

2021년 2학기 운영체제실습 2주차

Unix/Linux Commands

System Software Laboratory

School of Computer and Information Engineering

Kwangwoon Univ.

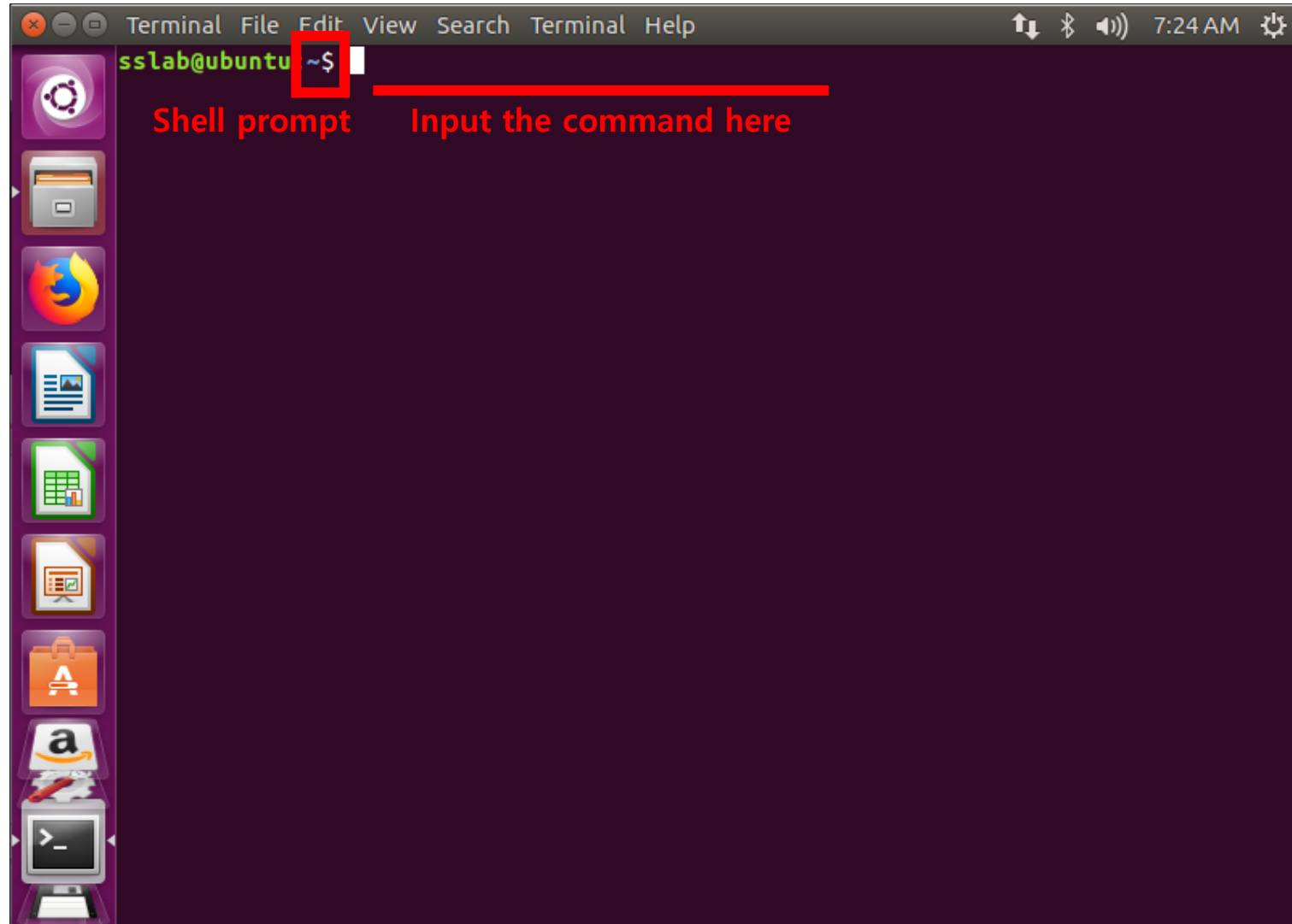
Preparation

- 같이 배포된 `splab_commands` 파일을 아래와 같이 실행
 - \$ `chmod +x splab_commands`
 - \$ `./splab_commands`
 - 실행하는 위치는 상관 없음

```
ssl@ubuntu:~$ ls -l
total 68
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Desktop
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Documents
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Downloads
-rw-r--r-- 1 ssl ssl 8980 Mar 17 05:32 examples.desktop
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Music
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Pictures
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Public
-rwx--x--x 1 ssl ssl 21261 Mar 18 01:13 splab_commands
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Templates
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Videos
ssl@ubuntu:~$ chmod +x splab_commands
ssl@ubuntu:~$ ./splab_commands
ssl@ubuntu:~$ ls -l
total 72
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Desktop
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Documents
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Downloads
-rw-r--r-- 1 ssl ssl 8980 Mar 17 05:32 examples.desktop
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Music
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Pictures
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Public
-rwx--x--x 1 ssl ssl 21261 Mar 18 01:13 splab_commands
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Templates
drwxr-xr-x 2 ssl ssl 4096 Mar 17 05:44 Videos
drwxrwxr-x 3 ssl ssl 4096 Mar 18 20:26 work
ssl@ubuntu:~$
```

Linux Terminal

- Command Line Interface



man

- **Format and display the on-line manual pages**

- Usage: man [option] name ...

- E.g.

- \$ man ls

- \$ man -k copy

//keyword search

- \$ man -a write

//all manuals

man

- e.g. `$ man kill`

```
KILL(1)                                User Commands                                KILL(1)

NAME
    kill - send a signal to a process

SYNOPSIS
    kill [options] <pid> [...]

DESCRIPTION
    The default signal for kill is TERM. Use -l or -L to list available signals. Particularly useful signals include HUP, INT, KILL, STOP, CONT, and 0. Alternate signals may be specified in three ways: -9, -SIGKILL or -KILL. Negative PID values may be used to choose whole process groups; see the PGID column in ps command output. A PID of -1 is special; it indicates all processes except the kill process itself and init.

OPTIONS
    <pid> [...]
        Send signal to every <pid> listed.

    -<signal>
    -s <signal>
    --signal <signal>
        Specify the signal to be sent. The signal can be specified by using name or number. The behavior of signals is explained in signal(7) manual page.

    -l, --list [signal]
        List signal names. This option has optional argument, which will convert signal number to signal name, or other way round.

    -L, --table
        List signal names in a nice table.

NOTES
    Your shell (command line interpreter) may have a built-in kill command. You may need to run the command described here as /bin/kill to solve the conflict.

EXAMPLES
    kill -9 -1
        Kill all processes you can kill.

    kill -l 11
        Translate number 11 into a signal name.

    kill -L
        List the available signal choices in a nice table.
```

man

- **Section description**

- (1) General commands
- (2) System calls
- (3) C library functions
- (4) Special files (usually devices) and drivers
- (5) File formats and conventions
- (6) Games and screensavers
- (7) Miscellanea
- (8) System administration commands and daemons

- **Examples**

- ls(1), open(2), fopen(3)
- write(1)/write(2)
 - \$ man 1 write
 - \$ man 2 write

man

- **Manual layout**

- NAME
 - name of the command or function
- SYNOPSIS
 - command: how to run,
 - functions: parameter list
- DESCRIPTION
 - description of the functioning of the command or function.
- EXAMPLES
 - some examples of common usage.
- SEE ALSO
 - list of related commands or functions.
- OPTIONS, EXIT STATUS, ENVIRONMENT, KNOWN BUGS, FILES, AUTHOR, REPORTING BUGS, HISTORY and COPYRIGHT.

uname

- **Display system information**

- usage: `uname [options]`
- e.g.

```
sslab@ubuntu:~$ uname
Linux
sslab@ubuntu:~$ uname -r
4.15.0-46-generic
sslab@ubuntu:~$ uname -m
x86_64
sslab@ubuntu:~$ uname -a
Linux ubuntu 4.15.0-46-generic #49~16.04.1-Ubuntu SMP Tue Feb 12 17:45:24 UTC 2019 x86_64
x86_64 x86_64 GNU/Linux
```

