
Introduction to Software Design

Lab1: C Environment Setting for Lectures & Assignments

Yoonsang Lee
Spring 2020

Topic Covered

- 우분투 설치
- 리눅스 기본 명령어
- Vim 기본 사용법
- "Hello World!" 예제

Development Environment

- Development environment for this course:
- Python : Python 3 on any OSs
- **C: gcc on Ubuntu**

추천 환경

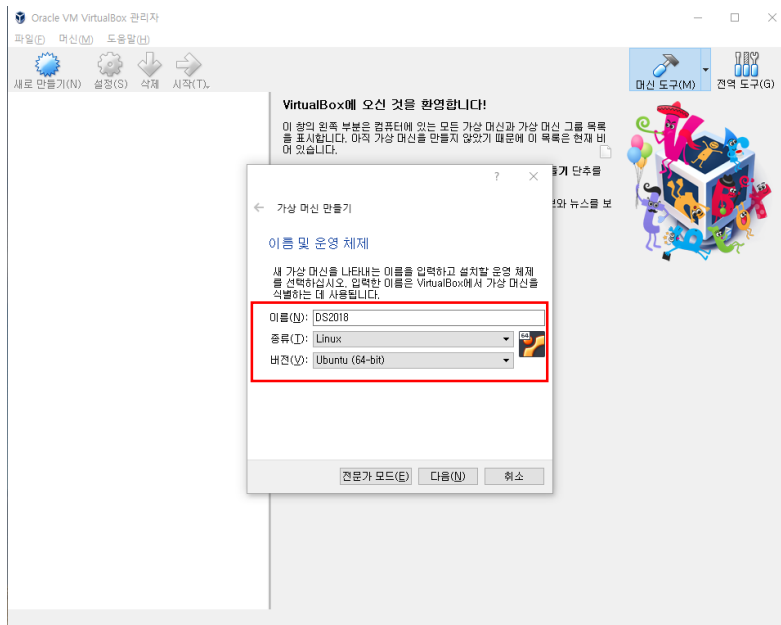
- (Ubuntu가 아닌 다른 OS라면) Virtual machine:
<http://www.virtualbox.org/>
- Ubuntu: <http://releases.ubuntu.com/18.04>
- Editor: vim을 추천하나 gedit 혹은 다른 editor를 사용해도 됨.

우분투 설치

Ubuntu 설치

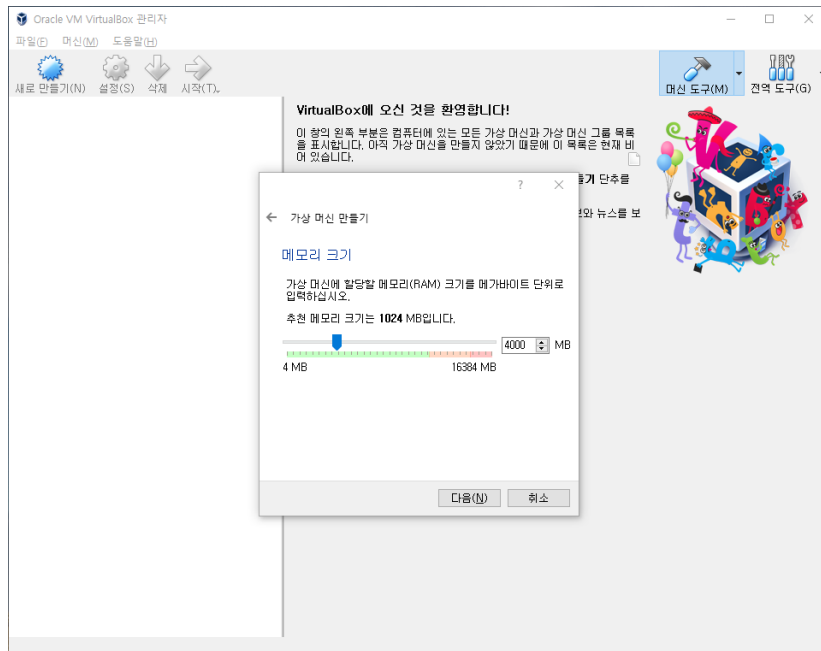
- 이미 Ubuntu가 설치된 컴퓨터를 사용하고 있다면 그대로 사용하면 됨.
- 이후 슬라이드는 다른 OS를 사용하고 경우를 가정해서 설명함.

