redhat CentOS

ENTERPRISE LINUX ADMIN GUIDE

디렉토리와 파일관리

단원목표

- 디렉토리 이동 관련 명령어
- 디렉토리 관리 명령어
- 파일 관리 명령어
- 파일 내용 확인 명령어
- 기타 관리용 명령어

디렉토리 이동 관련 명령어

- pwd 명령어

```
# pwd
```

PS1 변수 (# export PS1='\w@\h"\w# '

- cd 명령어

경로(PATH)
- 상대경로
- 절대경로

```
# cd
# cd ~fedora

#cd -
#cd ../dir1
```

1 pwd CMD (printing working directory)

이름

pwd - 현재/작업 디렉토리명을 출력한다

개요

```
pwd
pwd {--help,--version}
```

설명

이 맨페이지는 GNU 버전의 pwd 를 다룬다. pwd 는 현재 디렉토리의 완전한 이름을 출력한다. 즉 출력된 이름의 모든 구성 요소들이 실제 디렉토 리 를 의미한다. - 심볼릭 링크가 아니라.

대 부 분의 유닉스 쉘들은 내장 pwd 명령으로 비슷한 기능을 제공하고 있다. 따라서 대부분의 pwd 명령은 지금 이것이 아니라 내부에서 수행되는 것일 것이다.

파일 작업 중이나 자료의 위치로 이동 할 경우 현재 내가 작업하고 있는 디렉토리의 위치를 알고 이동한 디렉토리를 지정해야 한다. 디렉토리 이동 관련 명령어에는 pwd 명령어, cd 명령어가 있다.

현재 내가 작업하고 있는 디렉토리의 위치를 절대경로를 통해 /(root)부터의 전체 경로를 출력해 준다.

[명령어 형식]

```
# pwd
```

[EX1] pwd 명령어 실습

```
# cd
# pwd
```

```
/root
```

```
# cd /etc/sysconfig/network-scripts
# pwd              /* 위치가 변경되었다. */
```

```
/etc/sysconfig/network-scripts/
```

```
# cd
# pwd
```

```
/root
```

[참고] PS1 변수설정

PS1 변수는 쉘 프롬프트를 변경할 때 사용하는 변수다. # echo $PS1으로 현재 설정된 PS1 변수의 값을 확인가능하다.
pwd 명령어로 디렉토리 변경 할 때 마다 현재 디렉토리를 확인하는 것은 불편하다. 따라서 PS1 변수를 환경변수에 선언하여 사용하면 pwd 명령어를 굳이 사용하지 않아도 된다.

- PS1 변수: 쉘 프롬프트를 정의할 때 사용하는 변수

```
# echo $PS1
[\u@\h \W]\$      ->      [root@linux249 ~]#

# PS1='a  '

         # man bash
         /PROMPTING

# PS1='[\u@\h \w]\$ '          /* 쉘프롬프트에 $PWD 변수 추가 */

                                    \w , \W의 차이

                                    소문자는 (\w) 전체경로 출력
                                    대문자(\W)는 경로중의 마지막 디렉터리만 출력

# vi  ~/.bashrc     (# gedit ~/.bashrc)
```

```
..... (중략) .....
#
# Sfecific Configuration
#
PS1='[\u@\h \w]\$ '
export PS1
```

- 개인적인 설정내용은 파일의 아래에 설정하면 편하다.

```
# .  ~/.bashrc     (# source ~/.bashrc)
# echo $PS1
```

```
[\u@\h \w]\$
```

```
# man bash
이후 /PROMPTING 으로 검색해보면 각 매직넘버의 정의에 대해 확인할 수 있다.
```

2 cd CMD (change working directory)

NAME
cd - change the working directory

SYNOPSIS
cd [-L | -P] [directory]

cd -

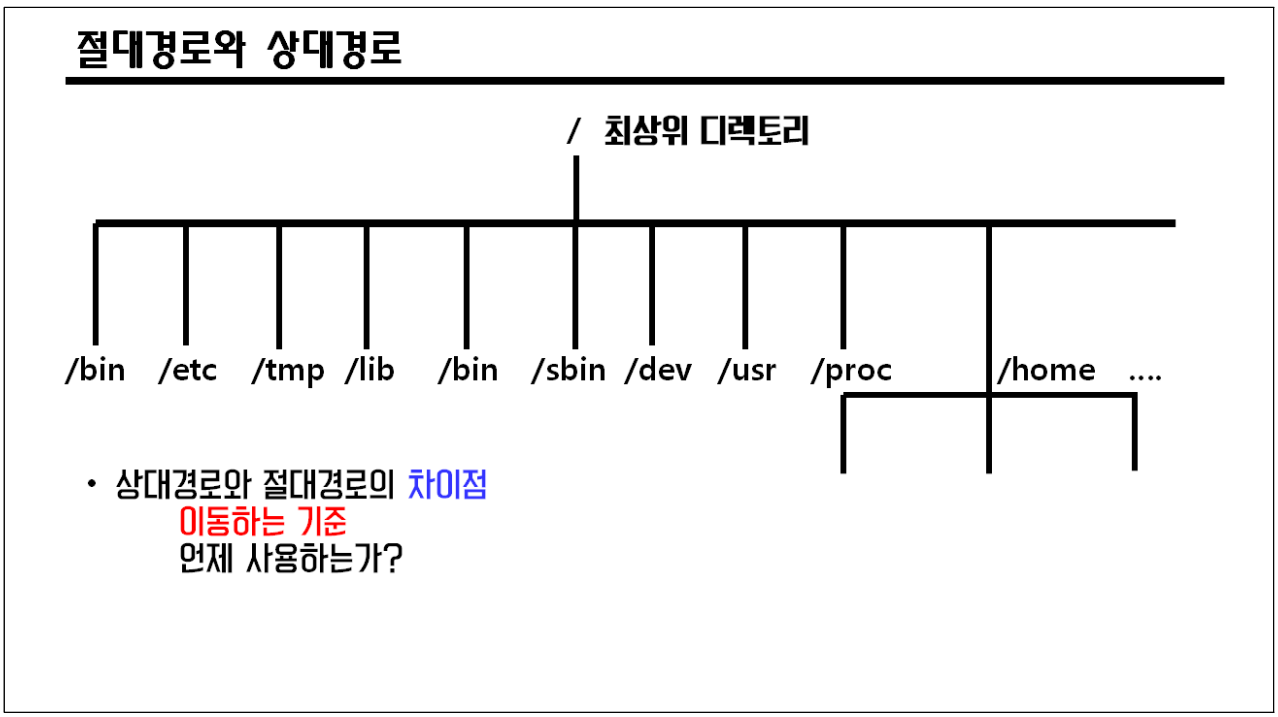
DESCRIPTION
The cd utility shall change the working directory of the current shell execution environment (see Shell Execution Environment) by executing the following steps in sequence. (In the following steps, the symbol curpath represents an intermediate value used to simplify the description of the algorithm used by cd. There is no requirement that curpath be made visible to the application.)

작업 디렉토리를 변경하고자 할 때 사용한다. 인자(Argument)인 디렉토리명은 이동하고자 하는 경로명을 나타낸다. 디렉토리 명을 지정하지 않고 cd 명령어를 단독으로 사용하면 현재 작업디렉토리가 어디에 있든지 사용자의 홈 디렉토리로 이동한다. cd 명령어는 디렉토리를 변경하는 명령어이다. cd 명령어를 사용하여 디렉토리 경로를 변경하는 경우 상대경로(Relative Path)나 절대경로(Absolute Path)를 사용 할 수 있다.

상대경로(Relatvie Path)는 이동하는 기준이 현재 디렉토리이며, 현재 디렉토리를 기준으로 위, 아래로 이동하는 할 때 사용한다. **절대 경로(Absolute Path)**는 이동하는 기준이 최상위 디렉토리(/)이며, root(/) 디렉토리를 기준으로 이동할 때 사용한다.

경로(PATH)
- 상대경로(Relative PATH) **# cd dir1**
- 절대경로(Absolute PATH) **# cd /dir1**

작업시에는 상대경로를 사용하여 이동하는 경우, 즉 작업 디렉토리안에 들어가서 직접 파일을 다루는 방법을 권장한다. 이것은 명령어 실수로 인해 불의의 사고를 예방할 수 있거나, 최소화 할 수 있기 때문이다.



(1) 상대경로(Relative PATH)

[EX1] 상대경로 사용하는 경우

```
# cd /etc
# pwd
```

```
/etc
```

```
# cd sysconfig
# cd network-scripts
# ls -a                                  /* .디렉토리 , ..디렉토리 */
```

```
./              ifdown-ipsec*   ifdown-tunnel*  ifup-ipx*      ifup-sl*
../             ifdown-ipv6*    ifup@           ifup-isdn@     ifup-tunnel*
ifcfg-eth0      ifdown-isdn@    ifup-aliases*   ifup-plip*     ifup-wireless*
ifcfg-lo        ifdown-post*    ifup-bnep*      ifup-plusb*    init.ipv6-global*
ifdown@         ifdown-ppp*     ifup-eth*       ifup-post*     net.hotplug*
ifdown-bnep*    ifdown-routes*  ifup-ippp*      ifup-ppp*      network-functions
ifdown-eth*     ifdown-sit*     ifup-ipsec*     ifup-routes*   network-functions-ipv6
ifdown-ippp*    ifdown-sl*      ifup-ipv6*      ifup-sit*
```

```
# cd .            /* 현재 디렉토리 (cd /etc/sysconfig/network-scripts/와 같은 뜻) */
# pwd
```

```
/etc/sysconfig/network-scripts/
```

```
# cd ..           /* 상위 디렉토리 (내 기준에서 상위 디렉토리로 이동) */
# pwd
```

```
/etc/sysconfig/
```

[참고] 디렉토리 표시
. 디렉토리는 현재 디렉토리
.. 디렉토리는 상위 디렉토리

```
# pwd
```

```
/etc/sysconfig/
```

```
# cd ../../       /* 상위 디렉토리, 상위 디렉토리 */
# pwd
```

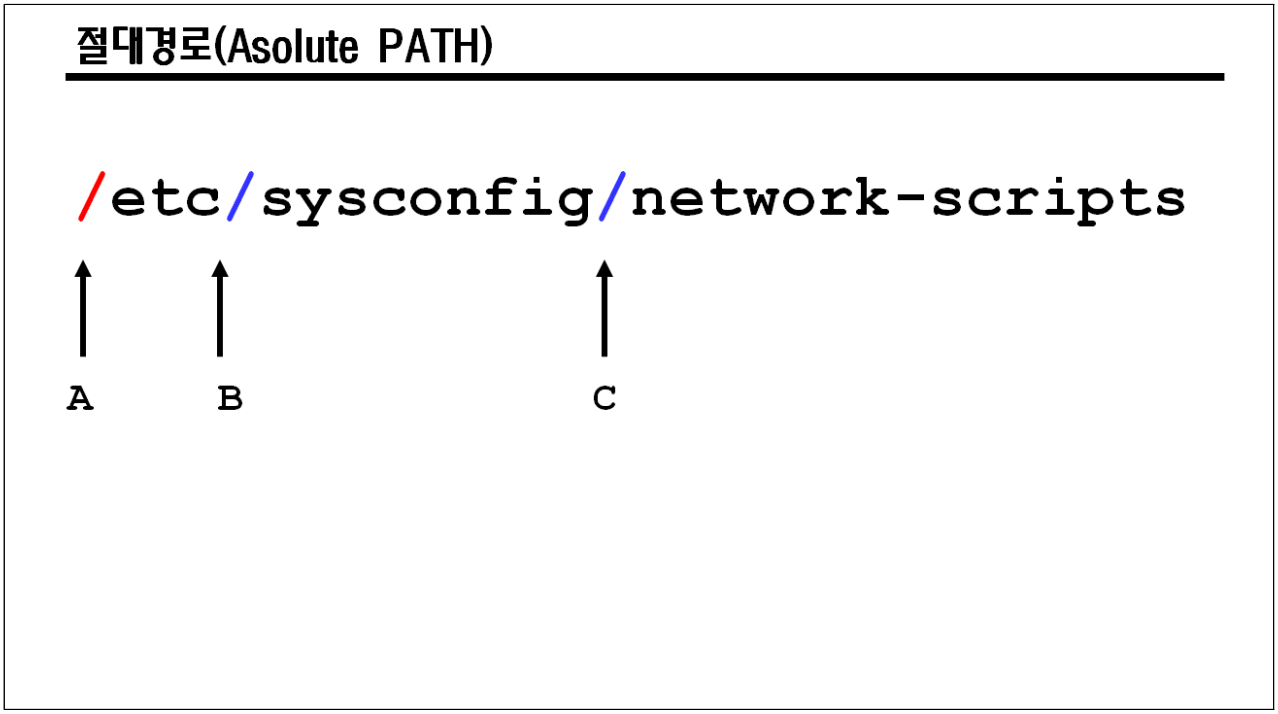
```
/
```

[EX2] 상위 디렉토리에 파일 만들기

```
# cd /etc/sysconfig/network-scripts
# touch ../../test.txt
# cd ../../
# pwd
```

```
/etc/
```

```
# ls -l test.txt
```



A는 최상위 디렉토리인 / 디렉토리를 뜻 한다.
B,C는 디렉토리의 구분자

처음 위치한 /는 최상위 디렉토리를 나타내지만 뒤에 나와 있는 /는 구분자 역할만 한다. 따라서 /etc/sysconfig/network-scripts 와 /etc/sysconfig/network-scripts/는 같은 뜻을 가지게 된다.

```
# cd /tmp , cd /tmp/
# cd /etc/sysconfig
# cd /usr
```

(3) 로그인 된 사용자의 홈 디렉토리로 이동

root 사용자로 로그인 하였기 때문에 root의 홈디렉토리로 이동 한다. 만약 fedora 사용자로 로그인 하였다면 /home/fedora 사용자로 이동하게 된다.

■ 사용자의 홈디렉토리
root 사용자 -> /root
일반 사용자 -> /home/$USER

(root 사용자인 경우)

```
# id
```

uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)

-> root 사용자인 경우

```
# cd                                    /* # cd ~ , # cd $HOME 과 동일하다. */
# pwd
```

/root

(fedora 사용자인 경우)

```
# ssh fedora@localhost
```

fedora 사용자로 로그인

```
$ id
```

uid=500(fedora) gid=500(fedora) groups=500(fedora)

```
$ cd
$ pwd
```

/home/fedora

```
$ exit
# cd ~fedora
```

[차이점]

```
# cd ~fedora
# cd ~/fedora
```

~fedora의 경우는 fedora의 홈 디렉터리 /home/fedora로 이동하는 경우의 명령어이며
~/fedora의 경우는 fedora의 홈 디렉터리로 이동하지 않고 현재 계정의 홈 디렉터리 하위의 fedora 디렉터리로 이동하는 /$HOME/fedora의 의미를 가진다.....

[참고]

```
$ cd
$ cd ~
$ cd $HOME
```

동일 하게 자신의 홈 디렉토리로 이동한다.

(4) 사용한 이전 디렉토리로 이동

[시나리오] 사용자가 /etc 디렉토리에서 작업을 하다가 /home/user01 디렉토리로 이동하는 경우 상대경로나 절대경로를 사용하는 경우는 불편하게 된다. 이런 경우 이전 디렉토리로 바로 이동하기 위해서 "cd - " 명령어를 사용한다.

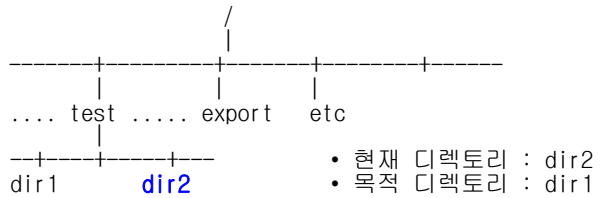
```
        /etc <------> /etc/sysconfig/network-scripts
```

```
# cd /etc
# ls


# cd /etc/sysconfig/network-scripts
# ls


# cd /etc
# ls


# cd -
# pwd <cd - 로 이동이후 현재 디렉토리를 확인한다.>
```

```
/etc/sysconfig/network-scripts
```

-> 반복적으로 사용해 본다.