- useful options
 - `-r` : print the kernel release
 - `-m`: print the machine hardware name
 - `-a` : print all information

passwd

- **Update a user's authentication tokens**
 - usage: passwd [options]
 - e.g.

```
ssl@ubuntu:~$ passwd
Changing password for ssl.
(current) UNIX password:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
ssl@ubuntu:~$
```

ls

- List directory contents

- usage: ls [OPTION]... [FILE]...
- useful options
 - a : hidden file을 포함한 모든 파일을 출력
 - F : 파일 종류 표시 (/는 디렉토리, *는 실행파일)
 - l : 파일 정보를 자세하게 출력

```
sslab@ubuntu:~$ ls
Desktop      Downloads    Music        Public        Templates    work
Documents    examples.desktop  Pictures     splab_commands  Videos
```

```
sslab@ubuntu:~$ ls -a
.              .config      .gconf       .profile      .viminfo
..             Desktop      .gnupg       Public         work
.bash_history  .dmrc        .ICEauthority splab_commands .Xauthority
.bash_logout   Documents    .local       .sudo_as_admin_successful .xsession-errors
.bashrc        Downloads    Music        Templates     .xsession-errors.old
.cache         examples.desktop  Pictures     Videos
```

```
sslab@ubuntu:~$ ls -F
Desktop/      Downloads/    Music/        Public/        Templates/    work/
Documents/    examples.desktop  Pictures/     splab_commands*  Videos/
```

```
sslab@ubuntu:~$ ls -al
total 140
drwxr-xr-x 16 sslab sslab  4096 Mar 22 19:38 .
drwxr-xr-x  3 root  root   4096 Mar 17 05:32 ..
-rw-r--r--  1 sslab sslab  1174 Mar 17 07:09 .bash_history
-rw-r--r--  1 sslab sslab   220 Mar 17 05:32 .bash_logout
-rw-r--r--  1 sslab sslab  3771 Mar 17 05:32 .bashrc
```

| pwd

- **Print name of current working directory**
 - usage: pwd [OPTION]
 - e.g.

```
ssl@ubuntu:~$ pwd
/home/ssl
ssl@ubuntu:~$
```

mkdir

- **Make directories**

- usage: mkdir [OPTION] DIRECTORY...
- e.g.

```
ssl@ubuntu:~$ cd work
ssl@ubuntu:~/work$ ls -l
total 20
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
ssl@ubuntu:~/work$ mkdir SP lecture
ssl@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
drwxrwxr-x 2 sslab sslab 4096 Aug 28 21:19 SP_lecture
ssl@ubuntu:~/work$
```

File permission

- 파일 허가 지정

- owner, group, others 세 종류에 대해 아래의 권한 부여
 - read
 - file: 파일 내용 열람/복사 가능, 수정/삭제 불가능
 - directory: 디렉토리 내의 파일이름 열람 가능(ls)
 - write
 - file: 파일 내용 수정/삭제 가능, 열람/복사 불가능
 - directory: 파일을 생성하거나 삭제할 수 있는 권리
 - execute
 - file: 실행시킬 권리의 유무
 - directory: 이동 가능 여부(cd)

chmod

- Change file access permissions

- usage 1: chmod [OPTION]... MODE[,MODE]... FILE...

- MODE

- 대상

- u: user (owner)

- g: group

- o: other

- a: all

- 연산

- +: 추가

- : 제거

- =: 할당

- 권한

- r: read

- w: write

- x: execution

- e.g.

- chmod a=rwx test → test: 모든 대상에게 모든 권한 부여

- chmod a+r,o-w test → test: 모든 대상에게 읽기 부여, other는 write 제거

```
sslab@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
drwxrwxr-x 2 sslab sslab 4096 Aug 28 21:19 SP_lecture
sslab@ubuntu:~/work$ chmod u-w,g-w,o-r hello.txt
sslab@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-r--r----- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
drwxrwxr-x 2 sslab sslab 4096 Aug 28 21:19 SP_lecture
sslab@ubuntu:~/work$
```

chmod

- Change file access permissions

- usage 2: chmod [OPTION]... **OCTAL-MODE** FILE...

- OCTAL-MODE**

- 8진수 숫자 세 개로 user(owner), group, other의 권한 표현
- 각 숫자는 다음의 합으로 표현

- 1: execute

- 2: write

- 4: read

- e.g.

- chmod 777 test → test: 모든 대상에게 모든 권한 부여

- chmod 701 test → test: 소유자에게 모든 권한, other는 execute만 가능

```
sslab@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-r--r----- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
drwxrwxr-x 2 sslab sslab 4096 Aug 28 21:19 SP_lecture
sslab@ubuntu:~/work$ chmod 664 hello.txt
sslab@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
drwxrwxr-x 2 sslab sslab 4096 Aug 28 21:19 SP_lecture
sslab@ubuntu:~/work$
```

rmdir

- **Remove empty directories**
 - usage: `rmdir [OPTION]... DIRECTORY...`
 - e.g.

```
sslab@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
drwxrwxr-x 2 sslab sslab 4096 Aug 28 21:19 SP_lecture
sslab@ubuntu:~/work$ rmdir SP_lecture/
sslab@ubuntu:~/work$ ls -l
total 20
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
sslab@ubuntu:~/work$
```


cd

- **Change the current directory**

- usage : `cd [-L|-P] [dir]`
- Special filenames
 - `.` : current directory
 - `..` : parent directory
- e.g.

```
sslab@ubuntu:~$ pwd
/home/sslab
sslab@ubuntu:~$ cd work
sslab@ubuntu:~/work$ pwd
/home/sslab/work
sslab@ubuntu:~/work$ cd .
sslab@ubuntu:~/work$ cd ..
sslab@ubuntu:~$ cd work
sslab@ubuntu:~/work$ cd ~
sslab@ubuntu:~$ cd -
/home/sslab/work
sslab@ubuntu:~/work$
```

touch

- Make an empty file or change modification time of the file to the current time
 - usage : touch [OPTION]... FILE...
 - e.g.

```
sslab@ubuntu:~/work$ ls
file1.txt file2.txt file3.txt hello.txt SP_lab
sslab@ubuntu:~/work$ touch empty.txt
sslab@ubuntu:~/work$ ls
empty.txt file1.txt file2.txt file3.txt hello.txt SP_lab
sslab@ubuntu:~/work$ ls -l
total 20
-rw-rw-r-- 1 sslab sslab  0 Aug 28 21:41 empty.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
sslab@ubuntu:~/work$ touch empty.txt
sslab@ubuntu:~/work$ ls -l
total 20
-rw-rw-r-- 1 sslab sslab  0 Aug 28 21:44 empty.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
sslab@ubuntu:~/work$
```

echo

- **Display a line of text**
 - usage: echo [OPTION]... [STRING]...
 - Display environment variable
 - e.g.

```
sslab@ubuntu:~/work$ echo helloworld
helloworld
sslab@ubuntu:~/work$ echo $HOME
/home/sslab
sslab@ubuntu:~/work$ echo ~
/home/sslab
sslab@ubuntu:~/work$
```

cat

- **Concatenate files and print on the standard output**
 - usage: cat [OPTION] [FILE]...
 - e.g.

```
ssl@ubuntu:~/work$ ls
empty.txt  file1.txt  file2.txt  file3.txt  hello.txt  SP_lab
ssl@ubuntu:~/work$ cat file1.txt
Hello This is file 1
ssl@ubuntu:~/work$ cat file2.txt
Hello This is file 2
ssl@ubuntu:~/work$ cat file1.txt file2.txt
Hello This is file 1
Hello This is file 2
ssl@ubuntu:~/work$
```

more

- **More is a filter for paging through text one screenful at a time**
 - usage: more [-options] [-num] [+ / pattern] [+ linenum] [file ...]
 - e.g.

