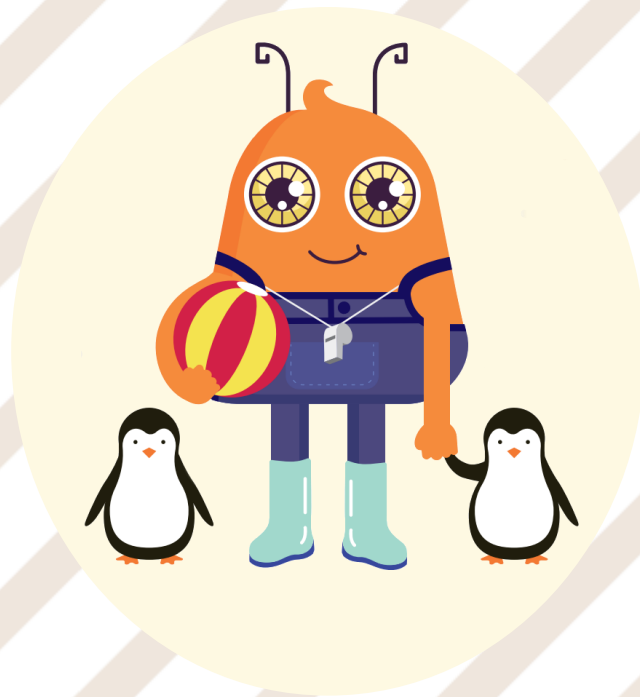


03

# CHAPTER

## 리눅스 기본 명령어



# C.ontents

---

**01** 리눅스 기본 명령어

**02** 파일 압축과 묶기

**03** 파일 명령어

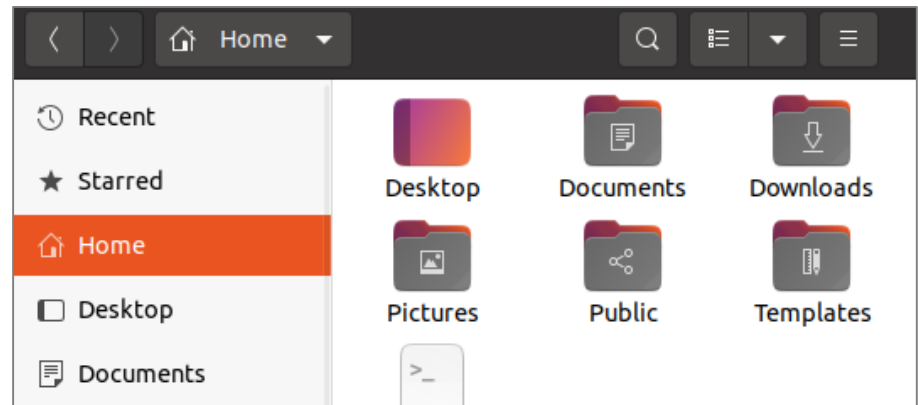
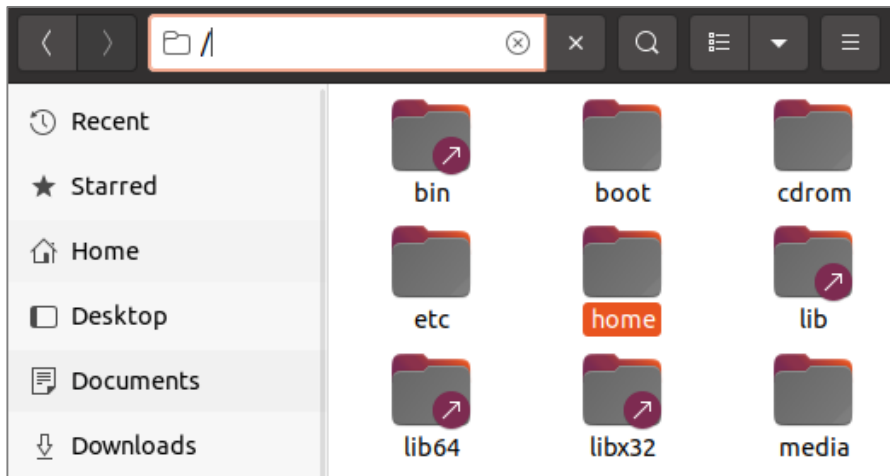
# 학습목표

---

- 리눅스의 기본 명령어를 익힌다.
- 파일 압축 및 묶기 명령어를 익힌다.
- 파일 관련 명령어를 익힌다.

