

수업 명 : 시스템프로그래밍

과제 이름 : Basic

학 과: 컴퓨터정보공학부

담당교수: 김태석 교수님

분 반: 월5, 수6

학 번: 2023202043

성 명: 최은준

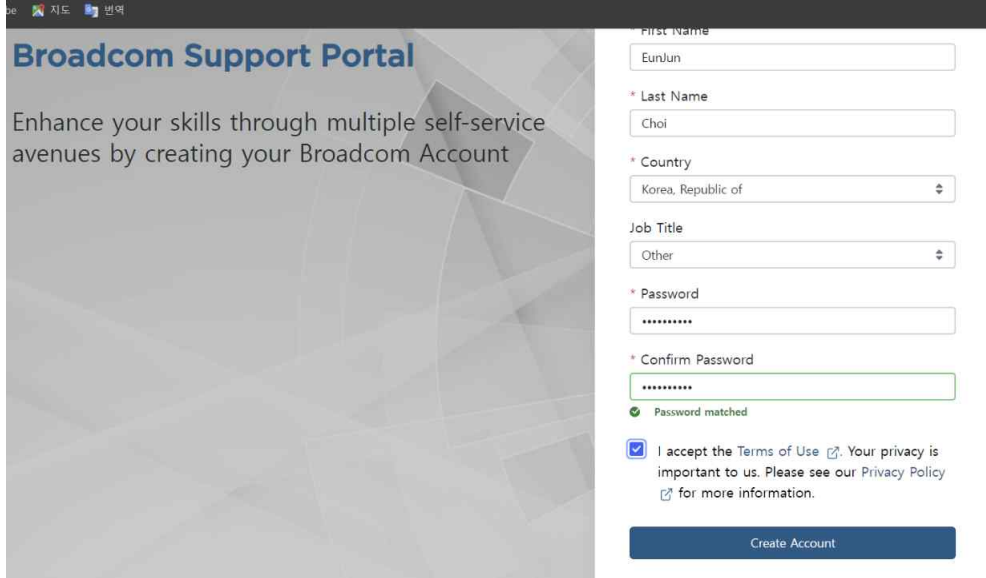
0. Introduction

이 과제에서는 Vmware와 Ubuntu를 설치하여 명령어를 실행 후 설명한다. 또한, vi 편집기와 Makefile을 사용하여 C program을 작성하고 실행하는 것이 목표이다. vi를 사용하여 다음의 내용을 입력하고 저장한다. 첫 번째 라인에는 학번, 두 번째 라인에는 이름, 세 번째 라인에는 Kwangwoon University를 입력한 뒤, 이를 복사하여 학번 다음 라인에 붙여넣는다. 이 때 라인 번호(:set number)를 표시하여야 하고, 저장명은 학번이다.

Makefile을 작성하여 kw_hello.c를 자동으로 컴파일하도록 한다. make 명령어 실행시 hello 실행 파일이 생성되도록 한다. ./hello를 실행하여 학번과 이름을 출력하는 kw_hello.c를 확인한다.

Basic-1. Ubuntu Installation

1. Vmware 설치 과정



Broadcom Support Portal

Enhance your skills through multiple self-service avenues by creating your Broadcom Account

First Name
EunJun

* Last Name
Choi

* Country
Korea, Republic of

Job Title
Other

* Password

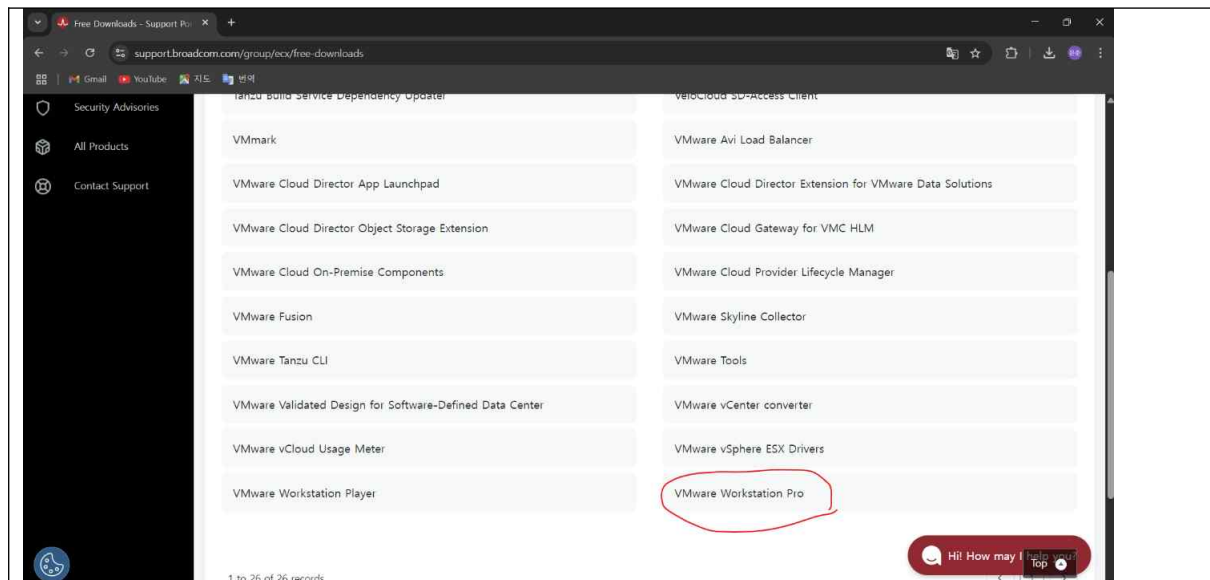
* Confirm Password

✔ Password matched

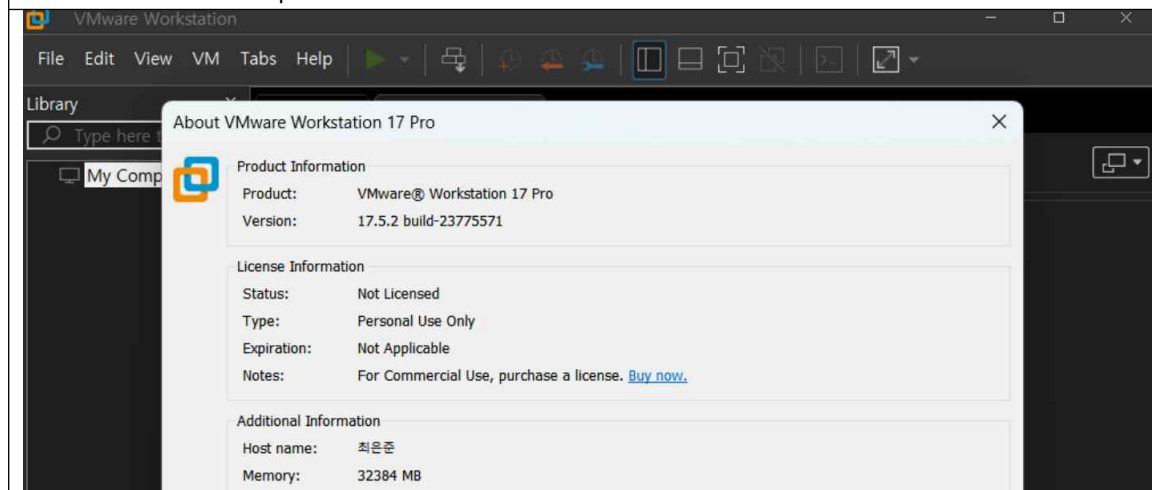
☒ I accept the Terms of Use [↗](#). Your privacy is important to us. Please see our Privacy Policy [↗](#) for more information.

Create Account

Broadcom Support Portal에서 회원가입을 완료한다.

