# Welcome to SWE2007-44
## System Software Experiment 2

Prof. Joonwon Lee(joonwon@skku.edu)
TA – Sungin Hong(sungin.h@skku.edu),
Sewan Ha(hsewan2495@gmail.com)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu

# Introduction

- **Schedule**
  - 18:00 – 21:45 (Thursday)
  - Lecture room: 27419B (2nd Engineering Building)

- **Course homepage**
  - http://csl.skku.edu/SWE2007F18/Overview

  Or
  - http://icampus.ac.kr

# About Professor

- **Joonwon Lee**
  - Professor @ CE & SW Dept.
  - Computer Systems Laboratory
  - Office: 산학협력센터 #85572 (5th floor)

  - Email: joonwon@skku.edu
  - URL: http://csl.skku.edu/People/Joon
  - Tel: 031-299-4592
  - Office hours: Monday & Wednesday
  - The best way to contact him is by **email**.

# About TA

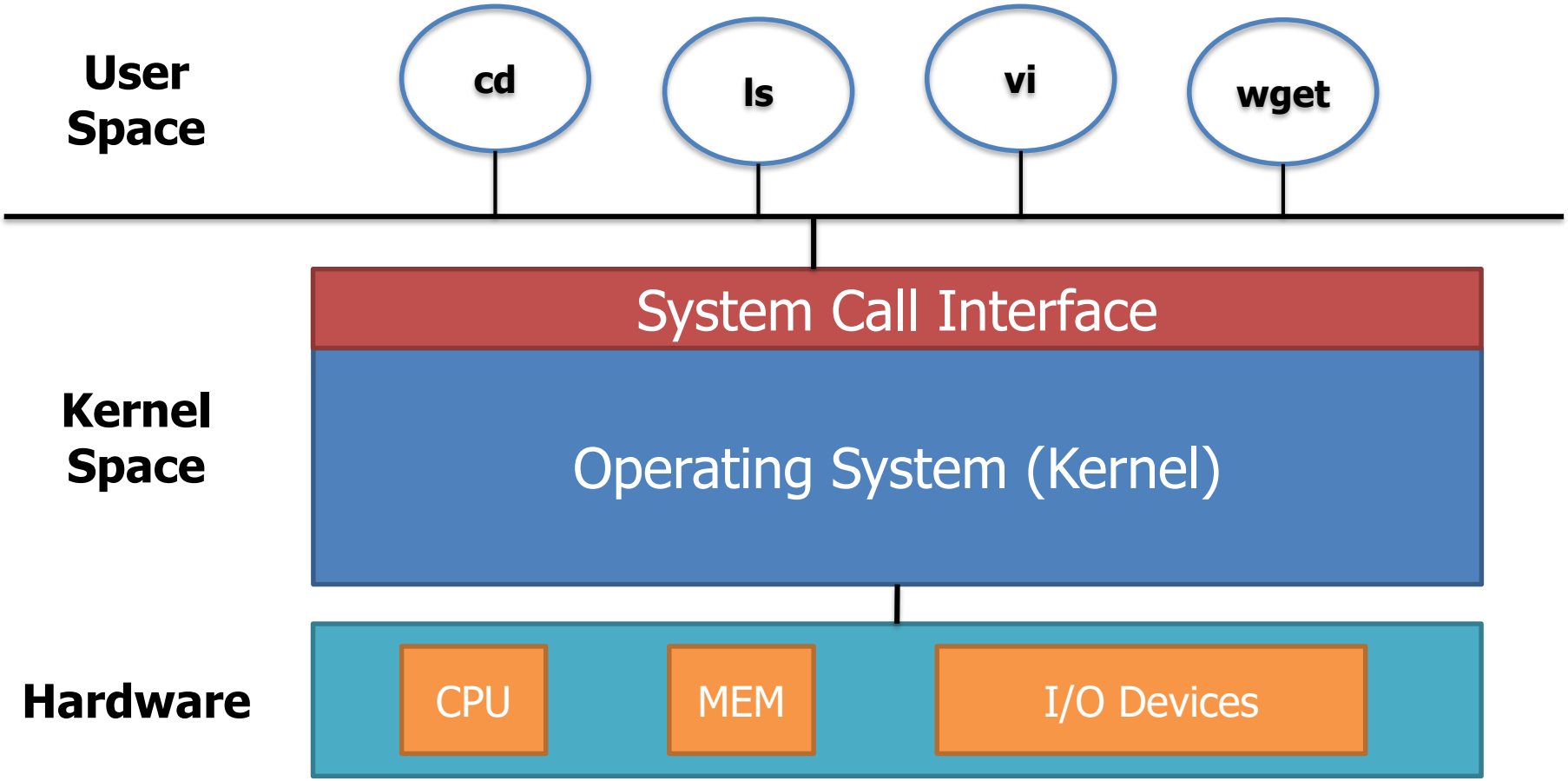
- **Sungin Hong**
  - Advanced Research on Compilers and Systems Laboratory
  - Office: 산학협력센터 #85565 (5th floor)
  - Email: sungin.h@skku.edu
- **Sewan Ha**
  - Infromation Security Group
  - Office: 제2공학관 #27306 (3rd floor)
  - Email: hsewan2495@gmail.com

**Please add "[SWE2007]" in the title**

# Course Outline (1)

- **Course description**
  - This course is intended to make students be familiar with Linux systems. We will learn how to install and setup your own Linux system and review the **basic Linux commands**. We move on to various **system calls provided by Linux systems for advanced programming**. No prior knowledge on the Linux system is required.

# Course Outline (2)



# Course Outline (3)

- **Why we use Linux?**
  - Used in many scientific and industrial settings
  - Internet servers and services run on Linux
  - It's free!
- **How to use Linux?**
- **How to make [advanced] programs on Linux?**
  - We will learn various system calls provided by Linux systems

# Course Outline (4)

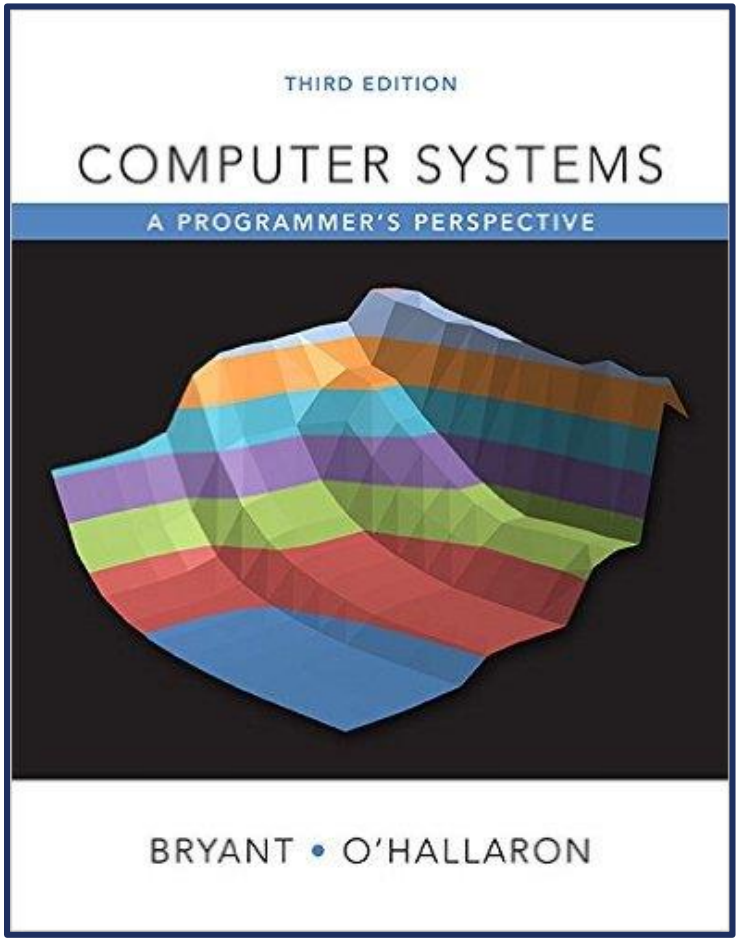
- **Very basic Linux usages (commands & tools)**
  - Shell, text editor, compiler
- **Basic Linux system calls**
  - File I/O, Process management
  - Inter-Process Communication (IPC)
- **Network programming**
  - Sockets
- **Concurrent programming**
  - Processes, Threads

# Reference

- **Computer Systems: A Programmer's Perspective**
  - Randal E. Bryant and David R. O'Hallaron, Third Edition,

    Pearson Education Inc. 2015.

