

## Ch 5~10

1. type : 지정된 명령어의 종류를 알려준다.

```
[s21500630@peace:~]$ type type
type is a shell builtin
[s21500630@peace:~]$ type ls
ls is aliased to `ls --color=auto'
[s21500630@peace:~]$ type cp
cp is /bin/cp
```

2. which : 명령어의 위치를 알려준다.

```
[s21500630@peace:~]$ which ls
/bin/ls
```

3. help : 명령어를 어떻게 사용하는지, 어떤 옵션이 있는지 알려준다.

```
[s21500630@peace:~]$ help cd
cd: cd [-L|[-P [-e]] [-@]] [dir]
    Change the shell working directory.

    Change the current directory to DIR.  The default DIR is the value of the
    HOME shell variable.

    The variable CDPATH defines the search path for the directory containing
    DIR.  Alternative directory names in CDPATH are separated by a colon (:).
    A null directory name is the same as the current directory.  If DIR begins
    with a slash (/), then CDPATH is not used.

    If the directory is not found, and the shell option `cdable_vars' is set,
    the word is assumed to be a variable name.  If that variable has a value,
    its value is used for DIR.

    Options:
      -L      force symbolic links to be followed: resolve symbolic links in
              DIR after processing instances of `..'
      -P      use the physical directory structure without following symbolic
              links: resolve symbolic links in DIR before processing instances
              of `..'
      -e      if the -P option is supplied, and the current working directory
              cannot be determined successfully, exit with a non-zero status
      -@      on systems that support it, present a file with extended attributes
              as a directory containing the file attributes

    The default is to follow symbolic links, as if `-L' were specified.
    `..' is processed by removing the immediately previous pathname component
    back to a slash or the beginning of DIR.

    Exit Status:
    Returns 0 if the directory is changed, and if $PWD is set successfully when
    -P is used; non-zero otherwise.
```

4. man : 명령어의 매뉴얼을 보여준다.

```

LS(1)                                User Commands                                LS(1)

NAME
    ls - list directory contents

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

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

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

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

    --author
        with -l, print the author of each file

    -b, --escape
        print C-style escapes for nongraphic characters

    --block-size=SIZE
        scale sizes by SIZE before printing them; e.g., '--block-size=M'
        prints sizes in units of 1,048,576 bytes; see SIZE format below

    -B, --ignore-backups
        do not list implied entries ending with ~

    -c      with -lt: sort by, and show, ctime (time of last modification of
Manual page ls(1) line 1/234 14% (press h for help or q to quit)

```

5. apropos : 매뉴얼 페이지에서 특정한 단어를 검색해서 어떤 매뉴얼에 몇페이지에 있는지 보여준다.

```

[s21500630@peace:~]$ apropos partition
addpart (8)          - tell the kernel about the existence of a partition
all-swaps (7)        - event signalling that all swap partitions have been ac...
cfdisk (8)           - display or manipulate a disk partition table
cgdisk (8)           - Curses-based GUID partition table (GPT) manipulator
delpart (8)          - tell the kernel to forget about a partition
fdisk (8)            - manipulate disk partition table
fixparts (8)         - MBR partition table repair utility
gdisk (8)            - Interactive GUID partition table (GPT) manipulator
mptable (1)          - partition an MSDOS hard disk
partprobe (8)        - inform the OS of partition table changes
partx (8)            - tell the kernel about the presence and numbering of on...
resizepart (8)       - tell the kernel about the new size of a partition
sfdisk (8)           - display or manipulate a disk partition table
sgdisk (8)           - Command-line GUID partition table (GPT) manipulator fo...
systemd-gpt-auto-generator (8) - Generator for automatically discovering and ...

