ROS TA Session

- Ubuntu basics
- ROS install
- ROS execution
- Create ROS node
- Concepts of compiler, g++, Makefile & cmake

2018 . 03. 19
Juhwan Seo

KAIST 2018
Capstone Design

Before Start

- The reason why you feel difficulties about Ubuntu is that you have no information.

The core purpose of this session!

- You will see lots of errors. To solve them, you should understand why the errors are appeared. To understand, you should know. To know, you should hear at least once

Best **Better** **Good enough!**

- Don’t believe me 100%. I could be wrong. Wrongly-delivered contents should be studied and corrected by yourself.

- If you installed Ubuntu in Korea, I recommend you to re-install in English for your future, not mine.

- This material is not kind. Sorry for that. Please concentrate on my explanation.

Contents

- Ubuntu
  -install 16.04 LTS
  -TTY(Tele-TYpewriter); brief history of computer
  -commands
    -file&folder management
    -compress/decompress
    -installation of applications; concept of mirror site
  -shell script
  -bashrc

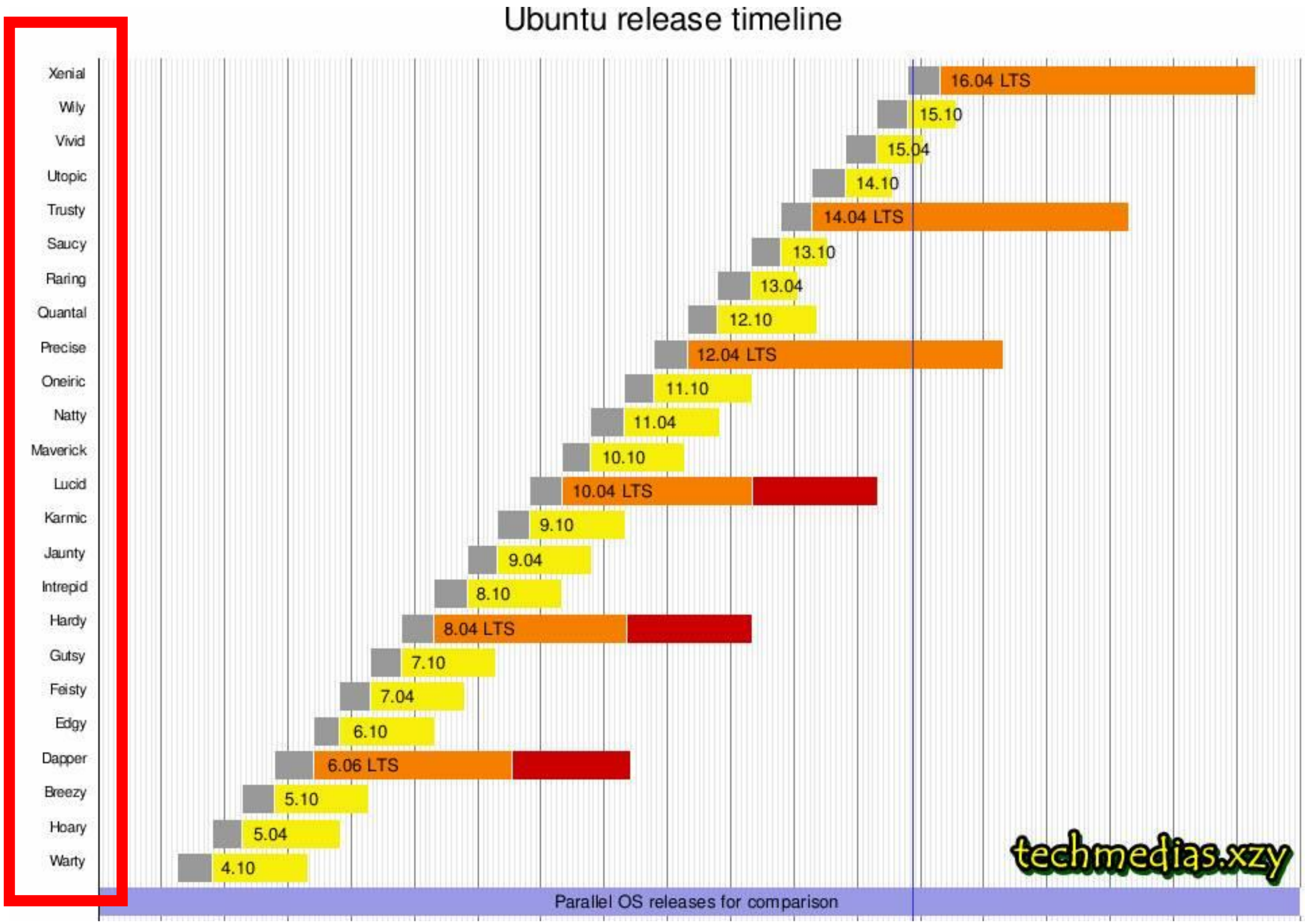
- Re-visit the questions on KakaoTalk group chat.