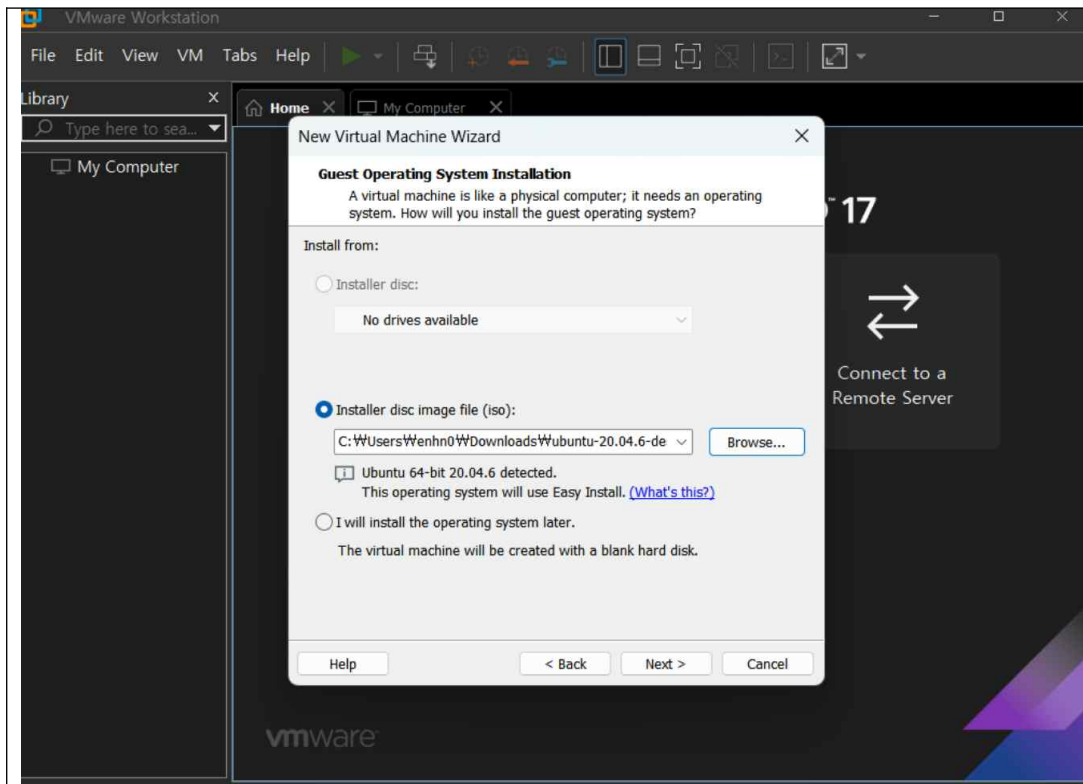


Vmware workstation pro 17.5.2 버전을 설치한다.

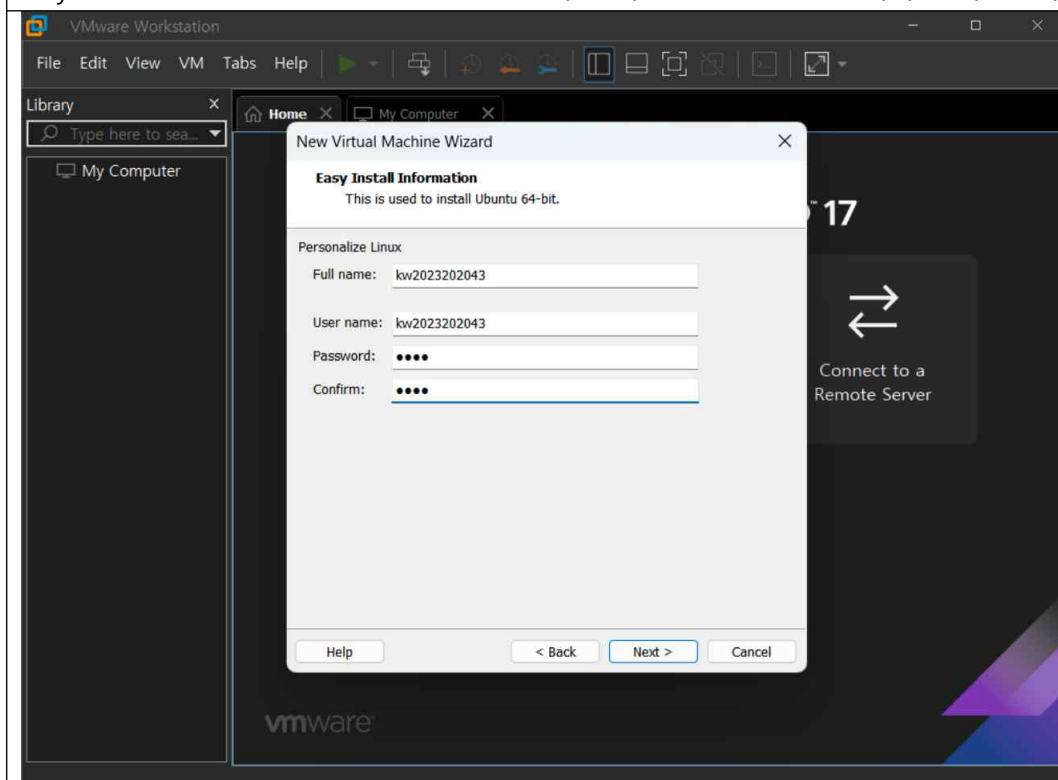


17.5.2 version으로 잘 설치된 모습이다.

