

# 용어정리

**Root :** root는 linux에서 관리자 계정을 뜻하기도 하고  
linux 디렉토리 내 최상위 디렉토리 "/"를 의미하기도 한다.  
ex) cd / 즉 root 디렉토리로 이동

## 정리하기

### 1-1) | (파이프)

명령어1 | 명령어2    명령어 1을 실행한 결과값에 명령어 2를 실행한다.

grep이라는 파일을 원하는 조건으로 골라 출력하는 명령어는 파이프와 함께 자주 쓰인다.  
(cat aa.txt | grep abc : aa.txt에서 abc를 포함하는 내용을 출력하라)

### 1-2) | (파이프)와 같이 여러개의 결과로 다른 명령어를 실행하거나 명령어의 결과를 파이프에 기록할 때 사용.

cat > aa : aa 파일이 없으면 새로 생성하고 내용 입력함. replace  
aa 파일이 존재한다면 기존 aa 파일에 새로 입력한 내용을 덮어씀

cat >> aa : aa 파일 마지막 줄에 내용 추가 add

cat < aa : aa 파일을 이용하여 명령어 실행  
= cat aa && 줄아 ' < ' 는 줄 안쓰임

### 1-3) | s - a | > text.txt    현대적으로 실제 파일 현황을 text에 기록

grep "ab" abc.txt >> text.txt    abc.txt에서 "ab"가 포함된 내용을 추출하여 text.txt로 내용 추가.

### 2-1) cmp (compare) a b    a와 b의 어디부터 차이가 나는지 byte 단위 라인 번호를 출력한다.

diff (difference) a b    왼쪽 파일을 기준으로 오른쪽 파일이 어떻게 다른지 출력. n c n    line 번호 < , > 으로 파일 표시  
a: add / c: change / d: delete

diff -D flag a b    왼쪽 파일을 기준으로 오른쪽 파일에 어떻게 다른지 diff 보다 상세히 출력 ifindif: 왼쪽 줄 / else: 오른쪽 줄은 다름  
ifindif: 왼쪽이 없는 오른쪽 추가 메타.

### 2-2) sort filename    파일 정렬하여 출력 (오름차순)

sort -r filename    역순으로 정렬하여 출력 (내림차순)

sort -kn    sort는 기본적으로 첫번째 항목을 기준으로 오름차순 정렬을 하지만 -kn을 해주면 뒤에서 k번째 항목 (단어)를 기준으로 정렬을 할 수 있다.

### 2-3) find dirname    -name ~이름이 들어간 file 찾기

-mtime @m/n/-n    m일전(modified) 파일 찾기

-atime @m/n/-n    추가 (add)된 파일 찾기

-perm    권한을 가진 파일 찾기

-user    사용자 명이 같은 파일 찾기

-group    그룹명이 같은 파일 찾기

-type    파일 type 이 같은 것 찾기

(b 문자 디렉토리 f 파일 등)

USER	GROUP	OTHER
rwX	rwX	rwX
111	111	111
7	6	4
-perm	nn5	등의 설명은

3-1) tar -cvf a .    현대적으로 (.)의 파일들을 a라는 이름으로 묶고 그 다음 출력  
-xvf a    묶음 풀기  
-tvf a    파일 내 목록 상세보기  
-r a filename a 묶음 내 file 추가  
-u a filename a 묶음 내 같은 파일 명만 파일 덮어쓰기  
-zcvf a    압축 하기  
-zxvf a    압축 풀기

3-2) compress    압축하기  
uncompress    압축 풀기  
} UNIX에서는 많이 쓰이지만 linux에서는 잘 쓰이지 않음  
sudo apt install ncompress로 설치하기 가능함

### 3-3) gzip    압축

gunzip    압축 해제

gzip -vd    압축 해제

gzip -l    목록 리드아웃

-vq    초미세한 압축

-r    현재 디렉토리 내 모든 파일까지 압축

⊖ gzip 보단  
gzip -v를 더 씀

## ● 3. 이해하기

### (1) 파일 필터

파일의 내용 중 원하는 부분을 검색하거나 파일에 내용을 쓰거나 출력하고또는 기존파일에 추가하여내용을 쓸 수 있는 데, 이는 파일 필터를 활용한 명령어를 이용한다