```

6. whatis : 명령어의 이름과 어떤 명령어인지 한줄로 간단하게 보여준다.

```

[s21500630@peace:~]$ whatis ls
ls (1)          - list directory contents

```

7. info : 더 상세한 매뉴얼 페이지를 보여준다.

```

Next: Introduction, Up: (dir)

GNU Coreutils
*****

This manual documents version 8.25 of the GNU core utilities, including
the standard programs for text and file manipulation.

Copyright © 1994–2016 Free Software Foundation, Inc.

Permission is granted to copy, distribute and/or modify this
document under the terms of the GNU Free Documentation License,
Version 1.3 or any later version published by the Free Software
Foundation; with no Invariant Sections, with no Front-Cover Texts,
and with no Back-Cover Texts. A copy of the license is included in
the section entitled “GNU Free Documentation License”.

* Menu:

* Introduction::          Caveats, overview, and authors
* Common options::       Common options
* Output of entire files:: cat tac nl od base32 base64
* Formatting file contents:: fmt pr fold
* Output of parts of files:: head tail split csplit
* Summarizing files::    wc sum cksum md5sum sha1sum sha2
* Operating on sorted files:: sort shuf uniq comm ptx tsort
* Operating on fields::  cut paste join
* Operating on characters:: tr expand unexpand
* Directory listing::    ls dir vdir dircolors
* Basic operations::     cp dd install mv rm shred
* Special file types::   mkdir rmdir unlink mkfifo mknod ln link readlink
* Changing file attributes:: chgrp chmod chown touch
* Disk usage::           df du stat sync truncate
* Printing text::        echo printf yes
* Conditions::           false true test expr
-----Info: (coreutils.info.gz)Top, 340 lines --Top-----

```

8. alias : 특정 명령어를 별명으로 지정해두고 계속 사용할 수 있다.

```

s21500630@peace:~$ alias foo='cd /usr; ls; cd -'
s21500630@peace:~$ alias name = 'string'
-bash: alias: name: not found
-bash: alias: =: not found
-bash: alias: string: not found
s21500630@peace:~$ alias name= 'string'
-bash: alias: string: not found
s21500630@peace:~$ alias name='string'
s21500630@peace:~$ foo
bin  games  include  lib  local  locale  sbin  share  src
/home/s21500630

```

9. ls -l /usr/bin > ls-output.txt : ls -l을 한 결과를 ls-output.txt 파일에 저장

```

s21500630@peace:~$ ls -l /usr/bin > ls-output.txt
s21500630@peace:~$ ls
a.out  calculator  examples.desktop  hello.txt  ls-output.txt

```

10. ls -l /usr/bin >> ls-output.txt : ls -l을 한 결과를 ls-output.txt 파일에 이어쓴다. 만약 파일이 존재하지 않는다면 새로운 파일을 만들어 저장한다.

11. cat [file...] : 파일을 터미널에 출력한다.

```

s21500630@peace:~$ cat ls-output.txt
total 275388
-rwxr-xr-x 1 root root      51920  3월  3  2017 [
lrwxrwxrwx 1 root root         8  1월  22  20:47 2to3 -> 2to3-2.7
-rwxr-xr-x 1 root root       96 11월  13  04:40 2to3-2.7
-rwxr-xr-x 1 root root       96 11월  13  01:18 2to3-3.5
-rwxr-xr-x 1 root root     10392  1월  31  2016 411toppm
-rwxr-xr-x 1 root root     19208  4월  15  2016 a11y-profile-manager-indicator
-rwxr-xr-x 1 root root     22696  9월  28  07:10 aa-enabled
-rwxr-xr-x 1 root root     19064  4월  15  2016 aconnect
-rwxr-xr-x 1 root root     15072  4월   9  2016 acpi_listen

```

12. cat : 명령어만 입력하고 터미널에 문자열을 입력하고 엔터를 누르면 그대로 출력된다.

```
s21500630@peace:~$ cat
the quck brown fox jumped over the lazy dog.
the quck brown fox jumped over the lazy dog.
```