명령어 : 스페이스 바, z

명령어 : q

[illegible][illegible][illegible]

find

- Search for files in a directory hierarchy
 - usage: find [-H] [-L] [-P] [path...] [expression]
 - e.g.

```
sslab@ubuntu:~/work$ ls
empty.txt  file1.txt  file2.txt  file3.txt  hello.txt  SP_lab
sslab@ubuntu:~/work$ cd SP_lab/
sslab@ubuntu:~/work/SP_lab$ ls
fileA.txt  fileC.txt
sslab@ubuntu:~/work/SP_lab$ cd ..
sslab@ubuntu:~/work$ ls
empty.txt  file1.txt  file2.txt  file3.txt  hello.txt  SP_lab
sslab@ubuntu:~/work$ find -name '*.txt'
./file2.txt
./file1.txt
./empty.txt
./file3.txt
./hello.txt
./SP_lab/fileA.txt
./SP_lab/fileC.txt
sslab@ubuntu:~/work$
```

grep

- Search the named input FILES (or standard input if no files are named, or the file name is given) for lines containing a match to the given PATTERN
 - usage: `grep [options] [PATTEN] [FILE...]`
 - e.g.

```
sslab@ubuntu:~/work$ ls
empty.txt  file1.txt  file2.txt  file3.txt  hello.txt  SP_lab
sslab@ubuntu:~/work$ cat hello.txt
hello world
My Name is N~~~
How are you?
sslab@ubuntu:~/work$ grep hello hello.txt
hello world
sslab@ubuntu:~/work$
```

WC

- **Print newline, word, and byte count for each file**
 - usage: wc [options]... [FILE]...
 - e.g.

```
sslab@ubuntu:~/work$ cat hello.txt
hello world
My Name is N~~~
How are you?
sslab@ubuntu:~/work$ wc hello.txt
 3  9 41 hello.txt
sslab@ubuntu:~/work$ wc -c hello.txt
41 hello.txt
sslab@ubuntu:~/work$ wc -w hello.txt
9 hello.txt
sslab@ubuntu:~/work$ wc -l hello.txt
3 hello.txt
sslab@ubuntu:~/work$
```


cp

- Copy files and directories

- usage: cp [OPTION]... SOURCE DEST
cp [OPTION]... SOURCE... DIRECTORY

- e.g.

```
sslab@ubuntu:~/work$ ls -l
total 20
-rw-rw-r-- 1 sslab sslab  0 Aug 28 21:44 empty.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab 41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
sslab@ubuntu:~/work$ cp hello.txt hello_copy.txt
sslab@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 sslab sslab  0 Aug 28 21:44 empty.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab 41 Aug 28 22:07 hello_copy.txt
-rw-rw-r-- 1 sslab sslab 41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
sslab@ubuntu:~/work$ cp SP_lab/* .
sslab@ubuntu:~/work$ ls -l
total 32
-rw-rw-r-- 1 sslab sslab  0 Aug 28 21:44 empty.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab 15 Aug 28 22:07 fileA.txt
-rw-rw-r-- 1 sslab sslab 15 Aug 28 22:07 fileC.txt
-rw-rw-r-- 1 sslab sslab 41 Aug 28 22:07 hello_copy.txt
-rw-rw-r-- 1 sslab sslab 41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
sslab@ubuntu:~/work$
```

mv

- **Move (rename) files**

- usage: mv [OPTION]... SOURCE DEST
- e.g.

```
sslab@ubuntu:~/work$ mkdir ex
sslab@ubuntu:~/work$ ls
empty.txt  file1.txt  file3.txt  fileC.txt      hello.txt
ex         file2.txt  fileA.txt  hello_copy.txt SP_lab
sslab@ubuntu:~/work$ mv hello_copy.txt /home/sslab/work/ex or mv hello_copy.txt ~/work/ex
sslab@ubuntu:~/work$ ls
empty.txt  ex  file1.txt  file2.txt  file3.txt  fileA.txt  fileC.txt  hello.txt  SP_lab
sslab@ubuntu:~/work$ cd ex/
sslab@ubuntu:~/work/ex$ ls
hello_copy.txt
sslab@ubuntu:~/work/ex$ cd ..
sslab@ubuntu:~/work$ mv ex LINUX
sslab@ubuntu:~/work$ ls
empty.txt  file1.txt  file2.txt  file3.txt  fileA.txt  fileC.txt  hello.txt  LINUX  SP_lab
sslab@ubuntu:~/work$
```

rm(1/3)

- Remove files or directories
 - usage: rm [OPTION]... FILE...
 - e.g.

```
sslab@ubuntu:~/work$ ls -l
total 32
-rw-rw-r-- 1 sslab sslab  0 Aug 28 21:44 empty.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab  15 Aug 28 22:07 fileA.txt
-rw-rw-r-- 1 sslab sslab  15 Aug 28 22:07 fileC.txt
-rw-rw-r-- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 22:16 LINUX
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
sslab@ubuntu:~/work$ rm fileA.txt
sslab@ubuntu:~/work$ ls -l
total 28
-rw-rw-r-- 1 sslab sslab  0 Aug 28 21:44 empty.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab  15 Aug 28 22:07 fileC.txt
-rw-rw-r-- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 22:16 LINUX
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
sslab@ubuntu:~/work$
```

rm(2/3)

- Remove files or directories
 - useful option
 - -r : remove the contents of directory recursively
 - e.g.

```
sslab@ubuntu:~/work$ ls -l
total 28
-rw-rw-r-- 1 sslab sslab  0 Aug 28 21:44 empty.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab 15 Aug 28 22:07 fileC.txt
-rw-rw-r-- 1 sslab sslab 41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 22:16 LINUX
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
sslab@ubuntu:~/work$ rm -r LINUX/
sslab@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 sslab sslab  0 Aug 28 21:44 empty.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab 15 Aug 28 22:07 fileC.txt
-rw-rw-r-- 1 sslab sslab 41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
sslab@ubuntu:~/work$
```

rm(3/3)

- Remove files or directories
 - useful option
 - -i : prompt before every removal
 - e.g.

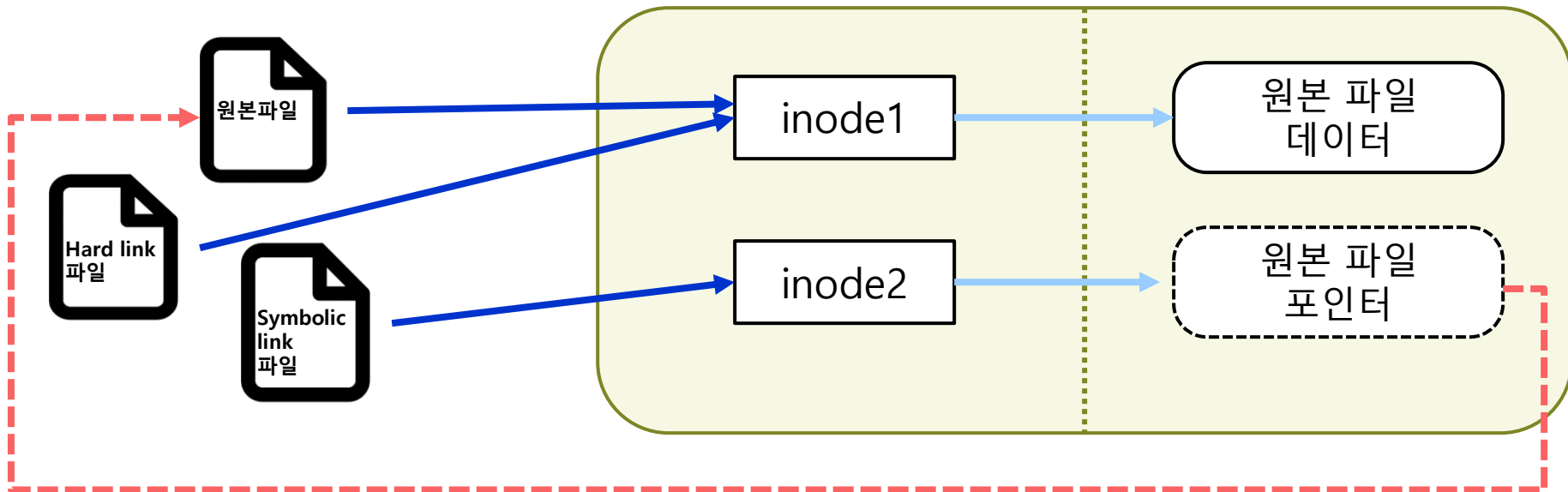
```
sslab@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 sslab sslab  0 Aug 28 21:44 empty.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file1.txt
-rw-rw-r-- 1 sslab sslab 21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab  15 Aug 28 22:07 fileC.txt
-rw-rw-r-- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
sslab@ubuntu:~/work$ rm -i *
rm: remove regular empty file 'empty.txt'? y
rm: remove regular file 'file1.txt'? y
rm: remove regular file 'file2.txt'? n
rm: remove regular file 'file3.txt'? n
rm: remove regular file 'fileC.txt'? n
rm: remove regular file 'hello.txt'? n
rm: cannot remove 'SP_lab': Is a directory
sslab@ubuntu:~/work$ ls -l
total 20
-rw-rw-r-- 1 sslab sslab  21 Aug 28 20:12 file2.txt
-rw-rw-r-- 1 sslab sslab 2001 Aug 28 20:12 file3.txt
-rw-rw-r-- 1 sslab sslab  15 Aug 28 22:07 fileC.txt
-rw-rw-r-- 1 sslab sslab  41 Aug 28 20:12 hello.txt
drwxrwxr-x 2 sslab sslab 4096 Aug 28 20:12 SP_lab
sslab@ubuntu:~/work$
```