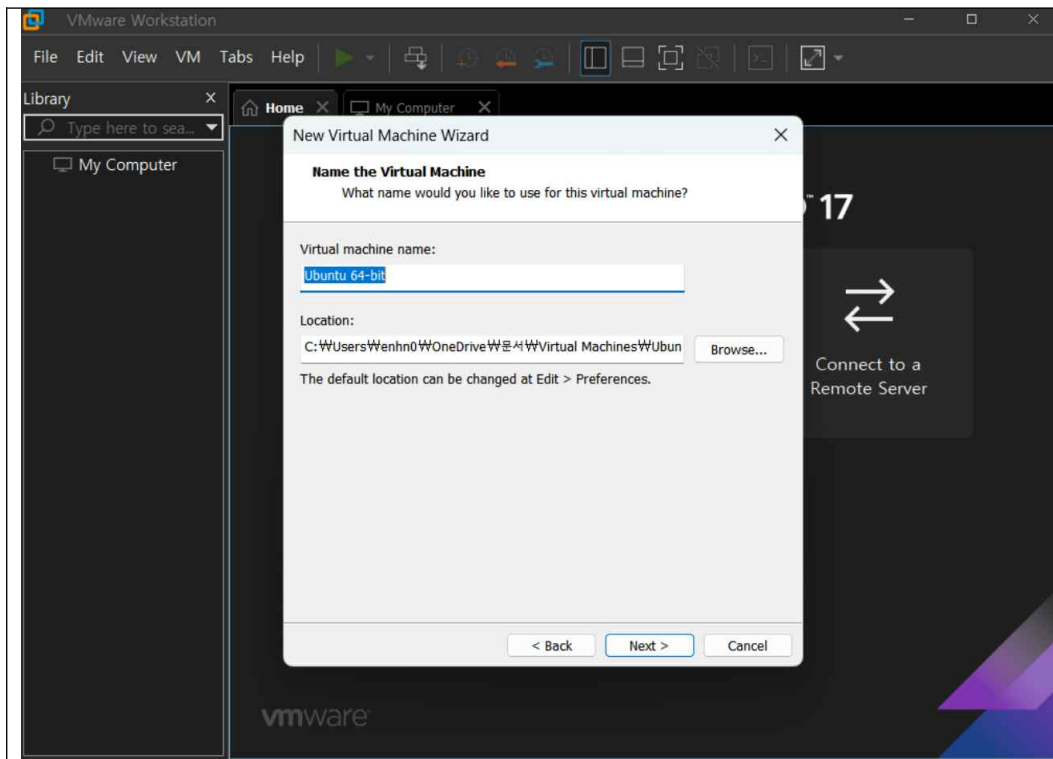
2. Ubuntu 설치 과정



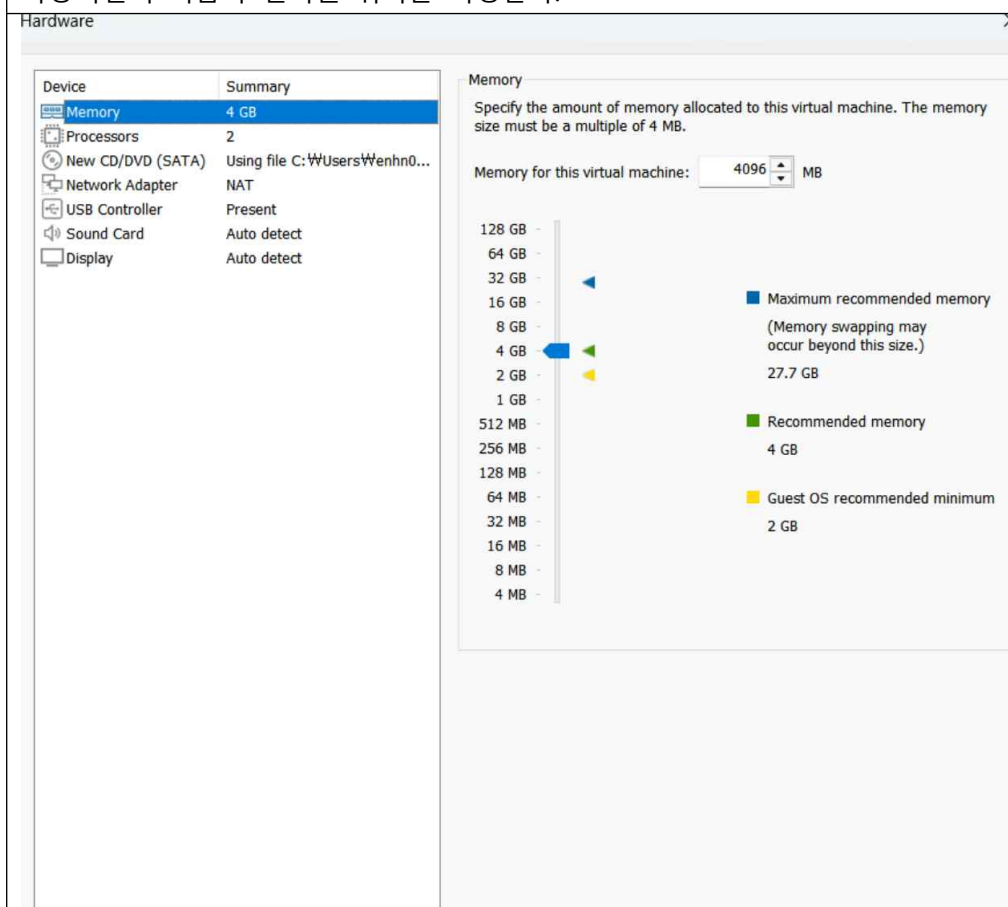
Player->File->New Virtual Machine 순으로 누른 후 Ubuntu ISO file 위치를 지정한다.



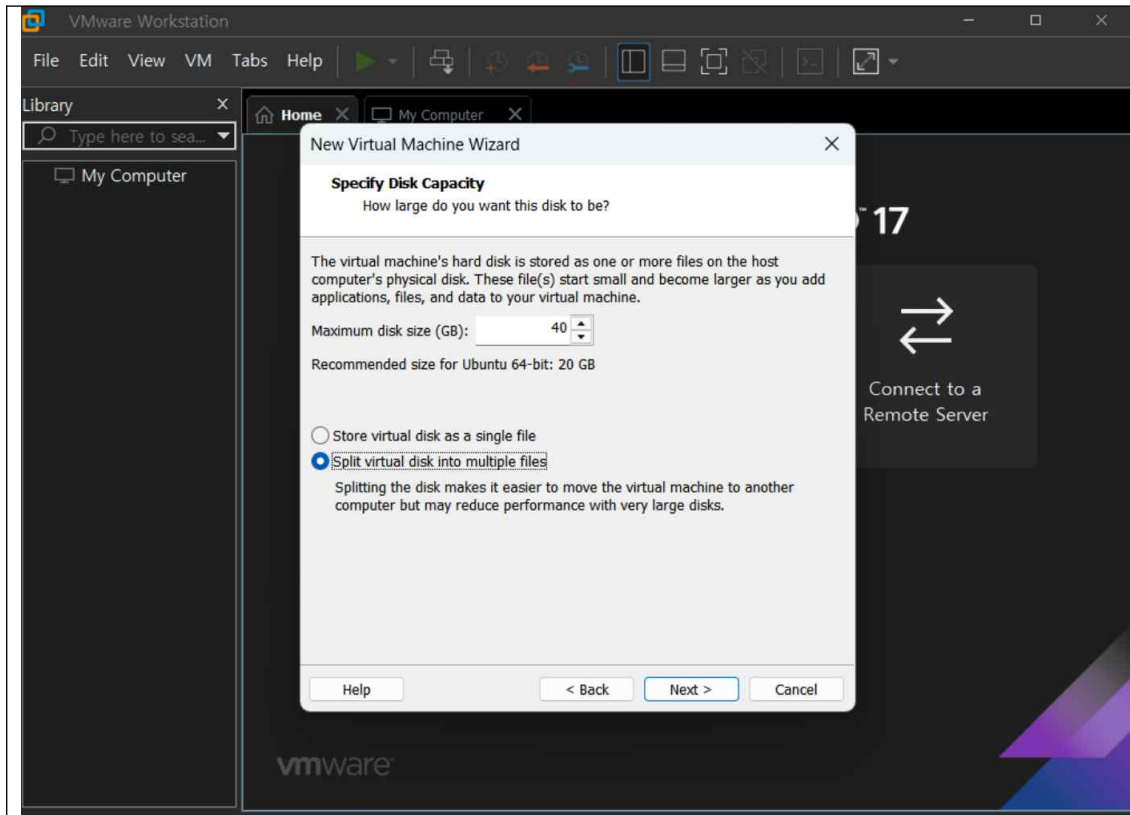
계정 ID를 kw+학번으로 설정하여 로그인한다.



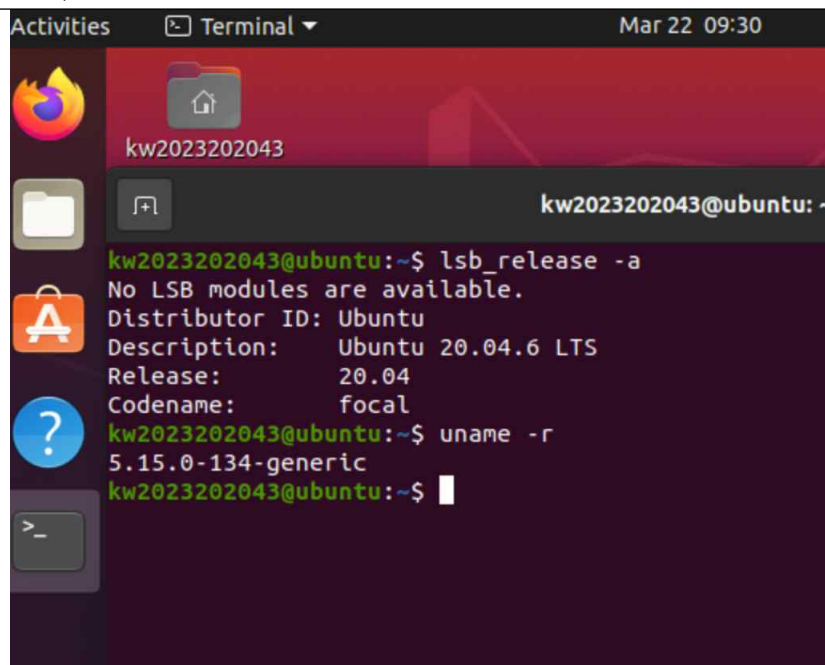
가상머신의 이름과 설치될 위치를 지정한다.