# 1-1 기본 명령어

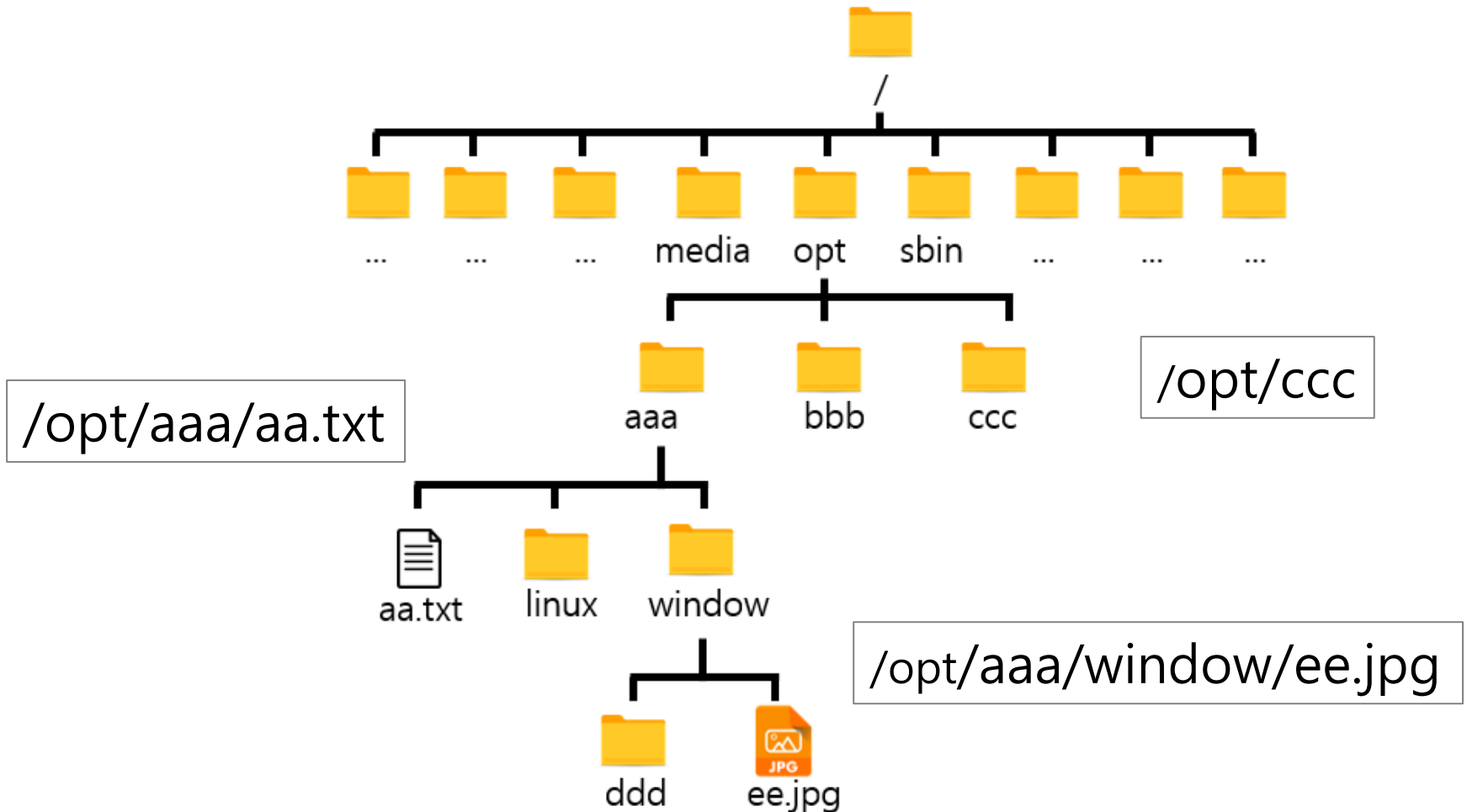
- 유닉스는 디렉토리, 파일로 구분되어 있음
  - 파일 – 확장자를 사용자가 만들어 낼수 있다.
  - 디렉토리 – 파일들을 모아놓는 논리적인 그룹
  - 최상위 디렉토리는 root directory라 한다.
    - root directory에 일반 user는 파일 삭제, 생성 권한이 없다
  - 사용자의 home은 /home/user\_id
    - Home에서 만든 파일과 디렉토리의 권한은 해당 사용자에게 있다.
  - Files 아이콘 누르고 **ctrl-I**, 직접 경로 작성 가능