- ROS install
  -what is ROS? → will be skipped. See Dr.Pyo’s lecture notes.
  -install ROS kinetic

- Create ROS node (*similar to bonus assignment)
  -step by step programming
  -concepts of compiler, g++, Makefile&cmake

0. Ubuntu – About version

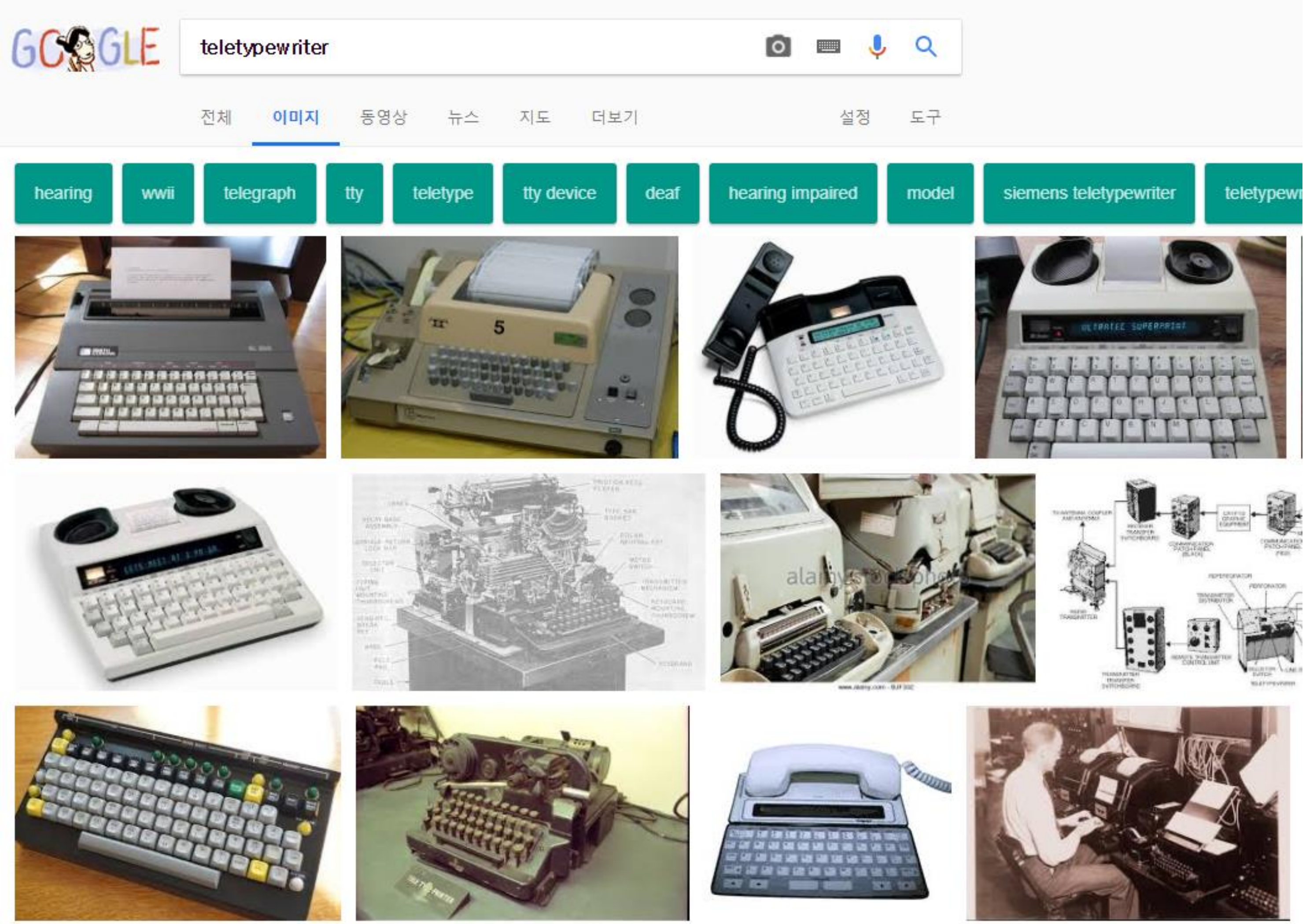
- There are many versions …
- Ubuntu yy.mm (LTS or nothing) yy: year, mm: month, LTS:Long-term support
  → Install LTS version!!