가상머신에 할당할 저장장치 크기와 메인 메모리 크기 등을 설정한다.



이때, 모든 요소는 시스템 자체에서 지원하는 크기보다 작아야 한다.



ubuntu의 version과 user name이 잘 설정된 모습이다.

Basic-2. Linux Commands

1. man

특정 명령어의 사용법과 옵션을 확인하는 명령어이다. 리눅스의 기본적인 도움말 시스템이며 사용 예시는 다음과 같다.

1) man ls -> ls 명령어의 설명서를 출력한다.

```
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Downloads
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Music
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Pictures
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Public
-rw-r--r-- 1 kw2023202043 kw2023202043 2690 Mar 22 06:44 splab_commands
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Templates
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Videos
-rwxrwxr-x 1 kw2023202043 kw2023202043 0 Mar 22 08:40 vmware.bundle
kw2023202043@ubuntu:~$ chmod +x splab_commands
kw2023202043@ubuntu:~$ ./splab_commands
kw2023202043@ubuntu:~$ ls -l
total 40
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Desktop
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Documents
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Downloads
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Music
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Pictures
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Public
-rwxr-xr-x 1 kw2023202043 kw2023202043 2690 Mar 22 06:44 splab_commands
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Templates
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Videos
-rwxrwxr-x 1 kw2023202043 kw2023202043 0 Mar 22 08:40 vmware.bundle
drwxrwxr-x 3 kw2023202043 kw2023202043 4096 Mar 22 09:37 work
kw2023202043@ubuntu:~$
```

2) man -k copy -> keyword를 search 한다.

```
kw2023202043@ubuntu:~$ man -k copy
bcopy (3) - copy byte sequence
c_files_range (2) - Copy a range of data from one file to another
c_files_sign (3) - copy sign of a number
copysignf (3) - copy sign of a number
copysignl (3) - copy sign of a number
cp (1) - copy files and directories
cpgr (8) - copy with locking the given file to the password or g...
cpio (1) - copy files to and from archives
cpw (8) - copy with locking the given file to the password or g...
dd (1) - convert and copy a file
debconf-copydb (1) - copy a debconf database
getunwind (2) - copy the unwind data to caller's buffer
getutmp (3) - copy utmp structure to utmpx, and vice versa
getutmpx (3) - copy utmp structure to utmpx, and vice versa
gvfs-copy (1) - (unknown subject)
install (1) - copy files and set attributes
ncopy (1) - copy MSDOS files to/from Unix
memccpy (3) - copy memory area
memcpy (3) - copy memory area
memmove (3) - copy memory area
mempcpy (3) - copy memory area
ntfscpy (8) - copy file to an NTFS volume.
objcopy (1) - copy and translate object files
rcp (1) - OpenSSH secure file copy
rsync (1) - a fast, versatile, remote (and local) file-copying tool
scp (1) - OpenSSH secure file copy
ssh-copy-id (1) - use locally available keys to authorise logins on a r...
stpcpy (3) - copy a string returning a pointer to its end
stpncpy (3) - copy a fixed-size string, returning a pointer to its end
strcpy (3) - copy a string
strncpy (3) - copy a string
va_copy (3) - variable argument lists
wcpcpy (3) - copy a wide-character string, returning a pointer to ...
wcpncpy (3) - copy a fixed-size string of wide characters, returnin...
wcscpy (3) - copy a wide-character string
wcsncpy (3) - copy a fixed-size string of wide characters
wmemcpy (3) - copy an array of wide-characters
wmemmove (3) - copy an array of wide-characters
wmemcpy (3) - copy memory area
x86_64-linux-gnu-objcopy (1) - copy and translate object files
```

3) man -a write -> all manual을 출력한다.


```
WRITE(1)          BSD General Commands Manual          WRITE(1)

NAME
  write - send a message to another user

SYNOPSIS
  write user [tty]

DESCRIPTION
  The write utility allows you to communicate with other users, by copying
  lines from your terminal to theirs.

  When you run the write command, the user you are writing to gets a mes-
  sage of the form:

      Message from yourname@yourhost on yourtty at hh:mm ...

  Any further lines you enter will be copied to the specified user's ter-
  minal. If the other user wants to reply, they must run write as well.

  When you are done, type an end-of-file or interrupt character. The
  other user will see the message 'EOF' indicating that the conversation
  is over.

  You can prevent people (other than the super-user) from writing to you
  with the mesg(1) command.

  If the user you want to write to is logged in on more than one terminal,
  you can specify which terminal to write to by specifying the terminal
  name as the second operand to the write command. Alternatively, you can
  let write select one of the terminals - it will pick the one with the
  shortest idle time. This is so that if the user is logged in at work
  and also dialed up from home, the message will go to the right place.

  The traditional protocol for writing to someone is that the string '-o',
  either at the end of a line or on a line by itself, means that it is the
  other person's turn to talk. The string 'oo' means that the person be-
  lieves the conversation to be over.

SEE ALSO
  mesg(1), talk(1), wall(1), who(1)

HISTORY
  Manual page write(1) line 1 (press h for help or q to quit)
```

2. ls

list로 ls [옵션] [파일] 형식의 명령어이다. 현재 디렉토리의 파일, 폴더 목록을 출력한다. 사용 예시는 다음과 같다.

1) ls -a -> 숨김 파일(.으로 시작하는 파일)을 포함한 모든 파일을 출력한다.

```
kw2023202043@ubuntu:~$ ls -a
.          .cache      .gnupg      Public      Videos
..         .config     .local      splab_commands  vmware.bundle
.bash_history Desktop    Music      .ssh        wget-hsts
.bash_logout Documents  Pictures   .sudo_as_admin_successful work
.bashrc    Downloads .profile   Templates
```

2) ls -F -> 파일 종류를 표시한다. 여기서 /는 디렉토리, *는 실행파일이다.

```
kw2023202043@ubuntu:~$ ls -F
Desktop/  Downloads/  Pictures/  splab_commands*  Videos/  work/
Documents/ Music/      Public/    Templates/      vmware.bundle*
```

3) ls -l -> 파일의 정보를 상세하게 출력한다.

```
kw2023202043@ubuntu:~$ ls -l
total 40
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Desktop
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Documents
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Downloads
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Music
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Pictures
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Public
-rwx--x--x 1 kw2023202043 kw2023202043 2690 Mar 22 06:44 splab_commands
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Templates
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Videos
-rwxrwxr-x 1 kw2023202043 kw2023202043 0 Mar 22 08:40 vmware.bundle
drwxrwxr-x 3 kw2023202043 kw2023202043 4096 Mar 22 09:37 work
```


3. pwd

Print working directory로 현재 디렉토리의 경로를 확인한다. pwd [OPTION] 형식이다. 사용 예시와 출력 예시는 다음과 같다.

```
kw2023202043@ubuntu:~$ pwd
/home/kw2023202043
```

4. cd

Change directory로 특정 디렉토리로 이동하는 명령어이다. cd [디렉토리명] 형식으로 사용예시는 다음과 같다.

```
kw2023202043@ubuntu:~$ pwd
/home/kw2023202043
kw2023202043@ubuntu:~$ ls
Desktop  Downloads  Pictures  splab_commands  Videos  work
Documents Music      Public    Templates        vmware.bundle
kw2023202043@ubuntu:~$ cd work
kw2023202043@ubuntu:~/work$ pwd
/home/kw2023202043/work
kw2023202043@ubuntu:~/work$ cd .
kw2023202043@ubuntu:~/work$ cd ..
kw2023202043@ubuntu:~$ cd work
kw2023202043@ubuntu:~/work$ cd ~
kw2023202043@ubuntu:~$ cd -
/home/kw2023202043/work
kw2023202043@ubuntu:~/work$
```

1) cd .. -> 한 단계 위의 상위 디렉토리로 이동한다.

2) cd ~ -> home 디렉토리로 이동한다.