#### 1) 파일 프 (l)와 grep 명령

파일내 지정된 패턴이 있는지 찾아내기 위하여 파일 프 필터(l)와 grep 명령을 사용 한다

##### ①grep

·grep(Globally find Regular-Expression and Print : 지정된 표현식이 전체에 있는지 찾아서 프린트 함)명령의 사용형식

·grep patten filename : 주어진 패턴이 있는지 해당 지정된 파일이나또는 확장파일형식으로 검색

##### Tip

예1) grep "gh" abc.txt : abc.txt파일안에 gh라는 문자가 있으면 출력

##### Tip

예2) grep "gh" \*.txt : 모든 .txt로 명칭이 끝나는 파일을 대상으로 파일안에 gh라는 문자가 있으면 출력하라

· ~ | grep patten : 필터를 이용하여 검색하는 방식으로 주어진 패턴을 검색하는 방법은 동일함

##### Tip

예) cat abc.txt|grep gh : abc.txt파일을 전부출력하고, 그 출력을 받아서 gh라는 글자 를 찾을. 결국 같은 결과가 나옴

##### ②grep 주요 옵션

## ● 3. 이해하기

· w: 전체 단어가 일치되는 경우 출력 equals .

· n: 라인넘버출력

· v: 단어가 일치하지 않는 경우출력 ex) abc def ~v def ~ 출력, abc

· l : 해당되는 파일명을출력

##### Tip

예) cat xinetd.conf|grep -nw telnetd : xinetd.conf파일내용 중 "telnetd"라는 단어가 정확히 일치하게 들어간 라인 수를 출력

### ● 3. 이해하기

#### 2) 리다이렉션(Redirection)

명령어의 결과를 다른 명령어의 입력으로 사용하는 경우나, 명령어의 결과를 파일에 기록하는 경우에 리다이렉션 기호를 사용한다.

##### ①파이프()

·명령 1 | 명령2: (파이프)어떤 명령의 결과를 받아 다른 명령을 실행

##### Tip

예) cat abc.txt | grep gh : 첫 번째 명령 abc.txt파일을 전부출력하고, 그 출력을 받아 서 두 번째 명령인 gh라는 글자를 찾음.

##### ②리다이렉션, 꺾쇠 (>, >>)

·명령 > filename : 어떤 명령의 결과를 지정된 명칭의 파일을 새로 생성하여 기록 > 은 덮어쓰기

·명령 >> filename : 어떤 명령의 결과를 지정된 명칭의파일 뒤로 계속 붙여서 기록함 >> 는 내용추가함

·명령 < filename : 어떤 명령의 입력으로 지정된 명칭의 파일을 사용함 cat file name == cat < filename 이라고 < 는 잘 쓰지않음

·예 1) cat xinetd.conf>a.file : xinetd.conf파일을 출력하고이 결과를 a.file에 기록, 만일 a.file의 명칭으로파일이 존재 하였다면, 이를 무시하고 새로 생성하고 기록함

##### Tip

예 1) cat xinetd.conf > a.file : xinetd.conf파일을 출력하고 이 결과를 a.file에 기록, 만일 a.file의 명칭으로 파일이 존재하였다면, 이를 무시하고 새로 생성하고 기록함

##### Tip

예 2) cat xinetd.conf >> a.file : xinetd.conf파일을 출력하고 이 결과를 a.file에 기록 하는데 기존 a.file의 맨끝부터 추가하여 기록함. 만일 a.file이 없다면 에러가 발생

### ● 4. 실습하기(1)

#### 1)파일필터

① grep hello abc.txt

② grep hello \*

③ cat abc.txt | grep hello

④ ps -ef | grep http

⑤ grep -wn hello abc.txt

⑥ grep -wnv hello abc.txt

⑦ grep -l hello abc.txt

⑧ grep -l hello \*

⑨ cat aa ; cat >aa; cat aa; cat >>aa;cat aa

## ● 3. 이해하기

### (2) 파일 비교, 정렬, 탐색

두 파일이 동일한 파일인지, 또는 다른 곳이 어디인지 알기 위하여 파일을 비교하거나, 원하는 파일이 어느 디렉토리에 위치해 있는지 파일을 찾기 위한 명령어를 다룬다.