How to install Ubuntu



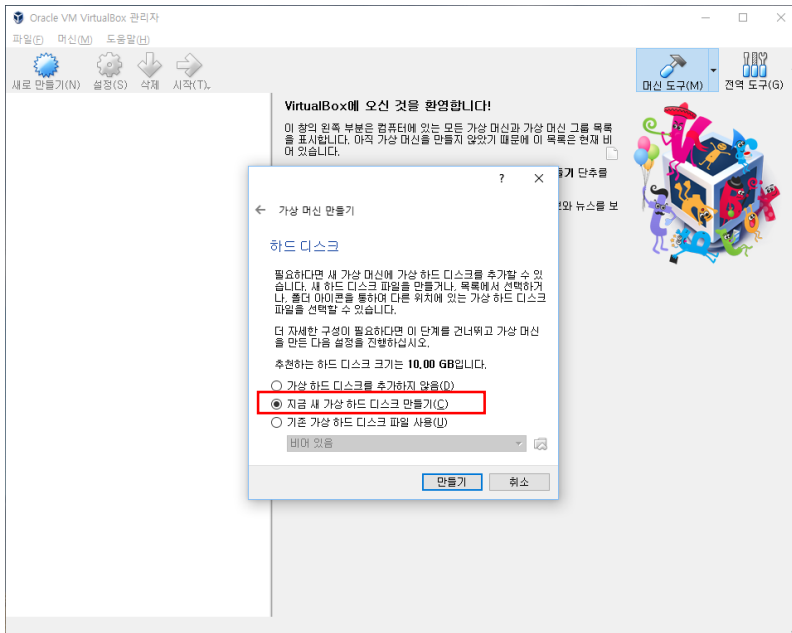
- 이름 : (임의 설정)
- 종류 : Linux
- 버전 : Ubuntu (64-bit)

How to install Ubuntu



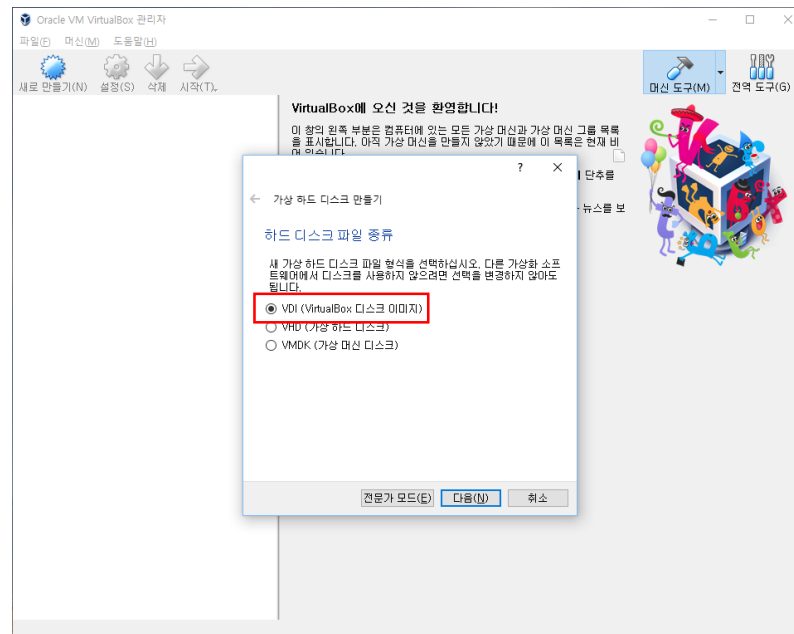
- 메모리 크기 : (임의 설정)