In (1/6)

- **Make links between files**
 - Hard link, Symbolic link
 - usage: `ln [OPTION]... TARGET [LINK_NAME]`
 - useful option
 - -s: 심볼릭 링크 생성

사용자에게 보여지는 디렉터리

내부적으로 동작



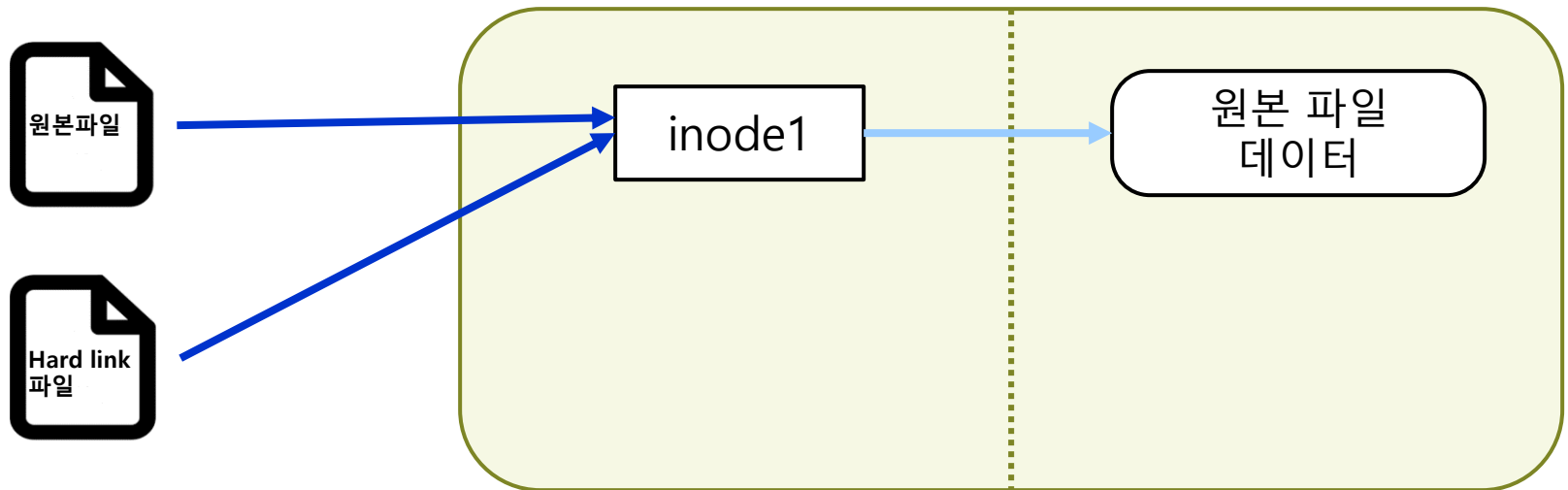
In (2/6)

- **hard link**

- usage: ln [원본파일] [생성할 하드 링크 파일]

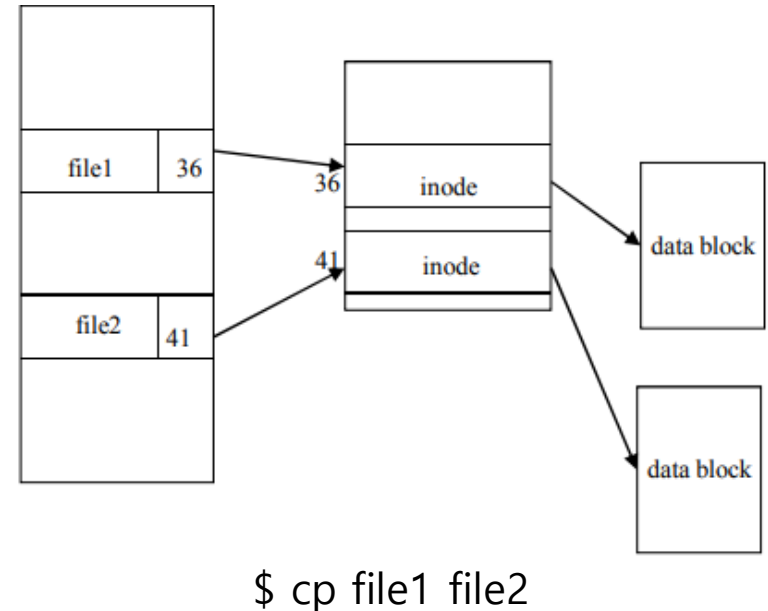
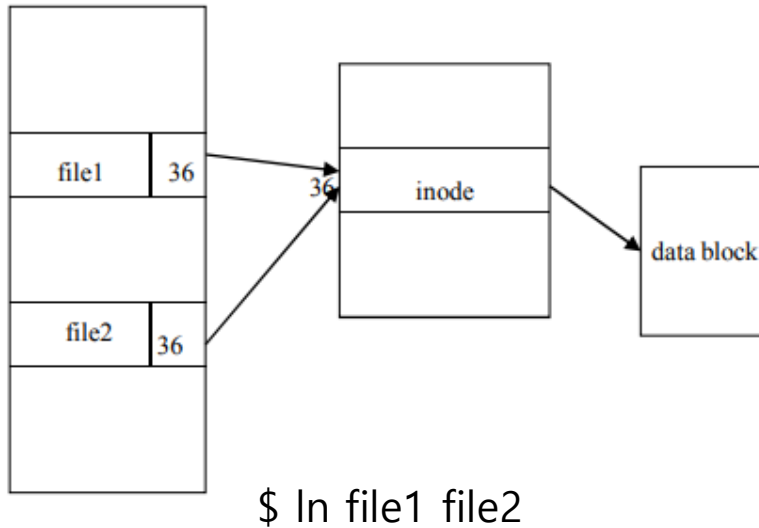
사용자에게 보여지는 디렉터리

내부적으로 동작



In (3/6)

- In(hard link) vs. cp



```
sslslab@ubuntu:~/work$ cd SP_lab/  
sslslab@ubuntu:~/work/SP_lab$ ls  
fileA.txt fileC.txt  
sslslab@ubuntu:~/work/SP_lab$ cat fileA.txt  
This is file A  
sslslab@ubuntu:~/work/SP_lab$ ln fileA.txt fileB.txt  
sslslab@ubuntu:~/work/SP_lab$ cat fileB.txt  
This is file A  
sslslab@ubuntu:~/work/SP_lab$ vi fileB.txt  
sslslab@ubuntu:~/work/SP_lab$ cat fileA.txt  
This is file B after the change.  
sslslab@ubuntu:~/work/SP_lab$ cat fileB.txt  
This is file B after the change.  
sslslab@ubuntu:~/work/SP_lab$
```

```
sslslab@ubuntu:~/work/SP_lab$ cat fileC.txt  
This is file C  
sslslab@ubuntu:~/work/SP_lab$ cp fileC.txt fileD.txt  
sslslab@ubuntu:~/work/SP_lab$ cat fileD.txt  
This is file C  
sslslab@ubuntu:~/work/SP_lab$ vi fileD.txt  
sslslab@ubuntu:~/work/SP_lab$ cat fileC.txt  
This is file C  
sslslab@ubuntu:~/work/SP_lab$ cat fileD.txt  
This is file D after the change.  
sslslab@ubuntu:~/work/SP_lab$
```