13. | (pipelines) : 첫번째 명령어를 다음 명령어의 입력 값으로 사용. ls -l 의 값이 less 의 입력값으로 전달되어 출력된다.

```
s21500630@peace:~$ ls -l /usr/bin | less
total 275388
-rwxr-xr-x 1 root root      51920  3월  3  2017 [
lrwxrwxrwx 1 root root         8  1월 22 20:47 2to3 -> 2to3-2.7
-rwxr-xr-x 1 root root       96 11월 13 04:40 2to3-2.7
-rwxr-xr-x 1 root root       96 11월 13 01:18 2to3-3.5
-rwxr-xr-x 1 root root    10392  1월 31  2016 411toppm
-rwxr-xr-x 1 root root    19208  4월 15  2016 a11y-profile-manager-indicator
-rwxr-xr-x 1 root root    22696  9월 28 07:10 aa-enabled
-rwxr-xr-x 1 root root    19064  4월 15  2016 aconnect
-rwxr-xr-x 1 root root    15072  4월  9  2016 acpi_listen
-rwxr-xr-x 1 root root   194288  3월 28  2017 activity-log-manager
-rwxr-xr-x 1 root root     6356 12월 19 02:08 add-apt-repository
-rwxr-xr-x 1 root root    18888  5월 17  2018 addpart
lrwxrwxrwx 1 root root         26 12월  8 04:49 addr2line -> x86_64-linux-gnu-a
ddr2line
-rwxr-xr-x 1 root root    35560  4월 15  2016 alsabat
-rwxr-xr-x 1 root root    73224  4월 15  2016 alsaloop
-rwxr-xr-x 1 root root    65496  4월 15  2016 alsamixer
-rwxr-xr-x 1 root root    10504  4월 15  2016 alsatplg
```

14. ls /bin /usr/bin | sort | less : 두개의 다른 경로의 리스트를 합쳐서 정렬하고 less로 출력한다.

15. uniq: 정렬과 함께 주로 사용. 중복되는 값 중 하나를 삭제하여 하나의 값만 남긴다. uniq -d를 하면 중복되는 값들을 보여준다.

16. wc : 파일안의 줄 수, 단어수, 바이트 수를 보여준다. wc -l을 사용하면 줄 수만 보여준다.

```
s21500630@peace:~$ wc ls-output.txt
 3608   33830 231912 ls-output.txt
```

17. grep pattern file : file 에서 pattern을 찾아 모두 보여준다.

```
s21500630@peace:~$ ls /bin /usr/bin | sort | uniq | grep zip
bunzip2
bzip2
bzip2recover
funzip
gpg-zip
gunzip
gzip
mzip
preunzip
prezip
prezip-bin
unzip
unzipsfx
zip
zipcloak
zipdetails
zipgrep
zipinfo
zipnote
zipsplit
```

18. echo : 텍스트를 출력해준다. \*을 붙이면 파일이름을 모두 출력한다..

```
s21500630@peace:~$ echo this is a test
this is a test
s21500630@peace:~$ echo *
a.out calculator examples.desktop hello.txt lazy_dog.txt lazydog.txt ls-error.tx
t ls-output.txt
```

19. \* : 경로 뒤 문자에 붙이면 해당 문자로 시작되는 파일들이, 앞에 붙이면 해당 문자로 끝나는 파일들을 출력

20. echo 로 연산결과도 출력 가능.  $((expression))$ 을 사용하면 연산이 가능하다.

```
[s21500630@peace:~]$ echo $((5*2)*3)
75
```

21. {} : 괄호 안의 콤마로 구분된 요소들을 앞 뒤 문자와 연결한다.

```
[s21500630@peace:~]$ echo Number_{1..5}
Number_1 Number_2 Number_3 Number_4 Number_5
[s21500630@peace:~]$ echo {01..15}
01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
```

22. {} 는 새로운 디렉토리를 일괄적으로 만들때 유용하게 사용가능

```
[s21500630@peace:~]$ mkdir Photos
[s21500630@peace:~]$ cd Photos
[s21500630@peace:~/Photos]$ mkdir {2007..2009}-{01..12}
[s21500630@peace:~/Photos]$ ls
2007-01 2007-05 2007-09 2008-01 2008-05 2008-09 2009-01 2009-05 2009-09
2007-02 2007-06 2007-10 2008-02 2008-06 2008-10 2009-02 2009-06 2009-10
2007-03 2007-07 2007-11 2008-03 2008-07 2008-11 2009-03 2009-07 2009-11
2007-04 2007-08 2007-12 2008-04 2008-08 2008-12 2009-04 2009-08 2009-12
```

23. 다른 명령문의 output을 expansion으로 사용 가능.

```
[s21500630@peace:~]$ echo $(ls)
a.out calculator examples.desktop hello.txt lazy_dog.txt lazydog.txt ls-error.tx
t ls-output.txt Photos
```

24. " " : 쌍따옴표를 사용하면 공백이 있는 변수나 파일을 오류없이 불러올 수 있음. 쌍따옴표가 없으면 리눅스에서 공백이 저절로 제거되기 때문. 공백을 유지하고 싶다면 쌍따옴표를 넣어준다.