5. cat

Concatenate 로 파일 내용을 출력한다. 또한 여러 파일을 연결하여 출력이 가능하다. cat [OPTION] [FILE] 형식으로 사용예시는 다음과 같다.

```
kw2023202043@ubuntu:~/work$ cat file1.txt
Hello This is file 1
kw2023202043@ubuntu:~/work$ cat file2.txt
Hello This is file 2
kw2023202043@ubuntu:~/work$ cat file1.txt file2.txt
Hello This is file 1
Hello This is file 2
kw2023202043@ubuntu:~/work$
```

6. chmod

Change mode로 파일 또는 디렉토리의 접근 권한을 변경한다.

- chmod [OPTION]... MODE[,MODE]... FILE... 형식이다. 대상은 다음과 같다. u: user (owner) g: group o: other a: all, 연산은 다음과 같다. +: 추가 -: 제거 =: 할당, 권한은 다음과 같다. r: read w: write x: execution

- chmod [OPTION]... OCTAL-MODE FILE... 형식이다. OCTAL-MODE는 8진수 숫자 세 개로 user(owner), group, other의 권한을 표현한다. 권한 숫자 표기법은 1: execute, 2: write, 4: read과 같다.

사용예시는 다음과 같다.

1) chmod a=rwx test -> test: 모든 대상에게 모든 권한 부여

```
kw2023202043@ubuntu:~/work$ ls -al
total 28
drwxrwxr-x 3 kw2023202043 kw2023202043 4096 Mar 22 09:37 .
drwxr-xr-x 16 kw2023202043 kw2023202043 4096 Mar 22 09:37 ..
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file1.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file2.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 2001 Mar 22 09:37 file3.txt
-r--r--r-- 1 kw2023202043 kw2023202043 41 Mar 22 09:37 hello.txt
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 22 09:37 SP_lab
kw2023202043@ubuntu:~/work$ chmod a=rwx hello.txt
```

2) chmod 777 test -> test: 모든 대상에게 모든 권한 부여

```
kw2023202043@ubuntu:~/work$ chmod a=rwx hello.txt
kw2023202043@ubuntu:~/work$ ls -l
total 20
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file1.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file2.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 2001 Mar 22 09:37 file3.txt
-rwxrwxrwx 1 kw2023202043 kw2023202043 41 Mar 22 09:37 hello.txt
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 22 09:37 SP_lab
kw2023202043@ubuntu:~/work$ chmod 777 hello.txt
```

7. mkdir

Make directory로 새로운 디렉토리를 생성하는 명령어이다. mkdir [OPTION] DIRECTORY 형식이다. 사용예시는 다음과 같다.

```

kw2023202043@ubuntu:~/work$ ls -l
total 20
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file1.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file2.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 2001 Mar 22 09:37 file3.txt
-rwxrwxrwx 1 kw2023202043 kw2023202043 41 Mar 22 09:37 hello.txt
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 22 09:37 SP_lab
kw2023202043@ubuntu:~/work$ mkdir SP_lecture
kw2023202043@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file1.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file2.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 2001 Mar 22 09:37 file3.txt
-rwxrwxrwx 1 kw2023202043 kw2023202043 41 Mar 22 09:37 hello.txt
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 22 09:37 SP_lab
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 23 05:47 SP_lecture

```

8. rmdir

Remove directory로 빈 디렉토리를 삭제하는 명령어이다. rmdir [OPTION] DIRECTORY 형식으로 사용예시는 다음과 같다.

```

kw2023202043@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file1.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file2.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 2001 Mar 22 09:37 file3.txt
-rwxrwxrwx 1 kw2023202043 kw2023202043 41 Mar 22 09:37 hello.txt
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 22 09:37 SP_lab
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 23 05:47 SP_lecture
kw2023202043@ubuntu:~/work$ rmdir SP_lecture
kw2023202043@ubuntu:~/work$ ls -l
total 20
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file1.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file2.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 2001 Mar 22 09:37 file3.txt
-rwxrwxrwx 1 kw2023202043 kw2023202043 41 Mar 22 09:37 hello.txt
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 22 09:37 SP_lab

```

9. rm

Remove로 파일 또는 디렉토리를 삭제하는 명령어이다. rm [OPTION] [FILE/DIRECTORY] 형식으로 사용예시는 다음과 같다.


```

kw2023202043@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file1.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file2.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 2001 Mar 22 09:37 file3.txt
-rwxrwxrwx 1 kw2023202043 kw2023202043 41 Mar 22 09:37 hello.txt
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 22 09:37 SP_lab
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 23 06:17 SP_lecture
kw2023202043@ubuntu:~/work$ rm -r SP_lecture
kw2023202043@ubuntu:~/work$ ls -l
total 20
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file1.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file2.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 2001 Mar 22 09:37 file3.txt
-rwxrwxrwx 1 kw2023202043 kw2023202043 41 Mar 22 09:37 hello.txt
d Show Applications 2023202043 kw2023202043 4096 Mar 22 09:37 SP_lab
kw2023202043@ubuntu:~/work$

```

10. cp

Copy로 파일 또는 디렉토리를 복사한다. cp [OPTION] SOURCE DEST/DIRECTORY 형식으로 사용예시는 다음과 같다.

```

kw2023202043@ubuntu:~/work$ ls -l
total 20
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file1.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file2.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 2001 Mar 22 09:37 file3.txt
-rwxrwxrwx 1 kw2023202043 kw2023202043 41 Mar 22 09:37 hello.txt
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 22 09:37 SP_lab
kw2023202043@ubuntu:~/work$ cp hello.txt hello_copy.txt
kw2023202043@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file1.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file2.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 2001 Mar 22 09:37 file3.txt
-rwxrwxr-x 1 kw2023202043 kw2023202043 41 Mar 23 06:20 hello_copy.txt
-rwxrwxrwx 1 kw2023202043 kw2023202043 41 Mar 22 09:37 hello.txt
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 22 09:37 SP_lab
kw2023202043@ubuntu:~/work$ ls -l
total 24
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file1.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file2.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 2001 Mar 22 09:37 file3.txt
-rwxrwxr-x 1 kw2023202043 kw2023202043 41 Mar 23 06:20 hello_copy.txt
-rwxrwxrwx 1 kw2023202043 kw2023202043 41 Mar 22 09:37 hello.txt
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 22 09:37 SP_lab
kw2023202043@ubuntu:~/work$ cp SP_lab/* .
kw2023202043@ubuntu:~/work$ ls -l
total 32
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file1.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 21 Mar 22 09:37 file2.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 2001 Mar 22 09:37 file3.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 15 Mar 23 06:20 fileA.txt
-rw-rw-r-- 1 kw2023202043 kw2023202043 15 Mar 23 06:20 fileC.txt
-rwxrwxr-x 1 kw2023202043 kw2023202043 41 Mar 23 06:20 hello_copy.txt
-rwxrwxrwx 1 kw2023202043 kw2023202043 41 Mar 22 09:37 hello.txt
drwxrwxr-x 2 kw2023202043 kw2023202043 4096 Mar 22 09:37 SP_lab
kw2023202043@ubuntu:~/work$

```

11. mv

Move로 파일을 이동하거나 이름을 변경하는 명령어이다. mv [OPTION] SOURCE DEST 형식으로 사용 예시는 다음과 같다.

```
kw2023202043@ubuntu:~/work$ ls
file1.txt  file3.txt  fileC.txt      hello.txt
file2.txt  fileA.txt  hello_copy.txt SP_lab
kw2023202043@ubuntu:~/work$ mv hello_copy.txt /home/kw2023202043/work/ex
kw2023202043@ubuntu:~/work$ ls
ex  file1.txt  file2.txt  file3.txt  fileA.txt  fileC.txt  hello.txt  SP_lab
```

12. ln

link로 하드링크 또는 심볼릭 링크를 생성하는 명령어이다. 하드링크와 심볼릭 링크의 차이는 다음과 같다.