Codenames(it will be good to know the codename of your Ubuntu system)

0. Ubuntu – About TTY

- Let’s press ctrl+alt+F1~F7
- TTY: Tele-TYpewriter
- Concept of computer at early stage…

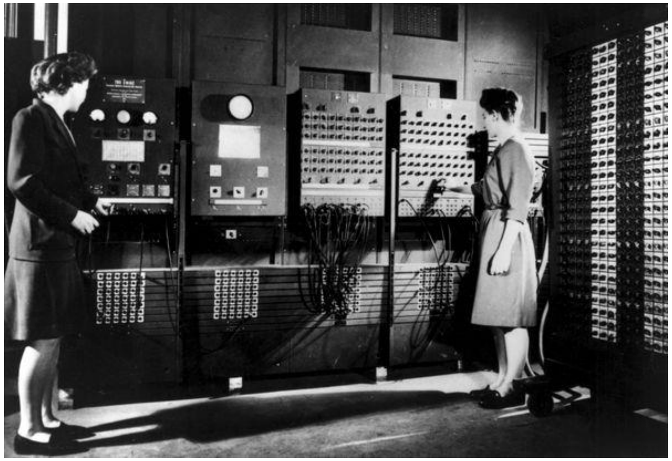


teletypewriter

0. Ubuntu – About TTY

- Let’s press ctrl+alt+F1~F7
- TTY: Tele-TYpewriter
- Concept of computer at early stage…

In the early stage….



*too big, cannot have it personally…
*connected via tele-typewriter

user1

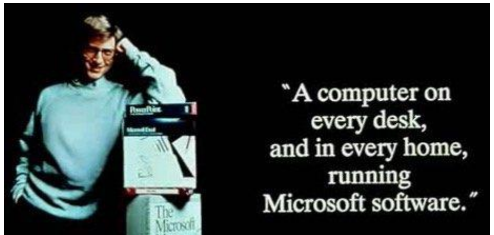
user2

user3

“There is no reason for any individual to have a computer in his home.”

Present

user1

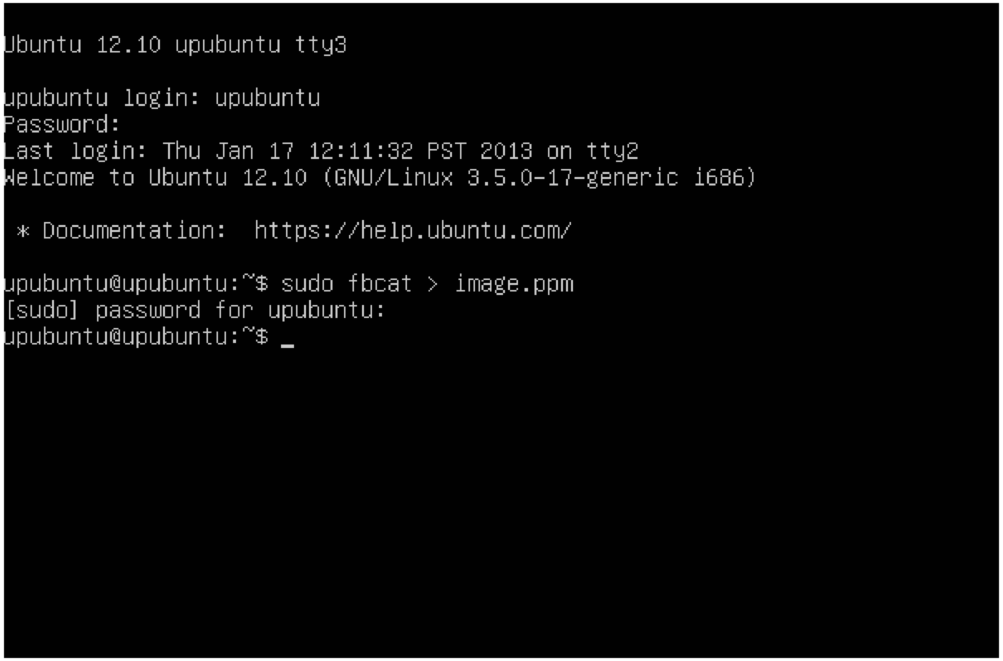


“A computer on every desk, and in every home, running Microsoft software.”