25. ctrl-a : 커서를 줄의 제일 처음으로 보낸다.

26. ctrl-e : 커서를 줄의 제일 끝으로 보낸다.

27. ctrl-f : 커서를 한 칸 앞으로 보낸다.

28. ctrl-b : 커서를 한 칸 뒤로 보낸다.

29. alt-f : 커서를 한 단어 앞으로 보낸다.

30. alt-b : 커서를 한 단어 뒤로 보낸다.

31. ctrl-l : 화면을 초기화하고 커서를 좌측 상단으로 보낸다.

32. ctrl-d : 커서 자리에 있는 글자를 지운다.

33. ctrl-t : 커서 자리에 있는 글자와 그 바로 앞자리에 있는 글자를 뒤바꾼다.

- 34. alt-l : 커서가 있는 자리부터 끝까지 모든 글자들을 소문자로 바꾼다.
- 35. alt-u : 커서가 있는 자리부터 끝까지 모든 글자들을 대문자로 바꾼다.
- 36. ctrl-k : 커서가 있는 자리부터 끝까지 모든 글자들을 지운다.
- 37. ctrl-u : 커서가 있는 자리부터 첫번째까지 모든 글자들을 지운다.
- 38. alt-d : 커서가 있는 자리부터 현재 단어의 끝까지 글자들을 지운다
- 39. alt-Backspace : 커서가 있는 자리부터 현재 단어의 가장 앞 글자까지의 글자들을 지운다.
- 40. 명령어를 입력한 상태에서 탭을 누르면 매치되는 파일 이름을 자동으로 완성시켜준다.
- 41. history : 입력한 명령어의 기록을 볼 수 있다. 최대 500개의 커맨드 기록을 저장.

```
s21500630@peace:~$ history
1  do-release-upgrade
2  exit
3  ls
4  als - al
5  as -al
6  ls -al
7  tree -al
8  cd
9  pwd
10 cd ..
11 ls
12 cd ..
13 ls
14 tree -al
15 vi
```

- 42. 특정한 디렉토리에서 사용한 명령기록을 보고 싶으면 파이프와 grep을 사용한다.

```
s21500630@peace:~$ history | grep /usr/bin
81  cd /usr/bin
85  /home/s21500630/usr/bin
91  cd /usr/bin
139 ls -l /usr/bin > ls-output.txt
143 ls -l /usr/bin >> ls-output.txt
161 ls -l /usr/bin | less
163 ls /bin /usr/bin | sort | uniq | grep zip
190 history | grep /usr/bin
```

- 43. ctrl-p : 이전 명령어 기록으로 이동
- 44. ctrl-n : 다음 명령어 기록으로 이동
- 45. alt-< : 첫 명령어 기록으로 이동
- 46. alt-> : 마지막 명령어 기록으로 이동
- 47. ctrl-r : 역순으로 기록 출력

48. ctrl-o : 현재 기록에 있는 명령을 실행

49. id : 현재 아이디의 정보를 보여준다.

```
[s21500630@peace:~$ id  
uid=1064(s21500630) gid=1065(s21500630) groups=1065(s21500630)
```

50. r: 읽기권한, w: 쓰기권한, x: 실행권한 // 유저-그룹-외부인

```
[s21500630@peace:~$ ls -l foo.txt  
-rw-rw-r-- 1 s21500630 s21500630 0 3월 25 01:46 foo.txt  
s21500630@peace:~$
```

51. rwx(111) 을 이진수로 표현해서 지정할 수 있고, 8진수로 각 권한은 표현할 수 있다.

7 - 111 - wrx //

chmod 600 foo.txt 라는 명령을 입력하면 foo.txt 의 권한을 rw----로 변경하라는 뜻.

```
[s21500630@peace:~$ ls -l foo.txt  
-rw----- 1 s21500630 s21500630 0 3월 25 01:46 foo.txt  
s21500630@peace:~$
```

52. u : 유저 // g : 그룹 // o : 외부인 // a : 전체

53. u+x : 유저에게 실행 권한을 추가

54. u-x: 유저에게 실행 권한을 주지 않는다.