하드링크	심볼릭 링크
파일에만 링크 가능	파일 또는 디렉토리에 링크 가능
존재하지 않는 파일에 대해 하드링크 작성 불가	존재하지 않는 파일에 대해 심볼릭 링크 작성 가능
연결되어 있는 파일인지 알기 어려움	연결되어 있는 파일을 찾기 용이
같은 파일 시스템 간에서만 작성 가능	다른 파일 시스템 간에서도 작성 가능
원본파일과 i-node 같음	원본파일과 i-node 다름

하드링크 생성 형식은 ln [원본파일] [생성할 하드 링크 파일]로 사용 예시는 다음과 같다.

```
kw2023202043@ubuntu:~$ ls
Desktop  Downloads  Music      Public      Templates  vmware.bundle
Documents file.txt   Pictures   splab_commands Videos      work
kw2023202043@ubuntu:~$ ln file.txt link.txt
kw2023202043@ubuntu:~$ ls -li file.txt link.txt
1835176 -rw-rw-r-- 2 kw2023202043 kw2023202043 0 Mar 23 06:35 file.txt
1835176 -rw-rw-r-- 2 kw2023202043 kw2023202043 0 Mar 23 06:35 link.txt
kw2023202043@ubuntu:~$
```

심볼릭 링크 생성 형식은 ln -s [원본파일] [생성할 심볼릭 링크 파일]로 사용 예시는 다음과 같다.


```

kw2023202043@ubuntu:~$ ln -s file.txt link.txt
kw2023202043@ubuntu:~$ ls -l
total 40
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Desktop
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Documents
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Downloads
-rw-rw-r-- 1 kw2023202043 kw2023202043 0 Mar 23 06:35 file.txt
lrwxrwxrwx 1 kw2023202043 kw2023202043 8 Mar 23 06:38 link.txt -> file.txt
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Music
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Pictures
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Public
-rwx--x--x 1 kw2023202043 kw2023202043 2690 Mar 22 06:44 splab_commands
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Templates
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Videos
-rwxrwxr-x 1 kw2023202043 kw2023202043 0 Mar 22 08:40 vmware.bundle
drwxrwxr-x 3 kw2023202043 kw2023202043 4096 Mar 23 06:22 work
kw2023202043@ubuntu:~$

```

13. touch

빈 파일을 생성하거나 수정 시간을 변경하는 명령어이다. touch [OPTION] FILE 형식으로 사용 예시는 다음과 같다.

```

kw2023202043@ubuntu:~$ ls
Desktop  Downloads  Pictures  splab_commands  Videos  work
Documents Music      Public    Templates        vmware.bundle
kw2023202043@ubuntu:~$ touch empty.txt
kw2023202043@ubuntu:~$ ls
Desktop  Downloads  Music      Public    Templates  vmware.bundle
Documents empty.txt  Pictures    splab_commands  Videos    work
kw2023202043@ubuntu:~$ ls -l
total 40
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Desktop
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Documents
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Downloads
-rw-rw-r-- 1 kw2023202043 kw2023202043 0 Mar 23 06:50 empty.txt
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Music
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Pictures
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Public
-rwx--x--x 1 kw2023202043 kw2023202043 2690 Mar 22 06:44 splab_commands
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Templates
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Videos
-rwxrwxr-x 1 kw2023202043 kw2023202043 0 Mar 22 08:40 vmware.bundle
drwxrwxr-x 3 kw2023202043 kw2023202043 4096 Mar 23 06:22 work
kw2023202043@ubuntu:~$ touch empty.txt
kw2023202043@ubuntu:~$ ls -l
total 40
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Desktop
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Documents
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Downloads
-rw-rw-r-- 1 kw2023202043 kw2023202043 0 Mar 23 06:51 empty.txt
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Music
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Pictures
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Public
-rwx--x--x 1 kw2023202043 kw2023202043 2690 Mar 22 06:44 splab_commands
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Templates
drwxr-xr-x 2 kw2023202043 kw2023202043 4096 Mar 22 08:27 Videos
-rwxrwxr-x 1 kw2023202043 kw2023202043 0 Mar 22 08:40 vmware.bundle
drwxrwxr-x 3 kw2023202043 kw2023202043 4096 Mar 23 06:22 work
kw2023202043@ubuntu:~$

```


14. ps

현재 실행 중인 프로세스 목록을 출력하는 명령어이다. ps [options] 형식으로 사용 예시는 다음과 같다.

1) ps -e -> 모든 프로세스 목록을 출력한다.

```
kw2023202043@ubuntu:~$ ps
  PID TTY          TIME CMD
 2785 pts/0        00:00:00 bash
 74302 pts/0        00:00:00 ps
kw2023202043@ubuntu:~$ ps -e
  PID TTY          TIME CMD
    1 ?            00:00:50 systemd
    2 ?            00:00:00 kthreadd
    3 ?            00:00:00 rcu_gp
    4 ?            00:00:00 rcu_par_gp
    5 ?            00:00:00 slub_flushwq
    6 ?            00:00:00 netns
    8 ?            00:00:00 kworker/0:0H-events_highpri
```

2) ps -f -> 상세 정보를 출력한다.

```
kw2023202043@ubuntu:~$ ps -f
UID          PID    PPID  C STIME TTY          TIME CMD
kw20232+    2785     2777  0 05:03 pts/0        00:00:00 bash
kw20232+    74639    2785  0 06:55 pts/0        00:00:00 ps -f
```

3) ps -ef -> 모든 프로세스 + 상세 정보를 출력한다.

```
kw2023202043@ubuntu:~$ ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root         1         0  0 04:45 ?            00:00:50 /sbin/init auto noprompt
root         2         0  0 04:45 ?            00:00:00 [kthreadd]
root         3         2  0 04:45 ?            00:00:00 [rcu_gp]
root         4         2  0 04:45 ?            00:00:00 [rcu_par_gp]
root         5         2  0 04:45 ?            00:00:00 [slub_flushwq]
root         6         2  0 04:45 ?            00:00:00 [netns]
root         8         2  0 04:45 ?            00:00:00 [kworker/0:0H-events_highp
root        10         2  0 04:45 ?            00:00:00 [mm_percpu_wq]
root        11         2  0 04:45 ?            00:00:00 [rcu_tasks_rude_]
root        12         2  0 04:45 ?            00:00:00 [rcu_tasks_trace]
root        13         2  0 04:45 ?            00:00:01 [ksoftirqd/0]
root        14         2  0 04:45 ?            00:00:05 [rcu_sched]
root        15         2  0 04:45 ?            00:00:00 [migration/0]
root        16         2  0 04:45 ?            00:00:00 [idle_inject/0]
root        18         2  0 04:45 ?            00:00:00 [cpuhp/0]
```

15. exit

터미널을 종료하는 명령어이다. 사용 예시는 다음과 같다.

```

kw2023202043@ubuntu:~$ sudo systemctl stop unattended-upgrades
[sudo] password for kw2023202043:
kw2023202043@ubuntu:~$ sudo apt-get install csh
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  csh
0 upgraded, 1 newly installed, 0 to remove and 91 not upgraded.
Need to get 243 kB of archives.
After this operation, 367 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu focal/universe amd64 csh amd64 20110502-5 [243 kB]
Fetched 243 kB in 4s (68.0 kB/s)
Selecting previously unselected package csh.
(Reading database ... 162653 files and directories currently installed.)
Preparing to unpack .../csh_20110502-5_amd64.deb ...
Unpacking csh (20110502-5) ...
Setting up csh (20110502-5) ...
update-alternatives: using /bin/bsd-csh to provide /bin/csh (csh) in auto mode
Processing triggers for man-db (2.9.1-1) ...
kw2023202043@ubuntu:~$ ps
  PID TTY          TIME CMD
  96709 pts/0        00:00:00 bash
  97727 pts/0        00:00:00 ps
kw2023202043@ubuntu:~$ csh
% ps
  PID TTY          TIME CMD
  96709 pts/0        00:00:00 bash
  97728 pts/0        00:00:00 csh
  97729 pts/0        00:00:00 ps
% exit
% exit
kw2023202043@ubuntu:~$ ps
  PID TTY          TIME CMD
  96709 pts/0        00:00:00 bash
  97731 pts/0        00:00:00 ps
kw2023202043@ubuntu:~$ █

```

16. kill

특정 프로세스 PID를 종료하는 명령어이다. kill [-s signal | -p] [-a] [--] pid 형식이며 ps aux | grep <name of program>으로 PID 확인이 가능하다. 사용 예시는 다음과 같다.

