알파벳 순서다.)
<b>-R</b>	하위 디렉터리까지 출력한다.
<b>-h</b>	용량 크기를 K, M, G 단위를 사용하여 사람이 보기 좋게 표시한다.

- 리눅스는 숨김파일(hidden file)이라는 속성이 별도로 존재하지 않는다. 파일 이름이나 디렉터리의 제일 앞글자를 '.'으로 하면 자동으로 숨김파일이 된다.

# 기본 명령어

## cd 명령어

- Change Directory의 약자로 디렉터리를 이동하는 명령어이다.

사용 예	설 명
cd or cd ~	현재 사용자의 홈 디렉터리로 이동.
cd ~yummy	yummy 사용자의 홈 디렉터리로 이동
cd .	현재 위치하고 있는 디렉터리
cd..	바로 상위의 디렉터리로 이동. ' .. '은 현 디렉터리의 부모 디렉터를 의미 예) /etc/systemd 면 상위 /etc
cd /etc/systemd	/etc/systemd 디렉토리로 이동(절대 경로)
cd ../etc/systemd	상대 경로로 이동. 현재 디렉터리의 상위(' .. ')로 이동한 후 다시 /etc/systemd로 이동

- ' .' (현재 디렉토리)와 ' .. ' (현재 디렉터리의 상위 디렉터리)는 자주 사용되는데, 리눅스 초보자 입장에서는 헷갈스러울 수 있으니 잘 기억해 두어야 한다.



# 기본 명령어

---

## pwd 명령어

- Print Working Directory의 약자로 현재 디렉터리의 전체 경로를 화면에 보여준다..

```
yummy@ubuntu:~$ pwd  
/home/yummy  
yummy@ubuntu:~$
```

# 기본 명령어

## touch 명령어

- 빈파일을 생성할 수 있는 명령어
- 내용은 수정하지 않고 수정 시간만 변경할 수 있는 명령어

```
yummy@ubuntu:~$ touch a.txt
yummy@ubuntu:~$ ls -la*
-rw-rw-r-- 1 yummy yummy 0 Aug 16 12:53 a.txt
yummy@ubuntu:~$ touch a.txt
yummy@ubuntu:~$ ls -la*
-rw-rw-r-- 1 yummy yummy 0 Aug 16 12:54 a.txt
yummy@ubuntu:~$
```

- 비어있는 파일을 생성한 후 touch 명령어를 다시 사용하면 수정 시간이 변경된 것을 볼 수 있다..

# 기본 명령어

## rm 명령어

- ReMove의 약자로 파일이나 디렉터리를 삭제한다. 삭제할 때는 현재 사용자가 파일이나 디렉터리를 삭제할 권한이 있어야 한다. 단, root 사용자는 모든 권한이 있으므로 이 명령에 제약이 없다.
- Root 사용자는 모든 권한이 있으므로 이 명령에 제약이 없다.

사용 예	설 명
<b>rm abc.txt</b>	해당 파일 삭제
<b>rm -i abc.txt</b>	삭제 시 정말 삭제할지 확인하는 메시지가 나옴
<b>rm -f abc.txt</b>	삭제 시 확인하지 않고 바로 삭제(f는 Force의 약자)
<b>rm -rf abc</b>	abc 디렉토리와 그 아래에 있는 하위 디렉터리를 강제로 전부 삭제. 편리 하지만 상당히 주의해서 사용해야 함 (r은 Recursive의 약자)

- 리눅스에서는 Windows의 휴지통 개념을 잘 사용하지 않으므로 삭제한 파일이나 폴더를 복구하기가 상당히 어렵다는 점을 기억하자.

# 기본 명령어

## CP 명령어

- CoPy의 약자로, 파일이나 디렉터를 복사한다. 새로 복사한 파일은 복사한 사용자의 소유가 된다. 그러므로 명령을 실행하는 사용자는 해당 파일의 읽기 권한이 필요하다.

사용 예	설 명
<code>cp abc.txt cba.txt</code>	abc.txt를 cba.txt라는 이름으로 바꿔서 복사
<code>cp -r abc cba</code>	디렉터리 복사. abc 디렉터리를 cba 디렉터리로 복사

- 리눅스에서는 Windows의 휴지통 개념을 잘 사용하지 않으므로 삭제한 파일이나 폴더를 복구하기가 상당히 어렵다는 점을 기억하자.

# 기본 명령어

## mv 명령어

- MoVe의 약자로, 파일이나 디렉터리의 이름을 변경하거나 다른 디렉터리로 옮길 때 사용한다.
- 일반 파일 하나의 이름을 변경할 때 사용하는 명령어.