How to install Ubuntu

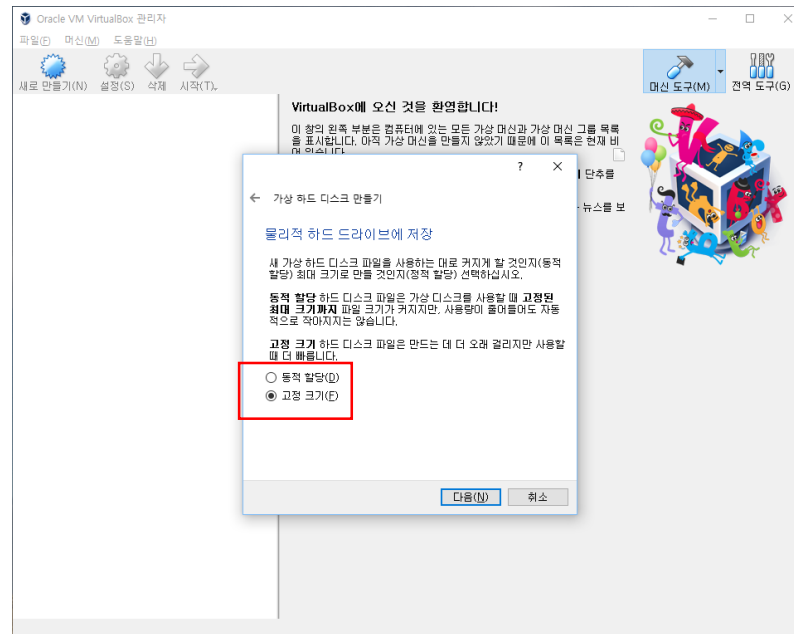


- 지금 새 가상 하드 디스크 만들기

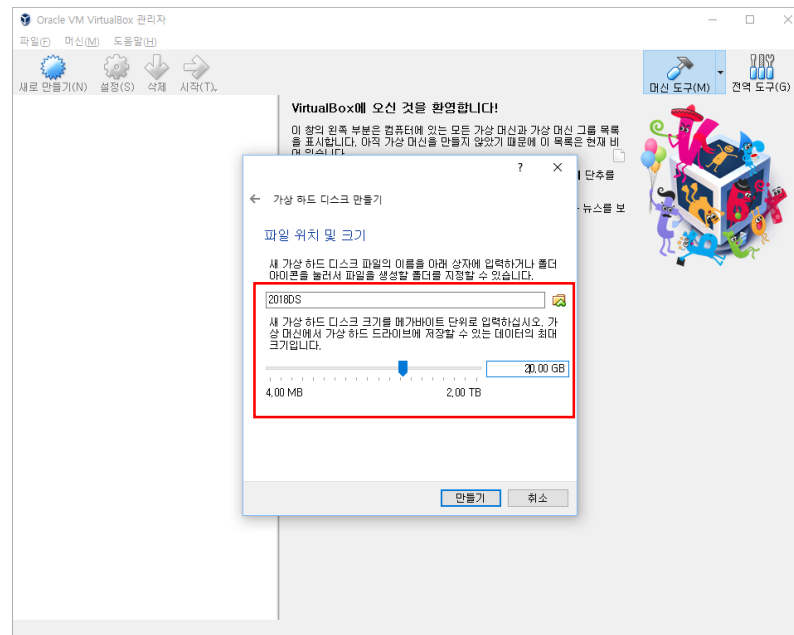
How to install Ubuntu



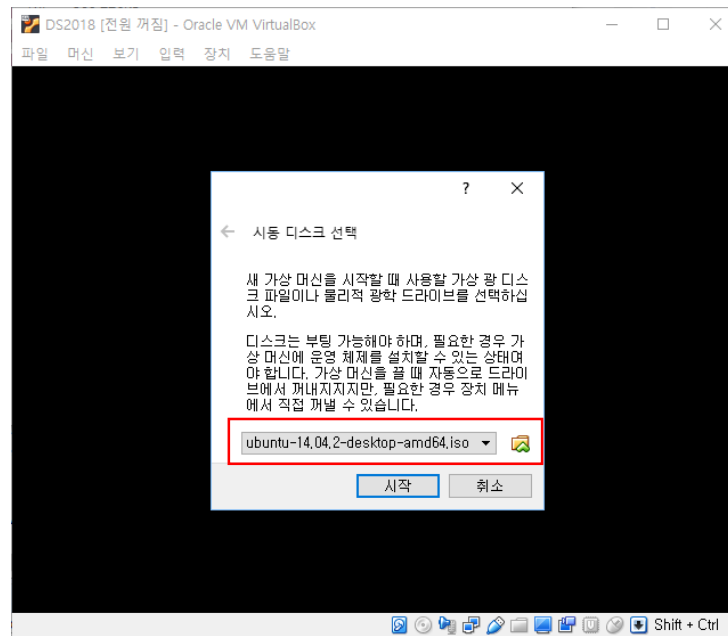
How to install Ubuntu



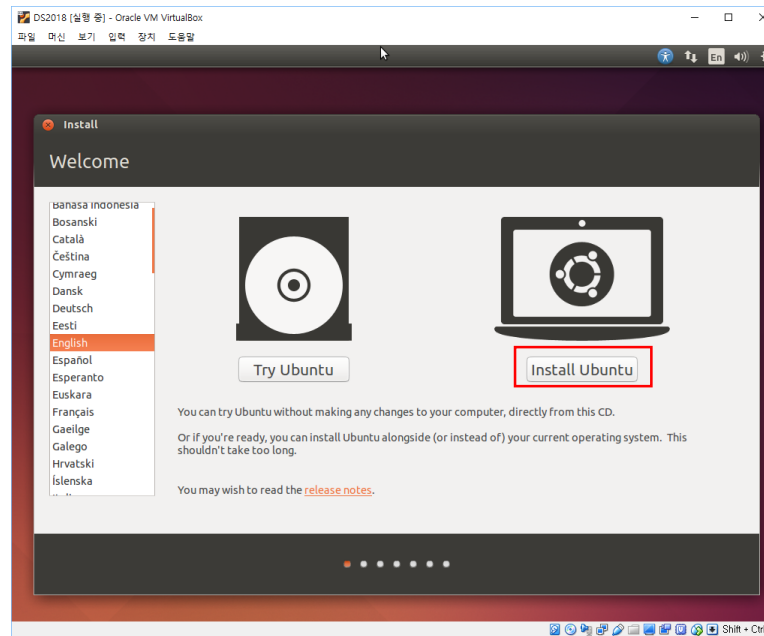
How to install Ubuntu



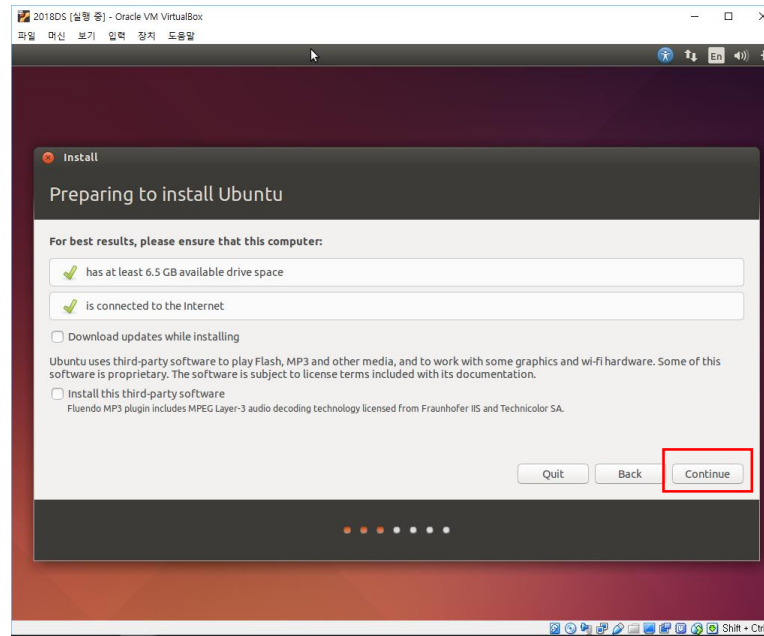
How to install Ubuntu



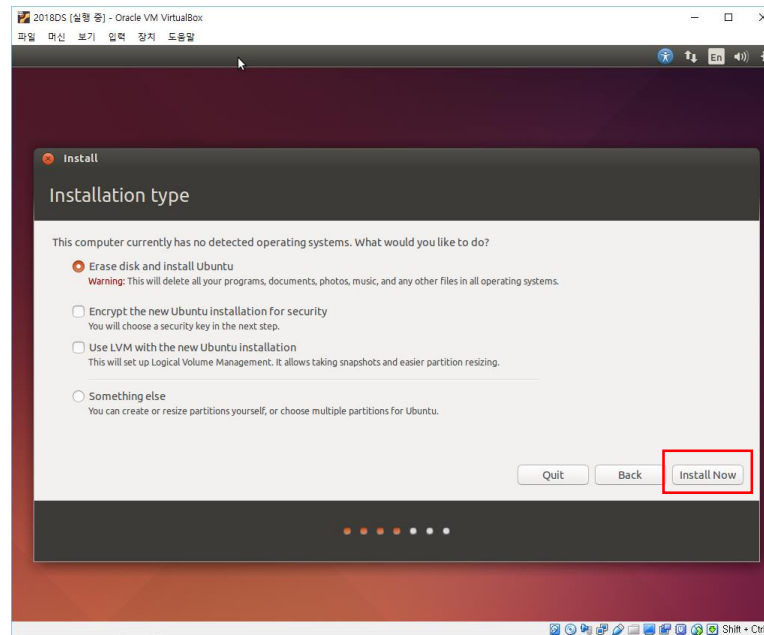
How to install Ubuntu



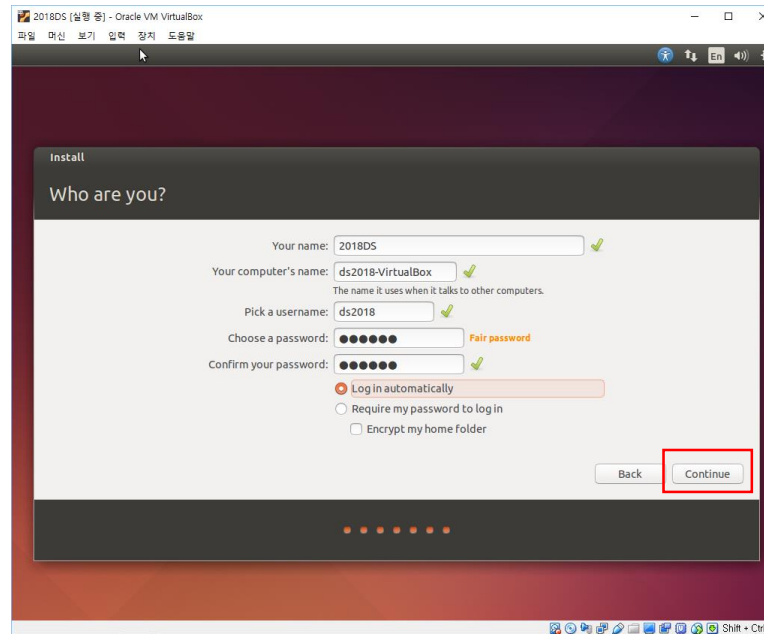
How to install Ubuntu



How to install Ubuntu



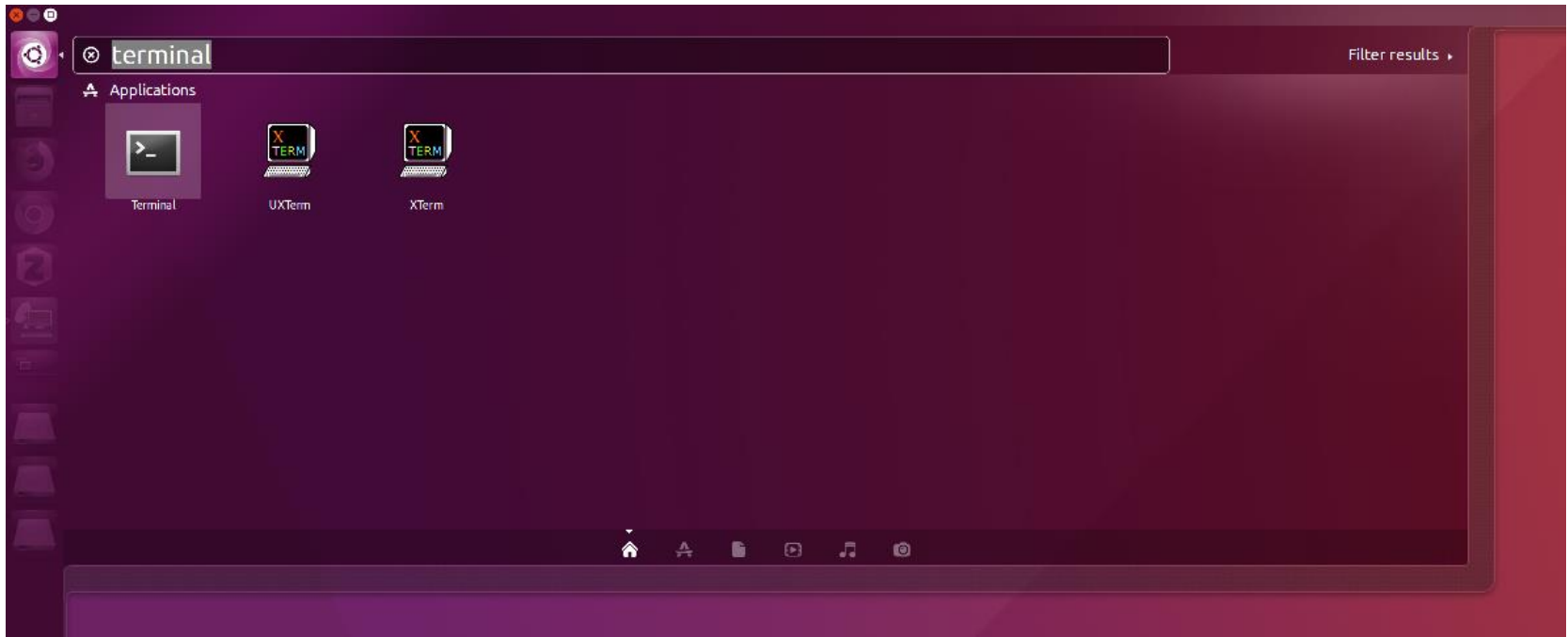
How to install Ubuntu



리눅스 기본 사용법

Terminal 실행

- Click Dash button (Start button)
- Type “terminal” and click Terminal
- or use Shortcut: CTRL + ALT + T



How to use Terminal

- Retrieve file on current directory

```
(Shell – home directory)
```

```
$ ls
```

- Current Location

```
(Shell – home directory)
