

ENTERPRISE LINUX ADMIN GUIDE

유용한 명령어

단원목표

- `cmp / diff` 명령어 [비교]
- `sort` 명령어 [정렬]
- `file` 명령어 [확인]

diff/sort/file 명령어

- diff 명령어

```
# diff file1 file2
# diff -recursive dir1 dir2
```

- sort 명령어

```
# CMD | sort -k 3
# CMD | sort -k 3 -r

# df -h
# du -sk /var
# cd /var ; du -sk * | sort -nr | more
```

- file 명령어

```
# cd /etc ; file *
```

1

cmp / diff CMD

(1) cmp 명령어

NAME

cmp - compare two files

SYNOPSIS

cmp [-l | -s] file1 file2 [skip1 [skip2]]

DESCRIPTION

The cmp utility compares two files of any type and writes the results to the standard output. By default, cmp is silent if the files are the same; if they differ, the byte and line number at which the first difference occurred is reported.

Bytes and lines are numbered beginning with one.

서로 다른 파일을 비교하여 다른 내용을 출력해 주는 명령어이다. `cmp(compare)`, `diff(different)`

[명령어 형식]

cmp file1 file2

/ 두 개의 파일에 대한 비교점(차이점의 시작 정도) 확인 */*

[명령어 옵션]

옵션	설명
-l	두 파일 내용을 비교함에 있어 틀린곳마다 byte 수 (10진수)와 틀린 byte 수(8진수)를 출력
-s	틀린 파일의 내용을 출력하지 않고 return code 만 변환한다

[EX1] cmp 명령어 실습

```
# cd /test
```

```
# cat > cmpfile1
```

```
11111
22222
33333
<Ctrl + D>
```

```
# cat > cmpfile2
```

```
11111
22222
33333
<Ctrl + D>
```

```
# cat > cmpfile3
```

```
11111
22222
44444
<Ctrl + D>
```

```
# cmp cmpfile1 cmpfile2
```

```
# cmp cmpfile1 cmpfile3
```

```
cmpfile1 cmpfile3 differ: char 13, line 3
```

(2) diff 명령어

이름

diff - 두 파일에서 차이점을 찾는다.

사용법

diff [options] from-file to-file

설명

간단한 예로, diff 는 from-file 와 to-file 의 두파일의 내용을 비교한다. - 의 파일명은 표준입력으로부터 읽어들이며 내용을 나타낸다. 특별한 경우로, diff - - 는 자기 자신을 표준입력으로 비교한다.

If from-file 이 디렉토리이고, to-file 이 아니라면 diff 는 to-file의 파일과 from-file안의 파일을 비교한다.

from-file 과 to-file 이 모두 디렉토리라면, diff 는 알파벳 순서로 두 디렉토리 안의 상응하는 파일을 비교한다; 이 비교는 -r이나 -recursive 옵션이 주어지지 않으면 재귀적이지 않다. diff 는 파일인양 실제 디렉토리를 비교하지는 않는다. 표준입력은 같은 이름을 가진 파일개념을 적용하지 않기때문에 꼭한 파일은 표준입력되지 않을 수 있다.

diff 은 -, 로 시작된다, 그래서 대개 from-file 과 to-file 은 - 로 시작되지 않을런지 모른다.

Options

아래는 그누 diff에서 쓰이는 모든 옵션들의 요약이다. 대부분의 옵션은 두개의 이름을 갖는다. 하나는 -에 앞서는 단일 문자이고, 다른 하나는 --에 앞서는 긴 이름이다. 복수의 단일 문자 옵션은 단일 명령행에 결합될 수 있다: -ac 는 -a-c와 같다. 긴 이름의 옵션은 그 이름의 특정부분만으로 줄여 쓸수가 있다. 대괄호 ([와]) 은 옵션이 임의의 인자를 갖는다는 것을 가리킨다.

-c 내용 출력폼을 사용한다.

-i ignore changes in case; consider upper- and lower-case letters equivalent.

--recursive

두 디렉토리를 비교할때, 모든 서브디렉토리는 재귀적으로(recursively) 비교한다.