사용 예	설 명
<code>mv abc.txt /etc/systemd/</code>	abc.txt을 /etc/systemd/ 디렉터리로 이동
<code>mv aaa bbb ccc ddd</code>	aaa, bbb, ccc 파일을 '/ddd' 디렉터리로 이동
<code>mv abc.txt www.txt</code>	abc.txt의 이름을 <a href="#">www.txt</a> 로 변경해서 이동

# 기본 명령어

## mkdir 명령어

- MaKeDIRectory의 약자로, 새로운 디렉토리를 생성한다. 생성된 디렉터리는 명령을 실행한 사용자의 소유가 된다.

사용 예	설 명
<b>mkdir abc</b>	현재 디렉토리 아래에 '/abc'라는 디렉터리 생성.
<b>mkdir -p /def/fgh</b>	/def/fgh 디렉터를 생성하는데, 만약 '/fgh'의 부모 디렉터리인 '/def' 디렉터리가 없다면 자동 생성해줌(p는 Parents의 약자)

# 기본 명령어

## rmmdir 명령어

- ReMove DIRectory의 약자로, 디렉터리를 삭제한다. 해당 디렉터리의 삭제 권한이 있어야 하며 디렉터리는 비어 있어야 한다.
- 파일이 들어있는 디렉터리를 삭제하려면 'rm-r'을 실행해야 한다.

사용 예	설 명
rmmdir abc	abc 디렉터리 삭제

# 기본 명령어

## cat 명령어

- conCATenate의 약자로, 파일 내용을 화면에 보여준다. 여러 개의 파일을 나열하면 파일을 연결해서 보여준다.

사용 예	설 명
cat a.txt b.txt	a.txt 와 b.txt를 연결해서 파일의 내용을 화면에 보여줌



# 기본 명령어

## head, tail 명령어

- 텍스트 형식으로 작성된 파일의 앞 10행 또는 마지막 10행만 화면에 출력한다.

사용 예	설 명
<b>head /etc/systemd/user.conf</b>	해당 파일의 앞 10행을 화면에 출력
<b>head -3 /etc/systemd/user.conf</b>	앞 3행만 화면에 출력
<b>tail -5 /etc/systemd/user.conf</b>	마지막 5행만 화면에 출력

# 기본 명령어

## more 명령어

- 텍스트 형식으로 작성된 파일을 페이지 단위로 화면에 출력한다.  를 누르면 다음 페이지로 이동하여  를 누르면 앞 페이지로 이동한다.  를 누르면 종료한다

사용 예	설 명
<code>more /etc/systemd/system.conf</code>	
<code>more +10 /etc/systemd/system.conf</code>	10행부터 출력

# 기본 명령어

## less 명령어

- more명령어와 용도가 비슷하지만 기능이 더 확장되어 있다. more명령어에서 사용하는 키도 사용할 수 있으며 추가로 화사표 키나  ,  도 사용할 수 있다.

사용 예	설 명
<code>less /etc/systemd/system.conf</code>	
<code>less +10 /etc/systemd/system.conf</code>	10행부터 출력

# 기본 명령어

## file 명령어

- 파일이 어떤 종류의 파일인지를 표시해주는 명령어

사용 예	설 명
<b>file /etc/systemd/system.conf</b>	system.conf는 텍스트 파일이므로 아스키 파일(ASCII)로 표시됨
<b>file /bin/gzip</b>	gzip은 실행 파일이므로 'ELF 64-bit LSB shared object' 파일로 표시됨

# 기본 명령어

## clear 명령어

- 현재 사용 중인 터미널 화면을 깨끗하게 지워준다.

사용 예	설 명
clear	Windows 의 cls 와 같은 명령.

# 기본 명령어

## alias 명령어

- 명령어를 단축키 형식으로 사용할 수 있는 명령어

사용 예	설 명
<code>alias 1="ls -l"</code>	숫자 1로 명령어 <code>ls -l</code> 를 단축키로 등록, 1를 입력 하고 실행하면 <code>ls -l</code> 이 실행

# 기본 명령어

## find 명령어

- 옵션과 조건을 기준으로 파일을 찾는 명령어

형식: find [경로] [옵션] [조건] [외부실행]

옵 션	옵션 설명	옵 션	옵션 설명
-name	이름으로 찾기	-gid	GID로 찾기
-user	소유자로 찾기	-group	소유 그룹명으로 찾기
-perm	권한으로 찾기	-atime	접근 시간으로 찾기
-size	파일 크기로 찾기	-ctime	권한 변경 시간으로 찾기
-uid	UID로 찾기	-mtime	수정 된 시간으로 찾기
-newer	지정한 파일 전 후 파일 찾기	-empty	용량이 0인 파일 찾기

- atime/ctime/mtime은 +숫자, 숫자, -숫자로 표기할 수 있다.
- +숫자는 숫자보다 더 오래된 값, 숫자는 딱 그 값, -숫자는 숫자보다 더 덜된 값을 의미한다.