(5) 옆에 있는 디렉토리로 이동하는 경우



- 현재 디렉토리 : dir2
- 목적 디렉토리 : dir1

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

# mkdir dir1 dir2
# cd dir2
# pwd
```

```
/test/dir2
```

```
# cd ..
# pwd
```

```
/test
```

```
# cd dir1
# pwd
```

```
/test/dir1
```

```
# cd ../dir2
# pwd
```

```
/test/dir2
```

```
# cd ../dir1
# pwd
```

```
/test/dir1
```

최상위 디렉터리에서부터 다시 이동하는 것보다는 근처에 있는 디렉터리라면 상대 경로를 통해 이동하는 것이 더 빠르게 이동할 수 있는 경우도 있다.

(실무 예) 프로그램 설치

```
/usr/local/apache2
        |
        +---- bin   (# cd ../bin)
        |
        +---- conf  (# cd ../conf)
        |
        +---- docs  (# cd ../docs)
        |
        +---- ....
```

디렉토리 관리 명령어

- **ls 명령어**

```
# ls -l dir1
# ls -ld dir1

OPTIONS : -a -l -d -t -F -r -I -R -h
```

- **mkdir 명령어**

```
# mkdir -p dir1/dir2/dir3
```

- **rmdir 명령어**

```
# rm -rf dir1
```

1 ls CMD

NAME

ls, dir, vdir - 경로의 내용을 나열한다.

SYNOPSIS

ls [-abcdfgiklmnpqrstuxABCFGLNQRSUX1] [-w cols] [-T cols] [-I pattern] [--all] [--escape] [--directory] [--inode] [--kilobytes] [--numeric-uid-gid] [--no-group] [--hide-control-chars] [--reverse] [--size] [--width=cols] [--tabsize=cols] [--almost-all] [--ignore-backups] [--classify] [--file-type] [--full-time] [--ignore=pattern] [--dereference] [--literal] [--quote-name] [--recursive] [--sort={none,time,size,extension}] [--format={long,verbose,commas,across,vertical,single-column}] [--time={atime,access,use,ctime,status}] [--help] [--version] [--color[={yes,no,tty}]] [--colour[={yes,no,tty}]] [name...]

DESCRIPTION

이 문서는 더이상 최신 정보를 담고 있지않다. 그래서, 몇몇틀릴 경우 도 있고, 부족한 경우도 있을 것이다. 완전한매뉴얼을 원하면, Texinfo 문서를 참조하기 바란다.

이 매뉴얼 페이지는 ls 명령의 GNU 버전에 대한 것이다. dir과 vdir 명령 은 ls 명령의 심블릭 파일로그 출력 양식을 다르게 보여주는 풀그림들이다. 인자로 파일이름이나, 경로 이름이 사용된다. 경로의 내용은 초기값으로 알파벳 순으로 나열된다. ls의 경우는 출력이 표준 출력(터미날 화면)이면, 세로로 정열된 것이 가로로 나열된다. 다른 방식의 출력이면 한줄에 하나씩 나열된다. dir의 경우는, 초기값으로 ls와 같으나, 모든 출력에서세로로 정열해서 가로로 나열한다.(다른 방식의 출력에서도 항상 같음) vdir 의 경우는, 초기값으로 목록을 자세히 나열한다.

OPTIONS

-a, --all
경 로안의 모든 파일을 나열한다. '.'으로 시작하는 파일들도 포함된다.

-d, --directory
경 로안의 내용을 나열하지 않고, 그 경로를 보여준다.(이것은 쉘 스크립트에서 유용하게 쓰인다.)

-i, --inode
파일 왼쪽에 색인 번호를 보여준다.

-r, --reverse
정렬 순서를 내림차순으로 한다.

-t, --sort=time
파일 시간 순으로 정열한다. 최근 파일이 제일 먼저. (access time)

-u, --time=atime, --time=access, --time=use
파일 사용 시간 순으로 정열한다. 자세한 나열이면, 시간 표시는 만들어진 날자대신, 사용된 날자를 보여준다.

-F, --classify
파 일 형식을 알리는 문자를 각 파일 뒤에 추가한다. 일반적으로 실행파일은 "*", 경로는 "/", 심블릭링크는 "@", FIFO는 "|", 소 켓 은 "=", 일반적인파일은 없다.

-R, --recursive
하위 경로와 그 안에 있는 모든 파일들도 나열한다.

--color, --colour, --color=yes, --colour=yes
파 일 의 형태에 따라 그 파일의 색깔을 다르게 보여주는 기능한다. 자세한 이야기는 아래 DISPLAY COLORIZATION 부분을 참조한다.

--color=tty, --colour=tty
--color 옵션과 같으나, 단지 표준 출력에서만 색깔을 사용한다. 이 옵 션 은 칼라 제어 코드를 지원하지않는 보기 풀그림을 사용하는 쉘 스크립트나, 명령행 사용에서아주 유용하게 쓰인다.

--color=no, --colour=no
색깔 사용하지 않는다. 이것이 초기값이다. 이옵션은 색깔 사 용 을 이미 하고 있다면, 이 값을 무시한다.

디렉토리에 있는 내용을 확인하고자 할 때 (ls 명령에 대해서 확인 : # man ls)

[명령어 형식]

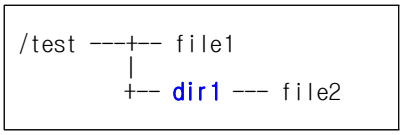
```
# ls -l
# ls -l dir1

# ls -ld
# ls -ld dir1

# ls -altr
```

[명령어 옵션]

옵션	설명
-a	모든 파일 표시, 여기에는 숨김 파일(점(.)으로 시작하는 파일)도 표시한다.
-l	디렉토리가 지정되는 경우 디렉토리의 내용을 자세히 보여준다. 또한 파일의 내용 이 지정되는 경우 파일의 속성 정보를 자세히 보여준다. 파일 종류, 링크 수, 소유자명, 그룹명, 파일 크기, 최종 수정일 및 용량, 파일명 표시한다.
-R	해당 디렉토리와 서브디렉토리의 모든 내용을 표시
-F	디렉토리인 경우에는 디렉토리 "/"표시를 하고 실행 파일인 경우는 뒤에 "*"표시를 한다.
-i	해당 파일의 inode 번호를 표시한다.
-n	파일의 소유자와 그룹을 숫자로(UID : GID)표시한다.
-d	찾고자 하는 디렉토리에 관한 정보만을 표시한다.

[EX1] ls 명령어의 기본 사용법 테스트

(실습용 구조)



(실습 준비)

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

# touch file1
# mkdir dir1
# touch dir1/file2
```

 # ls -IR (# find .)

```
.:
합계 4
drwxr-xr-x. 2 root root 4096 2015-08-12 15:37 dir1
-rw-r--r--. 1 root root    0 2015-08-12 15:36 file1

./dir1:
합계 0
-rw-r--r--. 1 root root 0 2015-08-12 15:37 file2
```

(ls -l 출력 **결과 해석**)

```
-                파일의 종류(File Type), -(일반파일), d(디렉토리파일)
rw-r--r--        퍼미션 모드(Permission Mode)
1                링크 수(Hard Link Count)
root             소유자(Owner)
root             그룹(Group)
0                파일의 크기(File Size), 기본 단위 : bytes
8월 15 20:37     수정 또는 생성 시간(Mtime: Modify Time)
file1            파일의 이름(File Name)
```

 # ls -ld /* 현 디렉토리 정보를 자세히 출력(.디렉토리 출력) */

```
drwxr-xr-x 3 root root 4096  8월 15 20:37 .
```

 # ls -l dir1 /* dir1디렉토리의 내용을 출력 */

```
-rw-r--r-- 1 root root 0  8월 15 20:38 file2
```

 # ls -ld dir1 /* dir1디렉토리의 정보를 출력 */

```
drwxr-xr-x 2 root root 4096  8월 15 20:38 dir1
```

 # ls -lR /test /* -R: Recursive, 하위 디렉토리까지 */

```
/test:   <해당하는 디렉토리명을 출력한뒤 디렉토리 내부의 파일과 디렉토리를 출력한다.>
합계 4
drwxr-xr-x 2 root root 4096  8월 15  2010 dir1
-rw-r--r-- 1 root root    0  8월 15  2010 file1

/test/dir1:
합계 0
-rw-r--r-- 1 root root 0  8월 15  2010 file2
```

[EX2] "ls -a" 옵션 실습
```
# cd
# pwd
```
```
/root
```

- 일반파일 출력

```
# ls -l
```
```
drwxr-xr-x 2 root root  4096  4월  6 12:17 Desktop
-rw------- 1 root root  4210  1월 25 22:54 anaconda-ks.cfg
-rw-r--r-- 1 root root     0  4월  7 12:38 file_0407.log
-rw-r--r-- 1 root root 54631  1월 25 22:54 install.log
-rw-r--r-- 1 root root  9641  1월 25 22:50 install.log.syslog
```

- 숨김파일과 일반파일 출력

```
# ls -al
```
```
drwxr-x--- 18 root root  4096  8월 15  2010 .
drwxr-xr-x 26 root root  4096  8월 15  2010 ..
-rw-------  1 root root  2395  8월 15  2010 .bash_history
-rw-r--r--  1 root root    24  1월  6  2007 .bash_logout
-rw-r--r--  1 root root   231  4월  7 10:44 .bash_profile
-rw-r--r--  1 root root   223  4월  7 10:33 .bashrc
..... (중략) .....
drwxr-xr-x  2 root root  4096  4월  6 12:17 Desktop
-rw-------  1 root root  4210  1월 25 22:54 anaconda-ks.cfg
-rw-r--r--  1 root root     0  4월  7 12:38 file_0407.log
-rw-r--r--  1 root root 54631  1월 25 22:54 install.log
-rw-r--r--  1 root root  9641  1월 25 22:50 install.log.syslog
```

[EX3] "ls -F" 옵션 실습
```
# cd /test
# rm -rf /test/*

# cp /etc/passwd file1    /* 일반 파일 */
# ln -s file1 file2       /* 링크 파일 */
# cp /bin/ls file3        /* 실행 파일 */
# mkdir dir1              /* 디렉토리 파일 */

# ls -F   (dir1/  file1  file2@  file3*)
# ls      (dir1   file1  file2   file3 )
```

[참고]
리눅스의 경우는 파일, 디렉터리등 종류가 다르다면 다른 색깔로 표시를 해주기 때문에 -F의 경우 그렇게 사용량이 많지는 않다. 하지만 다른 UNIX계열의 OS는 동일하게 표시되기 때문에 알아두자.

[EX4] 'ls -i' 옵션 실습

-i는 Index NODE인 inode를 표시해주기 위한 옵션이다. 파일의 속성정보를 담고있는 inode의 구분번호를 표시해 주고 있다.ls -l로 명령어를 입력한경우의 출력되는 결과들이 저장되어 있다. 데이터는 Data Block이라는 공간에 별도로 저장되어있다.
```
# ls -li /test/file1              /* inode 번호까지 확인 */
```
```
2606085 -rw-r--r-- 1 root root 0  8월 15  2010 /test/file1
```
```
# ls -ldi /test/dir1              /* inode는 각각 고유한 값을 지님 */
```
```
4312752 drwxr-xr-x 2 root root 4.0K Jan  8 10:18 /test/dir1/
```
inode - 파일과 디렉토리에 지정되어 있는 번호

[EX5] 파일 또는 디렉토리만 출력

alias(별칭) 명령어를 간편하게 사용하기 위해 사용한다
(선언) # alias ls='ls -h --color=tty'
(확인) # alias
(해제) # unalias ls

```
# alias lsf='ls -l | grep "^-"'    /* 파일인 경우 속성 정보에 -로 표시 */
# alias lsd='ls -l | grep "^d"'    /* 디렉토리인 경우 속성 정보에 d로 표시 */

# cd
# lsf
```
```
-rw------- 1 root root  4210 Jan 25 22:54 anaconda-ks.cfg
-rw-r--r-- 1 root root 54631 Jan 25 22:54 install.log
-rw-r--r-- 1 root root  9641 Jan 25 22:50 install.log.syslog
-rw-r--r-- 1 root root     0 Jan 26 07:34 local-host-names
-rw------- 1 root root 11665 Jan 26 07:32 mbox
```

lsd

```
drwxr-xr-x 2 root root  4096 Jan 26 02:13 Desktop
```

(참고) 선언된 alias 확인
alias alias의 전체목록을 확인
alias lsf 인자값으로 생성되어 있는 alias가 있다면 출력

```
# alias lsf                          /* lsf의 alias의 정보 확인 */
alias lsf='ls -l | grep "^-"'

# alias lsd                          /* lsd의 alias의 정보 확인 */
alias lsd='ls -l | grep "^d"'
```

[EX6] ls -h 옵션 설정(-h : human)
■ alias를 지정하지 않은 경우
```
# ls -l /etc/services
-rw-r--r-- 1 root root 362031 Feb 23  2006 /etc/services
```
이전과 달리 현재는 byte 단위로 용량을 보게 된다면 단위가 너무 많기 때문에 힘들다. 용량이 정확해야 하지 않는다면 -h 옵션을 사용하여 사람이 보기 쉬운형태로 변환하여 사용한다.

```
# vi ~/.bashrc     (# gedit ~/.bashrc)
..... (중략) .....
#
# Sfecific Configuration
#
alias ls='ls --color=tty -h'

# . ~/.bashrc     (# source ~/.bashrc)
```

■ alias를 지정한 경우
```
# ls -l /etc/services
-rw-r--r-- 1 root root 354K Feb 23  2006 /etc/services
```
보기 편한 형태로 용량이 변경되었다.

[EX7] "ls -altr" 명령어 실습
```
# ls -altr /tmp    /* -t : time sort, -r : reverse sort */
```
이전파일이 상단에 출력된다.

```
        # cd /Log_dir
        # ls -altr
```

2 mkdir CMD

NAME
mkdir - 경로 만들기

SYNOPSIS
mkdir [-p] [-m mode] [--parents] [--mode=mode] [--help] [--version] dir...

DESCRIPTION
이 문서는 더이상 최신 정보를 담고 있지않다. 그래서, 몇몇틀릴 경우 도 있고, 부족한 경우도 있을 것이다. 완전한매뉴얼을 원하면, Texinfo 문서를 참조하기 바란다.

이 매뉴얼 페이지는 mkdir 명령의 GNU 버전에 대한 것이다. mkdir 명 령은 주워진 이름으로 경로를 만든다. 초기값으로는 그 만들어지는 경로의 모드는 0777이다.

OPTIONS
-m, --mode mode
mode로 사용할 것은 chmod(1)에서 사용하는 기호형식이나, 숫자형 식이며, 이 값은 초기값으로 지정되는 모드를 무시한다.

-p, --parents
필 요 한 경 우 에 상위 경로까지 만든다. 이 말은 가령 'mkdir ~/dest/dir1' 명령을 사용했을 때, 만약 '~/dest' 경로가 없다면 오료 메시지를 보인다. 하지만, 이 옵션을 사용하면, '~/dest' 경로를 만들고, 그 다음, 그 안에서 'dir1' 경로를 만든다. 만들어 지는 상위경로의 모드값은 'u+wx'이다.

새로운 디렉토리를 생성하며, 빈 디렉토리를 생성한다. 옵션을 통하여 여러개의 디렉토리를 한꺼번에 생성 할 수도 있다.