#### 1) 파일 비교

파일을 비교하는 명령어는 `cmp`, `diff`의 명령이 있다. <그림 1-49>

① `cmp` → 비교 : 어디에서부터 다르냐 그 다른 지점의 정보 출력

· `cmp file1 file2` : 두 개의 (`file1`, `file2`) 파일을 비교하는 명령.

· 쉘 스크립트 프로그래밍에서 사용하기 적합함

· 같으면 `exit code = 0`을 반환 (셸 스크립트에서 사용)

· 다르면 `exit code = 1`을 반환 일치하지 않는 첫째 byte 출력

② `diff` → 어떻게 다르냐 비교

· `diff file1 file2` : 두 개의 (`file1`, `file2`) 파일의 차이를 보여 줌

· 두 파일 간 다른 부분을 각각 보여 주기 때문에 프로그램 작성시 프로그램 소스를 비교하는데 편리함

## ● 3. 이해하기

```
kopocte@kopocte:~/mydir$ cmp aa bb
aa bb differ: byte 4, line 2
kopocte@kopocte:~/mydir$ diff aabb 2c2
<aa      a a 2는 2임
---      change
bb 4c4
< sasa
---
cwsasa
kopocte@kopocte:~/mydir$
```

a = add  
d = delete  
> 2c2  
< 4c4

<그림 1-49> `cmp`, `diff` 명령어의 차이

다 1씩씩 else 2씩

2c2  
4c4  
다 1씩  
else 2씩

## ● 3. 이해하기

### 2) 파일 정렬

파일내용을 정렬하는 명령어는 sort의 명령이 있다. <그림 150>

#### ① sort

- 파일내용을 정렬 조건에 따라 정렬
- 정렬기준인 필드를 선택하여 오름차순이나 내림차순으로 전체 줄을 순서대로 정렬
- 필드 기준은 빈칸을 기준으로 수행함
- sort abc.txt : abc.txt파일을 오름차순으로 정렬함
- sort -r abc.txt : abc.txt파일을 내림차순으로 정렬함  
*reverse*
- sort -k2 abc.txt >result.txt : 2번째 필드를 기준으로 정렬  
*2번째 필드*  
*정렬*

## ● 3. 이해하기

### 1) 파일 검색

파일을 검색하는 명령어는 find 명령이 있다. <그림 151>

#### ① find

- 원하는 조건의 파일의 위치를 찾아줌 *전부*

#### 1) 사용법 : find dirname-option1 ...-optionN

- dirname은 주어진 디렉토리 및 이하 하위 디렉토리를 검색함
- option을 지정하여 다양한 조건과 실행법을 지정할 수 있음

#### 2) 주요 조건

- name : 파일이름이나 패턴을 지정하여 찾을

#### Tip

예1) find . -name abc.txt: 현재 디렉토리부터 하위 디렉토리 까지 abc.txt파일을 찾을

#### wildcard

#### Tip

예2)	find .	-name	*.txt	현재	디렉토리부터	하	디렉토리까지	맨끝이	.txt	끝나는
)	파일	찾음	:	재		위			로	
모	을			<i>.txt</i>	<i>이것은 파일 형식</i>					
든										

- perm : 파일의 퍼미션(권한)이 일치하는 파일을 찾을

### 3. 이해하기

aa.txt    <sup>이름</sup> kop:ab:kop <sup>read write & 실행 권한</sup> rwx rwx rwx  
USER GROUP OTHER

#### Tip

예) find . -perm 775 : 현재 디렉토리부터 하위 디렉토리까지 파일권한이 775인 파일을 찾음  
: 현재 디렉토리부터 하위 디렉토리까지 이름이 abc인 디렉토리를 찾음

-type : 파일의 파일유형이 일치하는 파일을 찾음

하위옵션) -type b : 문자로만 구성된 파일 찾기, -type d : 디렉토리 찾기, -type f : 일반적인 파일 찾기, -type l : 심볼릭 링크 찾기(심볼릭 링크는 추후 배움)