```

```
$ pwd  
/home/<user> # this is your Home Directory
```

How to use Terminal

- Directory type
 - Normal directory : <dir-name>
 - Current directory : .
 - Parent directory : ..
 - Root directory : /
 - Home directory : ~
- Path type
 - Absolute address : /<dir1>/<dir2> ..
 - Relative address: : <dir1>/<dir2>

How to use Terminal

- Make directory

(Shell)

```
$ mkdir <dir-name>
```

- Change (current working) directory

(Shell)

```
$ cd <destination directory>
```

- Remove file, directory

(Shell)

```
$ rm <file-name>
```

(Shell)

```
$ rm -rf <dir-name>
```

How to use Terminal

- Move source(s) to destination directory.

(Shell)

```
$ mv <source file> <destination directory>
```

(Shell)

```
$ mv <source directory> <destination directory>
```

- Rename SOURCE to DEST

(Shell)

```
$ mv <SOURCE> <DEST>
```

How to use Terminal

- Copy

(Shell)

```
$ cp <source file> <destination directory>
```

(Shell)

```
$ cp <source file> <destination file>
```

(Shell)

```
$ cp -r <source directory> <destination directory>
```


그 밖의 명령어

- “cat {file_name}” : 파일 내용 출력
- “vi {file_name}” : 해당 파일 vi편집기로 편집(파일이 없으면 생성)
- “gedit {file_name}” : 해당 파일 gedit편집기로 편집(파일이 없으면 생성)

명령어 실행 예제

- ls 명령어 실행 예제

```
seongil@seongil-VirtualBox:~$ ls
Desktop  Downloads  Music      Public  Templates test.c
Documents examples.desktop Pictures  seongil test      Videos
```

- cat 명령어 실행 예제

```
seongil@seongil-VirtualBox:~$ cat test.c
#include<stdio.h>
int main (void)
{
    printf("Apple\n");
    return 0;
}
```

- vi {test.c}, vi {test.py} 명령어 실행 예제

```
seongil@seongil-VirtualBox: ~
#include<stdio.h>
int main (void)
{
    printf("Apple\n");
    return 0;
}
```

```
seongil@seongil-VirtualBox: ~
print("Apple")
```