[명령어 형식]

```
# mkdir dir1                    /* 현 디렉토리에 dir1 디렉토리 1개 생성 */
# mkdir dir1 dir2               /* 현 디렉토리에 dir1, dir2 디렉토리 2개 생성 */
# mkdir -p dir3/dir2/dir1       /* dir3 디렉토리 안에 dir2를 생성하고 dir2 안에 dir1을 생성 */
```

[명령어 옵션]

옵션	설명
-m	디렉토리의 퍼미션 권한을 지정 (기본값 : 755)
-p	디렉토리 경로로 생성 (디렉토리를 만들 때 상위 디렉토리가 없을시 상위 디렉토리까지 생성)

[EX1] mkdir -p 옵션 사용법

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

```
/test
```

```
# mkdir dir4
# ls -l
```

```
drwxr-xr-x 2 root root 4096 Jan 26 09:20 dir4
```

```
# mkdir dir4/dir2/dir1
```

```
mkdir: cannot create directory `dir4/dir2/dir1': No such file or directory
```

```
# mkdir -p dir4/dir2/dir1
# ls -R  (# find .)
-> 출력 내용 생략
```

3 rmdir CMD

NAME
rmdir - 비어 있는 경로를 지운다.

SYNOPSIS
rmdir [-p] [--parents] [--help] [--version] dir...

DESCRIPTION
이 문서는 더이상 최신 정보를 담고 있지않다. 그래서, 몇몇 틀릴 경우도 있고, 부족한 경우도 있을 것이다. 완전한 매뉴얼을 원하면, Texinfo 문서 를 참조하기 바란다.

이 매뉴얼 페이지는 rmdir 명령의 GNU 버전에 대한 것이다. rmdir 명령은 비어있는 경로른 지운다. 아무 옵션 없이 사용되면 오류 번호를 돌려주 고, 종료된다.

OPTIONS
-p, --parents
상위 경로도 지운다. 이명령은 그 상위 경로 안에 물론 비어있어야지 가능하다.

만약 비어 있지 않으나, 그 경로와 그 안에 포함된 모든 파일과 하위 경로들을모조리 지우고자 한다면 rm -r 옵션을 사용한다.

빈 디렉토리를 삭제하는 명령어로, 여러 개의 디렉토리를 동시에 삭제 할 수 있다. 단, 디렉토리가 비어 있지 않으면 삭제 불가능 하다.

[명령어 형식]

```
# rmdir dir1                        /* dir1 디렉토리 1개 삭제 */
# rmdir dir1 dir2                   /* dir1, dir2 디렉토리 2개 삭제 */
# rmdir -p dir4/dir2/dir1           /* 경로에 포함되어 있는 디렉토리 삭제 (비워있어야 함) */
```

[명령어 옵션]

옵션	설명
-p	하위항목을 같이 지움 (조건 : 하위항목도 비워 있어야 함)

[참고] 비어 있지 않는 디렉토리 삭제

```
# rm -rf dir1       /* -r : recursive, -f : force */
```

파일 관리 명령어

- **touch 명령어**

```
# touch -t 08301300 file1
```

- **cp 명령어**

```
# cp file1 file2
# cp file1 dir1
# cp -r dir1 dir2
```

- **mv 명령어**

```
# mv file1 file2
# mv file1 dir2
# mv dir1 dir2
```

- **rm 명령어**

```
# rm -rf dir1
```

1 touch CMD

NAME
touch - 파일의 시간 정보를 바꾼다.

SYNOPSIS
touch [-acfm] [-r file] [-t MMDDhhmm[[CC]YY][.ss]] [-d time] [--time={atime,access,use,mtime,modify}] [--date=time] [--reference=file] [--no-create] [--help] [--version] file...

DESCRIPTION
이 문서는 더이상 최신 정보를 담고 있지않다. 그래서, 몇몇 틀릴 경우도 있고, 부족한 경우도 있을 것이다. 완전한 매뉴얼을 원하면, Texinfo 문서 를 참조하기 바란다.

이 매뉴얼 페이지는 touch 명령의 GNU 버전에 대한 것이다. touch 명령은 주어진 파일의 최근 사용시간과 최근 변경 시간(파일 내용이 바뀐 시간)을시스 템 의 현재 시간으로 바꾼다. 이때 그런 파일이 없으면 0 바이트 크기의 이름만 있는 파일을 만든다. 만약, 첫번째 오는 filename에 해서 -t 옵션이 사 용되고 나머지 filename에 대해서 특별히 -d, -r, -t 옵션이 사용되고 있지 않다면, 나머지 모든 파일들은그 첫번째 파일에서 사용할 시간값을 그 대로 사용하게 된다. 이런 것을 방지하려면, -- 옵션을 사용해서 각각의 옵션들을 분리 시킨다.

If changing both the access and modification times to the current time, touch can change the timestamps for files that the user running it does not own but has write permission for. Otherwise, the user must own the files.

OPTIONS
-d, --date time
현재 시간 대신 지정한 time 값을 사용하는데, 이것은 다양한 형식이 될 수 있다. 이 시간에는 월 이름, 지역, 'am', 'pm' 등이 포 함 될 수도 있다.

-m, --time=mtime, --time=modify
최근 파일 변경 시간(modify time)만 바꾼다.

-t MMDDhhmm[[CC]YY][.ss]
현 재 시간 대신 지정한 시간(MM:달, DD:날, hh:시, mm:분, [CC]YY:년, SS:초)으로바꾼다.

파일의 이름을 지정하일이름을 지정하였다면여 기존에 존재하지 않는 파 빈 파일을 만들어주고 기존에 존재했다면 지정된 파일이나 디렉토리의 수정시간(mtime, Modify Time)이나 접근시간(atime, Access Time)등을 현재 시간으로 업데이트 시켜준다.

만약 touch 명령어에 -t 옵션을 사용 하여 파일이나 디렉토리의 수정시간을 특정한 시간으로 변경 가능하다.
(해커가 침입하여 파일을 수정하고 아래의 옵션을 통해 수정과 접근 시간을 바꿔놓을 수도 있다.)

[명령어 형식]

```
# touch file2              /* file2 파일 1개 생성 */
# touch file1 file2        /* file1, file2 파일 2개 생성 */
# touch -t 08081230 file1  /* file1 수정 시간 변경(월,일,시,분) */
```

[명령어 옵션]

옵션	설명
-a	최근 파일 사용기간 만 변경
-c	파일을 생성하지 않는 명령어
-d [시간]	현재 시간 대신 지정한 시간(시분)으로 변경
-m	최근 파일 변경 시간만 변경 (파일 수정시간)
-r [파일]	현재 시간 대신 지정한 파일의 시간으로 변경
-t MMDDhhmm [[CC]YY][.SS]	현재 시간 대신 지정한 시간(월일시분)으로 변경

[EX1] 빈 파일 생성

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

# touch file1
# ls -l file1
-rw-r--r-- 1 root root 0  8월 11 12:30 file1
```

-> 파일의 크기 0

[EX2] 파일의 생성 시간 현재시간으로 변경

```
# cp -p /etc/passwd file2
# ls -l file2
-rw-r--r-- 1 root root 2.0K Jan  7 15:52 file2
```

-> 시간 정보를 확인한다.

```
# touch file2
# ls -l file2
-rw-r--r-- 1 root root 2.0K Jan  8 11:08 file2
```

-> 시간 정보를 확인한다.

```
# date
Wed Jan  8 11:08:40 KST 2014
```

-> 현재 시간을 확인한다.

[EX3] touch -t 옵션 사용

```
# touch -t 08301300 file2
# ls -l file2
-rw-r--r-- 1 root root 2.0K Aug 30  2014 file2
```

2 cp CMD

NAME
cp - 파일 복사

SYNOPSIS
cp [options] source dest
cp [options] source... directory

DESCRIPTION
이 문서는 더이상 최신 정보를 담고 있지않다. 그래서, 몇몇틀릴 경우 도 있고, 부족한 경우도 있을 것이다. 완전한매뉴얼을 원하면, Texinfo 문서를 참조하기 바란다.

이 매뉴얼 페이지는 cp 명령의 GNU 버전에 대한 것이다. 마지막 명령 행 인자로 경로가 지정되면, cp 명령은 지정한 source 파일들을 그 경로로 안으로 복사한다. 한편 명령행 인자로 두개의 파일 이름이 사용되면, 첫번째 파일 을 두번째 파일로 복사한다. 마지막 명령행 인자가 경로가 아니고, 두개 이상의 파일이 지정되면, 오류 메시지를 보여준다. 초기값으로 경로는 복 사하지 않는다.

OPTIONS
-a, --archive
원 본 파일의 속성, 링크 정보들을 그대로 유지하면서 복사한다. 이 옵션은 -dpR 옵션과 같은 역활을 한다.

-i, --interactive
만약 복사 대상 파일 이미 있으면 사용자에게 어떻게 처리 할 것인지 물어보는프롬프트를 나타나게 한다.

-p, --preserve
원본 파일의 소유주, 그룹, 권한, 시간정보들이 그대로 보존되어 복사된다.

-r 일반 파일이면, 그냥 복사되고, 만약 원본이 경로면, 그 경로와 함께 경로 안에 있는 모든 하위경로, 파일들이 복사된다.

-R, --recursive
경로를 복사할 경우에는 그 안에 포함된 모든 하위경로와 파일들을모두 복사한다.

파일이나 디렉토리의 내용을 다른 파일 또는 다른 디렉토리에 복사 할 때 사용. 파일을 복사하는 것은 물리적으로 새로운 파일을 하나 생성하며 새로운 파일의 이름과 새로운 inode, 복사된 데이터 블록을 가지게 된다.

[명령어 형식]

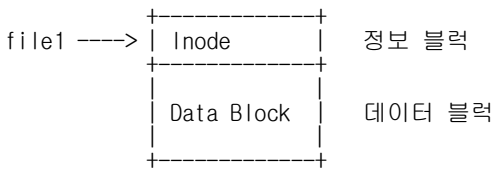
```
# cp file1 file2          /* file1 파일내용을 file2로 생성 */
# cp file1 dir1           /* file1 파일내용을 dir1디렉토리에 file1 생성 */
# cp -r dir1 dir2         /* dir1 디렉토리를 dir2디렉토리로 생성 */
```

[명령어 옵션]

옵션	내용
-a	원본 파일의 속성, 링크 정보를 유지 하면서 복사
-b	복사할 대상을 덮어쓰거나 지울 때를 대비해서 백업 파일 만듦 백업파일의 파일명 뒤에는 ~가 표시된다
-d	심볼릭 파일 자체를 심볼릭 정보와 함께 복사할 때 사용
-f	복사할 파일이 존재할 때 삭제하고 복사
-i	복사할 파일이 존재하는 경우 복사할 것인지 물어봄
-l	디렉토리가 아닌 경우 복사 대신 하드 링크로 만듦
-p	원본 파일의 소유, 그룹, 권한, 허용 시간을 보존한 채로 복사
-r	서브 디렉토리 내에 있는 모든 파일까지 통째로 복사
-s	디렉토리가 아닌 경우 복사 대신 심볼릭 링크로 만듦
-u	대상 파일보다 원본 파일이 새로운 것일 때 복사
-v	복사 상태를 보여줌

[EX1] 파일에 대한 inode를 확인

■ 파일의 일반적인 구조(EX: 일반 파일)



■ Inode Number ?

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

# touch file1
# ls -li file1
```

```
4039445 -rw-r--r-- 1 root root 0 Jan 26 11:34 file1
```

```
# cp file1 file2
# ls -li file2                    /* 복사하면서 inode가 바뀐 걸 볼 수 있음. */
```

```
4039452 -rw-r--r-- 1 root root 0 Jan 26 11:35 file2
```

[EX2] cp 명령어의 -r 옵션

```
# cd /test
# rm -r /test/*

# mkdir dir1
# touch dir1/file1 dir1/file2

# cp -r dir1 dir2
# ls -lR
```

```
drwxr-xr-x 2 root root 4096 Jan 26 11:36 dir1
drwxr-xr-x 2 root root 4096 Jan 26 11:37 dir2

./dir1:
total 0
-rw-r--r-- 1 root root 0 Jan 26 11:36 file1
-rw-r--r-- 1 root root 0 Jan 26 11:36 file2

./dir2:
total 0
-rw-r--r-- 1 root root 0 Jan 26 11:37 file1
-rw-r--r-- 1 root root 0 Jan 26 11:37 file2
```

```
# cp -r dir1 dir2
# ls -lR
        # cp -r /home/fedora /tmp
        # cp -r /test /tmp
```

[EX3] cp를 이용하여 같은 파일에 덮어 쓰기(Overwrite)하는 경우

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

# mkdir dir1
# echo "linux200" > file1
# cat file1
```

```
linux200
```

```
# touch dir1/file1                /* 동일한 파일명으로 파일 생성 */
# cat dir1/file1                  /* 이름만 동일 할 뿐 내용 다름 */
#

# cp file1 dir1                   /* 동일한 파일명에 덮어쓰기 */
```

```
cp: overwrite `dir1/file1'? y
```

```
# ls -lR
.:
total 8
drwxr-xr-x 2 root root 4096 Jan 26 23:39 dir1
-rw-r--r-- 1 root root    9 Jan 26 23:39 file1

./dir1:
total 4
-rw-r--r-- 1 root root 9 Jan 26 23:40 file1

# cat dir1/file1          /* copy 내용 확인 가능 */
linux200
```

[EX4] 원본 파일의 소유, 그룹, 권한, 허용 시간을 보존한 채로 복사(cp -p)

```
# touch file1
# chmod 777 file1                  /* 파일에 대한 퍼미션 변경 */
# ls -l
-rwxrwxrwx 1 root root 0  5월 11 20:30 file1

# cp file1 file2
# ls -l
-rwxrwxrwx 1 root root 0  5월 11 20:30 file1
-rwxr-xr-x 1 root root 0  5월 11 20:31 file2

# cp -p file1 file3
# ls -l                            /* 퍼미션 값과 생성 시간등을 그대로 이어 받아오고 있다 */
-rwxrwxrwx 1 root root 0  5월 11 20:30 file1
-rwxr-xr-x 1 root root 0  5월 11 20:31 file2
-rwxrwxrwx 1 root root 0  5월 11 20:30 file3

# chmod 777 /test                  /* 다른 사용자들이 접근하여 파일을 생성 할 수 있도록 퍼미션 변경 */
# su - fedora
$ id
uid=500(fedora) gid=500(fedora) groups=500(fedora)

$ pwd
/home/fedora

$ cd /test
$ cp file1 file4
$ ls -l
-rwxrwxrwx 1 root   root   0  5월 11 20:30 file1
-rwxr-xr-x 1 root   root   0  5월 11 20:31 file2
-rwxrwxrwx 1 root   root   0  5월 11 20:30 file3
-rwxrwxr-x 1 fedora fedora 0  5월 11 20:32 file4

$ cp -p file1 file5
$ ls -l
합계 0
-rwxrwxrwx 1 root   root   0  5월 11 20:30 file1
-rwxr-xr-x 1 root   root   0  5월 11 20:31 file2
-rwxrwxrwx 1 root   root   0  5월 11 20:30 file3
-rwxrwxr-x 1 fedora fedora 0  5월 11 20:32 file4
-rwxrwxrwx 1 fedora fedora 0  5월 11 20:30 file5

$ exit
#
```

(실무 예) 로그 파일(EX: file.log) 비우기

```
        # cp /dev/null file.log
        # cat /dev/null > file.log
        # > file.log

# cd /test
# cp /var/log/messages file.log
# cp /dev/null file.log
-> y
# ls -l file.log
-> 파일의 크기 '0' 확인
```

3 mv CMD

```
NAME
       mv - 파일 옮기기(rename)

SYNOPSIS
       mv [options] source dest
       mv [options] source... directory
       Options:
       [-bfiuv]  [-S backup-suffix] [-V {numbered,existing,simple}] [--backup]
       [--force] [--interactive] [--update] [--verbose]  [--suffix=backup-suf-
       fix]  [--version-control={numbered,existing,simple}]  [--help]  [--ver-
       sion]

DESCRIPTION
       이 문서는 더이상 최신 정보를 담고 있지않다. 그래서, 몇몇 틀릴 경우도 있
       고,  부족한 경우도 있을 것이다.  완전한 매뉴얼을 원하면, Texinfo 문서를
       참조하기 바란다.