- 1) | -> 이전 명령어의 출력을 다음 명령어의 입력으로 연결한다. tail -> 파일의 끝 부분부터 10개의 행을 출력한다.
- 2) kill -9 -> 강제 종료한다.

```

kw2023202043@ubuntu:~$ ps -e | tail
 85190 ?          00:00:01 kworker/0:5-events
 96631 ?          00:00:01 gnome-terminal-
 96709 pts/0      00:00:00 bash
 97735 ?          00:00:00 kworker/0:0-cgroup_destroy
 97739 ?          00:00:00 kworker/u256:1-events_unbound
 97742 ?          00:00:00 fprintd
 97743 ?          00:00:00 kworker/1:1-events
 97763 ?          00:00:00 nautilus
 97794 pts/0      00:00:00 ps
 97795 pts/0      00:00:00 tail
kw2023202043@ubuntu:~$ kill 96709
kw2023202043@ubuntu:~$ ps -e | tail
 85180 ?          00:00:00 kworker/1:0-rcu_par_gp
 85190 ?          00:00:01 kworker/0:5-events
 96631 ?          00:00:01 gnome-terminal-
 96709 pts/0      00:00:00 bash
 97735 ?          00:00:00 kworker/0:0-rcu_par_gp
 97739 ?          00:00:00 kworker/u256:1-events_unbound
 97743 ?          00:00:00 kworker/1:1-events
 97799 ?          00:00:00 kworker/0:1-events
 97800 pts/0      00:00:00 ps
 97801 pts/0      00:00:00 tail
kw2023202043@ubuntu:~$

```

17. passwd

사용자의 비밀번호를 변경하는 명령어이다. 현재 사용자 비밀번호를 변경하며 다른 사용자 비밀번호를 변경하는 명령어 형식은 `sudo passwd <사용자명>`이다. 사용 예시는 다음과 같다.

```

kw2023202043@ubuntu:~$ passwd
Changing password for kw2023202043.
Current password:
New password:
Retype new password:
You must choose a longer password
New password:
Retype new password:
passwd: password updated successfully
kw2023202043@ubuntu:~$ osswd

```

18. uname

시스템 정보를 확인하는 명령어이다. `uname [Option]` 형식으로 사용 예시는 다음과 같다.

- 1) `uname -r` -> 커널 버전을 출력한다.
- 2) `uname -m` -> 기계 하드웨어 name을 출력한다.
- 3) `uname -a` -> 전체 시스템 정보를 출력한다.

```
kw2023202043@ubuntu:~$ uname
Linux
kw2023202043@ubuntu:~$ uname -r
5.15.0-134-generic
kw2023202043@ubuntu:~$ uname -m
x86_64
kw2023202043@ubuntu:~$ uname -a
Linux ubuntu 5.15.0-134-generic #145~20.04.1-Ubuntu SMP Mon Feb 17 13:27:1
6 UTC 2025 x86_64 x86_64 x86_64 GNU/Linux
kw2023202043@ubuntu:~$
```

19. wc

word count로 파일 내 줄, 단어, 문자 개수 등을 출력하는 명령어이다. wc [options] [FILE] 형식으로 사용 예시는 다음과 같다.

- 1) wc -l 파일명 -> Line 수 출력
- 2) wc -w 파일명 -> Word 수 출력
- 3) wc -c 파일명 -> Byte 수 출력
- 4) wc -m 파일명 -> character 수 출력

```
kw2023202043@ubuntu:~$ cat hello.txt
hello world
My name is N~~~
How are you?
kw2023202043@ubuntu:~$ wc hello.txt
 3  9 41 hello.txt
kw2023202043@ubuntu:~$
```

20. echo

터미널에 문자열을 출력하는 명령어이다. 변수를 출력할 때 "\$"를 붙여 명령하며 사용 예시는 다음과 같다.

```
 3  9 41 hello.txt
kw2023202043@ubuntu:~$ echo helloworld
helloworld
kw2023202043@ubuntu:~$ echo $Home

kw2023202043@ubuntu:~$ echo ~
/home/kw2023202043
kw2023202043@ubuntu:~$
```

21. alias

명령어 단축어를 설정하는 명령어이다. 긴 명령어를 단축하여 사용할 수 있도록 하는 명령어로 사용 예시는 다음과 같다.

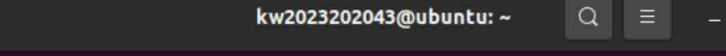
```
kw2023202043@ubuntu:~/work$ myls
Command 'mysls' not found, did you mean:
  command 'tyls' from deb terminology (1.6.0-2)
  command 'mmls' from deb sleuthkit (4.6.7-1build1)
Try: sudo apt install <deb name>

kw2023202043@ubuntu:~/work$ aliad myls='ls -al'
aliad: command not found
kw2023202043@ubuntu:~/work$ alias myls='ls -al'
kw2023202043@ubuntu:~/work$ myls
total 28
drwxrwxr-x  3 kw2023202043 kw2023202043 4096 Mar 23 08:57 .
drwxr-xr-x 15 kw2023202043 kw2023202043 4096 Mar 23 09:04 ..
-rw-rw-r--  1 kw2023202043 kw2023202043   21 Mar 23 08:57 file1.txt
-rw-rw-r--  1 kw2023202043 kw2023202043   21 Mar 23 08:57 file2.txt
-rw-rw-r--  1 kw2023202043 kw2023202043 2001 Mar 23 08:57 file3.txt
-rw-rw-r--  1 kw2023202043 kw2023202043   41 Mar 23 08:57 hello.txt
drwxrwxr-x  2 kw2023202043 kw2023202043 4096 Mar 23 08:57 SP_lab
kw2023202043@ubuntu:~/work$ 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$//'\''}"'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
alias myls='ls -al'
kw2023202043@ubuntu:~/work$
```

22. grep

특정 문자열을 찾는 명령어이다. 지정된 입력 파일에서 특정 문자열을 찾는데 사용된다. grep [options] [PATTEN] [FILE...] 형식으로 사용 예시는 다음과 같다.

```
kw2023202043@ubuntu:~$ cat hello.txt
hello world
My name is N~~~
H Help e you?
kw2023202043@ubuntu:~$ grep hello hello.txt
hello world
kw2023202043@ubuntu:~$
```

```
kw2023202043@ubuntu: ~  
1 2023202043  
2 Kwangwoon University  
3 Choi EunJun  
4 Kwangwoon University  
~  
~  
~  
"2023202043" 4L, 65C written                2,1                All
```



```
kw2023202043@ubuntu:~$ vi
kw2023202043@ubuntu:~$
```

:wq를 입력하여 저장 후 종료한다.

```
kw2023202043@ubuntu:~$ cat 2023202043
2023202043
Kwangwoon University
Choi EunJun
Kwangwoon University
kw2023202043@ubuntu:~$
```

잘 작성된 모습이다.