  - http://csapp.cs.cmu.edu



# Class Policies (1)

- **Grading Policy (subject to change)**
  - Class attendance (10%)
  - 5 Programming assignments, TBD (90%)

- **There will be no exam**

# Class Policies (2)

- **Cheating Policy (Important)**
  - Example of cheating
    - Turning in someone else's work as your own.
    - Allowing someone else to turn in your work as his or her own
    - Several people writing one solution and turning in multiple copies
  - Example of NOT cheating?
    - Getting or giving help on using systems or tools
    - High-level discussing on how to solve the problem
    - Getting or giving help on how to solve minor syntax error
  - Penalty for cheating:
    - Severe penalty on the grade and report to dept. chair
  - Feel free to ask me for help!

# Tentative Schedule

| Day | Topic | Reading | Assignment |
|---|---|---|---|
| 9/6 (T) | Course overview | | |
| 9/13 (T) | Introduction to Linux | | PA#0 |
| 9/20 (T) | File I/O | | PA#1 |
| 9/27 (T) | Process | | |
| 10/4 (T) | Signals | | PA#2 |
| 10/11 (T) | IPC (Pipes and FIFOs) | | |
| 10/18 (T) | Recitation session | | |
| 10/25 (T) | Midterm exam week | | |
| 11/1 (T) | Signals | | |
| 11/8 (T) | Sockets | | PA#3 |
| 11/15 (T) | Concurrent programming | | |
| 11/22 (T) | Pthreads | | PA#4 |
| 11/29 (T) | Pthreads (cont'd) | | PA#5 |
| 12/6 (T) | Threads Synchronization | | |
| 12/13 (T) | Course summary | | |
| 12/20 (T) | Final exam week | | |

# Any Questions?

# Coding in Linux

Prof. Joonwon Lee(joonwon@skku.edu)
TA – Sungin Hong(Sungin.h@skku.edu)
Computer Systems Laboratory
Sungkyunkwan University
http://csl.skku.edu

# Contents

- Coding standard
- Debugging tool
- Text editor

# Coding standard (1)

- **A rule for writing a source code**

- **Pros**
  - Specify a common format for the source code and comments
  - Allows developers to easily share code.
  - Looks better!

- **Cons**
  - It's bothering

# Coding standard (2)

- There are many coding standards
- The most important thing is consistency

- You can use "Linux kernel coding style"
  - https://github.com/torvalds/linux/blob/master/Documentation/process/coding-style.rst
  - https://wiki.kldp.org/wiki.php/LinuxKernelCodingStyle (in Korean)

# Debugging tool (1)

- **Still use "printf"?**
  - In multithread program?

```
sanghoon@test:~$ ./a.out
HHi, I'm Thread No.i, I'm Thread No.Hi, I'm Thread No.2
1
3
```

  - You have to recompile the source code every time
  - How to debug a "segmentation fault"?

- **GDB**
  - Debugging tool for GNU project
  - $ sudo apt-get install gdb
  - Compiler option '-g' is needed
  - Usage : gdb <Executable File>

# Debugging tool (2)

- **Commands for GDB**
  - R : Run program
  - B [FuncName/FileName:LineNum] : Set breakpoint
  - P : Print variables
  - S : Step (Go in to function)
  - N : Next (Skip function)
  - C : Continue until gdb meets breakpoint
  - Bt : Print backtrace of all stack frame
    (Use this when segmentation fault occurs)
  - Q : Quit
  - H : Help

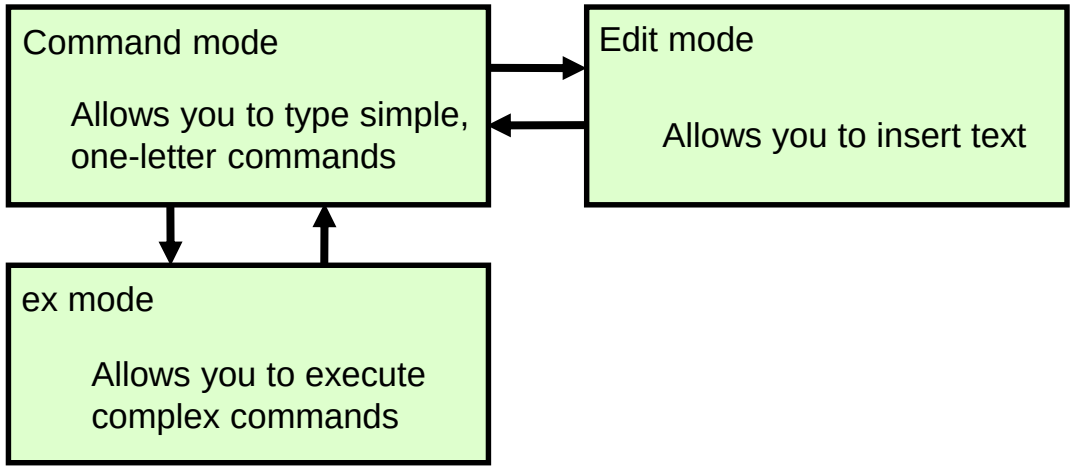
# Text editor – Vim (1)

- **Vi & Vim**
  - Vi is the default editor in all UNIX operating systems
  - It may be hard to learn, but it is useful
  - Vi in Linux is usually Vim (Vi Improved)
  - You can easily install Vim
    - $ sudo apt-get install vim

# Text editor – Vim (2)

- **Vi has three mode**
  - Edit mode (insert text)
  - Command mode (for simple, one-letter commands)
  - ex mode (for complicated commands)

- **You can easily change between modes.**



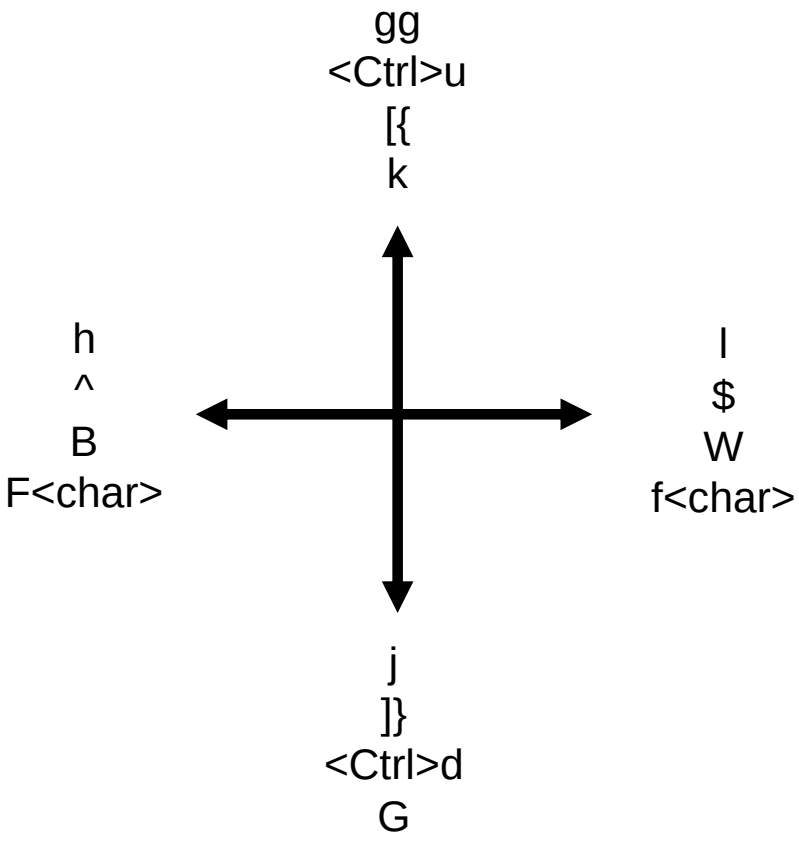
# Text editor – Vim (3)

- **Basic interface**
  - i, a, o, s : Insert mode
  - h, j, k, l : Cursor mode
  - ':' '/' : Command mode