Vim 기본 사용법

Vim



- **Vi IMproved**
- 기존의 Vi 편집기에 수많은 개선사항을 추가한 편집기 (Bram Moolenaar가 1991년에 첫 공개)
 - Vi: BSD의 핵심 개발자인 Bill Joy가 1976년 만든 편집기
 - 현재는 Vim이 훨씬 더 많이 쓰인다.
- 현재 대부분의 Linux 시스템에 포함된 기본 편집기
 - Vim을 다룰 줄 알면 Linux에서 작업 시 여러모로 편하다.
- Vim의 3가지 작동 모드:
 - Normal mode
 - Insert mode
 - Command-line mode

Normal Mode

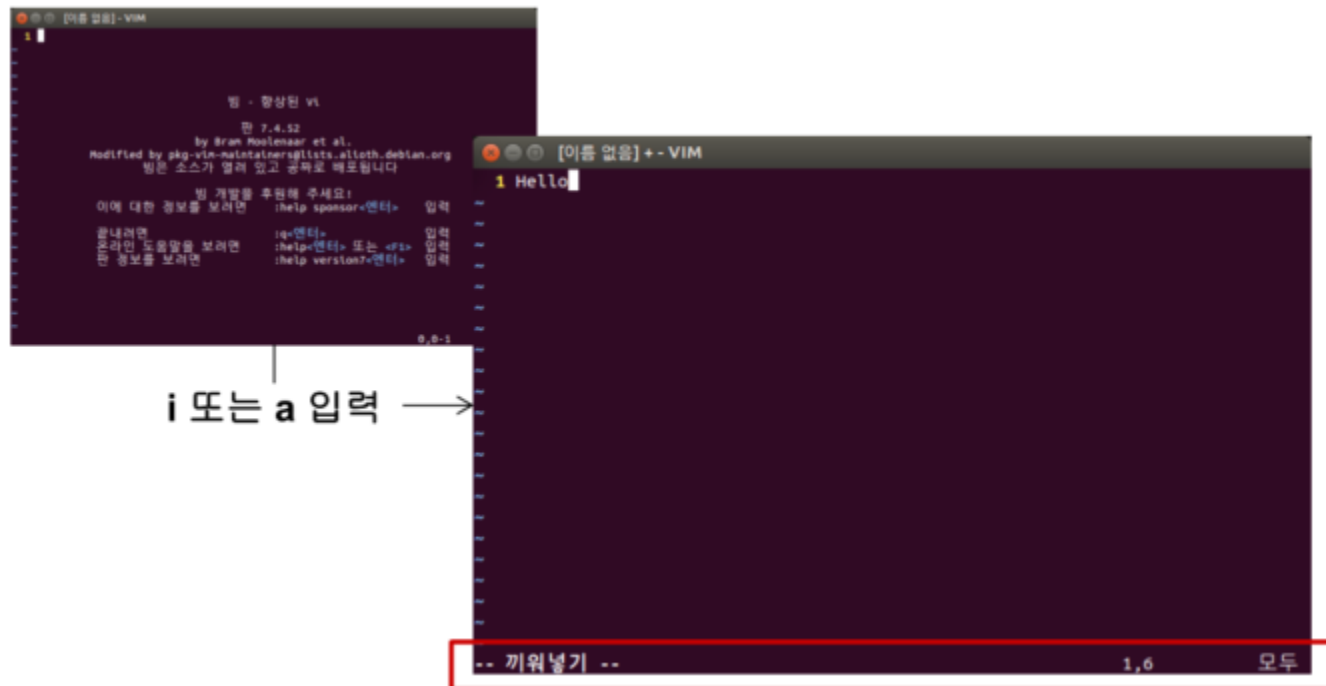
- Vim을 처음 실행하면 Normal mode 상태임.
- 복사, 붙이기, 삭제, 검색 등의 기능을 단축키를 통해 사용 가능한 모드.
- 아무것도 보이지 않지만 명령어를 기다리고 있는 상태임.



The screenshot shows a terminal window with a dark purple background. The title bar at the top reads "[아름 없음] +- VIM". The main area contains a single line of text "1 Hello" with a white cursor at the end. The bottom status bar is red and displays "1.6" and "10/11".