       이 매뉴얼 페이지는 mv 명령의 GNU 버전에 대한 것이다.  마지막 인자로  경
       로   이름이 사용되면, 원본 파일을 그 경로로 이름 똑같게 이동시킨다.  반
       면, 마지막 인자가 파일이름이면, 그 이름으로 바꾼다.  마지막 인자가 경로
       이 름이 아니거나, 원본과 대상이 파일 이름이아닌 경우는 오류 메시지를 보
       인다.  mv 명령은 파일 시스템에 접근이 가능할 때만 사용될 수 있다.  ( 일
       반 사용자는 DOS 파티션의 파일 시스템과 자신의 홈 경로 외에는읽기 접근을
       뺀 나머지는 불가능하다. 이런데, 바로 이런 파일시스템으로 파일을 옮기 고
       자 한다면, mv 명령은 오류 메시지를 낸다.)

       만 약에 파일 모드가 읽기 전용이고, 표준 입력이 tty이고, -f나 --force 옵
       션이 지정되지 않으면, mv 명령은 사용자에게 지정한 파일을 정말 지울것 인
       지 물어본다.  이때, 'y'나 'Y'를 입력해 주어야지만 그 파일을 옮긴다.

   OPTIONS
       -f, --force
              대상 파일이 이미 있어도 사용자에게 어떻게 처리할지를 묻지  않 는
              다.

       -i, --interactive
              대 상  파일이 이미 있어, 사용자에게 어떻게 처리할지를 물어 본다.
              이때, 'y'나 'Y'를 입력해 주어야지만 그 파일을 옮긴다.  (초기값)
```

파일과 디렉토리의 내용을 다른 파일 또는 다른 디렉토리로 옮길 때 사용하며 파일의 이름이나 디렉토리의 이름을 바꿀 수 있음. 같은 파티션 안에서 파일을 옮긴다는 것은 물리적으로 파일 이름만 변경하며, Inode 정보나 데이터 블록은 그대로 유지가 되고 다른 파티션으로 파일을 옮기는 경우는 새로운 파일 이름과 Inode, 데이터 블록을 할당 받게 됨.

[명령어 형식]

```
# mv file1 file2                    /* file1 파일이 이름이 file2로 변함 */
# mv file1 dir1                     /* file1 파일이 dir1 디렉토리에 하위경로로 이동 */
# mv dir1 dir2                      /* dir1 디렉토리가 dir2 디렉토리에 하위경로로 이동 */
```

[명령어 옵션]

옵션	내용
-b	복사할 대상을 덮어쓰거나 지울 때를 대비해서 백업 파일 만듦
-f	복사할 파일이 존재할 때 삭제하고 복사
-i	복사할 파일이 존재하는 경우 복사할 것인지 물어봄
-u	대상 파일보다 원본 파일이 새로운 것일 때 복사
-v	파일 옮기기 전의 과정을 보여 줌

[EX1] mv 명령어 사용법

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

# touch file1
# mkdir dir1
# touch dir1/file2
```

```
/test ---+--- file1
         |
         +--- dir1-----file2
```

```
# mv file1 file3
```

```
/test ---+--- file3
         |
         +--- dir1-----file2
```

```
# mv file3 dir1
```

```
/test --- dir1---+-----file2
                 |
                 +-----file3
```

```
# mv dir1 dir2
```

```
/test --- dir2---+-----file2
                 |
                 +-----file3
```

```
# cp -r dir2 dir1
```

```
/test -+- dir2---+-----file2
       |         |
       |         +-----file3
       +- dir1---+-----file2
                 |
                 +-----file3
```

```
# mv dir1 dir2
```

```
/test --- dir2 ---+---- file2
                  |
                  +---- file3
                  |
                  +--- dir1 --+-- file2
                              |
                              +-- file3
```

[EX2] Inode 확인

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

# touch file1
# ls -li file1                    /* Inode 확인 */
```

```
97774 -rw-r--r-- 1 root root 0 Jan 26 23:57 file1
```

```
# mv file1 file3
# ls -li file3                    /* Inode 가 변하지 않은 걸 알 수 있음 */
```

```
97774 -rw-r--r-- 1 root root 0 Jan 26 23:57 file3
```

단지 파일의 이름정보만 변경되었다.

[EX3] 여러 개의 파일을 동시에 이동

```
# cd /test
# touch file1 file2 file4
# ls
```

```
file1 file2 file3 file4
```

```
# mkdir dir1
# mv file* dir1                   /* file이라는 파일들을 dir1 디렉토리로 이동 */
# ls dir1
```

```
file1 file2 file3 file4
```

4 rm CMD

```
NAME
       rm - 파일 지우기

SYNOPSIS
       rm  [-dfirvR]  [--directory]  [--force]  [--interactive]  [--recursive]
       [--help] [--version] [--verbose] name...

DESCRIPTION
       이 문서는 더이상 최신 정보를 담고 있지않다.  그래서,  몇몇틀릴  경우 도
       있고,  부족한  경우도  있을  것이다.  완전한매뉴얼을 원하면, Texinfo 문
       서를 참조하기 바란다.

       이 매뉴얼 페이지는 rm 명령의 GNU  버전에  대한  것이다.  rm 명령은 지정
       한 파일을 지운다. 초기값으로는 경로는 지우지 않는다.

       만 약에 파일 모드가 읽기 전용이고, 표준 입력이 tty이고, -f나 --force 옵
       션이 지정되지 않으면, rm 명령은 사용자에게 지정한 파일을 정말 지울것 인
       지 물어본다.  이때, 'y'나 'Y'를 입력해 주어야지만 그 파일을 지운다.

       GNU  rm 명령과 같이 getopt(3) 함수를 사용하는 모든 풀그림에서는 -- 옵션
       다음에 오는 것은 옵션이 아닌 것으로 인식한다.  즉 파일 이름이 '-f' 라는
       파일을 지우고자 한다면, 다음 두 방법을 사용한다.
              rm -- -f
       또는
              rm ./-f
       유 닉스 rm 명령의 '-' 문자로 시작하는 옵션들 때문에이런 기능들이 고안된
       것이다.

   OPTIONS
       -f, --force
              지 울 파일이 없을 경우에 아무런 메시지를 보여주지 않고 그냥 넘어
              간다.  이 옵션은 쉘 스크립트 안에서 사용될 때 유용하게 쓰인다.

       -i, --interactive
              각 파일을 하나씩 지울 것인지 사용자에게 일일이 물어본다.   이 때
              'y' 나 'Y'를 눌러야지만 파일이 지워지다.

       -r, -R, --recursive
              일반 파일이면 그냥 지우고, 경로면, 그 하위 경로와 파일을 모두 지
              운다.
```

파일과 디렉토리를 지우고자 할 때 사용하며 한꺼번에 여러 개를 지울 수도 있으며 지운 파일들은 되살릴 수 없으므로 주위 해서 사용해야 하는데 -i 옵션을 사용하면 한번 더 묻게 되므로 부주의로 인한 파일 삭제를 막을 수 있으며, 옵션 -r를 사용 시 시스템의 모든 파일이 삭제되는 경우도 있으니 신중하게 사용해야 한다.

[명령어 형식]

```
# rm file1                          /* file1 파일 1개 삭제 */
# rm file1 file2                    /* file1, file2 파일 2개 삭제 */
# rm -r dir1                        /* dir1 디렉토리 하위경로까지 삭제 */
```

[명령어 옵션]

옵션	설명
-f	강제로 파일을 지우고 삭제할 파일이 없을 경우에도 아무런 메시지를 보여주지 않는다.
-i	파일을 삭제할 것인지 사용자에게 물어봄
-r, -R	일반파일이면 그냥 지우고 디렉토리일 경우 그 하위경로와 파일을 모두 지움
-v	삭제되는 파일의 정보를 보여줌

[참고] 비어 있지 않은 디렉토리 삭제

```
# rm -rf dir1
```

[참고] rm 명령어로 지운 파일 복구(100% 장담할 수 없음)

```
# debugfs /dev/sda3
        lsdel
        -잠시후 삭제한 목록이 출력
        -debugfs: dump <아이노드> 파일경로
         파일경로는 다른곳으로 복구 받는다.
         단 파일이 출력되지 않을수도 있다.
```

[EX1] rm 명령어 -r 옵션 사용에 대해서

```
# cd /test
# rm -r /test/*

# mkdir dir1
# touch dir1/file1 dir1/file2
```

```
# cat ~/.bashrc
```

```
# .bashrc

# User specific aliases and functions

alias rm='rm -i'     /* -i : interactive */
alias cp='cp -i'
alias mv='mv -i'

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

#
# Sfecific Configuration
#
export PS1='[Wu@Wh Ww]W$ '
alias ls='ls --color=tty -h'
```

[참고] alias 사용법
(선언) # alias cp='cp -i'
(확인) # alias (# alias cp)
(해제) # unalias cp

```
# rm -r dir1
```

```
rm: descend into directory `dir1'? <CTRL + C>
```

```
# ls
-> dir1 존재 확인

# rm -rf dir1    /* -f : force */
# ls
#
```

파일 내용 확인 명령어

- cat 명령어

```
# cat /etc/passwd
# cat -n /etc/passwd
# cat file1 file2 > file3
```

- more 명령어

```
# cat /etc/services
# more /etc/services

# CMD | more
# ps -ef | more
# rpm -qa | more
# cat /etc/services | more
```

파일 내용 확인 명령어

- head 명령어

```
# head /etc/passwd
# head -5 /etc/passwd
```

- tail 명령어

```
# tail /etc/passwd
# tail -5 /etc/passwd

# tail -f /var/log/messages
```

1 cat CMD

NAME
       cat - concatenate files and print on the standard output

SYNOPSIS
       cat [OPTION] [FILE]...

DESCRIPTION
       Concatenate FILE(s), or standard input, to standard output.

       -n, --number
              number all output lines

cat의 경우 파일이 화면이상으로 길게 작성된 경우 순식간에 지나가버린다.파일의 내용을 화면으로 출력. 파일의 내용을 화면에 연속적으로 출력하기 때문에 파이프(Pipe Line)을 사용하여 more 명령어에 연결하여 사용 가능.

[명령어 형식]

```
# cat file1                          /* file1 파일 내용을 출력 */
# cat file1 file2                    /* file1, file2 파일 내용을 출력 */
# cat -n file1                       /* file1 파일내용을 줄번호와 함께 출력 */
# cat file1 file2 > file3            /* file1, file2 출력 결과를 file3에 저장 */
```

[명령어 옵션]

옵션	설명
-e	제어 문자를 ^ 형태로 출력하며 끝에 $를 추가
-n	줄번호를 공백을 포함하여 화면 왼쪽에 나타냄
-s	중복되고 겹치는 빈 행은 하나의 빈 행으로 처리
-v	행바꿈 문자, tab를 제외한 제어문자를 ^ 형태로 출력
-E	각 행 끝에 $ 문자 출력
-T	tab 문자를 출력
-A	-vET 옵션과 동일

[EX1] /etc/passwd 파일 출력

```
# cat /etc/passwd
-> 출력내용 생략
```

```
# cat -n /etc/passwd | more (# nl /etc/passwd | more)        /* 내용 앞에 번호가 붙어 출력 */
```

```
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  adm:x:3:4:adm:/var/adm:/sbin/nologin
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     6  sync:x:5:0:sync:/sbin:/bin/sync
     7  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     8  halt:x:7:0:halt:/sbin:/sbin/halt
```

```
# cat /etc/passwd | grep fedora
```

```
fedora:x:500:500:fedora:/home/fedora:/bin/bash
```

[EX2] file1, file2 두 개의 파일을 하나의 file3으로 합치기

```
# echo 1111 > file1
# echo 2222 > file2
# cat file1 file2
```

```
1111
2222
```

```
# cat file1 file2 > file3
# cat file3
```

```
1111
2222
```

[참고] cat 명령어를 사용하여 바이너리 파일 확인 시

바이너리 파일을 cat 명령어로 보는 경우 비프(bepp)음이 계속 나오게 되며 이러한 경우 터미널 창을 종료 해야 함.

ex) # cat /bin/ls /* 바이너리 파일은 cat으로 열지 않는다 */
strings -f /bin/ls

man strings

```
NAME
       strings - print the strings of printable characters in files.

SYNOPSIS
       strings [-afov] [-min-len]
               [-n min-len] [--bytes=min-len]
               [-t radix] [--radix=radix]
               [-e encoding] [--encoding=encoding]
               [-] [--all] [--print-file-name]
               [-T bfdname] [--target=bfdname]
               [--help] [--version] file...

DESCRIPTION
       For  each  file  given,  GNU  strings  prints  the  printable character
       sequences that are at least 4 characters long (or the number given with
       the  options  below)  and are followed by an unprintable character.  By
       default, it only prints the strings from  the  initialized  and  loaded
       sections  of  object  files;  for  other  types of files, it prints the
       strings from the whole file.

       strings is mainly useful  for  determining  the  contents  of  non-text
       files.

OPTIONS
       -f
       --print-file-name
           Print the name of the file before each string.
```

확실하지 않은 파일의 경우는 파일의 형식을 확인해본다
ls -F

2 more CMD

NAME
more - 문자속성을 살린 파일 보기 풀그림

사용법
more [-dlfpcsu] [-num] [+/ 표현식] [+ 줄번호] [file ...]

설명
More 명령은 파일 보기 풀그림이다. 이 풀그림은 유닉스에서 처음부터사용한 풀그림이다. 하지만, 이 풀그림의 기능을 보다 보강한 less(1) 명령을 사용하기를 바란다.

옵션
다음은 이 풀그림에서 사용할 수 있는 옵션이며, 이 옵션은 MORE 환경변수로도 지정할 수 있다. 이미 이 환경 변수로 옵션들이 지정되어있는데, 명령행으로 옵션을 사용하면, 그 환경 변수는 무시된다.

-c Do not scroll. Instead, paint each screen from the top, clearing the remainder of each line as it is displayed.

큰 파일을 출력할 때 화면 크기 페이지 단위로 출력하며 하단에 "--More--(20%)"는 현재 내용을 20% 보았고 80% 남았다고 표현하며 화면에서 엔터(Enter)키를 누르면 한 개의 라인(line) 단위로 넘어가고 스페이스(space) 키를 누르면 한 페이지 단위로 넘어가는데 less 명령과 함께 사용 하면 더 효율적이다.

[명령어 형식]

```
# more file1                    /* file1 파일을 출력 */
# more -c file1                 /* file1 파일을 한행씩 지우면서 출력 (more와 같은 형태로 출력~) */
```

[명령어 옵션]

옵션	설명
-n(숫자)	출력 행수를 지정
-c	위에서부터 한 행씩 지운 후 한 행씩 출력
-d	스페이스나 q를 누르라는 프롬프트를 출력
-f	보통은 긴 칼럼의 행은 화면에서 행 바꿈을 하여 새로운 행으로 계산되는데 -f 옵션은 새로운 행으로 계산 하지 않으며 화면이 행이 아닌 논리적인 행 수를 계산
-s	여러 개의 빈 공백행은 하나로 취급
-p	스크롤하지 않으며 화면을 지우고 출력
-u	밑줄 치기를 금지

[참고] cat & more CMD 차이점

```
# cat /etc/services
# more /etc/services
```

[EX1] "CMD | more" 형식 실습

```
        # CMD        (EX: # help)              /* 쉘 내부(내장) 명령어의 목록 확인 */
        # CMD | more (EX: # help | more)

        # ps -ef | more
        # cat /etc/services | more
        # rpm -qa | more
        # chkconfig --list | more
```

```
# cat /etc/services | more                  / * 상단부터 출력한다 */
```

```
# /etc/services:
# $Id: services,v 1.42 2006/02/23 13:09:23 pknirsch Exp $
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1700, ``Assigned Numbers'' (October 1994).  Not all ports
# are included, only the more common ones.
--More--
```

```
# more /etc/services
```

```
# /etc/services:
# $Id: services,v 1.42 2006/02/23 13:09:23 pknirsch Exp $
#
# Network services, Internet style
#
# Note that it is presently the policy of IANA to assign a single well-known
# port number for both TCP and UDP; hence, most entries here have two entries
# even if the protocol doesn't support UDP operations.
# Updated from RFC 1700, ``Assigned Numbers'' (October 1994).  Not all ports
# are included, only the more common ones.
--More--(0%)                          /* 0% 확인 */
```

[참고] more 안에서 사용할 수 있는 여러 가지 키보드 키 값

more화면 창에서 h 또는 ?를 치면 아래와 같은 화면이 나오며 여러 가지 기능을 확인 할 수 있다.