- **Insert mode**
  - Indicated at left lower side
  - Press 'Esc' key to return

# Text editor – Vim (4)

- **Cursor movement in command mode**

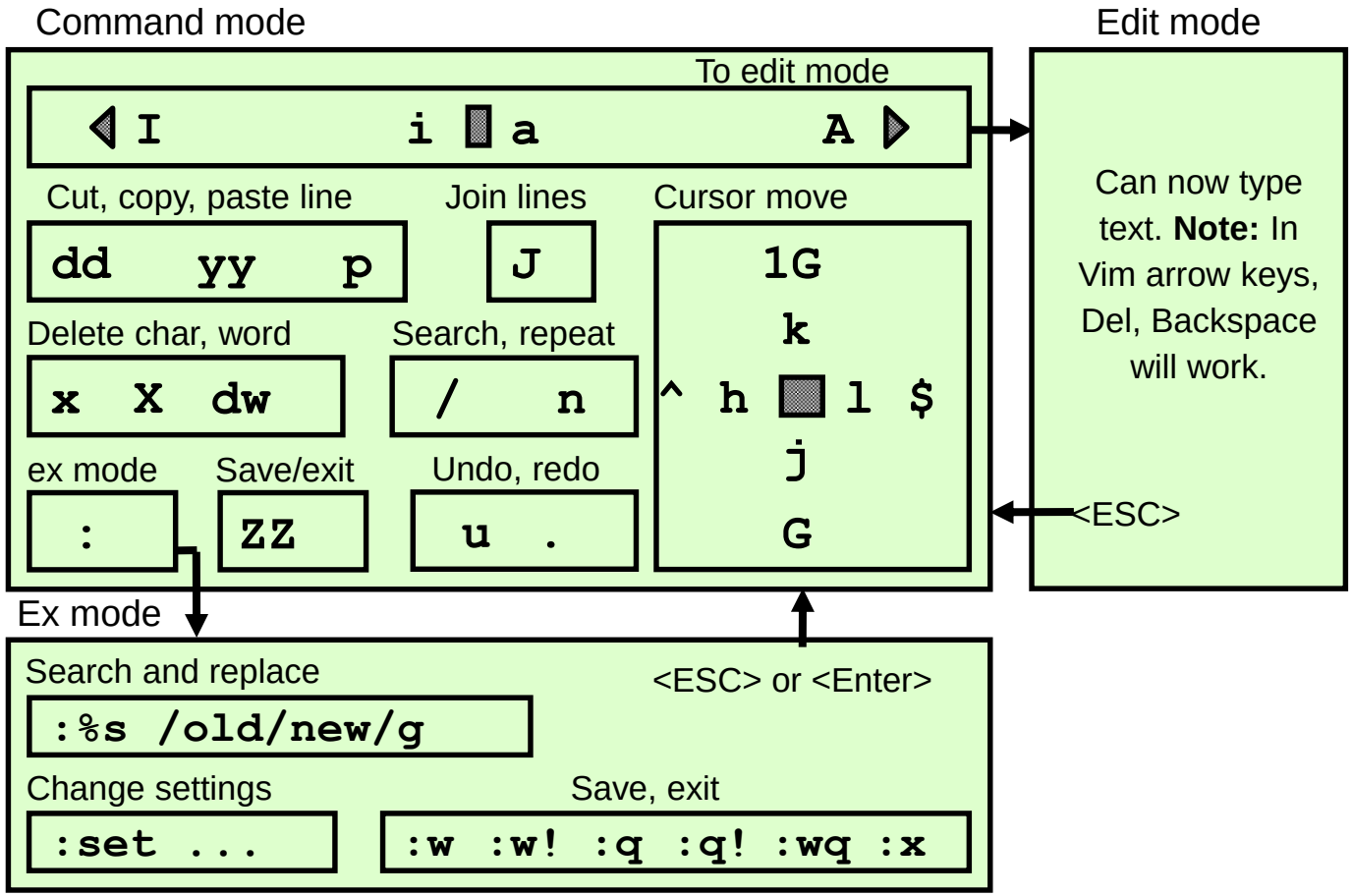


# Text editor – Vim (5)

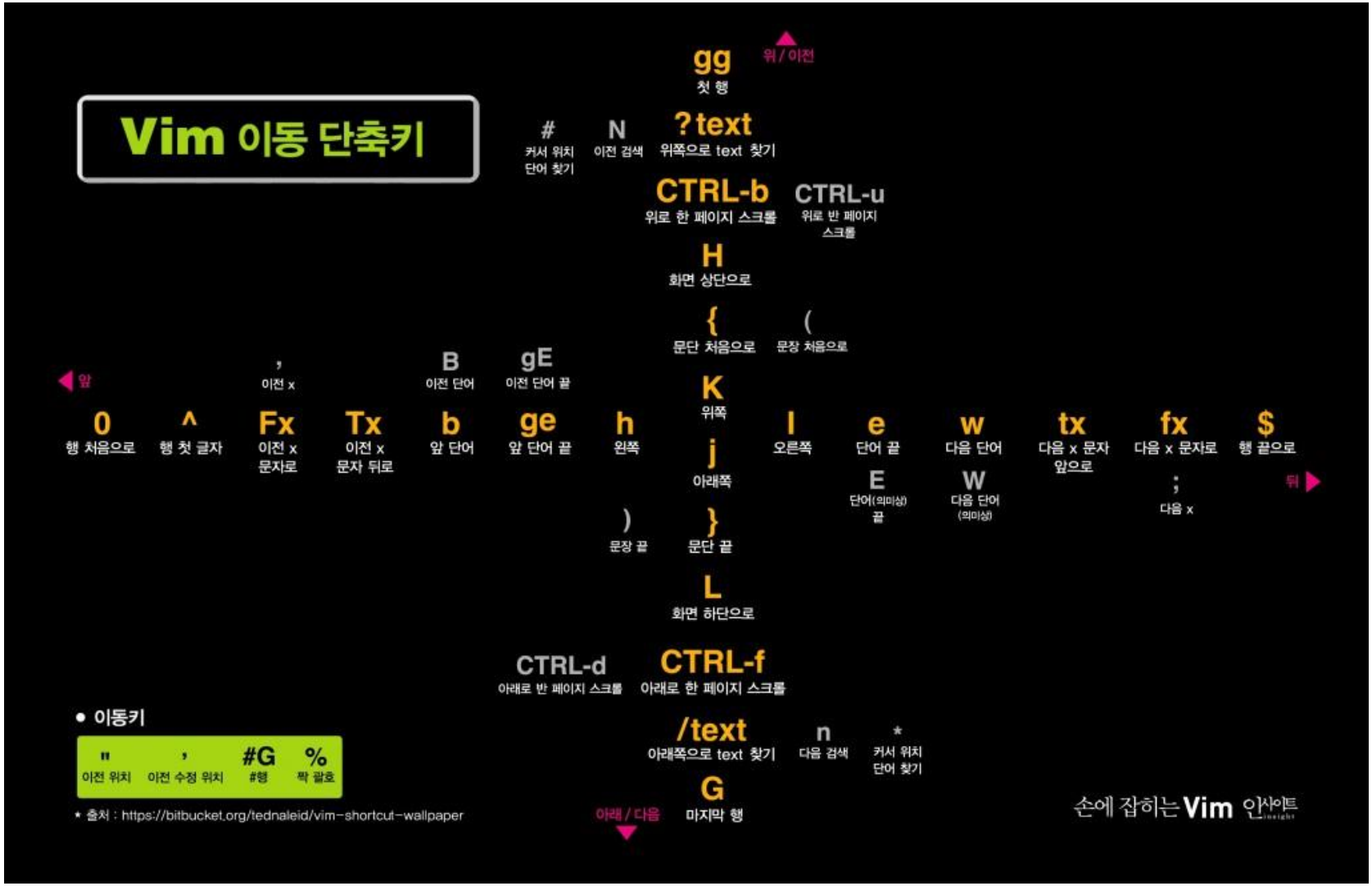
- **Exiting Vi**
  - To save in ex mode
    - `:w`
  - To quit without saving in ex mode
    - `:q`
  - To forcefully exit in ex mode without saving changes
    - `:q!`
  - To save and exit in ex mode (recommended)
    - `:wq`

# Text editor – Vim (6)

- Vi cheat sheet



# Text editor – Vim (7)



# Text editor – Vim (8)

# Text editor – Vim (9)

- **For learning Vim ..**
  - Vim Adventures (Game)
    - http://vim-adventures.com/
  - Vim Tutorial
    - http://www.openvim.com/tutorial.html
- **Repeat, repeat, and repeat.**