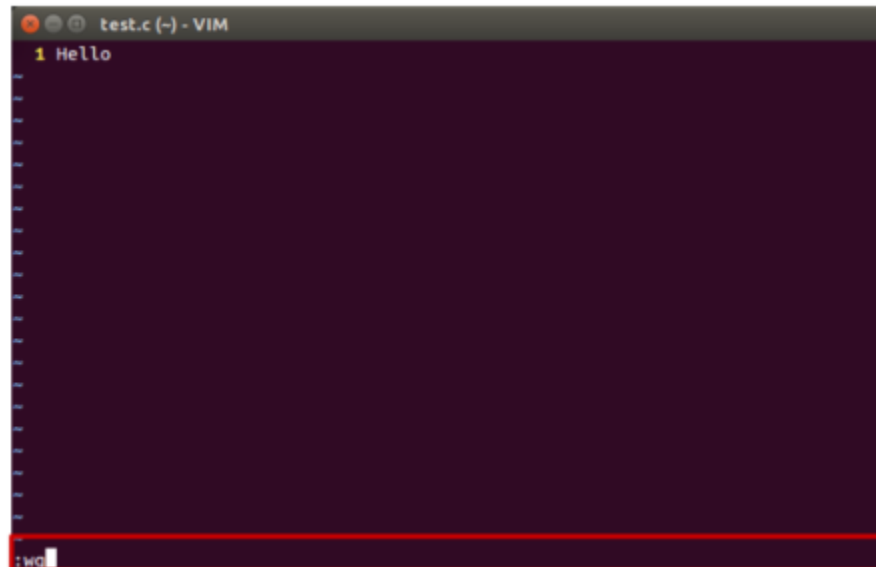
Insert Mode

- Normal mode에서 **a** 또는 **i**를 누르면 Insert mode로 전환.
- Insert mode에서는 일반적인 텍스트 편집기에서 작업하듯이 파일의 내용을 입력 및 수정 할 수 있음.
- Insert mode모드에서 **ESC** 를 누르면 Normal mode로 다시 돌아감.



Command-line Mode

- Normal mode에서 **:** 를 누르면 Command-line mode 로 진입.
- 명령줄에 명령어를 입력하여 vim의 여러 기능을 사용할 수 있는 모드.
 - w : 저장, q : 종료
 - ! : 강제명령 (ex : wq! , q!, 강제저장 및 강제종료)



The screenshot shows a terminal window titled "test.c (-) - VIM". The main area contains the text "1 Hello". At the bottom, a red bar highlights the command-line prompt ":wq", indicating the user is in command-line mode.

Vim 기본 사용법 참고 자료

- Vim에 익숙하지 않거나 처음 접하는 경우, Vim에 기본으로 포함된 tutorial인 vimtutor를 반드시 해볼 것

(Shell)

```
vimtutor
```

- 참고: System Settings – User Accounts – 본인 계정의 Language를 한국어로 변경 – logout & login하면 한글로 된 vimtutor를 볼 수 있다.
- 추가로 해볼만한 tutorial
 - Interactive Vim tutorial <http://www.openvim.com/tutorial.html>

Vim 명령어 참고 자료

- Vim Cheat Sheet : <https://vim.rtorr.com/lang/ko/>
 - 명령어가 잘 정리되어 있음
 - 앞으로 계속 참고할만한 자료
- Vim에서 `:help <명령어이름>`을 입력하면 해당 명령어에 대한 도움말을 볼 수 있다.

Vim 커서 이동 명령 요약

vim - movement commands

back

forward

previous
gg
first line

n ?text N
find word under cursor previous text find text next text

C-b C-u
page 1/2 page

H
screen

{ (paragraph sentence

;

previous x

B b
delimited word

gE ge
delimited end

k up

l right

e end

W word

tx before x

fx find x

\$ line

0 line

^ non-blank

Fx find x

Tx after x

b word

ge end

h left

j down

) sentence

} paragraph

E delimited end

W delimited word

;

next x

absolute movements

'' last location

'.' last edit

#G line #

% matching bracket

C-d C-f
1/2 page page

N /text n *
previous text find text next text find word under cursor

G
last line
next



Vim Advanced Usage

- Shell settings for convenient vim use
- `.vimrc` - vim configuration file
- Vim visual mode
- Vim windows
- Vim plug-ins
- Vim color schemes

- For details, see the supplementary material: *6-Lab1-VimAdvanced.pdf*

'Hello World!' 예제

Hello World 예제

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

```
int main(void)
```

```
{
```