[명령어 형식]

```
# diff file1 file2 /* 두 개의 파일에 대한 간략한 차이점 */
# diff -c file1 file2 /* 두 개의 파일에 대한 자세한 차이점 확인 */
# diff -i file1 file2 /* -i : 대소문자를 구분하지 말아라. A와 a는 같은 것으로 간주 */
```

[명령어 옵션]

옵션	설명
--brief	두 파일의 내용이 같은지 다른지 알아봄
-c	파일의 이름, 날짜 등 파일의 차이점을 상세히 출력
-d	두 디렉토리간의 차이점 출력
-H	용량이 큰 파일 비교 시 속도를 빠르게 비교
-i	대소문자 구분하지 않음

[EX1] 파일 비교

```
# cd /test
# cat > cmpfile1
```

```
11111
22222
33333
<Ctrl+D>
```

```
# cat > cmpfile2
```

```
11111
22222
33333
<Ctrl+D>
```

```
# cat > cmpfile3
```

```
11111
22222
44444
<Ctrl+D>
```

```
# diff cmpfile1 cmpfile2
```

```
#
```

```
# diff cmpfile1 cmpfile3 [ cmp와 달리 파일의 틀린 라인을 표시해 준다 좌 : < , 우 : > ]
```

```
3c3
< 33333
---
> 44444
```

```
# diff -c cmpfile1 cmpfile3
```

```
*** cmpfile1    Wed Jan 27 00:54:39 2010
--- cmpfile3    Wed Jan 27 00:55:05 2010
*****
*** 1,3 ****
    11111
    22222
! 33333
--- 1,3 ---
    11111
    22222
! 44444
```

[EX2] 대소문자 구별 없이 파일 비교

```
# vi file1
```

```
11111
22222
iiii
```

```
# vi file2
```

```
11111
22222
IIII
```

```
# diff file1 file2
```

```
3c3
< iiii
---
> IIII
```

```
# diff -i file1 file2      (-i : ignore case, 대소문자 구분하지 않음)
#
```

[EX3] 디렉토리 비교시

```
# cd /test
```

```
# rm -rf /test/*
```

```
# mkdir dir1
```

```
# vi dir1/file1
```

```
1111
```

```
# mkdir dir2
```

```
# vi dir2/file1
```

```
2222
```

```
# diff --recursive dir1 dir2
```

```
diff --recursive dir1/file1 dir2/file1
1c1
< 1111
---
> 2222
```

```
# vi dir2/file1
```

```
1111 <----- '2222 -> 1111' 변경
```

```
# touch dir2/file2
```

```
# diff --recursive dir1 dir2 두 디렉토리를 비교할 때 사용한다
```

```
Only in dir2: file2
```

(실무 예) 파일 비교

```
(원 본) -----> (백업 본)  
/source/* (server.xml)      /source/* (server.xml)
```

```
# diff server.xml servier.xml.old
```

```
[EX] /etc/httpd/conf/httpd.conf 파일  
# cd /test ; rm -rf /test/*  
# cp /etc/httpd/conf/httpd.conf /test  
# cp httpd.conf httpd.conf.old  
# vi httpd.conf  
/ServerName -> n  
[수정전]  
#ServerName www.example.com:80  
[수정후]  
ServerName 172.16.9.2XX:80  
# diff httpd.conf httpd.conf.old
```

(실무 예) 디렉토리 마이그레이션 작업

```
/was1/* ---- Migration ----> /was2/*
```

```
# ls -alR /was1 | wc -l  
# ls -alR /was2 | wc -l
```

```
# diff --recursive /was1 /was2
```

NAME

sort - sort lines of text files

SYNOPSIS

sort [OPTION]... [FILE]...

DESCRIPTION

Write sorted concatenation of all FILE(s) to standard output.

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

-d, --dictionary-order
consider only blanks and alphanumeric characters

-f, --ignore-case
fold lower case to upper case characters

-n, --numeric-sort
compare according to string numerical value

-r, --reverse
reverse the result of comparisons

-k, --key=POS1[,POS2]
start a key at POS1, end it at POS2 (origin 1)

SEE ALSO

The full documentation for sort is maintained as a Texinfo manual. If the info and sort programs are properly installed at your site, the command

info sort

should give you access to the complete manual.