# Installing Ubuntu on VirtualBox

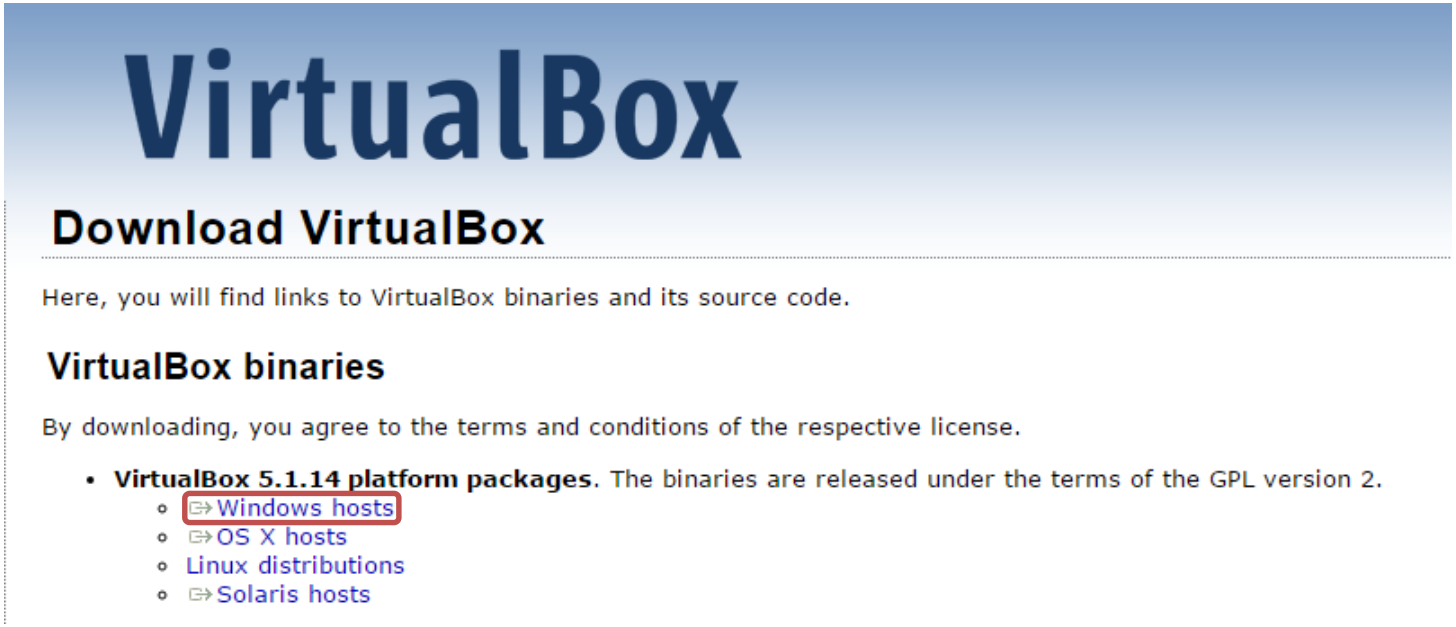
SK UNIVERSITY

# Steps

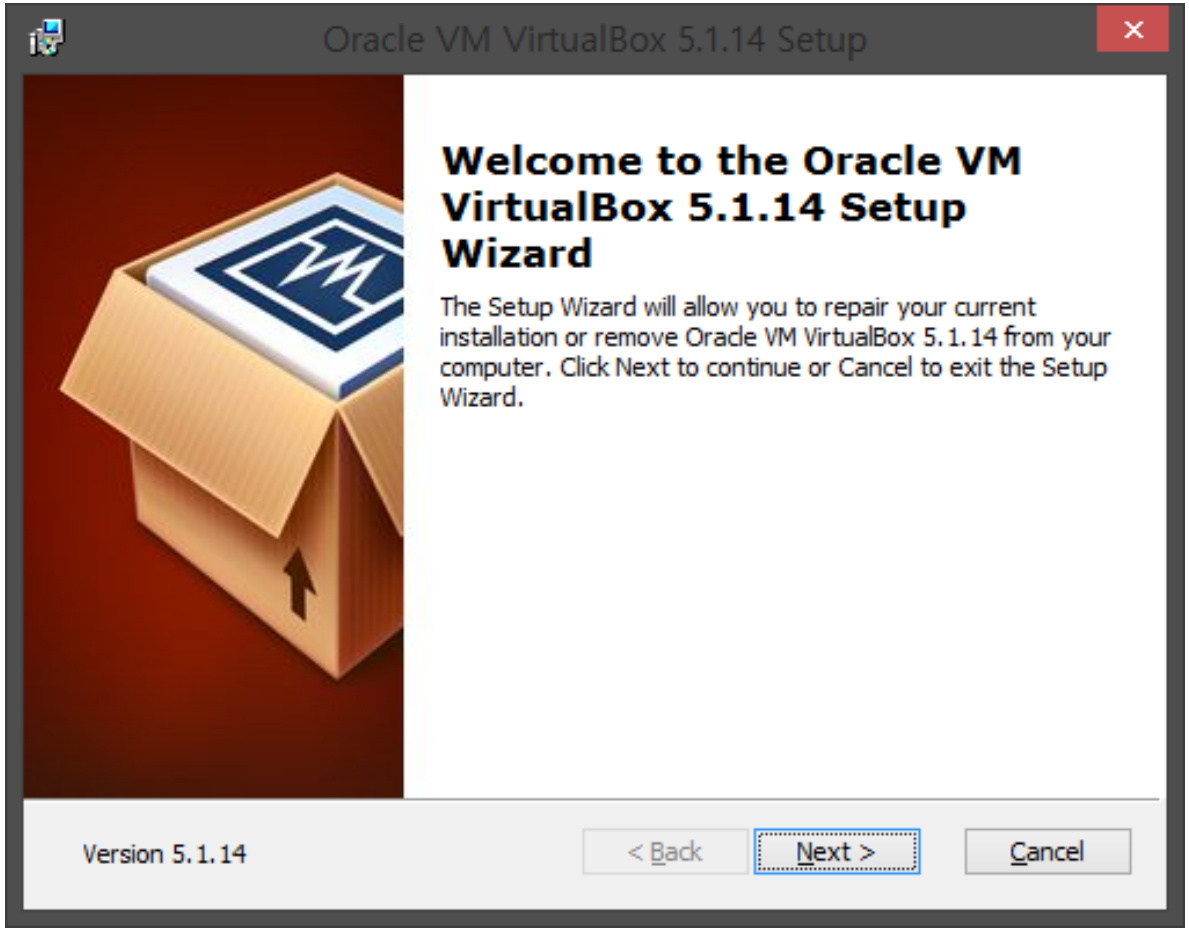
1. Install VirtualBox on your computer
2. Create a virtual machine (VM)
3. Install Ubuntu on the VM
4. Fun

# Installing VirtualBox (1)

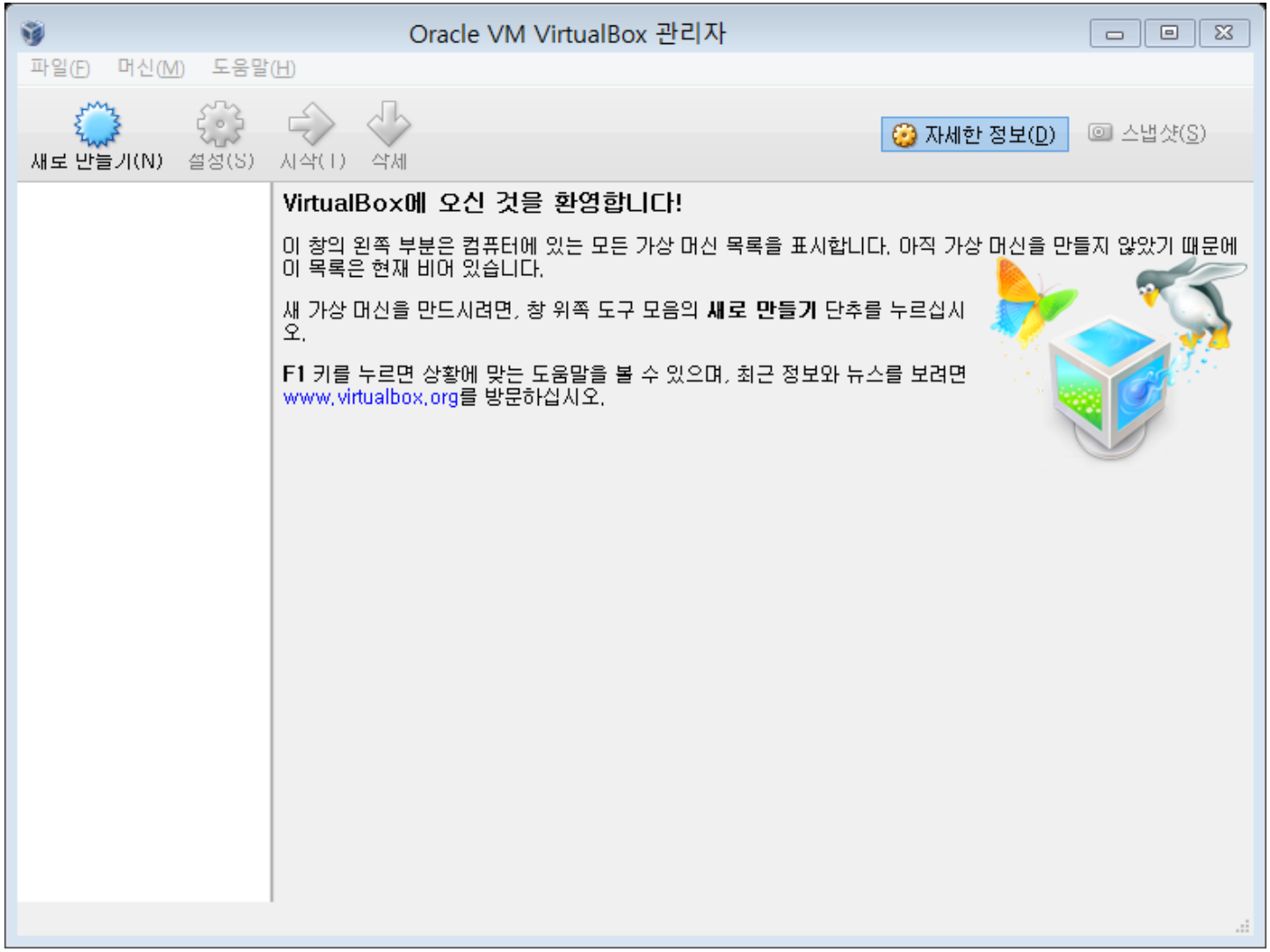
- **Go to VirtualBox's downloads page**
  - https://www.virtualbox.org/wiki/Downloads
- **Download installation binary**