# 1-1 기본 명령어

## ■ 절대경로

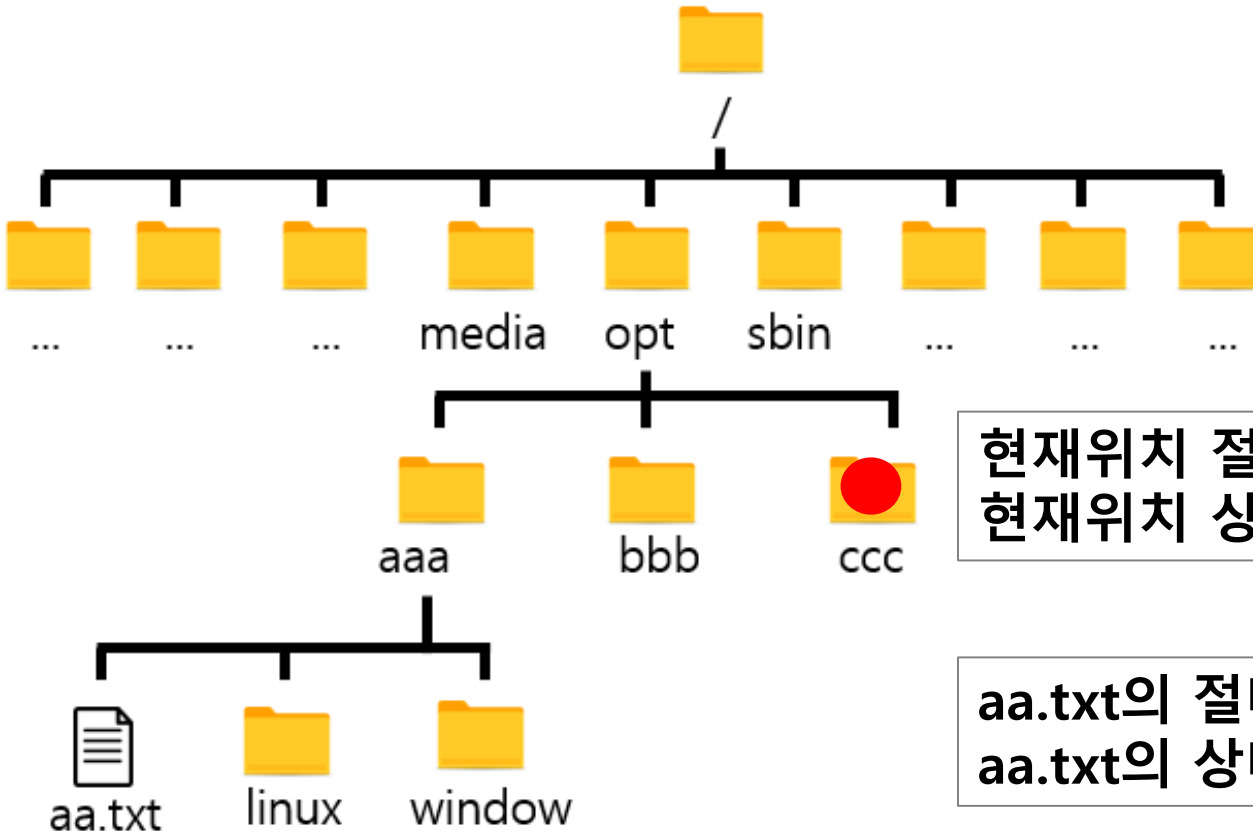
- Root directory로 부터 현재 파일이 위치한 디렉토리의 경로를 전체로 표기한 경로
- 예) /home/heejinlee/abc.txt