# 기본 명령어

## find 기본 사용 예

사용 예	설 명
<b>find /etc/ -name "*.conf"</b>	/etc 디렉터리 하위에서 확장명이 *.conf인 파일 검색
<b>find /home -user ubuntu</b>	/home 디렉토리 하위에서 소유자가 ubuntu인 파일 검색
<b>find ~ -perm 644</b>	현재 사용자의 홈 디렉터리 하위에서 허가권인 644인 파일 검색
<b>find /usr/bin -size +10k -size -100k</b>	/usr/bin 디렉터리 하위에서 파일 크기가 10kb~100kb인 파일 검색



# 기본 명령어

## find 고급 명령어

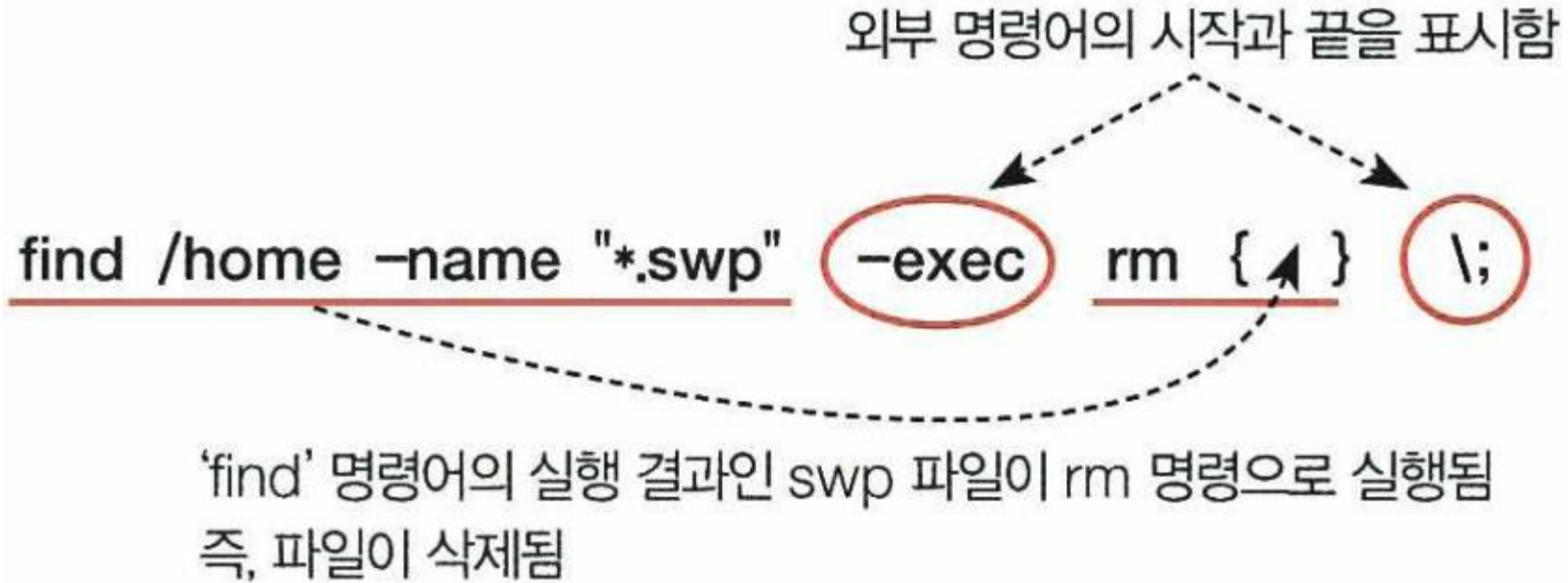
외부 실행 옵션	설 명
<b>-exec</b>	찾을 파일을 대상으로 강제로 명령을 실행 하는 옵션
<b>-ok</b>	찾은 파일을 대상으로 명령을 실행 시킬지 확인 하는 옵션

## find 고급 사용 예

사용 예	설 명
<b>find ~ -size 0k -exec ls -l {} \;</b>	현재 사용자의 홈 디렉토리 하위에서 파일 크기가 0인 파일의 목록을 상세히 출력
<b>find /home -name "*.swp" -exec rm {} \;</b>	/home 디렉토리 하위에서 확장명이 *.swp인 파일 삭제

# 기본 명령어

## find 고급 명령어 예



# 기본 명령어

## find 고급 명령어 예

```
yummy@ubuntu:~$ mkdir aaa
yummy@ubuntu:~$ mkdir aaa/aaa
yummy@ubuntu:~$ touch aaa/aaa/c.txt
yummy@ubuntu:~$ find /home/yummy/aaa -name "aaa" -exec tar cvfz aaa.tar.gz {} \;
tar: Removing leading `/' from member names
/home/yummy/aaa/
/home/yummy/aaa/aaa/
/home/yummy/aaa/aaa/c.txt
tar: Removing leading `/' from member names
/home/yummy/aaa/aaa/
/home/yummy/aaa/aaa/c.txt
yummy@ubuntu:~$ ls
aaa  aaa.tar.gz  a.txt  z
```