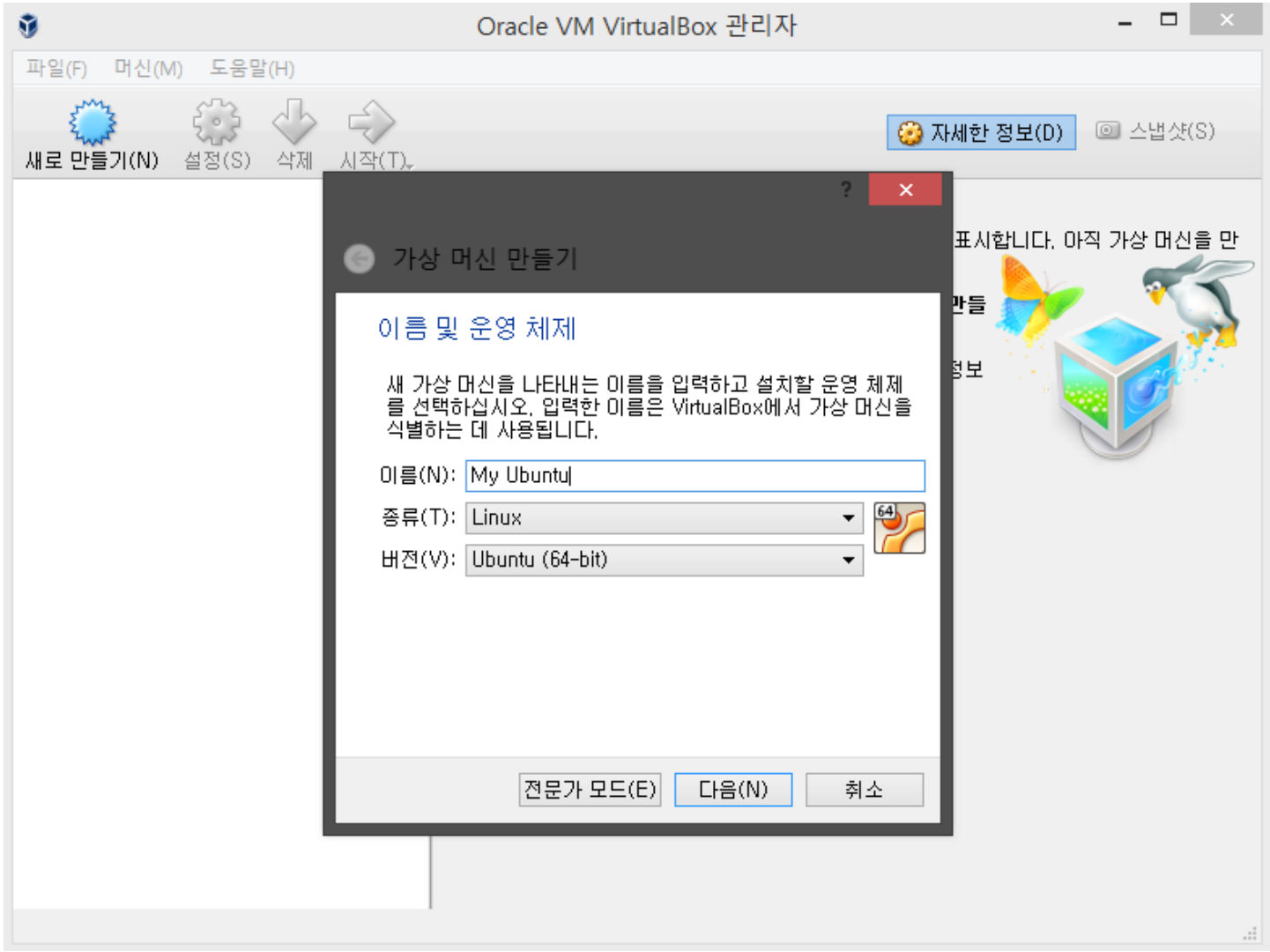
# Installing VirtualBox (2)



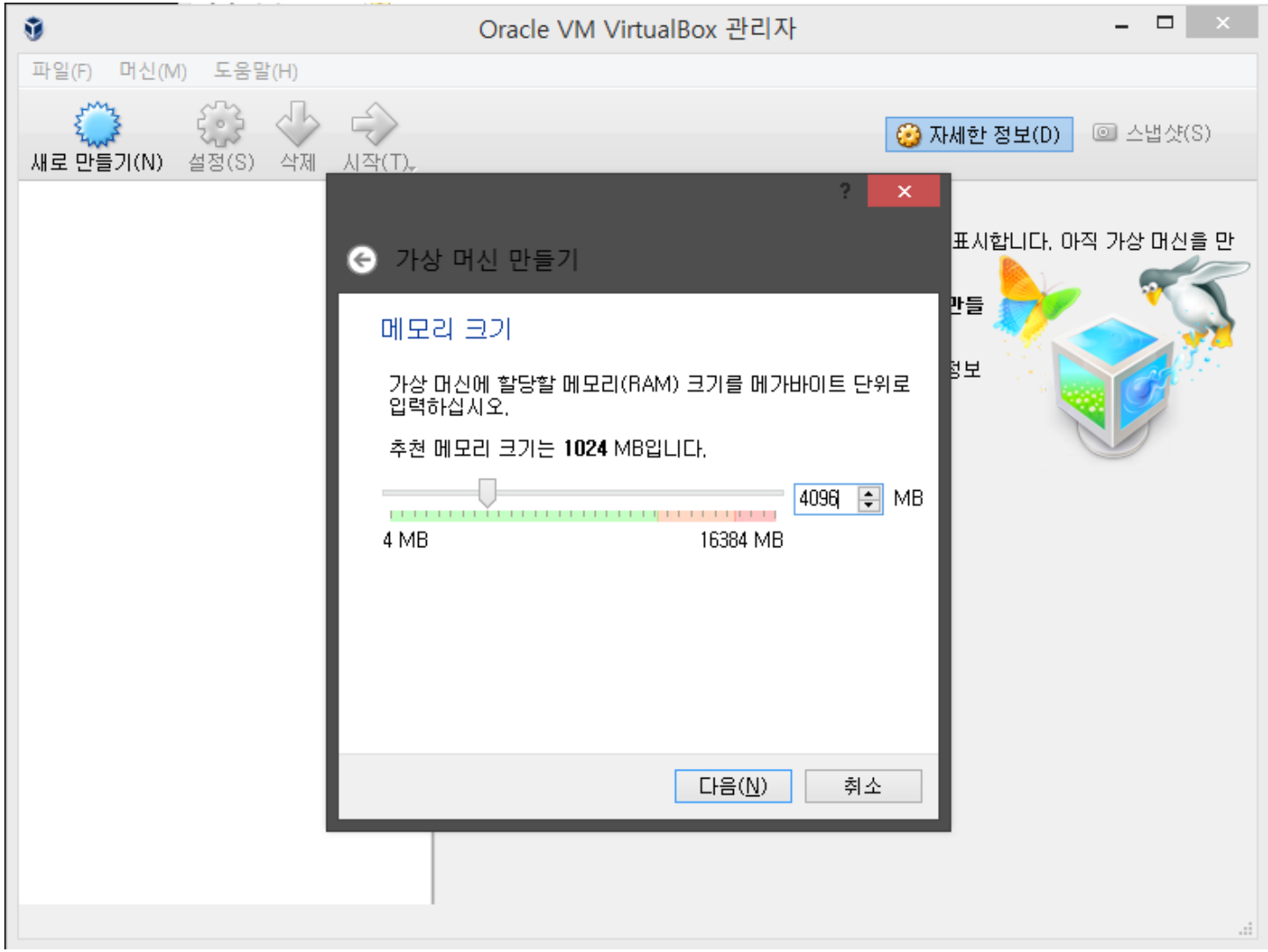
# Installing VirtualBox (3)