# 1-1 기본 명령어

## ■ 상대경로

- 현재 위치로부터 해당 파일이 있는 디렉토리를 표기한 경로
- 현재 위치가 바뀌면, 상대 경로도 변경
- 현재 위치를 점 한 개 "." 으로 나타냄, 상위디렉토리는 점 두개로 ".." 나타냄



현재위치 절대경로 - /opt/ccc

현재위치 상대 경로 - .

aa.txt의 절대경로: opt/aaa/aa.txt

aa.txt의 상대 경로: ../aaa/aa.txt

# 1-1 기본 명령어

## ■ 주요 기본 명령어

- ls: LiSt의 약자, 윈도우의 dir 명령어와 같은 기능(해당 디렉터리에 있는 파일 목록 나열)

```
# ls                -- 현재 디렉터리의 파일 목록을 보여줌
# ls /etc/systemd   -- /etc/systemd 디렉터리의 목록을 보여줌
# ls -a             -- 현재 디렉터리의 목록(숨김 파일 포함)을 보여줌
# ls -l             -- 현재 디렉터리의 목록을 자세히 보여줌(l은 L의 소문자)
# ls *.conf         -- 확장자가 conf인 목록을 보여줌
# ls -l /etc/systemd/b* -- /etc/systemd 디렉터리에 있는 목록 중 앞 글자가 b인 것을 자세히 보여줌
```

- cd: Change Directory의 약자, 디렉터를 이동하는 명령어

```
# cd -- 현재 사용자의 홈 디렉터리로 이동. 만약 현재 사용자가 root이면 /root 디렉터리로 이동
# cd ~ubuntu -- ubuntu 사용자의 홈 디렉터리로 이동
# cd .. -- 바로 상위의 디렉터리로 이동. '..'는 현재 디렉터리의 부모 디렉터를 의미(예: 현재 디렉터리가 /etc/systemd이면 /etc 디렉터리로 이동)
# cd /etc/systemd -- /etc/systemd 디렉터리로 이동(절대 경로)
# cd ../etc/systemd -- 상대 경로로 이동. 현재 디렉터리의 상위(..)로 이동한 후 /etc/systemd로 이동
```

- pwd: Print Working Directory의 약자, 현재 디렉터리의 전체 경로를 화면에 출력

```
# pwd -- 현재 작업 중인 디렉터리의 경로 출력
```

# 1-1 기본 명령어

- touch: 크기가 0인 새 파일을 생성하거나, 이미 파일이 존재한다면 파일의 최종 수정 시간을 변경

```
# touch abc.txt -- 파일이 없으면 abc.txt라는 빈 파일을 생성하고, abc.txt가 있으면 파일의 최종 수정 시간을 현재 시간으로 변경
```

- mkdir: MaKe DIRectory의 약자, 새로운 디렉터리를 생성  
생성된 디렉터리는 명령을 실행한 사용자의 소유

```
# mkdir abc -- 현재 디렉터리 아래에 /abc 디렉터리 생성  
# mkdir -p /def/fgh -- /def/fgh 디렉터리 생성. 만약 /fgh의 부모 디렉터리인 /def 디렉터리가 없으면 자동 생성(p: parents)
```

- rmdir: ReMove DIRectory의 약자, 디렉터를 삭제  
해당 디렉터리가 비어 있고 디렉터리에 대한 삭제 권한이 있어야 함  
파일이 들어 있는 디렉터를 삭제하려면 'rm -r' 실행

```
# rmdir abc -- /abc 디렉터리 삭제
```



# 1-1 기본 명령어

- cp: CoPy의 약자, 파일이나 디렉터리를 복사  
새로 복사된 파일은 복사한 사용자의 소유  
명령을 실행하는 사용자에게 해당 파일의 읽기 권한이 있어야 함

```
# cp abc.txt cba.txt -- abc.txt의 파일명을 cba.txt로 바꾸어 복사
# cp -r abc cba      -- 디렉터리 복사. abc 디렉터리를 cba 디렉터리로 복사
```