55. go=rw : 그룹과 외부인에게 읽기와 쓰기권한 부여

56. umask : 기본 권한 세팅 0000으로 권한 표현

```
[s21500630@peace:~$ > foo.txt  
[s21500630@peace:~$ ls -l foo.txt  
-rw-rw-r-- 1 s21500630 s21500630 0 3월 25 01:58 foo.txt  
[s21500630@peace:~$ rm foo.txt  
[s21500630@peace:~$ umask 0000  
[s21500630@peace:~$ > foo.txt  
[s21500630@peace:~$ ls -l foo.txt  
-rw-rw-rw- 1 s21500630 s21500630 0 3월 25 01:59 foo.txt  
s21500630@peace:~$
```

57. su - user : 다른 계정으로 셸 시작. 계정 접속 후 exit 명령어로 접속을 종료하면 이전에 로그인했던 계정으로 돌아온다.

```
[s21500630@peace:~$ su -s21500630  
Password: 
```

58. sudo : su 와 비슷하지만 sudo를 사용하면 관리자의 비밀번호를 알지 못해서 허용된 권한내에서 접근할 수 있다.

59. chown : sudo chown owner: file 을 하면 파일의 소유자를 변경할 수 있다.

60. passwd : 비밀번호를 변경할 수 있다.

61. ps : 현재 프로세스의 상태를 보여준다.

```
PID TTY          TIME CMD
14821 pts/9        00:00:00 bash
14926 pts/9        00:00:00 ps
```

62. ps -x : 모든 프로세스를 다 보여준다.

```
s21500630@peace:~$ ps x
  PID TTY          STAT TIME COMMAND
 14786 ?            Ss     0:00 /lib/systemd/systemd --user
 14787 ?            S       0:00 (sd-pam)
 14820 ?            R       0:00 sshd: s21500630@pts/9
 14821 pts/9        Ss     0:00 -bash
 15889 pts/9        R+     0:00 ps x
```

63. STAT 상태

1. R : 동작 중
2. S : 대기 중
3. D : 입출력 대기 중
4. T : 정지됨
5. Z : 자식 프로세스가 종료되었지만 아직 부모 프로세스에 의해 처리되지 않음
6. < : 최우선 프로세스
7. N : 낮은 우선순위의 프로세스

64. ps aux : 권한정보를 추가해서 프로세스들을 보여준다.

```
s21500630@peace:~$ ps aux
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root             1  0.0  0.0 185860 6524 ?        Ss   3월 19   0:25 /lib/systemd/
root             2  0.0  0.0      0     0 ?        S    3월 19   0:00 [kthreadd]
root             4  0.0  0.0      0     0 ?        I<   3월 19   0:00 [kworker/0:0H
root             6  0.0  0.0      0     0 ?        I    3월 19   0:00 [kworker/u80:
root             7  0.0  0.0      0     0 ?        I<   3월 19   0:00 [mm_percpu_wq
root             8  0.0  0.0      0     0 ?        S    3월 19   0:00 [ksoftirqd/0]
root             9  0.0  0.0      0     0 ?        I    3월 19   0:57 [rcu_sched]
root            10  0.0  0.0      0     0 ?        I    3월 19   0:00 [rcu_bh]
root            11  0.0  0.0      0     0 ?        S    3월 19   0:00 [initiation/0]
```

65. signals

1. (1) HUP : 터미널 세션을 끝낸다.
2. (2) INT : ctrl - c 와 같은 기능. 프로그램을 중지시킨다.
3. (9) KILL : 프로그램을 종료시킨다. 저장하지 않고 바로 종료시킴
4. (15) TERM : 프로그램이 시그널을 받을 수 있는 상태라면 종료시킨다.
5. (18) CONT : 중지된 프로세스를 다시 시작한다.
6. (19) STOP : 프로그램을 종료시키지 않고 일시정지 시킨다.
7. (20) TSTP : 프로그램을 정지시킨다. 하지만 프로그램이 정지할지 선택할 수 있다.

66. reboot : sudo reboot : 시스템을 재시작한다.

67. shutdown : sudo shutdown -h now : 시스템을 종료시킨다.