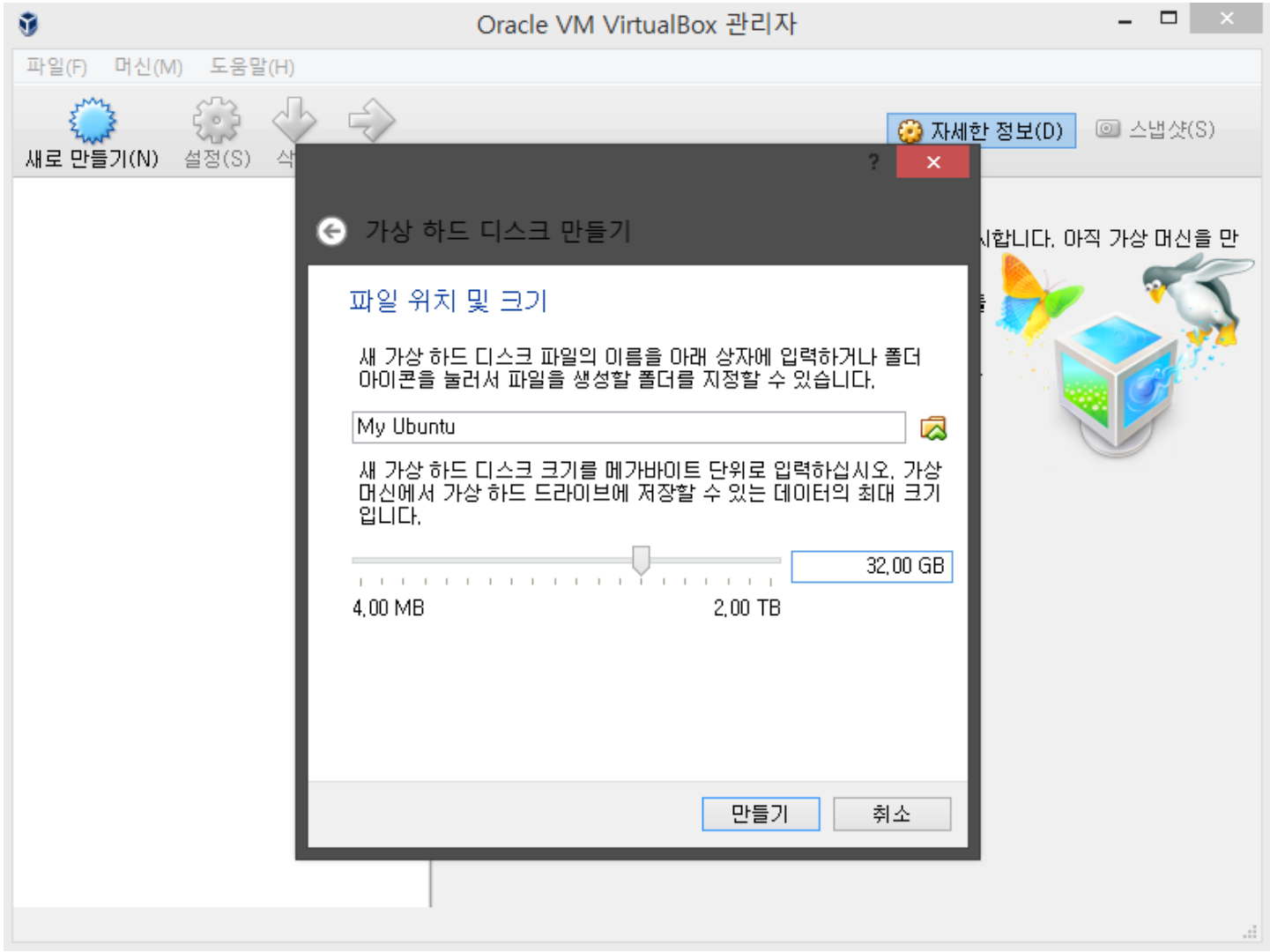
# Creating a VM (1)