2. Make

1) kw_hello.c file

```
kw2023202043@ubuntu:~$ cd Desktop
kw2023202043@ubuntu:~/Desktop$ ls
kw2023202043@ubuntu:~/Desktop$ vi kw_hello.c
```

vi를 실행하여 kw_hello.c 파일을 생성하고 편집한다.

```
kw2023202043@ubuntu: ~/Desktop
#include <stdio.h>

int main(){
    printf("2023202043\n");
    printf("Choi EunJun\n");
    return 0;
}
~
~
-- INSERT -- 7,2 All
```

i key를 눌러하여 Insert 모드로 진입한 뒤, 학번과 이름이 출력되도록 코드를 작성한다.
이때 printf()를 사용하여 코드를 작성한다.

```
kw2023202043@ubuntu: ~/Desktop
#include <stdio.h>

int main(){
    printf("2023202043\n");
    printf("Choi EunJun\n");
    return 0;
}
~
~
:wq
```

ESC key를 누르고 :wq로 저장 후 Enter key를 눌러 종료한다.

```
kw2023202043@ubuntu:~$ cd Desktop
kw2023202043@ubuntu:~/Desktop$ ls
a.out  hello  kw_hello.c  Makefile
kw2023202043@ubuntu:~/Desktop$ cat kw_hello.c
#include <stdio.h>

int main(){
    printf("2023202043\n");
    printf("Choi EunJun\n");
    return 0;
}
kw2023202043@ubuntu:~/Desktop$
```

잘 작성된 모습이다.

2) gcc 컴파일러

```
kw2023202043@ubuntu:~/Desktop$ sudo apt install gcc
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  binutils binutils-common binutils-x86-64-linux-gnu cp
```

sudo apt install gcc 를 입력하여 gcc를 설치한다.

```
kw2023202043@ubuntu:~/Desktop$ gcc --version
gcc (Ubuntu 9.4.0-1ubuntu1~20.04.2) 9.4.0
Copyright (C) 2019 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE
.
kw2023202043@ubuntu:~/Desktop$ ls
```

잘 설치된 모습이다.

```
kw2023202043@ubuntu:~/Desktop$ ls
kw_hello.c
kw2023202043@ubuntu:~/Desktop$ gcc kw_hello.c
kw2023202043@ubuntu:~/Desktop$ ls
a.out  Show Applications  kw_hello.c
kw2023202043@ubuntu:~/Desktop$
```

gcc kw_hello.c를 입력하여 컴파일을 완료한다.

```
kw2023202043@ubuntu:~/Desktop$ ls
kw_hello.c
kw2023202043@ubuntu:~/Desktop$ gcc kw_hello.c
kw2023202043@ubuntu:~/Desktop$ ls
a.out  kw_hello.c
kw2023202043@ubuntu:~/Desktop$ ./a.out
2023202043
Choi EunJun
kw2023202043@ubuntu:~/Desktop$
```

./a.out을 입력하여 kw_hello.c를 실행한다. 학번과 이름이 출력된 것을 확인할 수 있다.

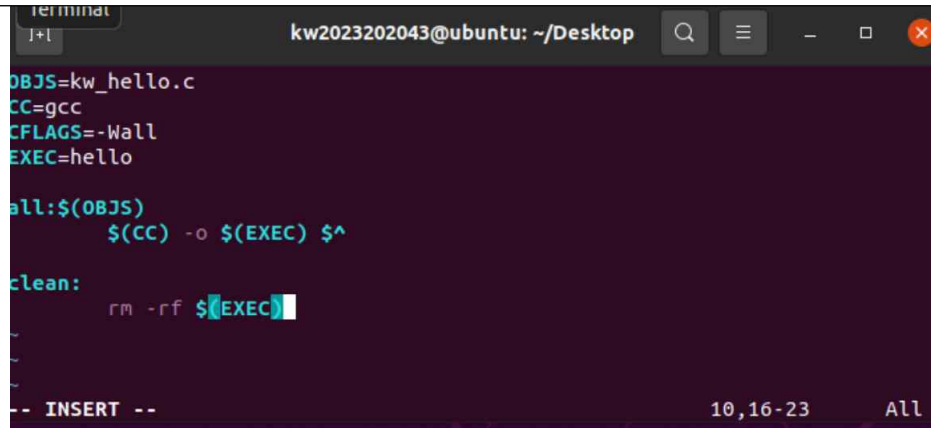
```
Choi EunJun
kw2023202043@ubuntu:~/Desktop$ gcc -o hello kw_hello.c
kw2023202043@ubuntu:~/Desktop$ ls
a.out  hello  kw_hello.c
kw2023202043@ubuntu:~/Desktop$
```

\$ gcc -o (실행파일명 = hello) (소스파일명 = kw_hello.c) 를 입력하여 실행파일명을 hello로 지정한다.

3) make

```
kw2023202043@ubuntu:~$ cd Desktop
kw2023202043@ubuntu:~/Desktop$ ls
a.out  hello  kw_hello.c
kw2023202043@ubuntu:~/Desktop$ vi Makefile
```

vi Makefile 을 입력하여 Makefile을 실행한다.



```
kw2023202043@ubuntu: ~/Desktop
OBJS=kw_hello.c
CC=gcc
CFLAGS=-Wall
EXEC=hello

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

clean:
    rm -rf $(EXEC)

-- INSERT --
```

변수명을 각각 설정하여 all이라는 타겟에 kw_hello.c인 OBJS를 하위 타겟으로 추가한다.

변수는 이름=값 형식으로 지정하며 타겟은 \$^로 대체할 수 있다. 변수는 \$(이름)로 사용한다.

```
kw2023202043@ubuntu:~/Desktop$ cat Makefile
OBJS=kw_hello.c
CC=gcc
CFLAGS=-Wall
EXEC=hello

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

clean:
    rm -rf $(EXEC)
kw2023202043@ubuntu:~/Desktop$
```

잘 작성된 모습이다.

```
kw2023202043@ubuntu:~/Desktop$ ls
a.out  hello  kw_hello.c  Makefile
kw2023202043@ubuntu:~/Desktop$ make
gcc -o hello kw_hello.c
kw2023202043@ubuntu:~/Desktop$ ls
a.out  hello  kw_hello.c  Makefile
kw2023202043@ubuntu:~/Desktop$ ./hello
2023202043
Choi EunJun
kw2023202043@ubuntu:~/Desktop$
```

Makefile을 컴파일하여 kw_hello.c를 출력한다. 출력된 모습이다.

4. 고찰

이번 리눅스 기반 프로그래밍 과제를 수행하며 기본적인 리눅스 명령어, vi 편집기와 Makefile 작성 등의 개념을 실습할 수 있었다.

기본적인 리눅스 명령어로 파일 및 디렉토리 관리 명령어, 파일 내용 확인 및 수정 명령어, 프로세스 및 시스템 관리 명령어 등을 알게 되었다.

처음에는 vi 편집기사용이 익숙하지 않아 어려웠다. v명령 모드, 입력 모드 등으로 동작하기 때문에 익숙해지기까지 시간이 걸렸다. i키를 눌러 입력 모드로 전환해야 텍스트를 입력할 수 있고, ESC키를 눌러 명령 모드로 전환해야 저장(:w), 종료(:q), 복사(yy), 붙여넣기(p) 등을 실행할 수 있는 것을 알게 되었다. 또한, yy(한 줄 복사), dd(한 줄 삭제), p(붙여넣기) 등의 단축키를 활용할 수 있었다. 그리고 :set number명령어를 입력하면 vi 편집기에서 줄 번호를 표시할 수 있어 코드 수정이 편리해짐을 알 수 있었다. 이렇게 처음에는 vi의 구조가 낯설었지만, 반복적으로 사용하면서 vi 편집기는 리눅스 환경에서 필수적인 도구라는 것을 깨달았다.

Makefile을 사용하여 C 프로그램을 컴파일하는 과정을 통해 Makefile을 활용하면 여러 개의 소스 파일이 있을 때도 자동으로 빌드를 수행할 수 있다는 점이 굉장히 유용함을 알 수 있었다. 특히 make clean을 활용한 불필요한 파일을 삭제할 수 있는 것과, 자동 컴파일을 통해 실수를 줄일 수 있음이 유용했다.

이렇게 Linux 명령어를 실습하면서 기본적인 이해가 깊어졌고, vi 편집기 사용법을 익히며 터미널에서 효율적으로 작업하는 방법을 배울 수 있었다. 또한, Makefile을 사용하여 C 프로그램을 컴파일하고 관리하는 방법을 익힐 수 있었다.