0. Ubuntu – About TTY

- Let’s press ctrl+alt+F1~F7
- TTY: Tele-TYpewriter
- Concept of computer at early stage…

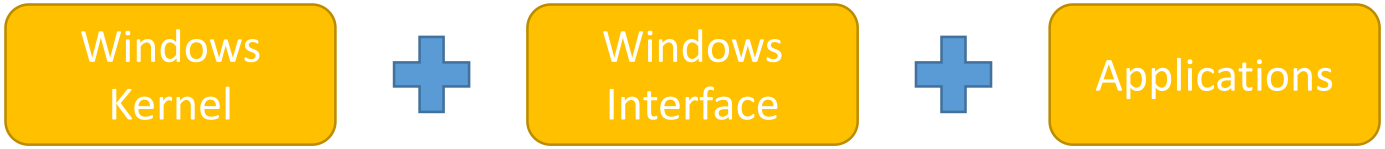
- Basically, there is no GUI. Everything is done in text-based console.
- Almost the same with terminal(next slide)
- There is no need to do something with this.
  However, sometimes you are needed to do something with TTY.
  Ex) installation of Graphic Driver
  (also, CUDA which will be required when installing Tensorflow in fall semester), …



```
Ubuntu 12.10 upubuntu tty3

upubuntu login: upubuntu
Password:
Last login: Thu Jan 17 12:11:32 PST 2013 on tty2
Welcome to Ubuntu 12.10 (GNU/Linux 3.5.0-17-generic i686)

 * Documentation:  https://help.ubuntu.com/

upubuntu@upubuntu:~$ sudo fbcat > image.ppm
[sudo] password for upubuntu:
upubuntu@upubuntu:~$ _
```

0. Ubuntu – About Linux & Ubuntu

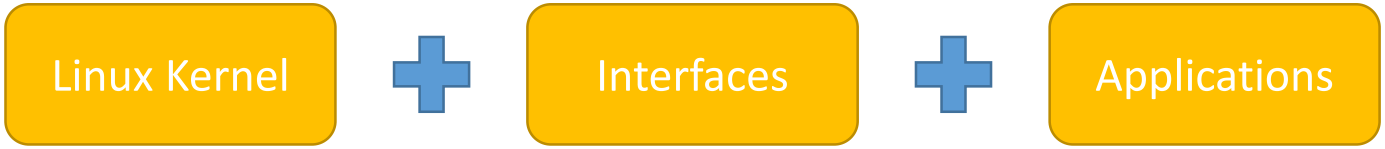
- OS(Operating System) consists of mainly three parts(kernels, interface and applications)
- In case of Windows,



Windows Kernel + Windows Interface + Applications

- All three parts are provided by Microsoft.

- In case of Linux,



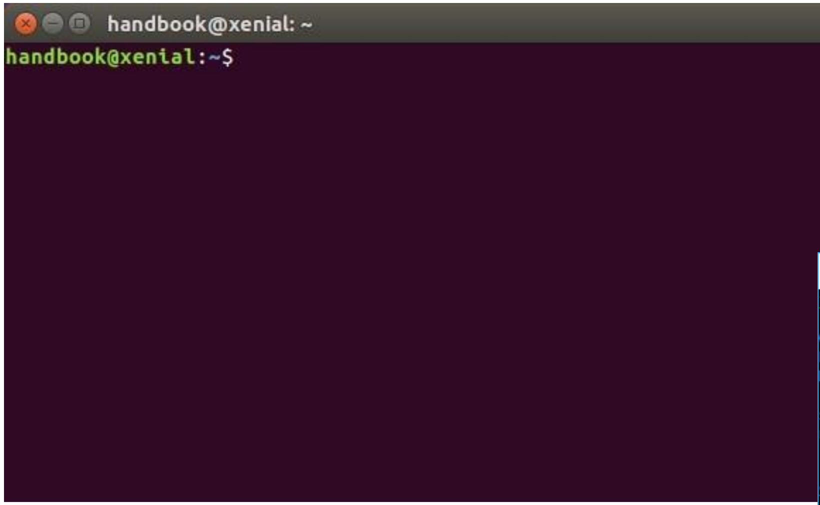
Linux Kernel + Interfaces + Applications

- Linux Kernel is provided by Linus Torvalds(Programmer)
- Interfaces are provided by many providers. (Ubuntu, CentOS, Linux Mint, Raspbian, ….)

- In conclusion, Ubuntu is one of Linux-based OS.
- For details, http://moordev.tistory.com/222