- rm: ReMove의 약자, 파일이나 디렉터리를 삭제  
사용자에게 해당 파일이나 디렉터리의 삭제 권한이 있어야 함  
root 사용자의 경우, 모든 권한을 가지고 있기 때문에 이 명령을 사용하는 데 제약 없음

```
# rm abc.txt      -- 해당 파일 삭제(내부적으로 rm -f로 연결됨)
# rm -i abc.txt   -- 삭제 시 정말 삭제할지 확인하는 메시지 출력
# rm -f abc.txt   -- 삭제 시 확인하지 않고 바로 삭제(f: force)
# rm -r abc       -- abc 디렉터리와 그 하위 디렉터리를 강제로 모두 삭제. 편리하지만 주의해서 사용
                  -- 해야 함(r: recursive)
```

- mv: MoVe의 약자, 파일이나 디렉터리 이름을 변경하거나 다른 디렉터리로 이동할 때 사용

```
# mv abc.txt /etc/systemd/ -- abc.txt를 /etc/systemd/ 디렉터리로 이동
# mv aaa bbb ccc ddd      -- aaa, bbb, ccc 파일을 /ddd 디렉터리로 이동
# mv abc.txt www.txt      -- abc.txt의 파일명을 www.txt로 변경
```

# 1-1 기본 명령어

- cat: conCATenate의 약자, 파일의 내용을 화면에 출력  
명령어 뒤에 여러 개의 파일명을 나열하면 파일을 연결하여 내용을 화면에 출력

```
# cat a.txt b.txt -- a.txt와 b.txt를 연결하여 파일의 내용을 화면에 출력
```

- head, tail: 텍스트 형식으로 작성된 파일의 앞 10행 또는 마지막 10행만 화면에 출력

```
# head /etc/systemd/bootchart.conf -- 해당 파일의 앞 10행을 화면에 출력  
# head -3 /etc/systemd/bootchart.conf -- 해당 파일의 앞 3행만 화면에 출력  
# tail -5 /etc/systemd/bootchart.conf -- 해당 파일의 마지막 5행만 화면에 출력
```

- more: 텍스트 형식으로 작성된 파일을 페이지 단위로 화면에 출력  
Space bar를 누르면 다음 페이지로 이동,  
B를 누르면 앞 페이지로 이동, Q를 누르면 종료

```
# more /etc/systemd/system.conf  
# more +10 /etc/systemd/ system.conf -- 해당 파일의 10행부터 출력
```

# 1-1 기본 명령어

- less: more 명령어와 용도가 비슷하지만 더 확장된 기능의 명령어  
more 명령어에서 사용하는 키도 사용할 수 있음  
추가로 ↑ , ↓ , ← , → , PageUp , PageDown도 사용 가능

```
# less /etc/systemd/system.conf  
# less +10 /etc/systemd/system.conf -- 해당 파일의 10행부터 출력
```

- file: 해당 파일이 어떤 종류의 파일인지 보여줌

```
# file /etc/systemd/system.conf -- system.conf는 텍스트 파일이므로 아스키 파일(ASCII)로 표시  
# file /bin/gzip -- gzip은 실행 파일이므로 ELF 64-bit LSB executable 파일로 표시
```

- clear: 현재 사용 중인 터미널 화면을 깨끗이 지워줌

```
# clear
```

## 2-1 파일 압축

### ■ xz

- 확장명 xz로 압축하거나 풀기
- 비교적 최신 압축 명령어이며 압축률이 뛰어나

```
# xz 파일명 -- '파일명.xz'라는 압축 파일 생성 및 기존 파일 삭제
# xz -d 파일명.xz -- '파일명.xz'의 압축을 풀어 '파일명'이라는 파일 생성(d: decompress)
# xz -l 파일명.xz -- '파일명.xz'에 포함된 파일 목록과 압축률 등 출력(l: list)
# xz -k 파일명 -- 압축 후 기존 파일을 삭제하지 않고 유지(k: keep)
```

### ■ bzip2

- 확장명 bz2로 압축하거나 풀기

```
# bzip2 파일명 -- '파일명.bz2'라는 압축 파일 생성
# bzip2 -d 파일명.bz2 == bunzip2 파일명.bz2 -- '파일명.bz2'의 압축을 풀어 '파일명'이라는 파일 생성
# bzip2 -k 파일명 -- 압축 후 기존 파일을 삭제하지 않고 유지(k: keep)
```