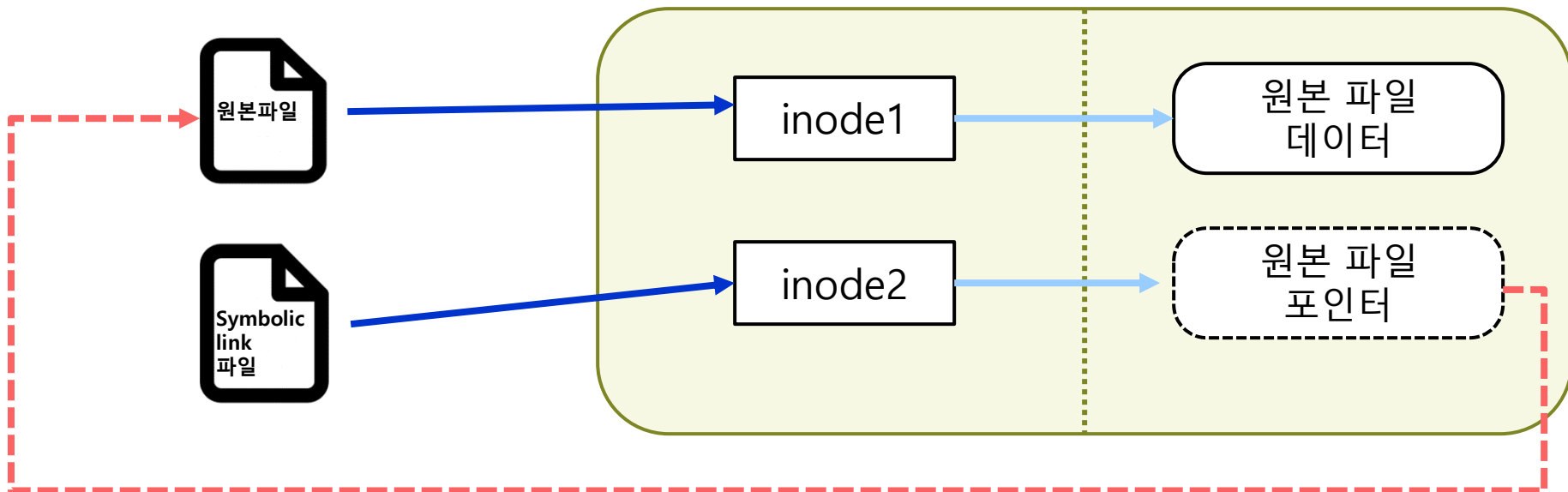

In (4/6)

- Symbolic link

- usage: `ln -s [원본파일] [생성할 심볼릭 링크 파일]`

사용자에게 보여지는 디렉터리

내부적으로 동작



In (5/6)

- **Symbolic link**

- usage: `ln -s [원본파일] [생성할 심볼릭 링크 파일]`
- e.g.

```
sslab@ubuntu:~/work/SP_lab$ cat fileC.txt
This is file C
sslab@ubuntu:~/work/SP_lab$ ln -s fileC.txt fileE.txt
sslab@ubuntu:~/work/SP_lab$ ls -l
total 16
-rw-rw-r-- 2 sslab sslab 32 Mar 23 12:25 fileA.txt
-rw-rw-r-- 2 sslab sslab 32 Mar 23 12:25 fileB.txt
-rw-rw-r-- 1 sslab sslab 15 Mar 23 07:42 fileC.txt
-rw-rw-r-- 1 sslab sslab 33 Mar 23 12:26 fileD.txt
lrwxrwxrwx 1 sslab sslab  9 Mar 23 12:34 fileE.txt -> fileC.txt
sslab@ubuntu:~/work/SP_lab$ cat fileE.txt
This is file C
sslab@ubuntu:~/work/SP_lab$ rm fileC.txt
sslab@ubuntu:~/work/SP_lab$ cat fileE.txt
cat: fileE.txt: No such file or directory
sslab@ubuntu:~/work/SP_lab$
```

In (6/6)

- hard link vs Symbolic link

hard link	symbolic link
파일에만 링크 가능	파일 또는 디렉터리에 링크 할 수 있음
존재하지 않는 파일에 대해 hard link를 작성 할 수 없음	존재하지 않는 파일에 대해 symbolic link을 작성할 수 있음
연결되어 있는 파일인지 알기 어려움	연결되어 있는 파일을 찾기 용이
같은 파일 시스템 간에서만 작성 가능	다른 파일 시스템 간에서도 작성 할 수 있음
원본파일과 i-node 같음	원본파일과 i-node 다름

ps

- **Report process status**

- usage: ps [options]
- e.g.

```
sslab@ubuntu:~/work/SP_lab$ cd ~
sslab@ubuntu:~$ ps
  PID TTY          TIME CMD
 2518 pts/11    00:00:00 bash
 4317 pts/11    00:00:00 ps
sslab@ubuntu:~$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1         0  0  20:10 ?        00:00:01 /sbin/init auto noprompt
root           2         0  0  20:10 ?        00:00:00 [kthreadd]
root           4         2  0  20:10 ?        00:00:00 [kworker/0:0H]
root           6         2  0  20:10 ?        00:00:00 [mm_percpu_wq]
root           7         2  0  20:10 ?        00:00:00 [ksoftirqd/0]
root           8         2  0  20:10 ?        00:00:00 [rcu_sched]
root           9         2  0  20:10 ?        00:00:00 [rcu_bh]
root          10         2  0  20:10 ?        00:00:00 [migration/0]
root          11         2  0  20:10 ?        00:00:00 [watchdog/0]
root          12         2  0  20:10 ?        00:00:00 [cpuhp/0]
root          13         2  0  20:10 ?        00:00:00 [cpuhp/1]
root          14         2  0  20:10 ?        00:00:00 [watchdog/1]
```

- useful options
 - -e : select all processes
 - -f : full format listing

pstree

- Display a tree of processes
 - usage: pstree [options]
 - e.g.

```
sslab@ubuntu:~$ pstree
systemd--NetworkManager--dhclient
                        --dnsmasq
                        --{gdbus}
                        --{gmain}
--VGAAuthService
--accounts-daemon--{gdbus}
                  --{gmain}
--acpid
--agetty
--avahi-daemon--avahi-daemon
--bluetoothd
--colord--{gdbus}
        --{gmain}
--cron
--cups-browsed--{gdbus}
               --{gmain}
--cupsd--dbus
--dbus-daemon
--fwupd--{GUsbEventThread}
        --{fwupd}
        --{gdbus}
        --{gmain}
--gnome-keyring-d--{gdbus}
                  --{gmain}
                  --{timer}
--irqbalance
```

exit

- Cause the shell to exit
 - usage: exit
 - e.g.