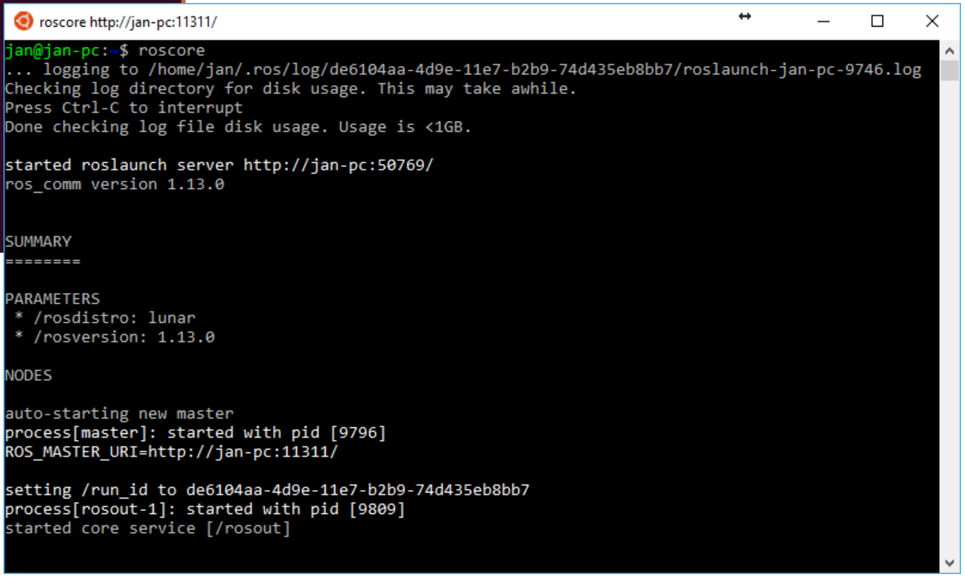
0. Ubuntu – About Terminal

- The thing you should master as soon as possible …
- Shortcuts) ctrl+alt+t : new terminal on new window
  ctrl+shift+t : new terminal on new tab
  ctrl+page up/down : tab change
- If a terminal is idle, you can do something with it.
  If a terminal is not idle, that means it is occupied by some process.



```
handbook@xenial:~$
```

Idle terminal

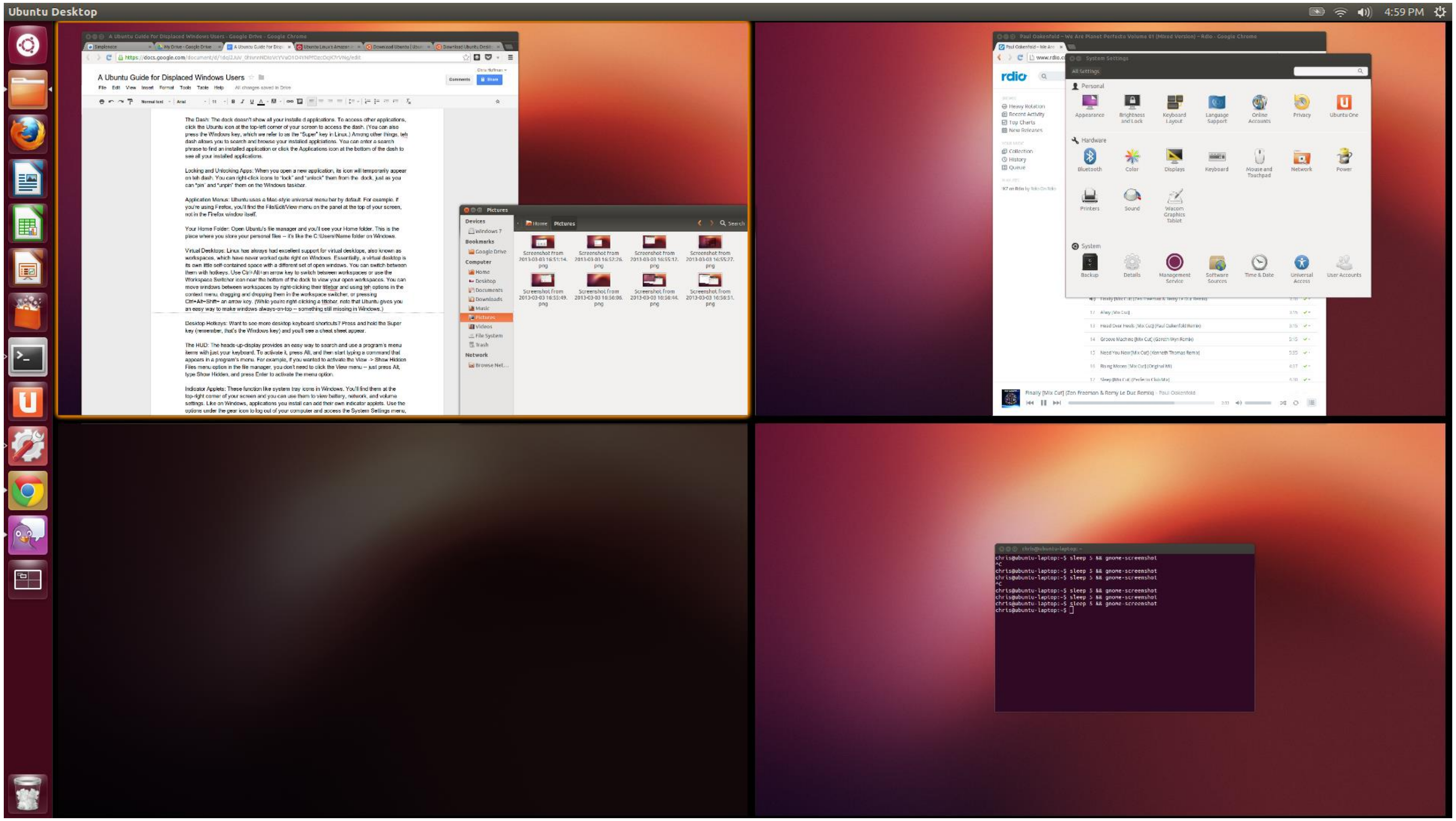


```
jan@jan-pc:~$ roscore
... logging to /home/jan/.ros/log/de6104aa-4d9e-11e7-b2b9-74d435eb8bb7/roslaunch-jan-pc-9746.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://jan-pc:50769/
ros_comm version 1.13.0


SUMMARY
========

PARAMETERS
 * /rosdistro: lunar
 * /rosversion: 1.13.0

NODES

auto-starting new master
process[master]: started with pid [9796]
ROS_MASTER_URI=http://jan-pc:11311/

setting /run_id to de6104aa-4d9e-11e7-b2b9-74d435eb8bb7
process[rosout-1]: started with pid [9809]
started core service [/rosout]
```