```
Star (*) indicates argument becomes new default.
-------------------------------------------------------------------------------
<space>                 Display next k lines of text [current screen size]
z                       Display next k lines of text [current screen size]*
<return>                Display next k lines of text [1]*
d or ctrl-D             Scroll k lines [current scroll size, initially 11]*
q or Q or <interrupt>   Exit from more
s                       Skip forward k lines of text [1]
f                       Skip forward k screenfuls of text [1]
b or ctrl-B             Skip backwards k screenfuls of text [1]
'                       Go to place where previous search started
=                       Display current line number
/<regular expression>   Search for kth occurrence of regular expression [1]
n                       Search for kth occurrence of last r.e [1]
!<cmd> or :!<cmd>       Execute <cmd> in a subshell
v                       Start up /usr/bin/vi at current line
ctrl-L                  Redraw screen
:n                      Go to kth next file [1]
:p                      Go to kth previous file [1]
:f                      Display current file name and line number
.                       Repeat previous command
-------------------------------------------------------------------------------
```

spacebar : 매뉴얼 페이지에서 한 화면 단위로 넘어 갈 때 사용
Enter : 매뉴얼 페이지에서 한 라인씩 넘어 갈 때 사용
b : Back Screen, 한 화면 전 화면으로 넘어갈 때 사용
/pattern : 특정한 패턴을 빨리 찾을 때 사용
n : Next, 특정한 패턴을 찾은 후 다음 번째 똑같은 문자열을 찾을 때 사용
h : help 매뉴얼 페이지 안에서 사용 할 수 있는 명령어 소개
q : quit, 빠져나옴

3 less CMD

```
NAME
       less - opposite of more

SYNOPSIS
       less -?
       less --help
       less -V
       less --version
       less [-[+]aBcCdeEfFgGiIJKLmMnNqQrRsSuUVwWX~]
            [-b space] [-h lines] [-j line] [-k keyfile]
            [-{oO} logfile] [-p pattern] [-P prompt] [-t tag]
            [-T tagsfile] [-x tab,...] [-y lines] [-[z] lines]
            [-# shift] [+[+]cmd] [--] [filename]...
       (See  the  OPTIONS section for alternate option syntax with long option
       names.)

DESCRIPTION
       Less is a program similar to more (1), but which allows backward  move-
       ment in the file as well as forward movement.  Also, less does not have
       to read the entire input file before  starting,  so  with  large  input
       files  it  starts  up  faster than text editors like vi (1).  Less uses
       termcap (or terminfo on some systems), so it can run on  a  variety  of
       terminals.   There is even limited support for hardcopy terminals.  (On
       a hardcopy terminal, lines which should be printed at the  top  of  the
       screen are prefixed with a caret.)

       Commands  are based on both more and vi.  Commands may be preceded by a
       decimal number, called N in the descriptions below.  The number is used
       by some commands, as indicated.
```

리눅스 시스템에서 more 명령어를 좀 더 보안한 명령어이다.

[명령어 형식]

```
# less file1                    /* file1 파일을 출력 */
# less -n file1                 /* file1 파일내용에 행 번호 출력 */
```

[명령어 옵션]

옵션	설명
-?	less에 대한 help
-a	마지막 라인이 화면에 출력되고 나서 탐색 시작
-c	필요할 때 전체 화면을 다시 갱신
-C	-c 옵션과 같지만 갱신할 때 화면 전체를 지우고 시작
-e	두 번째로 파일의 끝에 도달하면 자동적으로 종료
-E	파일의 끝에 도달하면 자동적으로 종료
-i	대소문자를 구분하여 탐색
-q	특정 에러가 없으면 소리 안냄
-Q	소리 안냄
-s	연속되는 공백 라인은 하나의 행으로 처리
-x n(숫자)	수치를 지정해서 탭 간격 조정

[EX1] less 명령어 실습

많이 사용되는 명령어 형식
CMD | more
CMD | less

```
# less /etc/passwd          /* q를 입력하지 않는 한 빠져나오지 않음 */
gdm:x:42:42::/var/gdm:/sbin/nologin
sabayon:x:86:86:Sabayon user:/home/sabayon:/sbin/nologin
fedora:x:500:500:fedora:/home/fedora:/bin/bash
virus:x:501:501::/home/virus:/bin/bash
(END)
```

```
# more /etc/passwd          /* 바로 빠져나옴 */
gdm:x:42:42::/var/gdm:/sbin/nologin
sabayon:x:86:86:Sabayon user:/home/sabayon:/sbin/nologin
fedora:x:500:500:fedora:/home/fedora:/bin/bash
virus:x:501:501::/home/virus:/bin/bash
```

```
# less -2 /etc/passwd       /* 한 화면 가득 보이다가 아래서 2줄씩 올려지는 것 확인 가능 */
gdm:x:42:42::/var/gdm:/sbin/nologin
sabayon:x:86:86:Sabayon user:/home/sabayon:/sbin/nologin
fedora:x:500:500:fedora:/home/fedora:/bin/bash
virus:x:501:501::/home/virus:/bin/bash
```

[참고] 매뉴얼 페이지 안에서 사용 할 수 있는 여러 가지 키보드 키 값

```
spacebar : 매뉴얼 페이지에서 한 화면 단위로 넘어 갈 때 사용
Enter    : 매뉴얼 페이지에서 한 라인씩 넘어 갈 때 사용
b        : Back Screen, 한 화면 전 화면으로 넘어갈 때 사용
/pattern : 특정한 패턴을 빨리 찾을 때 사용
n        : Next, 특정한 패턴을 찾은 후 다음 번째 똑같은 문자열을 찾을때 사용
h        : help 매뉴얼 페이지 안에서 사용할수 있는 명령어 소개
q        : quit, 빠져나옴
```

[EX2] less 명령어의 확장 사용

```
# less /etc/passwd /etc/shadow              /* 두개의 파일 동시에 열기 */
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
/etc/passwd (file 1 of 2)
:n                      <----- ":n" 입력    /* n(next), 다음 페이지(Next)로 이동 */
root:$1$i1ANqRGI$DRaSNHey0Tid6p8AnpW3r0:14739:0:99999:7:::
bin:*:14634:0:99999:7:::
daemon:*:14634:0:99999:7:::
adm:*:14634:0:99999:7:::
lp:*:14634:0:99999:7:::
/etc/shadow (file 2 of 2)
:e                      <----- ":e" 입력    /* e(examine), 새로운 파일 추가 */
Examine: /etc/hosts     <----- "/etc/hosts" 입력
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       localhost.localdomain   localhost
192.168.10.200  linux200        linux200.example.com
::1             localhost6.localdomain6 localhost6
/etc/hosts (file 3 of 3) (END)
:p                      <----- ":p" 입력    /* p(previous), 이전 페이지로 이동 */
root:$1$i1ANqRGI$DRaSNHey0Tid6p8AnpW3r0:14739:0:99999:7:::
bin:*:14634:0:99999:7:::
daemon:*:14634:0:99999:7:::
adm:*:14634:0:99999:7:::
lp:*:14634:0:99999:7:::
/etc/shadow (file 2 of 3)
q                       <----- "q" 입력     /* q(quit), 빠져 나가기 */
```

4 nl CMD

```
NAME
       nl - number lines of files

SYNOPSIS
       nl [OPTION]... [FILE]...

DESCRIPTION
       Write  each  FILE to standard output, with line numbers added.  With no
       FILE, or when FILE is -, read standard input.

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

파일의 내용을 확인 할 때 행번호 까지 출력 ("cat -n" 명령어와 같다.) 내가 작업한 문서가 몇째 줄 까지 작업이 되었는지 확인하고 싶을 때 사용한다.

[명령어 형식]

```
# nl file1
# nl file1 file2
# nl file1 file2 > files
```

[EX1] nl 명령어 실습

```
# nl /etc/passwd          /* # cat -n /etc/passwd 와 같다. */
     1  root:x:0:0:root:/root:/bin/bash
     2  bin:x:1:1:bin:/bin:/sbin/nologin
     3  daemon:x:2:2:daemon:/sbin:/sbin/nologin
     4  adm:x:3:4:adm:/var/adm:/sbin/nologin
     5  lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
     6  sync:x:5:0:sync:/sbin:/bin/sync
     7  shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
     8  halt:x:7:0:halt:/sbin:/sbin/halt
```

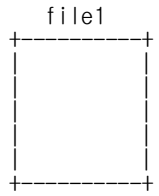
```
# nl /etc/passwd /etc/shadow        /* 두개의 파일이 동시에 보이면서 번호가 이어져서 출력 */
....
    59  fedora:x:500:500:fedora:/home/fedora:/bin/bash
    60  virus:x:501:501::/home/virus:/bin/bash                          /* /etc/passwd 파일 */
    61  root:$1$GUT1Ey/2$ltYR1Bjp6er6klmf1/4DV1:14634:0:99999:7:::      /* /etc/shadow 파일 */
    62  bin:*:14634:0:99999:7:::
....
```

```
# nl /etc/passwd /etc/shadow > passwd_shadow
# cat passwd_shadow
....
    59  fedora:x:500:500:fedora:/home/fedora:/bin/bash
    60  virus:x:501:501::/home/virus:/bin/bash                          /* /etc/passwd 파일 */
    61  root:$1$GUT1Ey/2$ltYR1Bjp6er6klmf1/4DV1:14634:0:99999:7:::      /* /etc/shadow 파일 */
    62  bin:*:14634:0:99999:7:::
....
```

5 head CMD

```
NAME
       head - output the first part of files

SYNOPSIS
       head [OPTION]... [FILE]...

DESCRIPTION
       Print  the  first  10 lines of each FILE to standard output.  With more
       than one FILE, precede each with a header giving the file  name.   With
       no FILE, or when FILE is -, read standard input.

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

       -c, --bytes=[-]N
              print the first N bytes of each  file;  with  the  leading  '-',
              print all but the last N bytes of each file

       -n, --lines=[-]N
              print  the first N lines instead of the first 10; with the lead-
              ing '-', print all but the last N lines of each file
```

파일의 처음 시작 부분의 몇 줄을 출력 하고 자 할 때 사용. 따라서 긴 파일의 내용의 앞 부분만을 출력 하고자 할 때 유용하게 사용 되며 head 명령어에 아무런 옵션 없이 사용된 경우 문서의 처음 10줄을 보여 준다.



[명령어 형식]

```
# head /etc/passwd          (# head -10 /etc/passwd, # head -n 10 /etc/passwd)
# head -n 5 /etc/passwd     /* 숫자에 해당하는 라인 번호 수 만큼만 출력 (기본은 10줄) */
# head -c 10 /etc/passwd    /* -c 옵션 다음에 오는 숫자 byte 수에 해당하는 만큼 출력 */
```

[명령어 옵션]

옵션	설명
-n (숫자)	위쪽 행에서부터 출력할 행수를 지정
-c (숫자)	byte 수 만큼만 출력

[EX1] "head -n #" 실습

 # head -n 5 /etc/passwd   (# head -5 /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
```

 # head -c 10 /etc/passwd

```
root:x:0:0
```

[EX2] head 명령어를 이용한 프로세스의 헤더 부분 출력

```
# ps -ef
# ps -ef | more
# ps -ef | grep inetd
```

```
root     27362     1  0 Jan26 ?        00:00:00 xinetd -stayalive -pidfile /var/run/xinetd.pid
```

```
# ps -ef | head -1
```

```
UID        PID  PPID  C STIME TTY          TIME CMD
```

```
# ps -ef | head -1 ; ps -ef | grep inetd  ( ; =  하나의 명령줄에 다수의 명령을 순차적으로 실행할 때)
```

```
UID        PID  PPID  C STIME TTY          TIME CMD
root      7048  6809  0 00:45 pts/4    00:00:00 grep inetd
root     27362     1  0 Jan26 ?        00:00:00 xinetd -stayalive -pidfile /var/run/xinetd.pid
```

```
# alias pps='ps -ef | head -1 ; ps -ef | grep $1'   /* $1 : 첫번째 인자(Argument) */
# pps inetd    (# pps syslogd)
```

```
UID        PID  PPID  C STIME TTY          TIME CMD
root      7048  6809  0 00:45 pts/4    00:00:00 grep inetd
root     27362     1  0 Jan26 ?        00:00:00 xinetd -stayalive -pidfile /var/run/xinetd.pid
```

```
# vi ~/.bashrc (# gedit ~/.bashrc)
```

```
..... (중략) .....
#
# Sfecific Configuration
#
alias pps='ps -ef | head -1 ; ps -ef | grep $1'
```

```
# . ~/.bashrc   (# source ~/.bashrc)
# pps syslogd
-> 정상 출력 내용 확인
```

[참고] Unix 계열의 초창기 head와 유사한 명령어

```
# sed 5q /etc/passwd        /* 파일의 5줄을 출력하고, 끝내라(q)는 뜻 */
```

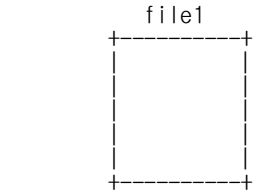
```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
```

```
# sed -n '1,5p' /etc/passwd
```

5 tail CMD

```
NAME
       tail - output the last part of files

SYNOPSIS
       tail [OPTION]... [FILE]...

DESCRIPTION
       Print  the  last  10  lines of each FILE to standard output.  With more
       than one FILE, precede each with a header giving the file  name.   With
       no FILE, or when FILE is -, read standard input.