#### Tip

예) find . -type d -name abc : 현재 디렉토리부터 하위 디렉토리까지 이름이 abc인 디렉토리를 찾음

-user : 파일의 사용자(user)가 일치하는 파일을 찾음

#### Tip

예) find / -user kopoctc : 루트(/) 디렉토리부터 하위 디렉토리까지 파일의 사용자(만든 자 user)가 kopoctc인 파일을 찾음

-group : 파일의 그룹(group)이 일치하는 파일을 찾음

#### Tip

예) find / -group kopoctc : 루트(/) 디렉토리부터 하위 디렉토리까지 파일의 그룹이 kopoctc인 파일을 찾음

### 3. 이해하기

- atime +n/-n : <sup>recent file</sup> 최근 n일 이전에 액세스된 파일을 찾아줌(accessed time)  
+n은 n일 또는 그보다 더 오래 전의 파일을, -n은 오늘부터 n일 전까지의 파일을, n은 정확히 n일 전에 액세스된 파일을 찾아줌.

#### Tip

예) find / -atime 2 : 루트(/) 디렉토리부터 하위 디렉토리까지 파일 중 2일 전에 접근한 (access)한 파일을 찾아줌

-print : 찾은 파일명을 출력. 일반적으로 옵션을 주지 않아도 기본적으로 수행

-ls : 찾은 파일에 대한 ls명령을 수행

## ● 4. 실습하기(3)

### 2)파일비교 정렬

- ① `cmp abc.txt abc2.txt`
- ② `diff abc.txt abc2.txt`
- ③ `diff -Dflag abc.c abc2.c`
- ④ `sort abc.txt , sort -r abc.txt : 내림차순`
- ⑤ `sort -k2 abc.txt >result.txt : 2번째 필드를 기준으로 정렬`

### 3)파일 찾기

- ① `find . -name '*.c' -print`
- ② `find . -mtime 14 -print`  
*→ 수정된 시간을 기준으로 검색 (modify)*
- ③ `find . \( -name '*.c' -o -name '*.txt' \) -print : -o: OR`
- ④ `find / |grep abc` 이름에 abc가 들어있는 것은 모두
- ⑤ `find / -name abc` 이름이 abc인 것.

## ● 3. 이해하기

### (3) 파일 보관, 압축

저장공간의 백업이나 자료를 옮기기 위하여 여러 파일을 묶어서 보관하거나 이를 압축하여 보관하고, 또한 파일의 압축을 풀 경우가 있는데, 이러한 파일의 보관과 압축을 위한 명령어를 다룬다.

#### 1) tar (현재는 zip을 많이 사용하나 필요할 때가 있음)

파일을 보관하거나 푸는 명령어는 tar의 명령이 있다. <그림 15>

##### ①일반적 기능

- 파일을 묶고 푸는 기능 (압축기능 없음)
- 고전적으로 가장 많이 사용
- 파일을 묶어서 테이프로 저장 등에 사용

##### ②파일을 묶을 때

- 사용법 : `tar -cvf tarFileName fileList`

✱ `-c`: fileList에 대한 tar형식의 백업 파일을 생성

- `-v`: 진행되는 상황을 설명 *-빈도↓*

✱ `-f`: tar형식의 백업 파일 이름을 지정(default: /dev/rmt/0 ...tape drive)

*tar 묶이기 ⊕ -c  
compress  
gzip 압축*

#### Tip

예) `tar -cvf abc mydir` : 만일 mydir이 디렉토리 일 경우 mydir디렉토리의 이하 모든 파일들을 abc라는 파일명으로 묶음

## 2) compress/uncompress 유닉스에서 많이 쓰이던 명령어. Ubuntu에서도 default pack.이 아님

install을 해야 쓸 수 있음

파일을 압축과 푸는 명령어는 compress/uncompress의 명령이 있다.