Occupied terminal

0. Ubuntu – About Workspace

- For a year with capstone design, you will utilize lots of terminals.
- Screen space might not be enough.
- Enable workspace from 'System Settings' → 'Appearance' → 'Behavior' → check 'Enable workspaces'
- Shortcuts) ctrl+alt+direction : change workspace
  ctrl+alt+shift+direction : change workspace & move current window



A Ubuntu Guide for Displaced Windows Users

The Dash: The dock doesn't show all your installe d applications. To access other applications, click the Ubuntu icon at the top-left corner of your screen to access the dash. (You can also press the Windows key, which we refer to as the "Super" key in Linux.) Among other things, teh dash allows you to search and browse your installed applciations. You can enter a search phrase to find an installed application or click the Applications icon at the bottom of the dash to see all your installed applications.

Locking and Unlocking Apps: When you open a new application, its icon will temporarily appear on teh dash. You can right-click icons to "lock" and "unlock" them from the dock, just as you can "pin" and "unpin" them on the Windows taskbar.

Application Menus: Ubuntu uses a Mac-style universal menu bar by default. For example, if you're using Firefox, you'll find the File/Edit/View menu on the panel at the top of your screen, not in the Firefox window itself.

Your Home Folder: Open Ubuntu's file manager and you'll see your Home folder. This is the place where you store your personal files -- it's like the C:\Users\Name folder on Windows.

Virtual Desktops: Linux has always had excellent support for virtual desktops, also known as workspaces, which have never worked quite right on Windows. Essentially, a virtual desktop is its own little self-contained space with a different set of open windows. You can switch between them with hotkeys. Use Ctrl+Alt+an arrow key to switch between workspaces or use the Workspace Switcher icon near the bottom of the dock to view your open workspaces. You can move windows between workspaces by right-clicking their titlebar and using teh options in the context menu, dragging and dropping them in the workspace switcher, or pressing Ctrl+Alt+Shift+ an arrow key. (While you're right-clicking a titlebar, note that Ubuntu gives you an easy way to make windows always-on-top -- something still missing in Windows.)

Desktop Hotkeys: Want to see more desktop keyboard shortcuts? Press and hold the Super key (remember, that's the Windows key) and you'll see a cheat sheet appear.