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

       -c, --bytes=N
              output the last N bytes

       -f, --follow[={name|descriptor}]
              output appended data as the file grows; -f, --follow, and --fol-
              low=descriptor are equivalent

       -n, --lines=N
              output the last N lines, instead of the last 10
```

tail은 텍스트파일이나 지정된 데이터의 마지막 몇 줄을 보여주는 데 사용하는 Unix 및 Unix계열 시스템에서의 프로그램이다. 파일의 끝 부분만 출력 하고자 할 때 사용하며, 아무런 옵션 없이 사용된 경우 문서의 마지막 10줄을 보여는데, 예를 들어서 사용자가 추가되면 /etc/passwd 파일에 마지막에 추가 된다. 이때 tail 명령어에 -1 옵션을 사용하여 사용자 추가를 확인 할 수 있다.



[명령어 형식]

```
# tail /etc/passwd     (# tail -10 /etc/passwd, # taill -n 10 /etc/passwd)
# tail -5 /etc/passwd
# tail +5 /etc/passwd

# tail -f /var/log/messages
```

[명령어 옵션]

옵션	설명
-c (숫자)	끝에서부터 지정된 수만큼의 바이트에 해당하는 정보를 보여준다.
-f	파일의 크기가 변할때마다 추가된 정보를 출력한다.
-F	위 -f옵션의 경우 파일크기가 변하여 5Mbytes 정도 되면 확장자에 숫자를 붙여 백업파일을 생성하고, 다시 본 파일은 0byte 부터 저장된다. 그래서 tail -f 파일명으로 실행중인 명령이 멈춰버린다는 문제가 생겨 재실행시켜줘야 하는 번거로움이 있다. -F옵션을 사용하면 이런 재실행문제 및 용량변화로 인한 문제를 걱정하지 않아도 된다. -f옵션때와 마찬가지로 Ctrl+C로 빠져나올 수 있다.
-(숫자)	끝에서부터 지정된 수만큼의 줄을 보여준다.
-q	출력결과에서 맨 윗줄에 입력파일명을 표시하지 않게 설정한다.
-v	-q와 반대로 출력결과에서 맨 윗줄에 입력파일명을 항상 표시해준다.
--help	도움말을 보여준다.
--version	버젼 정보를 보여준다.

[EX1] tail 명령어의 기본 사용법
tail -n 5 /etc/passwd (# tail -5 /etc/passwd)

```
tomcat:x:91:91:Tomcat:/usr/share/tomcat5:/bin/sh
gdm:x:42:42::/var/gdm:/sbin/nologin
sabayon:x:86:86:Sabayon user:/home/sabayon:/sbin/nologin
fedora:x:500:500:fedora:/home/fedora:/bin/bash
user01:x:501:501::/home/user01:/bin/bash
```

tail -c 15 /etc/passwd

```
er01:/bin/bash
```

[EX2] 로그 파일 모니터링(EX: tail -f CMD)
tail은 실시간으로 파일의 변화를 감지할 수 있게 해주는 -f옵션이라는 특별한 명령행옵션을 가지고 있다. 마지막 몇줄을 출력하고 끝내는 것에 그치지 않고, tail은 그 줄들을 표시하고 파일을 감독한다. 새 줄들이 다른 프로세스에 의해 그 파일에 추가될 때, tail의 -f옵션은 그 표시또한 실시간으로 업데이트한다. 이 옵션은 특히 로그파일(입출력정보파일)들을 감독할때 유용하다. 다음의 명령구문은 messages라는 파일의 마지막 열줄을 보여주고 새 줄들이 추가되면 그 줄들을 추가하여 보여준다.

[참고] telnet 서비스 Open 방법(일반사용자)
chkconfig —list
chkconfig —list | grep krb5-telnet

chkconfig krb5-telnet on
service xinetd restart

telnet localhost
fedora 사용자 로그인
$ id
$ exit

telnet localhost
root 사용자 로그인
-> 로그인이 되지 않음
<CTRL + D>
vi /etc/securetty (# gedit /etc/securetty)
.....
pts/1
pts/2
pts/3
.....
pts/10
pts/11 >> pts/0~11 까지 추가한다

telnet localhost
root 사용자로 로그인
id
exit

[참고]
CENTOS 6.X 버전의 경우에는 telnet 서비스가 기본적으로 설치되어 있지않기 때문에 설치하는 과정이 따로 필요하다.

yum -y install telnet-server
yum -y install telnet
service xinetd restart
chkconfig xinetd on
vi /etc/securetty pts/0 ~ pts/11 추가

[TERM1] 관리자용 윈도우
tail -f /var/log/messages
<ENTER>
<ENTER>
<ENTER> 이전 로그기록과의 간격을 만든다

[TERM2] 사용자 윈도우

(사용자 추가 방법)
useradd user01
passwd user01

telnet localhost
user01 사용자로 로그인

$ id
$ exit

[TERM1] 관리자용 윈도우

```
Apr  9 10:45:41 linux200 xinetd[18598]: START: telnet pid=18615 from=127.0.0.1
Apr  9 10:45:56 linux200 xinetd[18598]: EXIT: telnet status=0 pid=18615 duration=15(sec)
<CTRL + C>
```

[질문] 파일이 30 라인으로 되어져 있는데 파일 중간의 10라인만 출력해 볼 수 있나요?

```
# cp /etc/passwd /test/filename
# cd /test
# head -20 filename | tail -10
  head로 상단에서 20개 라인을 출력하고 PIPE로 받아 tail로 아래에서 10개 라인 출력
# nl filename | head -20 | tail -10
    11  uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
    12  operator:x:11:0:operator:/root:/sbin/nologin
    13  games:x:12:100:games:/usr/games:/sbin/nologin
    14  gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
    15  ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
    16  nobody:x:99:99:Nobody:/:/sbin/nologin
    17  vcsa:x:69:69:virtual console memory owner:/dev:/sbin/nologin
    18  apache:x:48:48:Apache:/var/www:/sbin/nologin
    19  exim:x:93:93::/var/spool/exim:/sbin/nologin
    20  rpc:x:32:32:Portmapper RPC user:/:/sbin/nologin
```

[EX3] 서버를 실시간적으로 모니터링

```
[TERM1] # top                          (# gnome-system-monitor)
[TERM2] # tail -f /var/log/messages  (# gnome-system-log)

# tail -f /var/log/messages | grep -i DHCP
# tail -f /var/log/messages | grep -i DNS
# tail -f /var/log/messages | grep oracle
# tail -f /var/log/messages | grep wasuser  (특정 서비스에 대한 모니터링)
```

또 다른 활용방안

```
# cat /var/log/messages | egrep -i '(warn|err|alert|emerg)'
```

기타 관리용 명령어

- wc 명령어

```
# wc -l /etc/passwd

# ps -ef | wc -l
# cat /etc/passwd | wc -l
# rpm -qa | wc -l
```

[참고] 데이터수집

```
# ps -ef | grep httpd | wc -l
# df -h | tail -1 | awk '{print $5}'
# cat /var/log/messages | grep 'START: telnet \
> | wc -l
```

1 wc CMD

```
NAME
       wc - print the number of newlines, words, and bytes in files

SYNOPSIS
       wc [OPTION]... [FILE]...

DESCRIPTION
       Print newline, word, and byte counts for each FILE, and a total line if
       more than one FILE is specified.  With no FILE, or when FILE is -, read
       standard input.

       -c, --bytes
              print the byte counts

       -m, --chars
              print the character counts

       -l, --lines
              print the newline counts

       -w, --words
              print the word counts
```

파일 내의 문자수, 단어 수 그리고 라인수를 확인하고자 할 때 사용한다. wc 명령어를 사용하여 프로세스의 수, 시스템에 설치된 패치의 수, 시스템에 설치된 패치의 수 등을 확인 할 때 사용 할 수 있다. wc 명령어에 -l 옵션은 쉘스크립트나 파일에 대한 무결성 체크 등 많은 곳에서 활용이 가능하다.

[명령어 형식]

```
# wc /etc/passwd
# wc -l /etc/passwd
# wc -w /etc/passwd
# wc -c /etc/passwd
```

[명령어 옵션]

옵션	설명
-c	문자수만 출력
-l	라인수만 출력
-w	단어수만 출력
-L	가장 긴줄 한줄만 출력

[EX1] wc 명령어 사용법

```
# wc /etc/passwd
  59     104    2902     /etc/passwd
```

```
# wc -l /etc/passwd
  59       /etc/passwd
```

```
# wc -w /etc/passwd                /* 단어수 (word) */
 104       /etc/passwd
```

```
# wc -c /etc/passwd                /* 문자수 (byte) */
2902       /etc/passwd
```

[EX2] 시스템 사용자 수 확인

```
# cat /etc/passwd | wc -l
59
```

```
# wc -l /etc/passwd
59 /etc/passwd
```

[EX3] 실행중인 프로세스의 수 확인

```
# ps -ef | wc -l
98
```

[EX4] 설치된 패키지 수 확인

```
# rpm -qa | wc -l
1451
```

-> 어떠한 프로그램이 설치 되어있는지 시스템에 설치된 패키지 수, 제어판 -> 프로그램 추가 삭제

[EX5] 시스템성능/사용량 카운트 수집 : Data Gathering

```
Apache Webserver
- apache 1.3.X    (Process 방식)
- apache 2.X      (Thread 방식)

Web Client ------------> Web Server(www.daum.net)
http://www.daum.net      httpd
```

```
# service httpd restart
Stopping httpd:                                            [FAILED]
Starting httpd: httpd: apr_sockaddr_info_get() failed for linux249.example.com
httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for
ServerName
                                                           [  OK  ]
```

-> 에러메세지 무시

```
# ps -ef | grep httpd
root      8459     1  0 12:39 ?        00:00:00 /usr/sbin/httpd
apache    8461  8459  0 12:39 ?        00:00:00 /usr/sbin/httpd
apache    8462  8459  0 12:39 ?        00:00:00 /usr/sbin/httpd
apache    8463  8459  0 12:39 ?        00:00:00 /usr/sbin/httpd
apache    8464  8459  0 12:39 ?        00:00:00 /usr/sbin/httpd
apache    8465  8459  0 12:39 ?        00:00:00 /usr/sbin/httpd
apache    8466  8459  0 12:39 ?        00:00:00 /usr/sbin/httpd
apache    8467  8459  0 12:39 ?        00:00:00 /usr/sbin/httpd
apache    8468  8459  0 12:39 ?        00:00:00 /usr/sbin/httpd

# ps -ef | grep httpd | wc -l > apache.count
# cat apache.count
9
```

[EX6] 디스크 사용량을 모니터링

```
# df -k
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda1             26960852   3299332  22269892  13% /
/dev/sda8               497829     10544    461583   3% /data1
/dev/sda7               497829     10544    461583   3% /data2
/dev/sda6               497829     10544    461583   3% /data3
/dev/sda5               497829     10544    461583   3% /data4
/dev/sda3               497861     10573    461584   3% /home
tmpfs                  1557668         0   1557668   0% /dev/shm

# df -k /
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/sda1             26960852   3299332  22269892  13% /

# df -k / | tail -1
/dev/sda1             18284608   6093720  11247072  36% /

# df -k / | tail -1 | awk '{print $5}' > df.count
# cat df.count
36%

# df -k / | tail -1 | awk '{print $5}' | awk -F% '{print $1}' > df.count
# cat df.count
36
```

[참고] awk 명령어

awk는 라인을 받아와서 구분자를 통해 구분하고 print 명령으로 출력하게 된다.

awk 명령어를 사용하기 전에 꼭 알아두어야 할 것은 기본적으로 탭 또는 공백으로 구분되는 각각의 단어들을 하나의 변수로 처리한다는 것이다. 탭과 공백을 무시하고 특정문자 콜론(:)이나, 세미콜론(;) 콤마(,) 등을 구분문자로 사용하고자 한다면 -F 옵션을 사용하면 된다.

명령어 형식 = awk [option] [-F] ['{script}']

```
# cd /test
# touch file1 file2 file3
# ls -l | grep "^-" | awk '{print "vi "$9}'
vi file1
vi file2
vi file3
```

ls -l 명령어로 현재 디렉토리의 목록을 출력하여 나온 결과를 grep 명령을 통해 "^-" 에 해당되는 값을 출력하여 awk 명령으로 "vi "$9 포맷형식으로 출력한 내용을 test.sh 파일에 복사하여 넣어라 라는 것입니다.

기타 관리용 명령어

- **su 명령어**

```
# su user01
# su - user01
```

- **id 명령어**

```
# id
# id user01
```

- **groups 명령어**

```
# groups
# groups user01
# groups user01 root
```

2 su CMD

이름

su - 사용자와 그룹 ID 를 교체하여 쉘을 실행한다.(Switching User)

개요

su [-flmp] [-c 명령] [-s 쉘] [--login] [--fast] [--preserve-environment] [--command=명령] [--shell=쉘] [-] [--help] [--version] [사용자 [인수...]]

설명

이 맨페이지는 GNU 버전의 su 를 설명한다. su 는 한 사용자가 잠시 다른 사용자가 될 수 있도록 해준다. 실제 사용자 ID, 그룹 ID, USER의 보충적인 그 룹 으로 쉘을 실행한다. USER가 주어지지 않으면 기본적으로 수퍼유저인 root 로 설정된다. 실행되는 쉘은 USER의 패스워드 목록에서 찾아오거나 없으 면 /bin/sh 를 수행한다. 만약 USER에 패스워드가 있다면 su 는 실제 사용자 ID 0 (수퍼유저)가 아닌 한 패스워드를 물어온다.

기본적으로, su 는 현재 디렉토리를 변경하지 않는다. USER 의 패스워드 항목 으로부터 'HOME', 'SHELL' 등의 변수를 설정하고 만약 수퍼유저가 아니라면 'USER'와 'LOGNAME'을 USER로 설정한다. 기본적으로 이 쉘은 로그인 쉘이 아니다.

만약 한 개 이상의 인수가 주어지면 쉘에 대한 인수로 전달된다.

su 는 /bin/sh나 다른 쉘을 특별히 다루지는 않는다. (argv[0]를 "-su"로 하고 -c 를 특정 쉘로 지정하지 않는 한...)

syslog를 가지고 있는 시스템에서는, su 가 실패하는 경우 보고를 하 도 록, 그 리 고 성공의 경우에는 선택적으로 보고하도록컴파일하면 su 가 syslog를 사용한다.

옵션

-c COMMAND, --command=COMMAND
대화형 쉘을 시작하지 않고 -c 옵션을 쉘에 주어서 한 개의 명령만을 수행하도록 한다.

다른 사용자의 권한으로 쉘을 실행한다. 서버에 접속한 상태에서 로그아웃 없이 다른 사용자로 전환할 수 있음 (windows의 사용자 전환 - terminal service (application)를 자동으로 만들어 놓아야 사용자 전환이 실행가능) 원격 접속시에 다른 사용자로 로그인하기 위해 로그아웃하면 접속이 종료가 되는데 접속이 종료 되지 않은 상태에서 다른 사용자로 로그인하고자 할 때 사용한다.
한명의 관리자가 여러 계정을 사용하는 경우 유용하다.

업데이트 필요

일반사용자가 다른 사용자가 되는 것을 권한이 높아지는 것이기 때문에 전환(Switching) 되는 사용자의 암호를 맞추어야만 전환이 가능하고, root 사용자가 다른사용자로 전환하는 경우에는 권한이 낮아지는 것이기 때문에 암호입력 없이 전환이 가능하다.

■ (사용자 전환의 예)
일반사용자(user01) ----> 다른 일반사용자(user02)
일반사용자(user01) ----> 관리자(root)
관리자(root) ----> 일반사용자(user01)

su 명령어 다음에 전환하고 싶은 사용자 이름이 없는 경우 root 사용자로 전환된다. 그리고 su 명령어에 '-' 기호 없이 다른 사용자로 전환하는 경우 지정된 사용자로 전환이 되지만 이전 사용자가 쓰고 있던 export 변수들의 설정이 그대로 따라온다. su 명령어에 '-' 기호를 붙이고 다른 사용자로 전환하는 경우 지정된 사용자로 새로 로그인한 것 처럼 동작을 시켜서 사용자 환경변수로 모두 변경한다.(DB를 운영하는 oracle 사용자)

[명령어 형식]
su [fedora]
su - [fedora]

[EX1] su 명령어의 전환 형식 실습
 # cd /etc
 # pwd

```
/etc
```

 # su fedroa
$ id

```
uid=500(fedora) gid=500(fedora) groups=500(fedora)
```

$ pwd

```
/etc
```

$ echo $PATH

```
/usr/kerberos/sbin:/usr/kerberos/bin:/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin:/root/bin
```

$ exit

 # cd /etc
 # pwd

```
/etc
```

 # su - fedora
$ id

```
uid=500(fedora) gid=500(fedora) groups=500(fedora)
```

$ pwd

```
/home/fedora
```

$ echo $PATH

```
/usr/kerberos/bin:/usr/local/bin:/bin:/usr/bin:/home/fedora/bin
```

$ exit
#

변경필요.

[EX2] fedora 사용자가 다른 사용자로 전환하는 경우

 # ssh fedora@localhost

```
fedora@localhost's password: (fedora)
Last login: Wed Jan  8 12:34:36 2014 from localhost.localdomain
```

-> 암호 입력

$ cat /etc/shadow