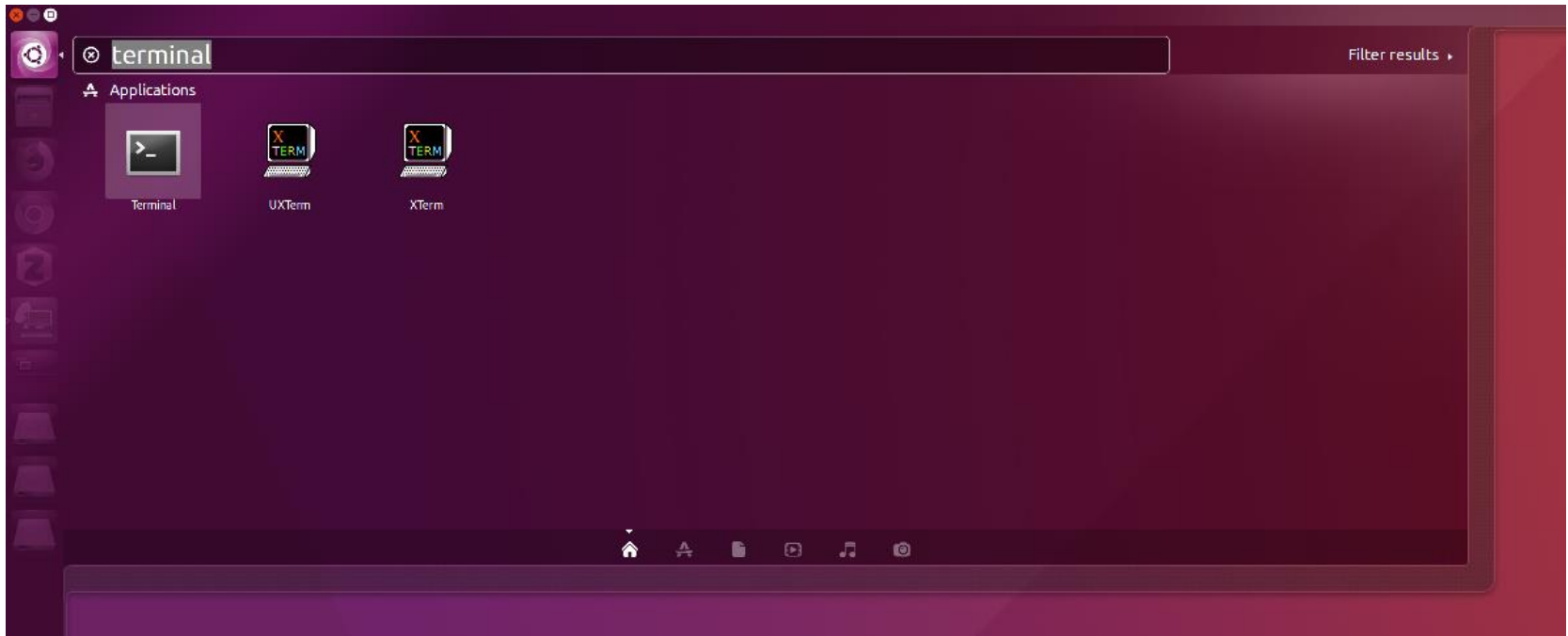
```
    printf("Hello world! \n");
```

```
    return 0;
```

```
}
```

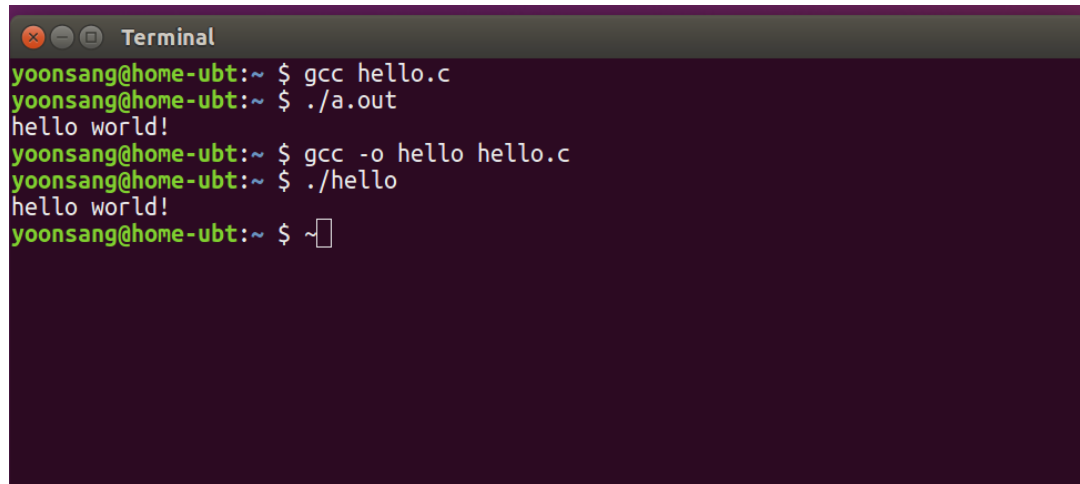
Hello World 컴파일 & 실행 (Ubuntu)

- Click Dash button (Start button)
- Type “terminal” and click Terminal



Hello World 컴파일 & 실행 (Ubuntu)

- `$ gcc hello.c` (컴파일 & 링크)
- `$./a.out` (생성된 실행파일 실행)
- 혹은
- `$ gcc -o hello hello.c` (실행파일의 이름을 hello로 지정)
- `$./hello` (생성된 실행파일 실행)



```
Terminal
yoonsang@home-ubt:~ $ gcc hello.c
yoonsang@home-ubt:~ $ ./a.out
hello world!
yoonsang@home-ubt:~ $ gcc -o hello hello.c
yoonsang@home-ubt:~ $ ./hello
hello world!
yoonsang@home-ubt:~ $ ~
```


Hello World 컴파일 & 실행 (Ubuntu)

- vim 대신 gedit와 같은 다른 text editor를 사용해서 편집 및 저장한 후 terminal에서 gcc로 컴파일 및 실행할 수도 있다.
- 하지만 vim에 익숙해지면 Linux/Unix 시스템 사용 시 도움이 되는 부분이 많기 때문에, 본 강좌의 실습에서는 vim을 사용하는 것을 권함.