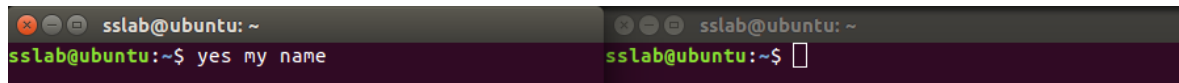
```
sslab@ubuntu:~$ sudo apt-get install csh
[sudo] password for sslab:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically in-
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
The following NEW packages will be install
  csh
```

```
sslab@ubuntu:~$ ps
  PID TTY          TIME CMD
 2518 pts/11        00:00:00 bash
 4811 pts/11        00:00:00 ps
sslab@ubuntu:~$ csh
% ps
  PID TTY          TIME CMD
 2518 pts/11        00:00:00 bash
 4812 pts/11        00:00:00 csh
 4813 pts/11        00:00:00 ps
% exit
% exit
sslab@ubuntu:~$ ps
  PID TTY          TIME CMD
 2518 pts/11        00:00:00 bash
 4814 pts/11        00:00:00 ps
sslab@ubuntu:~$
```

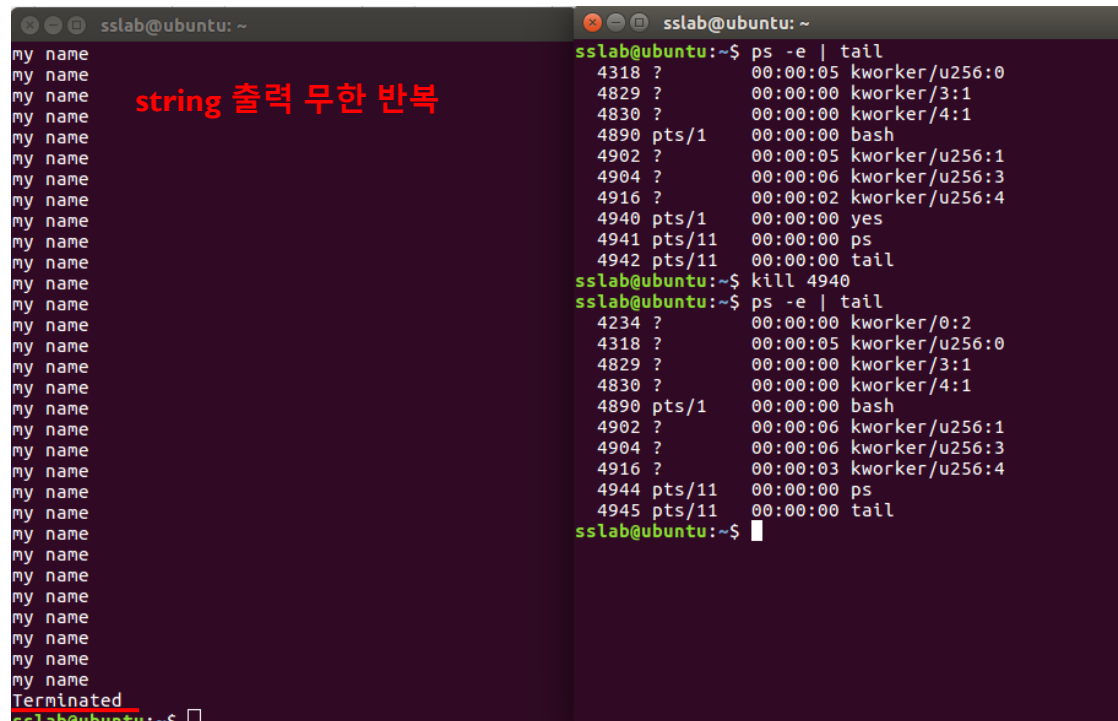
- **Send a signal to a process**

- usage: kill [-s signal | -p] [-a] [--] pid ...
 - The default signal for kill is TERM. (i.e. Terminate process)
- e.g.

Terminal 2개 출력



- | : 이전 명령어의 output을 다음 명령어의 Input으로 연결
- tail : 파일의 끝 부분부터 10개의 행 출력



kill(2/2)

- Send a signal to a process
 - KILL signal (-9)
 - -9 : SIGKILL (process 강제 종료)
 - e.g.

Ctrl + Z

```
sslab@ubuntu:~$ ps
  PID TTY          TIME CMD
 2518 pts/11        00:00:00 bash
 4869 pts/11        00:00:00 ps
sslab@ubuntu:~$ vi hello
[1]+  Stopped                  vi hello
sslab@ubuntu:~$ ps
  PID TTY          TIME CMD
 2518 pts/11        00:00:00 bash
 4870 pts/11        00:00:00 vi
 4874 pts/11        00:00:00 ps
sslab@ubuntu:~$ kill -9 4870
sslab@ubuntu:~$ ps
  PID TTY          TIME CMD
 2518 pts/11        00:00:00 bash
 4875 pts/11        00:00:00 ps
[1]+  Killed                  vi hello
sslab@ubuntu:~$
```


time

- **Time a simple command or give resource usage**

- usage: time [options] command [arguments...]
- e.g.

```
sslab@ubuntu:~$ time ps
  PID TTY          TIME CMD
  4890 pts/1        00:00:00 bash
  4959 pts/1        00:00:00 ps

real    0m0.006s
user    0m0.001s
sys     0m0.005s
sslab@ubuntu:~$
```

- Output data
 - real : 실제 소요 시간
 - user : user 영역에서 소비된 CPU 시간
 - sys : 커널에서 소비된 CPU 시간

alias

- Enable a replacement of a word by another string
 - e.g. `$ alias myls='ls -al'`

```
sslab@ubuntu:~$ myls
No command 'mysls' found, did you mean:
  Command 'mmls' from package 'sleuthkit' (universe)
  Command 'tyls' from package 'terminology' (universe)
mysls: command not found
sslab@ubuntu:~$ alias myls='ls -al'
sslab@ubuntu:~$ myls
total 128
drwxr-xr-x 16 sslab sslab 4096 Aug 28 22:52 .
drwxr-xr-x  3 root  root  4096 Aug 24 02:06 ..
-rw-r--r--  1 sslab sslab 3466 Aug 28 23:02 .bash_history
-rw-r--r--  1 sslab sslab  220 Aug 24 02:06 .bash_logout
-rw-r--r--  1 sslab sslab 3771 Aug 24 02:06 .bashrc
```

```
sslab@ubuntu:~$ ls -al
total 128
drwxr-xr-x 16 sslab sslab 4096 Aug 28 22:52 .
drwxr-xr-x  3 root  root  4096 Aug 24 02:06 ..
-rw-r--r--  1 sslab sslab 3466 Aug 28 23:02 .bash_history
-rw-r--r--  1 sslab sslab  220 Aug 24 02:06 .bash_logout
-rw-r--r--  1 sslab sslab 3771 Aug 24 02:06 .bashrc
```

```
sslab@ubuntu:~$ alias
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\''")'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
alias myls='ls -al'
sslab@ubuntu:~$
```

Unix commands

File and file system management	<code>cat</code> · <code>cd</code> · <code>chmod</code> · <code>chown</code> · <code>chgrp</code> · <code>cmp</code> · <code>cp</code> · <code>du</code> · <code>df</code> · <code>file</code> · <code>fsck</code> · <code>ln</code> · <code>ls</code> · <code>mkdir</code> · <code>mount</code> · <code>mv</code> · <code>pwd</code> · <code>rm</code> · <code>rmdir</code> · <code>touch</code>
Process management	<code>exit</code> · <code>kill</code> · <code>killall</code> · <code>nice</code> · <code>ps</code> · <code>pstree</code> · <code>sleep</code> · <code>time</code> · <code>top</code> · <code>wait</code>
User management /environment	<code>finger</code> · <code>mesg</code> · <code>passwd</code> · <code>su</code> · <code>sudo</code> · <code>uname</code> · <code>w</code> · <code>wall</code> · <code>who</code> · <code>whoami</code> · <code>write</code>
Text processing	<code>awk</code> · <code>comm</code> · <code>ed</code> · <code>ex</code> · <code>head</code> · <code>less</code> · <code>more</code> · <code>sed</code> · <code>sort</code> · <code>tail</code> · <code>uniq</code> · <code>wc</code> · <code>xargs</code>
Shell programming	<code>alias</code> · <code>echo</code> · <code>expr</code> · <code>false</code> · <code>printf</code> · <code>test</code> · <code>true</code> · <code>unset</code>
Communications	<code>inetd</code> · <code>netstat</code> · <code>ping</code> · <code>rlogin</code> · <code>traceroute</code>
Searching	<code>find</code> · <code>grep</code> · <code>strings</code>
Miscellaneous	<code>dd</code> · <code>lp</code> · <code>man</code> · <code>size</code> · <code>yes</code>

2021년 2학기 운영체제실습 2주차

Programming Tools part 1

System Software Laboratory

School of Computer and Information Engineering

Kwangwoon Univ.