## 2-1 파일 압축

### ■ gzip

- 확장명 gz로 압축하거나 풀기

```
# gzip 파일명 -- '파일명.gz'라는 압축 파일 생성  
# gzip -d 파일명.gz == gunzip 파일명.gz -- '파일명.gz'의 압축을 풀어 '파일명'이라는 파일 생성
```

### ■ zip/unzip

- 윈도우와 호환되는 확장명 zip로 압축하거나 풀기

```
# zip 새파일명.zip 파일명 -- '새파일명.zip'라는 압축 파일 생성 및 기존 파일 유지  
# unzip 파일명.zip -- '파일명.zip'의 압축을 풀어 '파일명'이라는 파일 생성
```

## 2-2 파일 묶기

- Tar : 묶음 파일 만들거나 풀기
  - c(소문자): 새로운 묶음 파일 생성
  - x : 묶음 파일 풀기
  - v : visual을 의미, 파일을 묶거나 푸는 과정을 보여줌(생략 가능)
  - f(필수): 묶음 파일명을 지정
- t : 묶음 파일을 풀기 전에 묶인 경로를 보여줌
- C(대문자): 지정된 디렉터리에 묶음 파일 풀기 or 묶음 파일이 있는 디렉터리에 풀기
- J : tar+xz
- z : tar+gzip
- j : tar+bzip2
- 압축 풀기
  - tar xvfJ 파일명.tar.xz
  - tar xvfj 파일명.tar.bz2
- 압축하기
  - Tar cvf aaa.tar aaa -> aaa폴더를 aaa. Tar로 압축 묶음
  - Tar cvfz aaa.tar.gz aaa -> aaa 폴더를 gz으로 압축하고 tar로 묶음

## 2-2 파일 묶기

```
# tar cvf my.tar /etc/fonts/      -- 묶기
# tar cvfJ my.tar.xz /etc/fonts/  -- 묶기+xz로 압축
# tar cvfz my.tar.gz /etc/fonts/  -- 묶기+gzip로 압축
# tar cvfj my.tar.bz2 /etc/fonts/ -- 묶기+bzip2로 압축
# tar tvf my.tar                  -- 파일 확인
# tar xvf my.tar                  -- tar 풀기

# tar Cxvf newdir my.tar -- newdir에 tar 풀기
# tar xfJ my.tar.xz       -- xz 압축 풀기+tar 풀기
# tar xfz my.tar.gz       -- gzip 압축 풀기+tar 풀기
# tar xfj my.tar.bz2      -- bzip2 압축 풀기+tar 풀기
```

## 3-1 파일 위치 검색

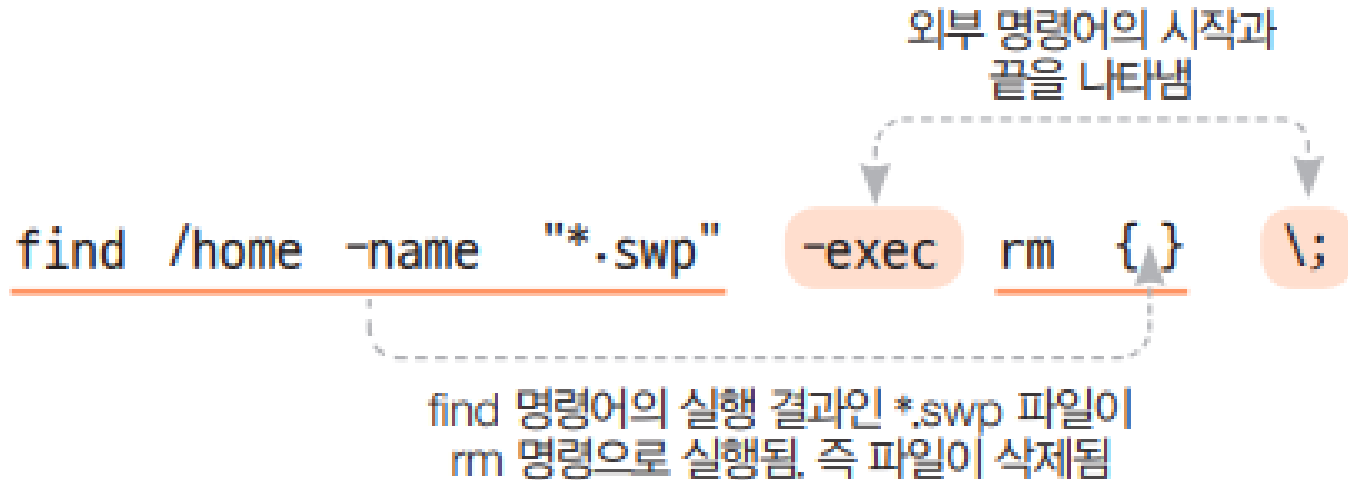
- find 경로 옵션 조건
  - 옵션: -name, -user(소유자), -newer(전, 후), -perm(허가권), -size(크기)

```
# find /etc -name "*.conf" -- /etc 디렉터리 하위에 있으며 확장명이 .conf인 파일 검색
# find /home -user ubuntu -- /home 디렉터리 하위에 있으며 소유자가 ubuntu인 파일 검색
# find ~ -perm 644 -- 현재 사용자의 홈 디렉터리 하위에 있으며 허가권이 644인 파일 검색
# find /usr/bin -size +10k -size -100k -- /usr/bin 디렉터리 하위에 있으며 크기가 10~100KB인
                                     파일 검색
```