# Creating a VM (2)

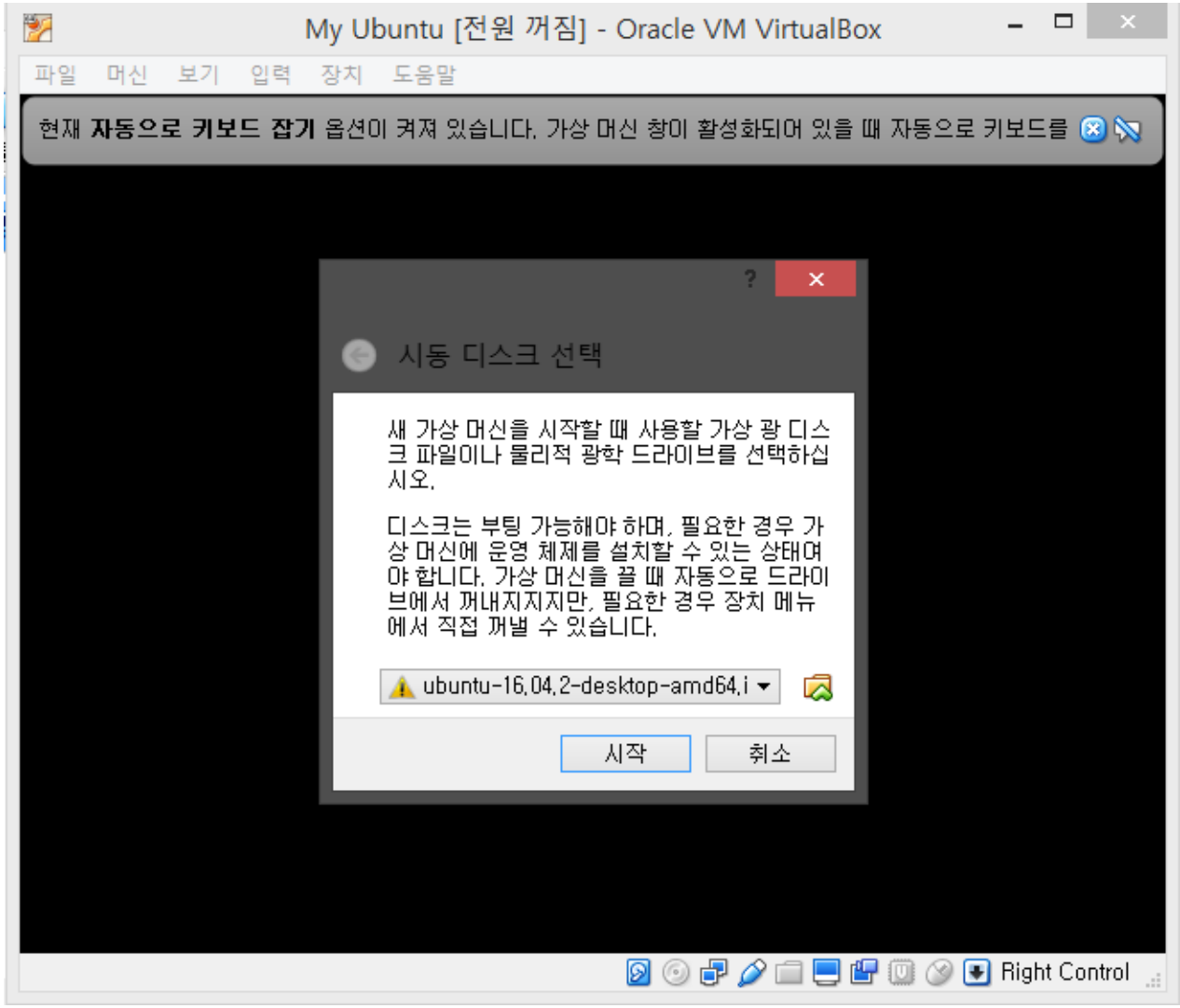


# Creating a VM (3)



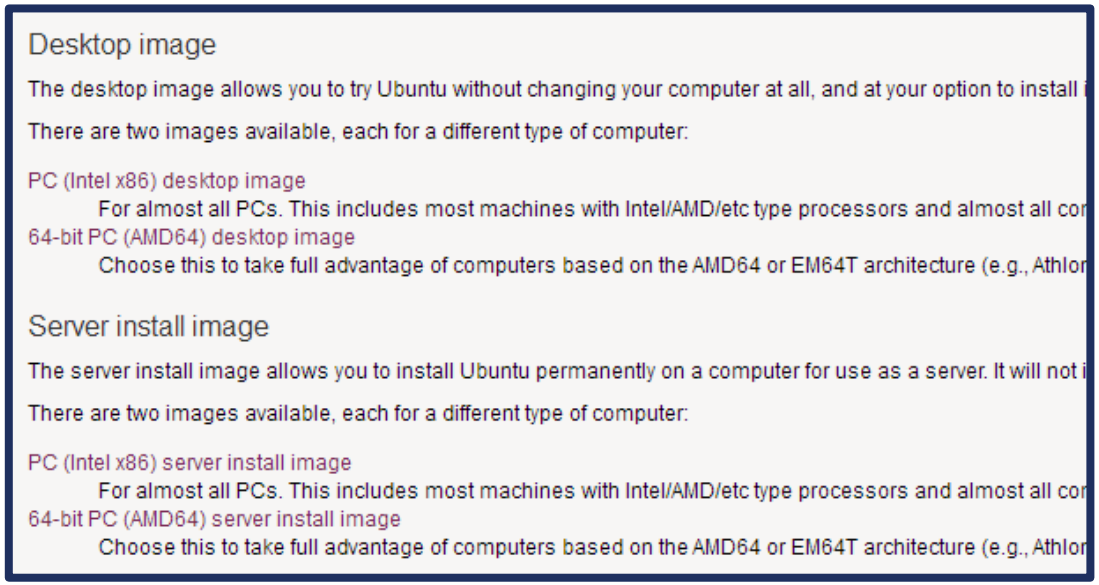
# Creating a VM (4)

# Creating a VM (5)

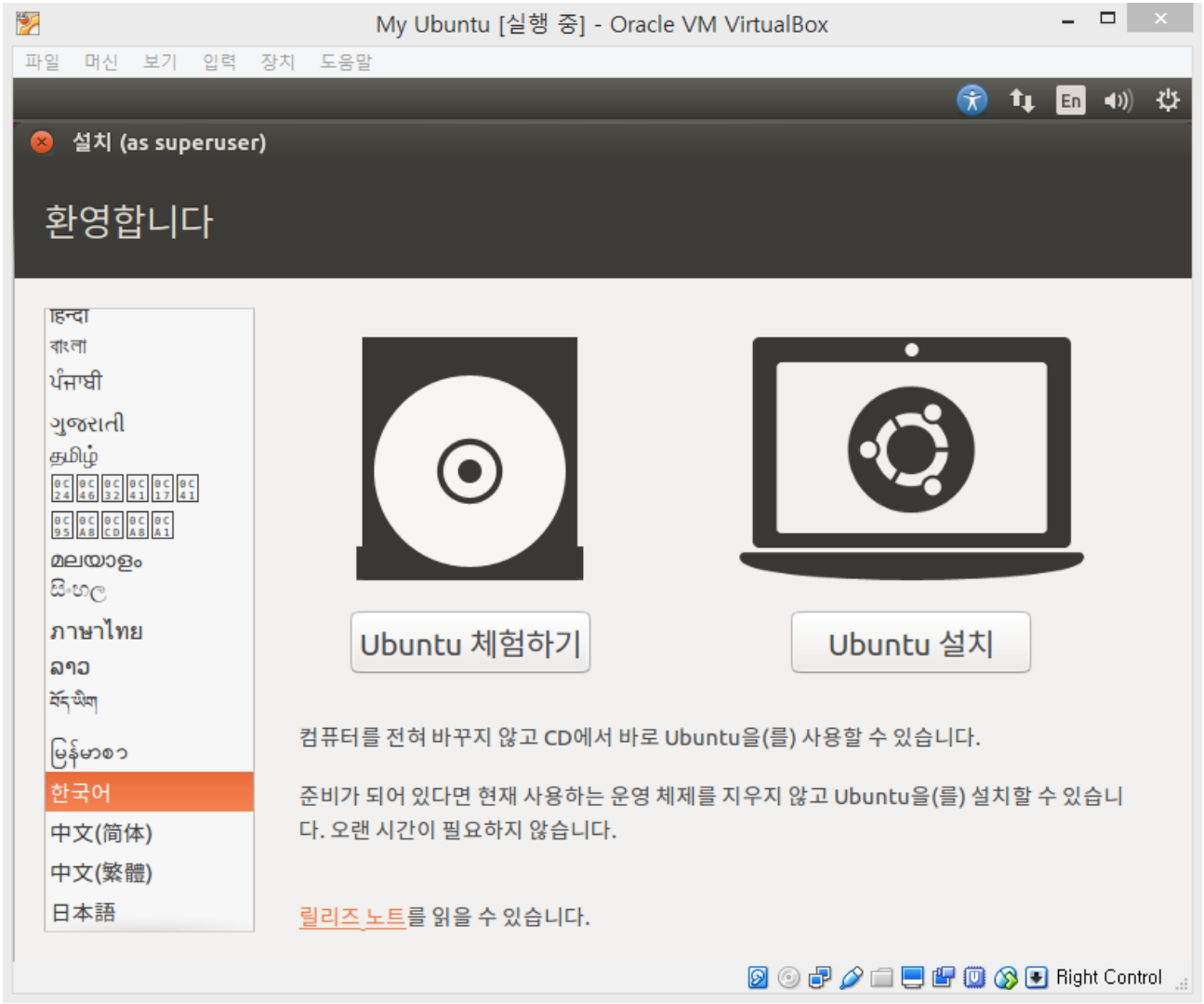


# Installing Ubuntu on the VM (1)

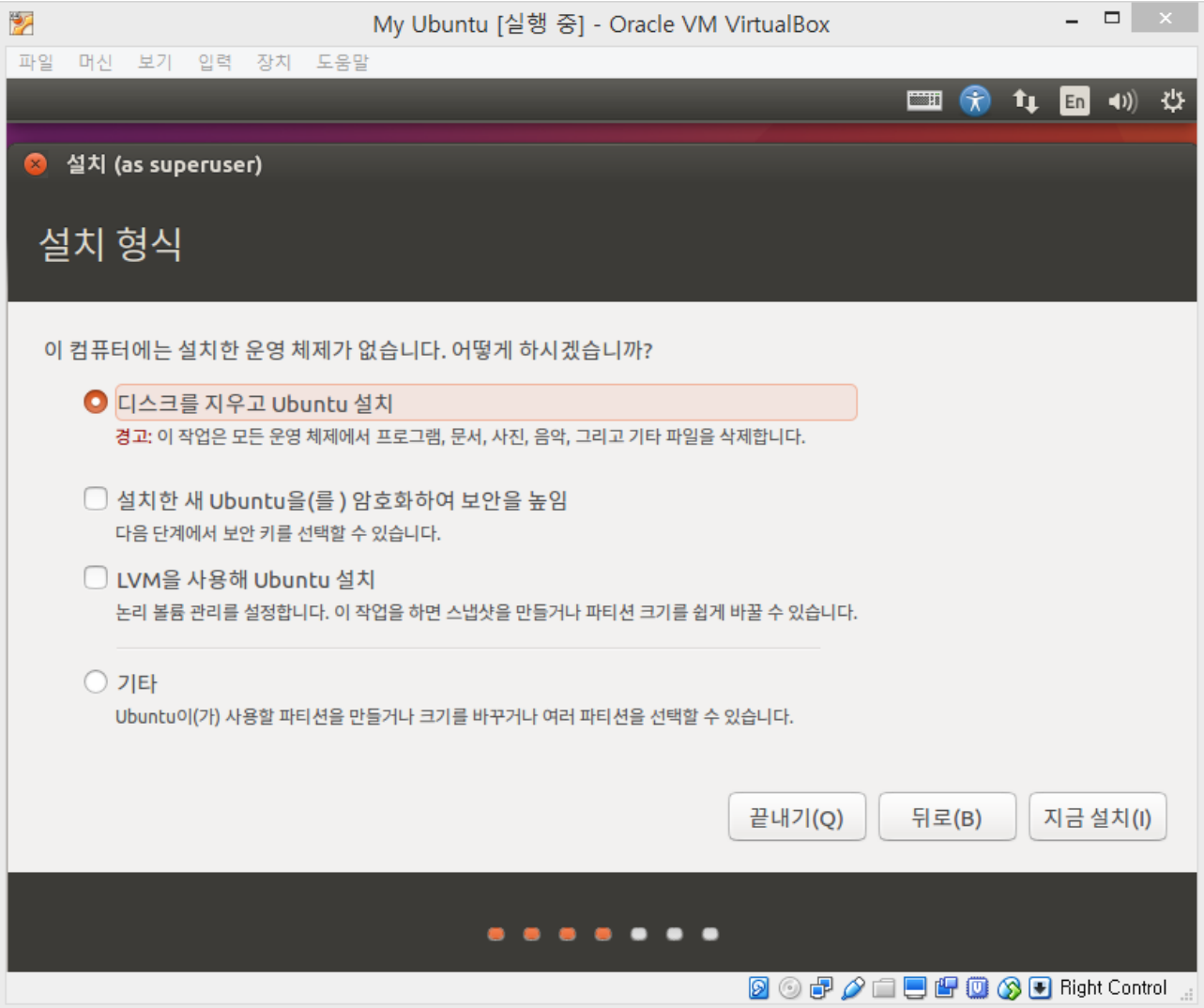
- Go to http://www.ubuntu.com/download/desktop or http://ftp.daumkakao.com/ubuntu-releases
- Download a desktop image