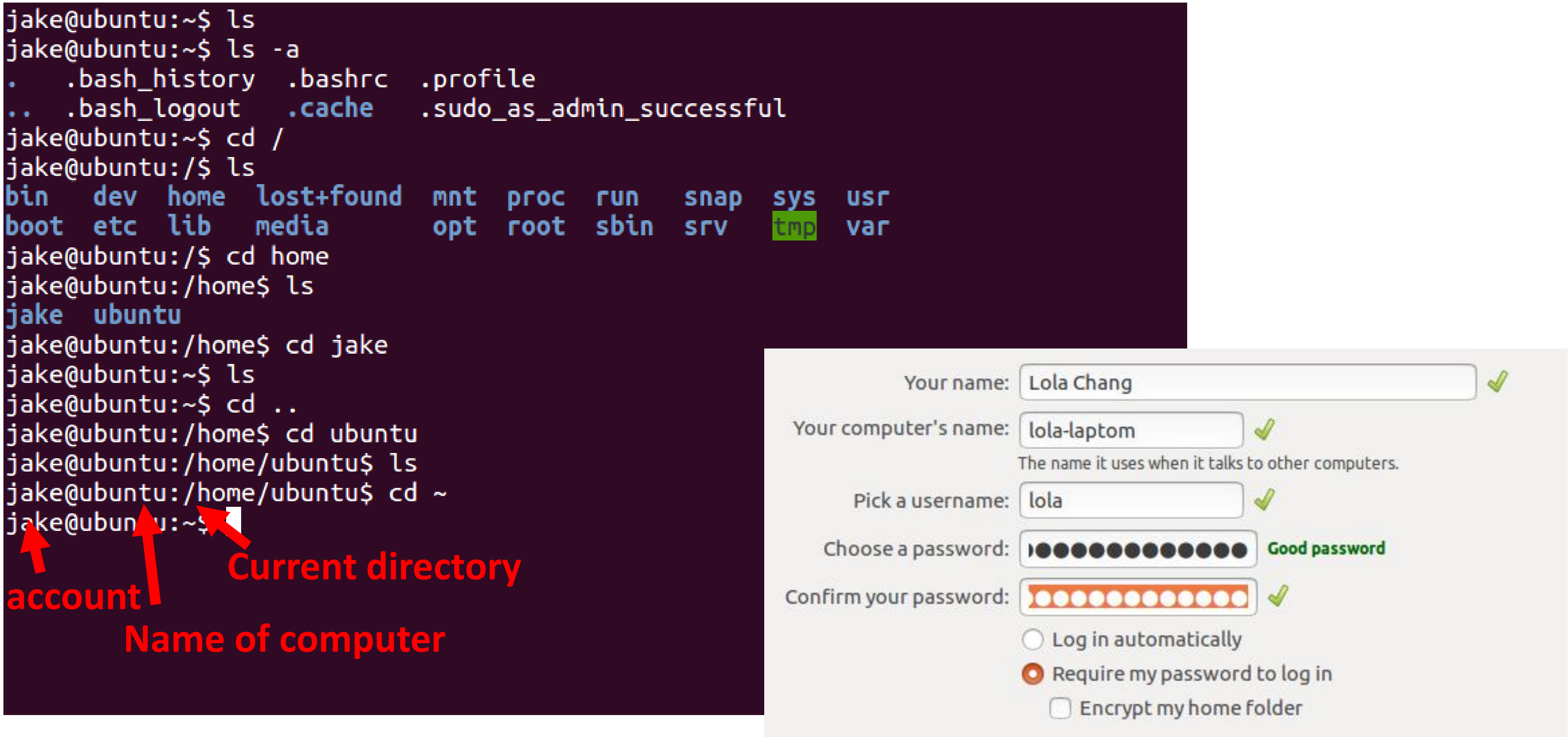
The HUD: The heads-up-display provides an easy way to search and use a program's menu items with just your keyboard. To activate it, press Alt, and then start typing a command that appears in a program's menu. For example, if you wanted to activate the View -> Show Hidden Files menu option in the file manager, you don't need to click the View menu -- just press Alt, type Show Hidden, and press Enter to activate the menu option.

Indicator Applets: These function like system tray icons in Windows. You'll find them at the top-right corner of your screen and you can use them to view battery, network, and volume settings. Like on Windows, applications you install can add their own indicator applets. Use the options under the gear icon to log out of your computer and access the System Settings menu,

```
chris@ubuntu-laptop:~$ sleep 5 && gnome-screenshot
^C
chris@ubuntu-laptop:~$ sleep 5 && gnome-screenshot
chris@ubuntu-laptop:~$ sleep 5 && gnome-screenshot
^C
chris@ubuntu-laptop:~$ sleep 5 && gnome-screenshot
chris@ubuntu-laptop:~$ sleep 5 && gnome-screenshot
chris@ubuntu-laptop:~$ sleep 5 && gnome-screenshot
chris@ubuntu-laptop:~$ 
```

0. Ubuntu – About File&Folder management

- GUI is easy… but you still need to know this
- Shorcuts) **cd** : change directory, **ls**: list directory contents, **mv**: file move
  rm : remove, **mkdir**: make directory, **rmdir**: remove directory, **cp**: copy
  there are more and each command has options.
  Study by your self! (if you just type the command in a terminal, help will appear)
- ~: home folder, /: the most basic part of Ubuntu system
- sudo apt-get install tree & tree -d