### ① 압축 및 푸는 기능

- 파일을 압축 (파일 크기를 줄임)하고 푸는 명령어 compress
- 압축하는 경우 `compress abc`로 명령하면 abc파일이 압축되어 abc.z의 해당 이름의 .z의 확장자가 붙은 압축파일이 생성됨
- 압축을 푸는 경우 `uncompress abc.z`로 명령하면 abc.z파일이 압축이 풀려서 abc로 복원된 파일이 생성됨
- 파일을 묶어서 압축하는 기능은 `gzip.gunzip` 명령어를 많이 사용하는데 이는 개인용PC에서 많이 쓰는 ZIP압축방식을 사용.
- `compress/uncompress`는 유닉스, 리눅스 서버환경에서 전통적으로 많이 사용되는 유틸리티 임.
- `zip` 유틸리티보다 일반적으로 압축효율은 떨어짐
- 우분투 리눅스에서 `compress/uncompress`의 명령은 `apt-get install ncompress`로 유틸리티 설치 후 사용가능

### Tip

예1) `compress *.txt` : 모든 \*.txt를 확장자로 가진 파일들이 파일 하나하나 별로 압축 이 수행되며 해당 파일명+.Z의 명칭을 가진 파일이 여러 개 생성

### Tip

예2) `uncompress *.Z` : 모든 Z 확장자를 가진 각각의 파일에 대하여 uncompress 풀기를 실행

## 3) gzip/gunzip

파일을 압축과 푸는 명령어로 `gzip/gunzip`의 명령이 있다.

### ① 압축 및 푸는 기능

- 많이 사용되는 파일을 압축 (파일 크기를 줄임)하고 푸는 명령어
- 압축하는 경우 `gzip abc`로 명령하면 abc파일이 압축되어 abc.gz의 해당 이름의 .gz의 확장자가 붙은 압축파일이 생성됨

- 압축을 푸는 경우 `gunzip abc.z`로 명령하면 abc.gz 파일이 압축이 풀려서 abc로 복원된 파일이 생성됨

- gunzip 명령은 `gzip -d` 명령과 동일

### ② gzip 명령어의 주요 옵션

- `-d`: 압축을 푼다
- `-l`: 현재 압축된 파일의 내용을 보여준다.
- `-r`: 현재 디렉토리부터 하위 디렉토리까지 전부를 압축한다.
- `-t`: 압축된 파일의 완전성을 검사한다.
- `-v`: 압축 진행상황을 보여준다.
- `-9`: 최대한 압축한다.

### Tip

`gzip *` / `gunzip *`로 전체 파일을 대상으로 압축과 풀기를 실행 <그림 1-53>



● 4. 실습하기(3)

#### 4) tar명령어

- 1 tar -cvf tarfile .
- 2 tar -tvf tarfile
- 3 tar -rvf tarfile reverse.c
- 4 tar -uvf tarfile reverse.c
- 5 mkdir ./tmp
- 6 cd tmp
- 7 tar -xvf ../tarfile
- 8 tar -xvf ../tarfile | tar -tf ../tarfile | grep 'lady\*'
- 9 tar -xvf ../tarfile | reverse.c
- 10 tar -cvf tarfile /dev/rmt0, tar -xvf /dev/rmt0

- 4. 실습하기(3)

## 5) compress, uncompress 명령어

- ① `cp abc abc2`
- ② `compress abc // abc.Z`과 `abc2`와 사이즈비교..
- ③ `uncompress abc.z`
- ④ `compress *.txt` : 수많은 `Z`파일
- ⑤ `uncompress *.Z` : 모든 `Z` 확장자 파일 `uncompress`

## 6) gzip, gunzip 명령어

- ① `gzip -v9 tarfile` 압축률이 가장 높게 압축
- ② `ls tarfile.gz`
- ③ `gzip -l tarfile.gz`
- ④ `gzip -vd tarfile.gz`
- ⑤ `gunzip tarfile.gz`
- ⑥ `gzip -vd tarfile.gz > tar.log &` 콘피임을 처리할 때  
백그라운드로 돌릴 것  
↓  
다른 명령어 실행이 가능함

- **4. 실습하기 (4 – Jump Up)**

- 1) 금일 배운 명령어를 man으로 찾아보고 각종 옵션에 대하여 조사 후 실습
  - 2) 금일 배운 명령어를 조합하여 복수의 명령어를 한번에 실행하는 방법을 실습
- ; , while : do done

while true; do cat aa ; sleep2 ; done