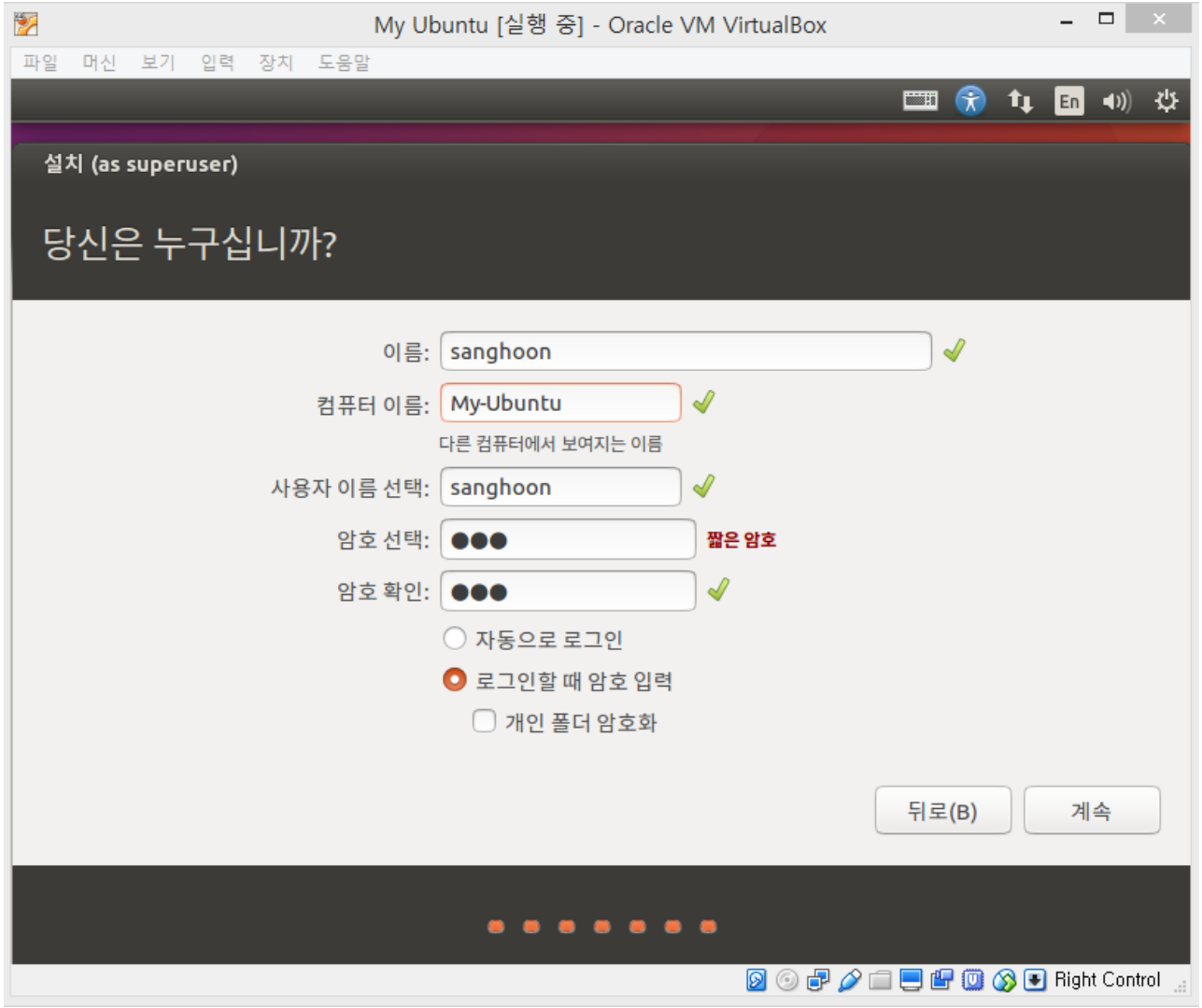
# Installing Ubuntu on the VM (3)



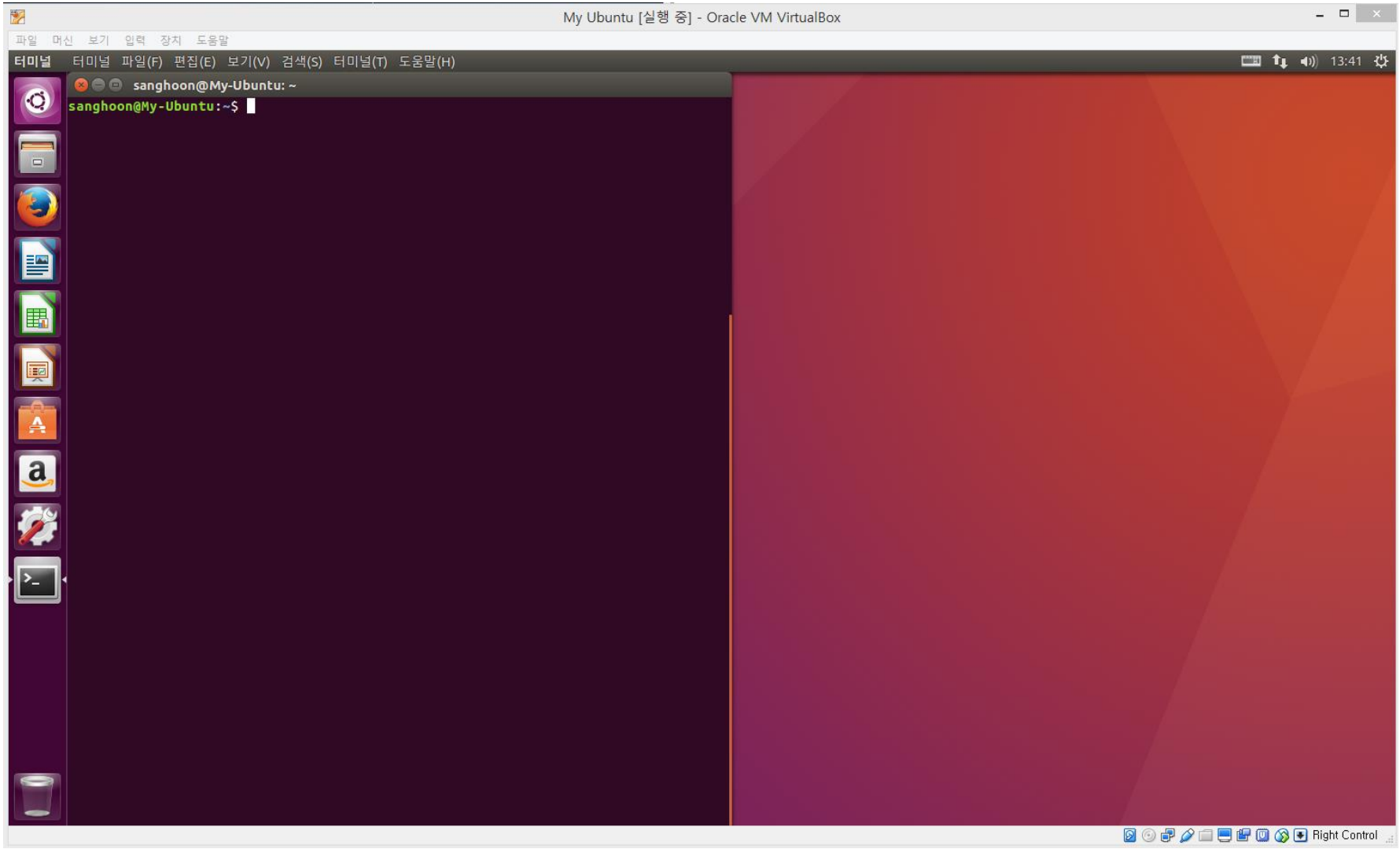
# Installing Ubuntu on the VM (4)



# Installing Ubuntu on the VM (5)



# Installing Ubuntu on the VM (6)



# Installing Ubuntu on the VM (7)

- Press *Ctrl + Alt + T* to launch a terminal (shell)

- Type the following commands:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

# Any Questions?