파일의 정렬. 하나 또는 그 이상의 파일의 텍스트 줄을 스크린 상에서 정렬 하고자 할 때 사용한다. 즉 출력 내용을 정렬하여 표현하고자 할 때 사용한다. sort 명령어는 아무런 옵션 없이 사용되면 숫자나 알파벳 순으로 정렬하여 준다. 기본 적으로 sort 명령어는 공백 문자(white space: space, Tab)를 필드 구분자로 인식한다.

[명령어 형식]

```
# sort /etc/passwd
# sort -r /etc/passwd          /* -r : reverse */
# sort -k 3 filename          /* -k : key */
# sort -t : -k 3 -n /etc/passwd /* -t : seperate, -n : numeric */
```

[명령어 옵션]

옵션	설명
-n	숫자로 정렬한다.
-r	내림차순으로 정렬한다. 기본은 오름차순으로 정렬하는 것이다.
-o	출력 결과를 파일에 저장한다.
-t	필드 구분자를 지정한다. (기본값은 공백이 기준이 된다)
-k	정렬할 필드를 지정한다.

[EX1] sort 정렬 실습

vi sortfile1

```
linux100 10 20 31 50
linux200 20 25 31 20
linux300 30 20 30 40
linux400 50 20 30 80
```

- 오름차순 정렬
 - 1필드->2필드->3필드-> ... 정렬
 - 문자열 정렬
- \longleftrightarrow (-r) 내림차순 정렬
 \longleftrightarrow (-k) 필드 지정 정렬
 \longleftrightarrow (-n) 숫자열 정렬

sort sortfile1 /* 첫 번째 필드를 기준으로 정렬 */

```
linux100 10 20 31 50
linux200 20 25 31 20
linux300 30 20 30 40
linux400 50 20 30 80
```

sort -r sortfile1 /* 내림차순으로 정렬 */

```
linux400 50 20 30 80
linux300 30 20 30 40
linux200 20 25 31 20
linux100 10 20 31 50
```

sort -k 3 -n sortfile1 /* 세 번째 필드를 기준으로 숫자로 정렬 */

```
linux100 10 20 31 50
linux300 30 20 30 40
linux400 50 20 30 80
linux200 20 25 31 20
```

sort -t : -k 1 /etc/passwd /* -t (seperate) 구분자로 :사용, 기준으로 첫번째 행 정렬 */

```
adm:x:3:4:adm:/var/adm:/sbin/nologin
amanda:x:33:6:Amanda user:/var/lib/amanda:/bin/bash
apache:x:48:48:Apache:/var/www:/sbin/nologin
avahi-autoipd:x:101:107:avahi-autoipd:/var/lib/avahi-autoipd:/sbin/nologin
avahi:x:70:70:Avahi daemon:/:/sbin/nologin
bin:x:1:1:bin:/bin:/sbin/nologin
cyrus:x:76:12:Cyrus IMAP Server:/var/lib/imap:/bin/bash
..... (중략) .....
```

sort -t : -k 3 -n /etc/passwd

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
..... (중략) .....
```

[EX2] sort 명령어 활용

```
# CMD | sort
# CMD | sort -r
# CMD | sort -nr

# ps -ef | head | sort
# ps -ef | head | sort -r
# ps -ef | head | sort -k 3
# ps -ef | head | sort -k 3 -r

# df -h | sort -k 4
# df -h | sort -k 4 -r
```

df -h

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda1	18G	5.5G	12G	33%	/
tmpfs	252M	0	252M	0%	/dev/shm

du -sk /var /* -s : sum, -k : Kbytes */

191416	/var
--------	------

cd /var

du -sk * | sort

12	account
146552	lib
16	empty
16	ftp
20	gdm
248	named
28	db
30948	cache
3164	log
32	kerberos
356	arpwatch
4	mail
40	yp

du -sk * | sort -n /* 첫번째 열을 숫자로 정렬 */

4	mail
8	crash
8	cvs
8	games
8	local
8	nis
8	opt
8	preserve
8	racoon
8	tmp
8	tux
12	account
16	empty
16	ftp
20	gdm

du -sk * | sort -nr | more /* 첫번째 필드열 역순으로 정렬 */

146552	lib
30948	cache
8920	www
3168	log
464	run
436	spool
356	arpwatch
248	named

3 file CMD

NAME

file - determine file type

SYNOPSIS