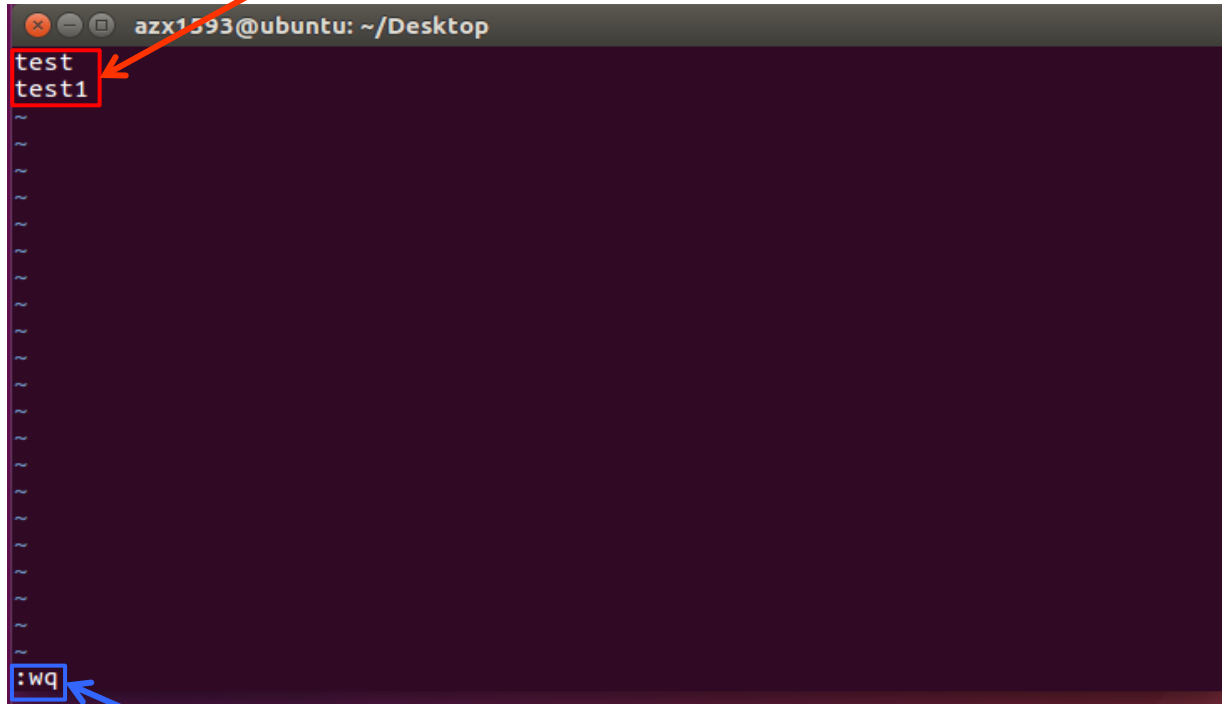
Contents

- **vi Editor**
- **make**
 - 실습: make 사용

vi Editor (1/2)

- **vi 에디터**
 - Linux 기본 편집기
 - 실행 화면

입력 모드 시, 소스 코드 등의 내용을 입력



- vim 패키지 설치 명령 모드 시, 콜론(:) or 슬래시(/) 등으로 시작하는 vi 명령어 입력
 - 기본 설치된 vi는 기능이 제한적이므로 vim 패키지 설치 필요
 - \$ **sudo apt-get install vim**

vi Editor (2/2)

- **vi 모드**

- 명령모드
 - 한/두 문자로 구성된 vi 전용 명령어를 사용하는 모드
 - vi 진입 시 기본 모드
- 입력모드에서 [esc] 키로 진입 가능
- 입력모드
 - vi 편집 화면에서 문자를 입력 할 수 있는 상태를 의미
 - 명령 모드에서 [esc] 후 [i] 등의 키로 진입 가능

- **vi 시작과 종료**

- 시작
- vi file_name → vi를 시작하여 지정한 파일 편집
- 종료
 - :wq → 데이터를 저장하고 종료
 - :q → 데이터를 저장하지 않고 종료
 - :q! → 데이터를 저장하지 않고 강제 종료

vi Editor Options (1/2)

■ 커서 이동

h	← 이동
j	↓ 이동
k	↑ 이동
l	→ 이동
Backspace	커서가 있는 행에서 커서를 왼쪽으로 옮김
Space	커서가 있는 행에서 커서를 오른쪽으로 옮김

■ 텍스트 변경

r	커서가 있는 문자를 변경
R	커서가 있는 부분부터 글자 덮어서 씀
s	한 글자를 삭제 한 후 문장 삽입
S	커서가 있는 문장을 삭제하고 문장 삽입
C	커서가 있는 행에서 커서를 왼쪽으로 옮김

■ 문자, 행, 삽입

i	커서 앞으로 문장 삽입
I	행의 시작 부분에서 문장 삽입
a	커서 뒤로 문장 삽입
A	행의 끝 부분에서 문장 삽입
o	커서가 위치한 행의 아래에 문장 삽입
O	커서가 위치한 행의 위에 문장 삽입

■ 텍스트 삭제

cc	현재 행 삭제 후 문장 입력
cw	커서가 있는 문자 삭제 후 문장 입력
x	커서가 있는 한 글자 삭제
X	커서 앞 한 글자 삭제
D	커서가 있는 부분의 뒷 부분의 행을 삭제
dd	커서가 있는 한 행 삭제

vi Editor Options (2/2)

■ 복사 및 붙여 놓기

nY	커서가 있는 행 부터 n행 만큼 복사
yy	커서가 있는 행 복사
p	커서 아래 복사된 문자열 붙임
P	커서 위에 복사된 문자열 붙임

■ 파일 불러오기

:e 파일 이름	vi 를 종료하지 않고 해당 파일 편집
:e!	현재 편집하고 있는 파일 다시 부르기
:e#	한 글자를 삭제 한 후 문장 삽입

■ 파일 저장

:w	vi 파일을 저장
:w 파일 이름	파일이름으로 저장
:wq	저장 후 종료
:wq!	저장 후 강제 종료

■ 행 번호 설정

:set number	편집기의 라인 표시
:set nu	편집기의 라인 표시
:set nonu	편집기의 라인 표시 없애기

vi Editor 단축키

version 1.1
April 1st, 06

vi / vim 단축키 모음

Esc
명령 모드

~ 대소문자 전환	! 외부 명령	@ 매크로 실행	# 이전 검색	\$ 줄끝으로 이동	% 일치하는 괄호 찾기	^ 줄의 첫 글자	& :s 반복	* 다음 검색	(문장 시작) 문장 끝	아래줄로 이동	+ 다음 줄
\ 매크로 이동	1	2	3	4	5	6	7	8	9	0 줄의 처음	- 이전 줄	= 자동 들여쓰기

Q 실행 모드	W WORD	E 끝 검색	R 수정 모드	T 뒤로 검색	Y 줄단위 복사	U 줄 단위 실행 취소	I 줄 시작에서 삽입	O 행 위에 삽입	P 커서 이전에 붙여넣기	{ 문단 시작	} 문단 끝
q 매크로 기록	w 다음 단어	e 단어 끝	r 한 문자 교체	t 한 문자 검색	y 복사	u 실행 취소	i 편집 모드	o 행 아래에 삽입	p 커서 이후에 붙여넣기	[기타] 기타

A 줄 끝에 덧붙이기	S 줄 삭제후 편집모드	D 줄 끝까지 삭제	F 뒤로 검색	G 파일끝/줄로 이동	H 화면 상단	J 줄 합치기	K 다음행	L 화면 하단	: 명령줄	ex 레지스터 지정	열 이동
a 덧붙이기	s 단어 삭제후 편집모드	d 1,3 삭제	f 한 문자 찾기	g 확장 명령	h ←	j ↓	k ↑	l →	t/T/t/F 명령 반복	' 매크로 이동	\ 사용 안함

Z 종료	X 백스페이스	C 줄 끝까지 바꾸기	V 줄단위 비수정 모드	B 이전 WORD	N 이전 (찾기)	M 화면 가운데	< 3 내어쓰기	> 3 들여쓰기	? 찾기 (뒤로)
z 확장 명령	x 글자 삭제	c 1,3 바꾸기	v 비수정 모드	b 이전 단어	n (찾기)	m 마크 설정	t/T/t/F 명령 반복	/ 찾기	

동작 커서를 이동하거나, 연산자가 동작할 범위를 지정합니다.