```
cat: /etc/shadow: Permission denied
```

-> 에러 메세지 확인

$ ls -l /etc/shadow

```
----------. 1 root root 1336 Jan  8 12:39 /etc/shadow
```

자신의 계정이상으로 권한이 필요한 파일,디렉토리의 접근,수정에 사용한다.

$ su - ($ su - root)

```
Password: (soldesk1.)
```

-> root 사용자의 암호 입력

 # id

```
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
```

 # cat /etc/shadow

-> 파일 내용 확인

 # exit
$ exit
#

3 id CMD

이름
id - 실제, 유효 UID와 GID를 출력한다.

개요
id [-gnruG] [--group] [--name] [--real] [--user] [--groups] [--help] [--version] [username]

설명
이 맨페이지는 GNU 버전의 id 를 다룬다. id 는 주어진 사용자, 또는 사 용자가 주어지지 않는 경우 프로세스의 주인에 대한정보를 출력한다. 기본적으로 실제 사용자 ID, 실제 그룹 ID, 만약 실제 사용자 ID와 다르다면 유효 사용 자 ID를, 마찬가지로 실제 그룹 ID와 다르다면 유효 그룹 ID를 출력하고 추가 그룹 ID를 출력한다. 각 항목은 식별 문자 그리고 괄호 안에해당 하 는 사용자 또는 그룹명으로 표현된다.

id 에 옵션을 주면 위에서 열거한 정보의 일부만 보여준다.

옵션
-g, --group
오로지 그룹 ID만 출력한다.

-G, --groups
추가 그룹만 출력한다.

--help 표준출력으로 사용법을 출력하고 정상적으로 종료한다.

-n, --name
ID 번호 대신 사용자명, 그룹명을 출력한다. -u, -g, 또는 -G 를 필요로 한다.

사용자의 uid, gid, groups을 보여줌 (사용자 이름의 길이가 긴 경우 메모리 기억공간의 낭비가 심함. but 정수로 사용자의 이름을 구분하는 경우 메모리 기억공간을 효율적으로 사용가능. 정수로 사용하여 4byte씩 사용될 경우 0 ~ 4294967295명까지 구분이 가능)

[명령어 형식]
```
# id
# id -u root
# id -g root
```

[명령어 옵션]

옵션	기능
-g	기본 그룹의 gid를 출력 (사용자가 소속되어 있는 기본 그룹)
-G	사용자가 속한 모든 그룹의 gid를 출력
-u	사용자의 uid를 출력
-n	-u과 함께 사용하여 숫자 대신 이름 출력

[EX] id 명령어 사용

 # id
```
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel),103(pkcs11)
```

 # id -g root
```
0
```

 # id -G root
```
0 1 2 3 4 6 10 103
```

 # id -Gn root
```
root bin daemon sys adm disk wheel pkcs11
```

4 groups CMD

이름

groups - 사용자가 속한 그룹들을 출력한다

개요

groups [사용자명...]
groups {--help,--version}

설명

이 맨페이지는 GNU 버전의 groups 를 다룬다. groups 는 주어진 각 user-name 또는 프로세스가 속한 추가 그룹의 이름을 출력해준다. 만약 사용자명이 주어졌다면 각 사용자명이 소속된 그룹 목록 앞에 표시된다.

그룹 목록은 'id -Gn'의 결과와 같다.

사용자가 속한 그룹의 정보를 보여준다. 일반적으로 id 명령어로 확인하는 편이고 gruops 명령어는 명령어 라인상에서는 자주 사용되는 명령어는 아니다.

[명령어 형식]

```
# groups                          /* 현재 사용중인 사용자의 그룹을 보여줌 */
# groups user1                    /* user1의 그룹을 보여줌 */
# groups user1 user2              /* user1, user2의 그룹을 보여줌 */
```

[EX1] groups 명령어 사용법

 # groups

```
root bin daemon sys adm disk wheel pkcs11
```

 # groups fedora

```
fedora : fedora
```

 # groups fedora root

```
fedora : fedora
root : root bin daemon sys adm disk wheel pkcs11
```

 # cat /etc/group

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
..... (중략) ....
sabayon:x:86:
fedora:x:500:
user01:x:501:
```

-> 시스템에 선언된 모든 그룹들에 대한 기본적인 정보를 표시한다.

group에 사용자를 추가할 경우
adduser -G [GROUP] [USER]

기타 관리용 명령어

- **last 명령어**

```
# last
# last user01
# last reboot
# last -20
# last -f /var/log/wtmp.0
```

- **lastlog 명령어**

```
# lastlog
# lastlog -u user01
# lastlog -t 10
```

- **lastb 명령어**

```
# lastb
# lastb -20
```

5 last CMD

NAME
last, lastb - 사용자들의 마지막으로 로그인한 기록을 보여준다.

SYNOPSIS
last [-R] [-num] [-n num] [-adx] [-f file] [name...] [tty...]
lastb [-R] [-num] [-n num] [-f file] [-adx] [name...] [tty...]

DESCRIPTION
Last 명령은 /var/log/wtmp 파일이나, -f 옵션에서 지정한파일을 통해서 사용자의 로그인, 로그아웃 시간을 보여준다. 특정 사용자의 이름이나, tty를 지정할 수 있으며, 이렇게 지정되면, 해당 사용자의 로그인 정보만을 보여준다. tty의 이름은 생략될 수 있어, last 0 명령은 last tty0 명령과 같은 기능을 한다. last가 실행 중에 SIGINT(인터럽트 글쇠, 주로 Ctrl-C에 의해서 만들어지는 신호)가 감지되거나, SIGQUIT(마침 글쇠, 주로 Ctrl-W에 의해서 만들어지는 신호)가감지되면, 실행을 마친다.

접속 사용자 이름에 reboot라고 나오는 것은 시스템이 reboot된 것을 나타낸다. 즉 last reboot 명령으로기록 파일이 만들어진 후부터 reboot 되었는 기록을 모두 볼 수 있다.

Lastb 명령은 접속 실패를 기록하는 파일인 /var/log/btmp 파일을 대상으로 한다는 것을 제외하고는 last 명령과 같다.

NOTES
wtmp 파일이나, btmp 파일이 없을 수도 있다. 시스템은 이런 파일이 있을 경우에만, 로그인 정보를 그 파일에 기록한다. 만약 없는데, 이제부터 last 명 령 을 사용하고 싶다면, 단지 간단하게 touch 명령을 사용해서 각 파일을 만들어 놓으면 된다. (예를 들면, touch /var/log/wtmp).

서버를 이용하는 각 계정사용자들의 로그인 정보를 보여주는 명령어이다. 흔히 관리자는 각 계정별로 서버에 접속한 시간과 IP주소 등을 확인해야 할 경우가 있다. 또한 특정 계정의 서버 접속정보를 확인해야 할 때 에도 마찬가지이다. 다양한 방법으로 사용자들의 로그인정보를 조사한다.

[명령어 형식]

```
# last
# last root
# last -5                    /* 5행의 결과만을 확인 */
# last -R                    /* last의 결과에서 호스트(IP주소)접속기록을 제외한 결과만을 출력 */
# last -a                    /* last의 결과중 호스트(IP주소)정보를 맨 마지막에 출력 */
```

[명령어 옵션]

옵션	설명
-n	(num) 지정한 num 만큼의 줄만 보여준다.
-f	(file) 지정한 파일에서 정보를 불러온다.
-R	보여주는 목록에서 hostname(IP주소)필드는 보여주지 않는다.
-a	보여주는 목록에서 hostname(IP주소)필드를 마지막 필드에 보여준다.
-d	다른 호스트에서 접속한 것만 보여준다.
-x	shutdown이 일어난 상태나, run level이 바뀐 상태를 보여준다.

[EX1] last 명령어 사용
[TERM1] fedora 사용자의 터미널
 # ssh fedora@localhost
-> fedora 사용자의 암호 입력
$ tty

```
/dev/pts/#
```

[TERM2] 관리자 터미널
 # last /* 최근기록이 상단에 출력된다. head 명령어와 같이 사용된다 */

```
fedora   pts/2        192.168.0.1      Wed Feb 10 10:27   still logged in
root     pts/1        192.168.0.1      Wed Feb 10 10:25   still logged in
reboot   system boot  2.6.18-164.el5   Wed Feb 10 10:23          (00:08)
root     pts/2        linux200         Wed Jan 27 14:11 - 14:28  (00:16)
root     pts/1        192.168.0.1      Wed Jan 27 14:11 - 14:28  (00:17)
fedora   pts/2        192.168.0.1      Wed Jan 27 13:52 - 14:10  (00:18)
root     pts/1        192.168.0.1      Wed Jan 27 13:40 - 14:10  (00:29)
root     pts/4        :0.0             Wed Jan 27 13:39 - 13:39  (00:00)
root     pts/3        192.168.0.1      Wed Jan 27 13:30 - 13:39  (00:09)
root     pts/3        192.168.0.1      Wed Jan 27 13:29 - 13:30  (00:00)
root     :0                            Wed Jan 27 13:27 - crash (13+20:55)
root     :0                            Wed Jan 27 13:27 - 13:27  (00:00)
```

 # last -5 (# last | head -5) /* 상위 5개 라인 출력 */

```
fedora   pts/2        192.168.0.1      Wed Feb 10 10:27   still logged in
root     pts/1        192.168.0.1      Wed Feb 10 10:25   still logged in
reboot   system boot  2.6.18-164.el5   Wed Feb 10 10:23          (00:09)
root     pts/2        linux200         Wed Jan 27 14:11 - 14:28  (00:16)
root     pts/1        192.168.0.1      Wed Jan 27 14:11 - 14:28  (00:17)

wtmp begins Tue Jan 26 01:25:16 2010
```

[EX2] /var/log/wtmp.# 파일 읽기

```
# file /var/log/wtmp                    /* data 파일 타입을 가지고 있다 */
/var/log/wtmp: data

# file /etc/passwd                      /* text 파일의 경우에 확인 가능 */
/etc/passwd: ASCII text

# cat /var/log/wtmp                     /* 파일을 열수 있는 형식 아님 */
퀽:0root?rKNJ퀽:0:0root?r?
pts/1R.rKP(mpts/3S.smpts/4S.rK?쭊pts/1ts/1root192.168.0.1^.rKNp웸npts/1횲rKblpts/2JNrK懲
...... (중략) ......
```

```
# last -f /var/log/wtmp    (# last)
root     pts/1        192.168.0.1      Wed Feb 10 15:47   still logged in
root     pts/1        192.168.0.1      Wed Feb 10 12:56 - 14:40  (01:44)
root     :0                            Wed Feb 10 12:44   still logged in
root     :0                            Wed Feb 10 12:44 - 12:44  (00:00)
root     pts/4        linux200         Wed Feb 10 12:43 - 12:56  (00:12)
root     pts/3        linux200         Wed Feb 10 12:32 - 12:56  (00:23)
root     pts/2        192.168.0.1      Wed Feb 10 11:53 - 15:12  (03:18)
fedora   pts/2        192.168.0.1      Wed Feb 10 10:27 - 11:33  (01:05)
root     pts/1        192.168.0.1      Wed Feb 10 10:25 - 12:56  (02:30)
```

[참고] last -f 옵션 사용하는 경우

```
# cd /var/log
# ls wtmp*
wtmp  wtmp.0  wtmp.1  wtmp.2 ....

# last   -----> /var/log/wtmp
# last -f /var/log/wtmp.0  < 정책에 의해 로그가 일정 기간이 지나면 다른 파일로 저장된다 >
                           /etc/logrotate.conf 파일에 의해 결정
                           기간이 지난 파일의 로그인기록을 확인할 경우 사용한다.
```

[EX3] last 명령어의 사용 예

(개발자 요청 내용) 어제 파일(예: file.log)을 삭제한 사용자를 검색해 달라.

(정보1) 어제 파일이 지워졌다.

```
# last | grep 'Jun  8'
```

(정보2) 지워진 파일의 이름 : file.log

```
# cat ~/.bash_history
# cat ~/.bash_history | grep 'file.log' | grep rm
```

(작업 준비)

```
# cd /test
# chmod 777 /test
# touch file.log

# telnet localhost
```

fedora 사용자로 로그인

```
$ rm -rf /test/file.log
$ ls /test
```

-> file.log 파일이 지워졌는지 확인

```
$ exit

# last | grep 'Sep  9'
fedora   pts/4        linux249         Fri Sep  9 10:29 - 10:30  (00:00)
root     pts/3        :0.0             Fri Sep  9 10:11   still logged in
fedora   pts/2        linux249         Fri Sep  9 10:11   still logged in
root     pts/2        linux249         Fri Sep  9 10:10 - 10:10  (00:00)
root     pts/1        :0.0             Fri Sep  9 10:00   still logged in
root     pts/1        :0.0             Fri Sep  9 09:59 - 10:00  (00:00)
root     :0                            Fri Sep  9 09:59   still logged in
root     :0                            Fri Sep  9 09:59 - 09:59  (00:00)
reboot   system boot  2.6.18-164.el5   Fri Sep  9 09:56          (00:35)

# egrep -l "file.log" /home/*/.bash_history
/home/fedora/.bash_history

# egrep "file.log" /home/fedora/.bash_history
rm -rf /test/file.log
```

6 lastlog CMD

```
NAME
       lastlog - reports the most recent login of all users or
       of a given user

SYNOPSIS
       lastlog [options]

DESCRIPTION
       lastlog formats and prints the contents of the last
       login log /var/log/lastlog file. The login-name, port,
       and last login time will be printed. The default (no
       flags) causes lastlog entries to be printed, sorted by
       their order in /etc/passwd.

OPTIONS
       -t, --time DAYS
          Print the lastlog records more recent than DAYS.

       -u, --user LOGIN
          Print the lastlog record for user with specified
          LOGIN only.

       The -t flag overrides the use of -u.

       If the user has never logged in the message ** Never
       logged in** will be displayed instead of the port and
       time.

NOTE
       The lastlog file is a database which contains info on
       the last login of each user. You should not rotate it.
       It is a sparse file, so its size on the disk is usually
       much smaller than the one shown by "ls -l" (which can
       indicate a really big file if you have in passwd users
       with a high UID). You can display its real size with "ls
       -s".

FILES
       /var/log/lastlog
          Database times of previous user logins.
```

사용자의 마지막 로그인 정보만을 출력 해준다. lastlog는 /var/log/lastlog라는 파일의 내용을 출력해 준다. 로그인할 때 마지막 로그인 정보가 출력되는데 이때 출력되는 정보가 /var/log/lastlog의 정보이다. 사용자 계정을 지우게 되면 /var/log/lastlog 파일에도 사용자계정에 해당 정보도 삭제된다.

[명령어 형식]

```
# lastlog
# lastlog -u feodra                 /* 지정한 로그인명의 lastlog 정보만을 보여준다. */
# lastlog -t 3                      /* 지정한 날짜기간 안에 로그인한 정보만을 보여준다 */
```

[명령어 옵션]

옵션	설명
-u	지정된 사용자의 lastlog 기록을 보여줌
-t	지정된 날짜기간 안의 로그인 정보만 출력해 준다.