```
file [ -bchikLnNprsvz ] [ -f namefile ] [ -F separator ]  
[ -m magicfiles ] file ...  
file -C [ -m magicfile ]
```

DESCRIPTION

This manual page documents version 4.17 of the file command.

File tests each argument in an attempt to classify it. There are three sets of tests, performed in this order: filesystem tests, magic number tests, and language tests. The first test that succeeds causes the file type to be printed.

The type printed will usually contain one of the words text (the file contains only printing characters and a few common control characters and is probably safe to read on an ASCII terminal), executable (the file contains the result of compiling a program in a form understandable to some UNIX kernel or another), or data meaning anything else (data is usually 'binary' or non-printable). Exceptions are well-known file formats (core files, tar archives) that are known to contain binary data. When modifying the file /usr/share/file/magic or the program itself, preserve these keywords. People depend on knowing that all the readable files in a directory have the word 'text' printed. Don't do as Berkeley did and change 'shell commands text' to 'shell script'. Note that the file /usr/share/file/magic is built mechanically from a large number of small files in the subdirectory Magdir in the source distribution of this program.

file 명령어는 파일의 종류(File Type)을 알 수 있는 명령어이다. 많이 사용되는 명령어는 아니지만 특별한 경우에 사용 될 수 있다. 예를 들어 인터넷 상에서 다운로드 받은 파일이 정확히 어떤 종류인지를 확인할 때 사용한다.

Unix와 리눅스에서는 확장자(Extention)는 특별한 의미를 가지고 있지 않은 경우가 대부분이기 때문이다. 인터넷상에서 다운로드 받는 대부분의 프로그램 파일들은 filename.tar.gz 형태로 되어져 있는 경우가 많다. filename.tar.gz 라는 이름은 만든 사람이 잘못 배포하는 경우 filename.tar로 배포 되는 경우가 발생할 수 있다. 이런 경우 다운로드 받은 파일의 형태를 정확히 알지 못하는 경우 프로그램을 사용할 수 없게 된다. file명령어를 사용하여 파일의 형식(File Type)을 정확히 알아서 원본 확인의 확장자(Extention)로 복구한 후 압축을 풀고 아카이브(Archive)를 풀어 사용하게 되면 해결된다.

[참고] Windos / Unix 계열의 파일 확장자에 대해서

Windows	-> 파일에 확장자(예: file.txt, hwp.exe,)
Unix, Linux	-> 파일의 확장자가 거의 의미가 없다.

[명령어 형식]

file /etc/passwd

[EX1] 파일의 종류 확인

```
# file /etc/passwd          /* ASCII파일 (# cat /etc/passwd) */
```

```
/etc/passwd: ASCII text
```

```
# file /bin/ls              /* Binary 파일 (# strings /bin/ls) */
```

```
/bin/ls: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), for GNU/Linux 2.6.9, dynamically  
linked (uses shared libs), for GNU/Linux 2.6.9, stripped
```

```
# file /var/run/utmp        /* Date 파일 */
```

```
/var/run/utmp: data
```

```
# file /etc/rc.d/init.d/sendmail /* Script 파일 */
```

```
/etc/rc.d/init.d/sendmail: Bourne-Again shell script text executable
```

```
# file /etc/hosts /etc     /* 여러개의 파일 지정 */
```

```
/etc/hosts: ascii text
```

```
/etc: directory          /* Directory 파일 */
```

```
# cd /etc
```

```
# file *
```

```
DIR_COLORS:             ASCII English text  
DIR_COLORS.xterm:       ASCII English text  
Muttrc:                  ASCII English text  
Muttrc.local:           empty  
NetworkManager:         directory  
Pegasus:                 directory  
X11:                     directory  
a2ps-site.cfg:           ASCII English text  
a2ps.cfg:                ASCII English text  
acpi:                   directory  
adjtime:                 ASCII text  
aide.conf:               ASCII English text  
..... (중략) .....
```