## 3-1 파일 위치 검색

- find 경로 옵션 조건 action
  - 옵션: -name, -user(소유자), -newer(전, 후), -perm(허가권), -size(크기)
  - action: -print(기본 값), -exec(외부 명령 실행)



```
# find ~ -size 0k -exec ls -l { } \; -- 현재 사용자의 홈 디렉터리 하위에 있으며 크기가 0인 파일의 목록을 상세히 출력
```

```
# find /home -name "*.swp" -exec rm { } \; -- /home 디렉터리 하위에 있으며 확장명이 *.swp인 파일 삭제
```

## 3-1 파일 위치 검색

- which 실행파일명
  - PATH에 설정된 디렉터리와 절대 경로를 포함한 위치 검색
- whereis 실행파일명
  - 실행 파일과 소스, man 페이지 파일까지 검색
- locate 파일명
  - 실행 파일 말고 모든 파일 검색
  - **sudo apt-get install mlocate**
  - **updatedb** 명령을 한 번 실행해야 사용 가능

```
root@server:~# which  gzip
/bin/gzip
root@server:~# whereis  gzip
gzip: /bin/gzip /usr/share/man/man1/gzip.1.gz /usr/share/info/gzip.info.gz
root@server:~# locate  gzip
/bin/gzip
/snap/core/6350/bin/gzip
/snap/core/6350/usr/lib/klibc/bin/gzip
/snap/core/6350/usr/lib/python3.5/gzip.py
```

## 3-2 표준 입출력 변경

- 프로그램마다 기본적으로 표준 입출력 방법이 있다.
- 운영체제가 프로그램에 미리 할당함

파일디스크립터	설 명	기본
0	표준 입력 (standard input)	Keyboard를 통한 입력
1	표준 출력 (standard output)	모니터에 출력
2	표준 에러 (standard error)	모니터에 출력

- 표준 입출력을 변경하는 방법
  - 파이프 - |
  - 리다이렉션 - >, >>, <<, <

## 3-2 표준 입출력 변경 - 파이프, 필터, 리디렉션

### ■ 파이프

- 두 프로그램을 연결하는 연결 통로를 의미, '|' ( Shift + ₩ )를 사용

```
# ls -l /etc | less -- ls -l /etc 명령을 입력하면 파일이 너무 많아 한 페이지에 모두 담을 수 없기  
때문에 한 페이지씩 나누어 보겠다는 의미
```

### ■ 필터

- 필요한 것만 걸러주는 명령으로는 grep, tail, wc, sort 등이 있음
- 주로 파이프와 같이 사용

```
# ps -ef | grep bash -- ps -ef 명령을 입력하면 모든 프로세스 번호를 출력하고 bash라는 글자가 들  
어 있는 프로세스만 출력
```