명령 바로 동작하는 명령, 빨간색은 편집 모드로 변경됩니다.

연산자 이동 관련 문자(숫자나 커서 이동)와 함께 사용되어야 하며, 커서의 위치부터 목적지까지 연산합니다.

확장 특별한 키 합수로, 추가적인 키 입력이 필요합니다.

q 입력후 (숫자를 제외한)으로 끝낼수 있는 글자를 입력하여야 합니다.

words: 구분자로 공백, 특수기호 모두 사용
WORDS: 구분자로 공백 문자만 사용

words: quux(foo, bar, baz)
WORDS: quux(foo, bar, baz)

주요 명령행 명령 ('ex'):
:w (저장), :q (종료), :q! (저장하지 않고 종료)
:e f (파일 f 열기), :%s/x/y/g (파일 전체에서 'x' 를 'y' 로 교체), :h (vim 도움말), :new (새 파일)

그외 중요한 명령들:
CTRL-R: 재실행 (vim), CTRL-F/B: 페이지 위로/아래로, CTRL-E/Y: 줄 스크롤 위로/아래로, CTRL-V: 블록-비수정 모드 (vim 전용)

비수정 모드: 커서를 움직여 지정한 범위에 연산자를 적용합니다. (vim 전용)

참고:
(1) 복사/붙여넣기/자르기 명령어를 사용하기 전에 "x"를 입력하여 레지스터(클립보드)를 지정하세요. (x는 a에서 z 또는 * 을 사용할 수 있음) (예: "ay\$를 입력하면 현재 커서에서 라인 끝까지의 내용을 레지스터 'a'에 저장합니다.)
(2) 어떤 명령을 입력하기 전에 횟수를 지정하면, 횟수만큼 반복하게 됩니다.(예: 2p, d2w, 5l, d4j)
(3) 연속으로 입력하는 명령은 현재의 라인에 반영됩니다. 예시: dd(현재 라인 지우기), >>(들여쓰기)
(4) ZZ는 저장후 종료, ZQ는 저장하지 않고 종료.
(5) zt: 커서가 위치한 곳을 제일위로 올리기, zb: 바닥으로, zz: 가운데로
(6) gg: 파일의 처음으로(Vim 전용), gf: 커서가 위치한 곳의 파일 열기(Vim 전용)

vi/vim 에 대한 더 많은 강좌나 팁을 얻으려면 www.viemu.com (ViEmu, MS 비주얼 스튜디오를 위한 vi/vim 에뮬레이션)을 방문하십시오.

Reference

- 영문 버전: http://www.viemu.com/a_vi_vim_graphical_cheat_sheet_tutorial.html
- 한글 버전: <https://kldp.org/node/102947>

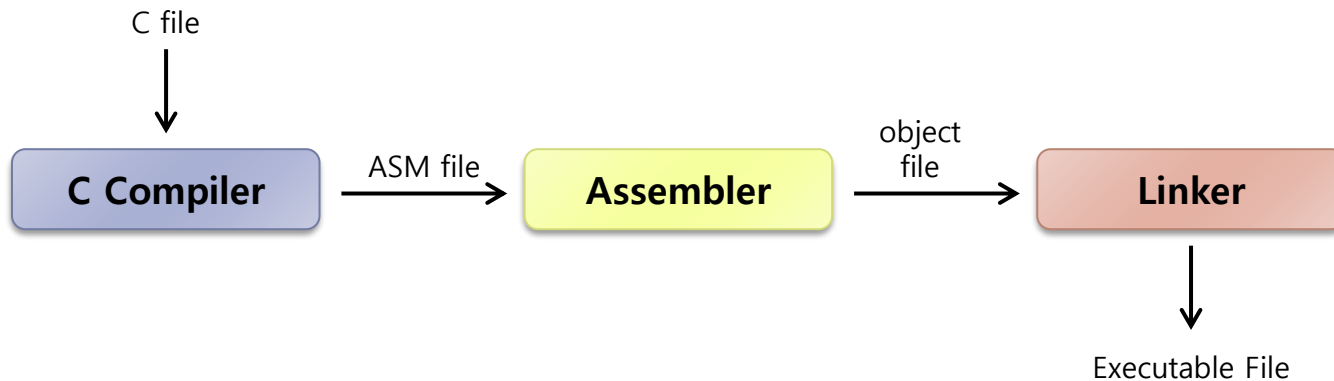
make (1/5)

- **필요성**

- 컴파일 과정을 자동화하기 위해 사용
- gcc 컴파일러의 다양한 옵션들을 컴파일 할 때마다 입력한다면 vi 실행/편집/종료, 컴파일을 반복하는데 많은 시간이 소요될 것임.

- **Makefile**

- 컴파일 할 소스 파일과 컴파일 옵션에 관해 정의해 놓은 스크립트 파일.
- 아래 그림과 같이 빌드는 여러 과정으로 구성되어 있는데, 이를 통해 한 번 만에 빌드를 수행할 수 있도록 함.



make (2/5)

Makefile (cont'd)

- 작업할 대상 → 타겟
 - 타겟 명을 의미하기 위해 이름 뒤에 ':'
 - 타겟 명 다음 줄에는 실행할 명령을 명시

타겟 → `hello:` `hello.c` ← 타겟 명
`gcc hello.c -o hello` ← 명령어

- 여러 개 빌드
 - "all:" 이라는 타겟 뒤에 새로운 하위 타겟("hello1 hello2") 추가
 - \$ make는 "all" 타겟에서 새로운 타겟 "hello1 hello2" 를 확인하고 실행
 - hello2만 빌드하고 싶다면, "\$make hello2" 실행

```
all: hello1 hello2

hello1: hello1.c
    gcc hello1.c -o hello1

hello2: hello2.c
    gcc hello2.c -o hello2
```

타겟(Target)	명령어가 수행되어 나온 결과를 저장한 파일
타겟 명(Target Name)	소스 파일 명
명령어(Command)	실행 명령어

make (3/5)

- **Makefile (cont'd)**

- make 수행 시 타겟을 명시하지 않는 경우
 - i.e. \$ make
 - Makefile의 제일 첫 번째 타겟에 해당하는 명령 수행
- make 수행 시 타겟을 명시하는 경우
 - i.e. \$ make hello1
 - 해당 타겟의 명령 수행

make (4/5)

■ 변수사용과 대체

- 변수는 '이름 = 값'의 형태로 지정
- 변수는 '\$(이름)' 형태로 사용
- 타겟은 '\$@'로 대체, 타겟 명은 '\$^'로 대체
- Example

```
OBJS = hello1.c hello2.c
CC = gcc
EXEC = test

all: $(OBJS)
    $(CC) -o $(EXEC) $^

clean:
    rm -rf $(EXEC)
```

변수 명 ←

→ 값

make (5/5)

- 프로그램 실행 하기
 - 소스 코드

```
#include <stdio.h>
#include "hello2.h"

int main(int argc, char **argv)
{
    printf("Hello World! 1 \n");
    hello2_func();
    return 0;
}
```

hello1.c

```
#include <stdio.h>

void hello2_func()
{
    printf("Hello World! 2 \n");
}
```

hello2.c

```
#ifndef __HELLO2_H__
#define __HELLO2_H__
int hello2_func(void);
#endif
```

hello2.h

- Makefile

```
OBJS = hello1.o hello2.o
CC = gcc
EXEC = test

all: $(OBJS)
    $(CC) $^ -o $(EXEC)

clean:
    rm -rf $(EXEC)
```

- 실행 결과

```
sslab@ubuntu:~/Desktop$ ls
hello1.c hello2.c hello2.h Makefile
sslab@ubuntu:~/Desktop$ make
gcc -o test hello1.c hello2.c
sslab@ubuntu:~/Desktop$ ls
hello1.c hello2.c hello2.h Makefile test
sslab@ubuntu:~/Desktop$ ./test
Hello world! 1
Hello world! 2
sslab@ubuntu:~/Desktop$
```

Assignment 1

- 제출 기한: 2022. 09.01(목) ~ 2022.09.22(목) 23:59:59
- Delay 없음
- 업로드 양식에 어긋날 경우 감점 처리
- Hardcopy 제출하지 않음