[EX1] lastlog 명령어 사용법

```
# telnet localhost (# ssh localhost)
login as: root
root@192.168.0.200's password:
Last login: Wed Feb 10 10:25:22 2010 from 192.168.0.1
# exit
```

 # lastlog

```
Username         Port     From             Latest
root             pts/2    linux249         Fri Sep  9 10:41:24 +0900 2011
bin                                        **Never logged in**
daemon                                     **Never logged in**
adm                                        **Never logged in**
lp                                         **Never logged in**
sync                                       **Never logged in**
shutdown                                   **Never logged in**
halt                                       **Never logged in**
mail                                       **Never logged in**

..... (중략) .....

rpcuser                                    **Never logged in**
named                                      **Never logged in**
hsqldb                                     **Never logged in**
sshd                                       **Never logged in**
haldaemon                                  **Never logged in**
avahi-autoipd                              **Never logged in**
xfs                                        **Never logged in**
gdm                                        **Never logged in**
sabayon                                    **Never logged in**
fedora           pts/4    linux249         Fri Sep  9 10:29:54 +0900 2011
user01           pts/3    linux249         Thu Sep  8 10:26:39 +0900 2011
```

 # lastlog -u fedora

```
Username         Port     From             Latest
fedora           pts/2    192.168.0.1      Wed Feb 10 10:27:47 +0900 2010
```

7 lastb CMD

NAME
last, lastb - 사용자들의 마지막 로그인했는 기록 목록을 보여준다.

SYNOPSIS
last [-R] [-num] [-n num] [-adx] [-f file] [name...] [tty...]
lastb [-R] [-num] [-n num] [-f file] [-adx] [name...] [tty...]

DESCRIPTION
Last 명령은 /var/log/wtmp 파일이나, -f 옵션에서 지정한파일을 통해서 사용자의 로그인, 로그아웃 시간을 보여준다. 특정 사용자의 이름이나, tty를 지정할 수 있으며, 이렇게 지정되면, 해당 사용자의 로그인 정보만을 보여준다. tty의 이름은 생략될 수 있어, last 0 명령은 last tty0 명령과 같은 기능을 한다. last가 실행 중에 SIGINT(인터럽트 글쇠, 주로 Ctrl-C에 의해서 만들어지는 신호)가 감지되거나, SIGQUIT(마침 글쇠, 주로 Ctrl-W에 의해서 만들어지는 신호)가감지되면, 실행을 마친다.

접속 사용자 이름에 reboot라고 나오는 것은 시스템이 reboot된 것을 나타낸다. 즉 last reboot 명령으로기록 파일이 만들어진 후부터 reboot 되었는 기록을 모두 볼 수 있다.

Lastb 명령은 접속 실패를 기록하는 파일인 /var/log/btmp 파일을 대상으로 한다는 것을 제외하고는 last 명령과 같다.

접속 실패 로그를 출력해준다. /var/log/btmp파일에 로그가 저장된다. cat이라는 명령어를 통해서 안의 내용을 열어보면 파일의 내용이 제대로 나오지 않는다. 이 파일의 내용을 보기 위해서 우리가 사용하는 명령어가 바로 lastb라는 명령어다. 보통 블루투스 공격(임의의 사용자 계정과 임의의 패스워드로 내 시스템에 접속하려는 것)을 당하는지, 당했는지와 어떤 사용자가 잘못 로그인을 했는지와 같은 정보를 확인 할 수 있다.

[명령어 형식]
lastb

[명령어 옵션]

옵션	설명
-n	(num) 지정한 num 만큼의 줄만 보여준다.
-f	(file) 지정한 파일에서 정보를 불러온다.
-R	보여주는 목록에서 hostname(IP주소)필드는 보여주지 않는다.
-x	shutdown이 일어난 상태나, run level이 바뀐 상태를 보여준다.

[EX1] lastb 명령어 실습

(전제 조건) user01 사용자가 존재해야 한다.
 # cat /etc/passwd | grep user01
 # useradd user01
 # passwd user01

 # ssh user01@localhost

```
user01@localhost's password: (1)        <------ 잘못된 암호 입력
Permission denied, please try again.
user01@localhost's password: (2)
Permission denied, please try again.
user01@localhost's password: (3)
Permission denied (publickey,gssapi-with-mic,password).
```

 # lastb

```
user01   ssh:notty    localhost.locald Sat Mar 23 15:49 - 15:49  (00:00)
user01   ssh:notty    localhost.locald Sat Mar 23 15:49 - 15:49  (00:00)
user01   ssh:notty    localhost.locald Sat Mar 23 15:49 - 15:49  (00:00)
rppt     :0                           Wed Feb 13 17:48 - 17:48  (00:00)
root     ssh:notty    172.16.9.1       Mon Feb  4 15:35 - 15:35  (00:00)
root     ssh:notty    172.16.9.1       Mon Feb  4 15:35 - 15:35  (00:00)
ftp      ssh:notty    172.16.9.1       Mon Feb  4 15:35 - 15:35  (00:00)
..... (중략) .....
```

```
# lastb -5              /* 지정된 만큼만 기록을 보여줌 */
user01   ssh:notty    localhost.locald Sat Mar 23 15:49 - 15:49  (00:00)
user01   ssh:notty    localhost.locald Sat Mar 23 15:49 - 15:49  (00:00)
user01   ssh:notty    localhost.locald Sat Mar 23 15:49 - 15:49  (00:00)
rppt     :0                           Wed Feb 13 17:48 - 17:48  (00:00)
root     ssh:notty    172.16.9.1       Mon Feb  4 15:35 - 15:35  (00:00)

btmp begins Wed Apr 11 13:33:02 2012
```

```
# lastb | grep user01
user01   ssh:notty    localhost.locald Sat Mar 23 15:49 - 15:49  (00:00)
user01   ssh:notty    localhost.locald Sat Mar 23 15:49 - 15:49  (00:00)
user01   ssh:notty    localhost.locald Sat Mar 23 15:49 - 15:49  (00:00)
```

```
# lastb | grep user01 | wc -l
3
```

기타 관리용 명령어

- who 명령어
```
# who
```
- w 명령어
```
# w
# w user01

# while true
> do
> CMD
> sleep
> done
```

8 who CMD

이름

who - 로그인한 사람들을 보여준다.

개요

who [-imqsuwHT] [--count] [--idle] [--heading] [--help] [--message] [--mesg] [--version] [--writable] [file] [am i]

설명

이 맨페이지는 GNU 버전의 who 를 다룬다. 옵션 아닌 인수가 없을 때 who 는 현재 로그인한 각 사용자에 대하여 다음 정보를 출력한다:

로그인명
터미널 라인
로그인 시간
원격 호스트 또는 X 디스플레이

옵션 아닌 인수가 하나 주어지면, who 는 /etc/utmp 를 사용하지 않고 그 인수를 로그인한 사용자 이름을 담고 있는화일로 간주하여 사용한 다. 보 통 /etc/wtmp 를 주면 who 는이전에 로그인했던 사람들의 목록을 보여준다.

옵션 아닌 인수가 2 개 주어지면, who 는 실행한 사용자에 대한 정보만 출력한다.(표준입력으로부터 결정) 전통적으로 두 개의 인수는 'am i' 로 서 'who am i' 가 된다.

옵션

-H, --heading
칼럼 헤더 라인을 출력한다.

who 명령을 통해 누가 로그인해 있는지, 어떤 장치를 이용하고 있는지, 언제 로그인했는지, 어디에서 로그인했는지 등의 정보를 알 수 있다. /var/run/utmp의 내용을 보여줌!

[명령어 형식]

```
# who                   /* 현재 시스템에 접속 중인 모든 사용자 */
# who -r                /* 현재 사용자의 Runlevel 확인 */
# who am i              /* 로그인한 사용자 정보 확인 */
# who -H                /* 헤드라인과 같이 출력 */
# whoami                /* 현재 사용자명 확인 , 유효사용자를 확인한다*/
```

[명령어 옵션]

옵션	기능
-i	idle time 과 함께 사용자 출력
-m	who 명령을 실행한 사용자 표시
-q	사용자 이름과 사용자수 출력
-w, -T	각 사용자의 메시지 설정 상태 출력
-H	헤드라인 정보 표시
-r	run-level 확인

[EX1] who 명령어 실습

[TERM1] fedora 사용자의 윈도우
 # telnet localhost
-> fedora 사용자로 로그인

$ id

```
uid=500(fedora) gid=500(fedora) groups=500(fedora)
```

$ pwd

```
/home/feodora
```

$ tty

```
/dev/pts/4
```

[TERM2] root 사용자의 윈도우
 # who

```
root     pts/1        Feb 10 15:47 (192.168.0.1)          /* Putty로 로그인하여 작업중 */
root     pts/2        Feb 10 16:34 (192.168.0.1)
fedora   pts/4        Feb 10 22:50 (192.168.0.1)
root     :0           Feb 10 12:44
root     pts/3        Feb 10 21:33 (192.168.0.1)
```

(명령어 출력 결과 해석)

```
------------------+-------------------------
필드 설명          |
------------------+-------------------------
root              |       사용자 정보
pts/1             |       제어 터미널
Feb 10 15:47      |       로그인 시간
192.168.0.1       |       원격호스트
------------------+-------------------------
```

[TERM1] fedora 사용자의 윈도우
$ exit
#

[EX2] who 명령어 실습

```
# ssh 172.16.9.252
The authenticity of host '172.16.9.252 (172.16.9.252)' can't be established.
RSA key fingerprint is 3b:25:5b:9a:ae:0f:ba:fc:85:66:73:d4:fc:e2:78:c6.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.9.252' (RSA) to the list of known hosts.
root@172.16.9.249's password: (centos)
Last login: Thu Jan  9 11:38:41 2014 from 172.16.9.202

# hostname
linux252.example.com

# id
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)

# tty
/dev/pts/4

# who
root     :0           2014-02-12 22:29
root     pts/1        2014-02-12 22:29 (:0.0)
root     pts/2        2014-02-13 10:29 (linux205.example.com)
root     pts/3        2014-02-13 10:31 (linux206.example.com)
root     pts/4        2014-02-13 10:29 (linux249.example.com)
root     pts/5        2014-02-13 10:29 (linux208.example.com)
root     pts/6        2014-02-13 10:29 (linux218.example.com)
root     pts/7        2014-02-13 10:29 (linux201.example.com)
root     pts/8        2014-02-13 10:30 (linux207.example.com)
root     pts/9        2014-02-13 10:30 (linux219.example.com)
root     pts/10       2014-02-13 10:30 (linux214.example.com)
root     pts/11       2014-02-13 10:31 (linux200.example.com)
..... (중략) .....

# exit
#
```

9 w CMD

```
NAME
       w - Show who is logged on and what they are doing.

SYNOPSIS
       w - [husfV] [user]

DESCRIPTION
       w  displays  information  about the users currently on the machine, and
       their processes.  The header shows, in this order,  the  current  time,
       how  long  the  system  has  been running, how many users are currently
       logged on, and the system load averages for the past 1, 5, and 15  min-
       utes.

       The  following entries are displayed for each user: login name, the tty
       name, the remote host, login time, idle time, JCPU, PCPU, and the  com-
       mand line of their current process.

       The  JCPU  time  is the time used by all processes attached to the tty.
       It does not include past background jobs, but  does  include  currently
       running background jobs.

       The  PCPU  time  is  the time used by the current process, named in the
       "what" field.
```

시스템에 login한 사용자가 어떤 명령어를 실행하고 있는지 알아보는 명령어이며, /proc 디렉토리로 부터 사용자에 대한 정보와 실행중인 명령어에 대한 정보를 추출해 낸다.

[명령어 형식]
 # w
 # w user01

 # w

```
 10:34:14 up 28 min,  3 users,  load average: 0.00, 0.01, 0.01
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root     :0                        10:07   ?xdm?  12.60s  0.19s /usr/bin/gnome-ses
root     pts/1    :0.0             10:10    0.00s  0.03s  0.00s w
root     pts/3    :0.0             10:27    7:00   0.02s  0.02s bash
```

[EX1] w 명령어 실습
[TERM1] root 사용자 윈도우
-> putty 프로그램을 통해 리눅스 서버에 접속
login : **root**
passwd : **(centos)**

[TERM2] fedora 사용자 윈도우
 # ssh fedora@localhost

```
fedora@localhost's password: (fedora)
Last login: Thu Feb 13 10:26:29 2014 from localhost.localdomain
```

[TERM3] user01 사용자 윈도우
 # telnet localhost
user01 사용자로 로그인
$ vi /etc/passwd

[TERM4] root 사용자 윈도우(분석하는 터미널)
 # w

```
 10:43:47 up 38 min,  7 users,  load average: 0.00, 0.03, 0.01
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root     :0                        10:07   ?xdm?  14.78s  0.19s /usr/bin/gnome-ses
root     pts/1    :0.0             10:10    2:56   0.03s  0.00s ssh fedora@localho
root     pts/3    :0.0             10:27   48.00s  0.02s  0.00s telnet localhost
root     pts/2    172.16.9.1       10:39    3:43   0.01s  0.01s -bash
fedora   pts/4    localhost.locald 10:40    2:51   0.01s  0.01s -bash
user01   pts/5    localhost.locald 10:42   48.00s  0.04s  0.03s vim /etc/passwd
root     pts/6    :0.0             10:43    0.00s  0.01s  0.00s w
```

[EX2] 악의적인 사용자 로그아웃 시키기
[TERM4] 관리자 윈도우

```
# w
 10:43:47 up 38 min,  7 users,  load average: 0.00, 0.03, 0.01
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
root     :0                        10:07   ?xdm?  14.78s  0.19s /usr/bin/gnome-ses
root     pts/1    :0.0             10:10    2:56   0.03s  0.00s ssh fedora@localho
root     pts/3    :0.0             10:27   48.00s  0.02s  0.00s telnet localhost
root     pts/2    172.16.9.1       10:39    3:43   0.01s  0.01s -bash
fedora   pts/4    localhost.locald 10:40    2:51   0.01s  0.01s -bash
user01   pts/5    localhost.locald 10:42   48.00s  0.04s  0.03s vim /etc/passwd
root     pts/6    :0.0             10:43    0.00s  0.01s  0.00s w

# w user01
 10:48:23 up 42 min,  7 users,  load average: 0.00, 0.02, 0.00
USER     TTY      FROM              LOGIN@   IDLE   JCPU   PCPU WHAT
user01   pts/5    localhost.locald 10:42    5:24   0.04s  0.03s vim /etc/passwd

# while true
> do
> w user01
> sleep 2
> done

# ps -u user01    (# ps -fu user01)
  PID TTY          TIME CMD
 8064 pts/5    00:00:00 bash
 8100 pts/5    00:00:00 vim

# kill -9 8064
#
```

[참고] 원격서버에 접속시
- 원격서버에 접속하여 접속한 서버의 이름, 사용자, 작업 디렉토리 확인

```
# ssh 172.16.9.252
```
root 사용자로 로그인

```
# hostname
# id
# pwd
```

[참고] 관리 서버에 접속시
- 자신이 관리하는 서버에 접속하여 오늘 로그인/로그아웃 사용자 정보 확인
- 현재 접속되어져 있는 사용자 확인
- 현재 접속된 사용자가 수행하는 명령어 확인

```
# last
# who
# w
```

10 exit CMD

NAME
exit - cause the shell to exit

SYNOPSIS
exit [n]

DESCRIPTION
The exit utility shall cause the shell to exit with the exit status specified by the unsigned decimal integer n. If n is specified, but its value is not between 0 and 255 inclusively, the exit status is undefined.

A trap on EXIT shall be executed before the shell terminates, except when the exit utility is invoked in that trap itself, in which case the shell shall exit immediately.

현재의 프로세스(현재쉘)를 종료한다.

[명령어 형식]
exit [Number] /* 값을 지정해준다면 0은 정상종료, 1~255는 비정상 종료 */

[EX1] 서브 쉘 종료 해보자
ps

```
  PID TTY          TIME CMD
21224 pts/11   00:00:00 bash
```

bash
ps

```
  PID TTY          TIME CMD
21224 pts/11   00:00:00 bash
21316 pts/11   00:00:00 bash
```

exit

```
exit
```

ps

```
  PID TTY          TIME CMD
21224 pts/11   00:00:00 bash
```

[EX2] 서버에 접속 후 로그 아웃
ssh 172.16.9.252
root 사용자로 접속

hostname
id
pwd

ps
exit
#

[EX3] 사용자 전환후에 원래 사용자로 전환
# su - user01
$ ps
$ exit
#