- wc 사용법을 익혀보자. 매뉴얼 찾기
- sort 사용법을 익혀보자. 매뉴얼 찾기

## 3-2 표준 입출력 변경 - 파이프, 필터, 리디렉션

- 리디렉션

- 표준 입출력의 방향을 바꾸는 것
- 표준 입력은 키보드이고 표준 출력은 화면이지만, 이를 파일로 처리하고 싶을 때 주로 사용

```
# ls -l > list.txt -- ls -l의 결과를 화면에 출력하지 않고 list.txt 파일에 저장. 만약 list.txt  
파일이 있으면 덮어씀
```

```
# ls -l >> list.txt -- ls -l의 결과를 화면에 출력하지 않고 list.txt 파일에 저장. 만약 list.txt  
파일이 있으면 기존의 내용에 이어짐
```

```
# sort < list.txt -- list.txt 파일을 정렬하여 화면에 출력
```

```
# sort < list.txt > out.txt -- list.txt 파일을 정렬한 후 out.txt 파일에 저장
```

역순으로 정렬하고 싶다면? **sort -r**

## 3-2 표준 입출력 변경 - Standard Output - 1

- 입력은 한가지라서 문제가 없는데, 출력은 두가지 방법이라서 어떻게 구분하지?
- Output으로 찍히는 것을 1로 표현

```
$ [명령문] 1> file명  
$ ls 1> file1
```

```
user1@ubuntu:~$ ls  
aa          Documents  Music     newfile    Pictures   Templates  
Desktop     Downloads  ne        nnn        Public     Videos  
user1@ubuntu:~$ ls 1> file1  
user1@ubuntu:~$ cat file1  
aa  
Desktop  
Documents  
Downloads  
file1  
Music  
ne  
newfile  
nnn
```

리눅스프로그래밍 p. 172

## 3-3 표준 입출력 변경 - Standard Error - 2

- Error 문구는 2로 표현함.

```
$ [명령문] 2> file명
```

```
$ ls 2> file2
```

//이경우, ls에서 error문이 안나서 file2에 아무 값도 없음

```
heejinlee@ubuntu:~$ des 2>stderr
heejinlee@ubuntu:~$ des 2
```

Command 'des' not found, did you mean:

```
command 'den' from snap den (1.2.0-0)
command 'yes' from deb coreutils (8.30-3ubuntu2)
command 'dms' from deb anacrolix-dms (1.1.0-1)
command 'nes' from deb fceux (2.2.2+dfsg0-1build1)
command 'nes' from deb mednafen (1.22.2+dfsg-1build1)
command 'nes' from deb nestopia (1.50-1build1)
command 'dcs' from deb drbl (2.30.5-1)
command 'dep' from deb go-dep (0.5.4-3)
command 'les' from deb atm-tools (1:2.5.1-4)
command 'mes' from deb mes (0.22-1)
command 'dex' from deb dex (0.8.0-2)
```

See 'snap info <snapname>' for additional versions.

```
heejinlee@ubuntu:~$ cat stderr
```

Command 'des' not found, did you mean:

```
command 'den' from snap den (1.2.0-0)
command 'les' from deb atm-tools (1:2.5.1-4)
command 'dep' from deb go-dep (0.5.4-3)
command 'dcs' from deb drbl (2.30.5-1)
command 'nes' from deb fceux (2.2.2+dfsg0-1build1)
command 'nes' from deb mednafen (1.22.2+dfsg-1build1)
command 'nes' from deb nestopia (1.50-1build1)
command 'dex' from deb dex (0.8.0-2)
command 'mes' from deb mes (0.22-1)
command 'yes' from deb coreutils (8.30-3ubuntu2)
command 'dms' from deb anacrolix-dms (1.1.0-1)
```

See 'snap info <snapname>' for additional versions.

## 3-3 Standard Output/Error

- Error가 없으면 2> 를 했을때, 아무것도 프린트 되지 않는다.

```
heejinlee@ubuntu:~$ echo this is echo 1>stdout
heejinlee@ubuntu:~$ cat stdout
this is echo
```

```
heejinlee@ubuntu:~$ touch newfile
heejinlee@ubuntu:~$ echo this is file 2>newfile
this is file
heejinlee@ubuntu:~$ cat newfile
heejinlee@ubuntu:~$
```



## [과제]

- PPT에 있는 예제 다 해보기
- 리눅스 프로그래밍 1단원 리눅스 명령어 기초 실습 예제 하기



Thank You

---