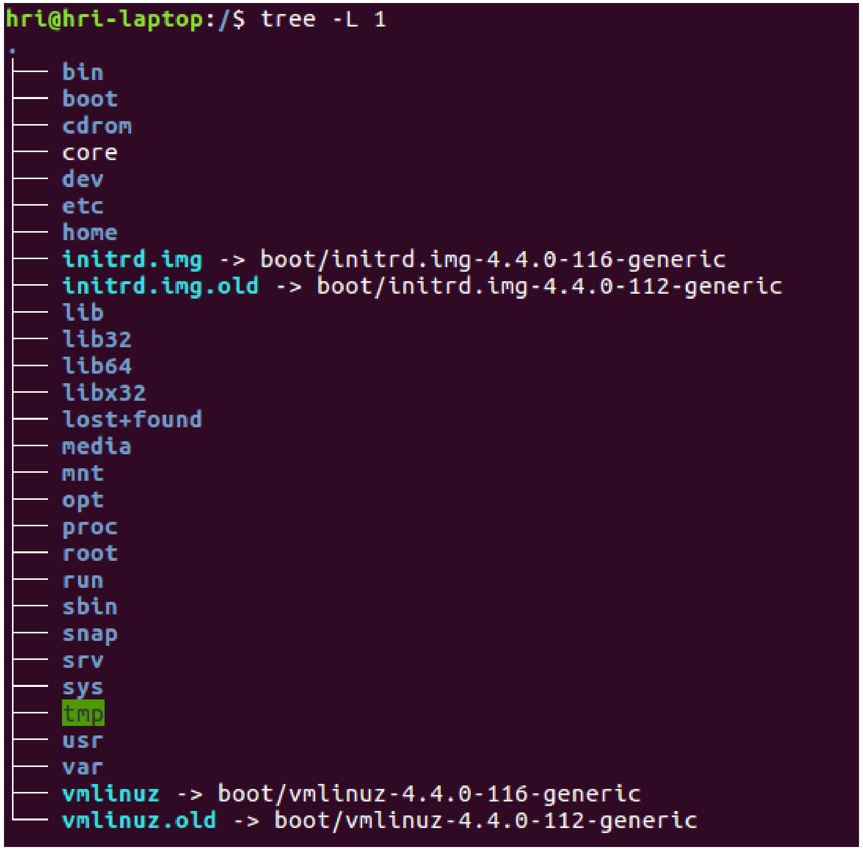


```
jake@ubuntu:~$ ls
jake@ubuntu:~$ ls -a
.   .bash_history  .bashrc   .profile
..  .bash_logout   .cache    .sudo_as_admin_successful
jake@ubuntu:~$ cd /
jake@ubuntu:/$ ls
bin   dev  home  lost+found  mnt  proc  run   snap  sys  usr
boot  etc  lib   media       opt  root  sbin  srv   tmp  var
jake@ubuntu:/$ cd home
jake@ubuntu:/home$ ls
jake  ubuntu
jake@ubuntu:/home$ cd jake
jake@ubuntu:~$ ls
jake@ubuntu:~$ cd ..
jake@ubuntu:/home$ cd ubuntu
jake@ubuntu:/home/ubuntu$ ls
jake@ubuntu:/home/ubuntu$ cd ~
jake@ubuntu:~$
```

Your name:	Lola Chang ✓
Your computer's name:	lola-laptom ✓ The name it uses when it talks to other computers.
Pick a username:	lola ✓
Choose a password:	●●●●●●●●●●●●● Good password
Confirm your password:	●●●●●●●●●●●●● ✓
	○ Log in automatically
	● Require my password to log in
	□ Encrypt my home folder

0. Ubuntu – About File&Folder management

- GUI is easy… but you still need to know this
- Shorcuts) **cd** : change directory, **ls**: list directory contents, **mv**: file move
  rm : remove, **mkdir**: make directory, **rmdir**: remove directory, **cp**: copy
  there are more and each command has options.
  Study by your self! (if you just type the command in a terminal, help will appear)
- ~: home folder, /: the most basic part of Ubuntu system
- sudo apt-get install tree & tree -d



```
hri@hri-laptop:/$ tree -L 1
.
├── bin
├── boot
├── cdrom
├── core
├── dev
├── etc
├── home
├── initrd.img -> boot/initrd.img-4.4.0-116-generic
├── initrd.img.old -> boot/initrd.img-4.4.0-112-generic
├── lib
├── lib32
├── lib64
├── libx32
├── lost+found
├── media
├── mnt
├── opt
├── proc
├── root
├── run
├── sbin
├── snap
├── srv
├── sys
├── tmp
├── usr
├── var
├── vmlinuz -> boot/vmlinuz-4.4.0-116-generic
└── vmlinuz.old -> boot/vmlinuz-4.4.0-112-generic
```

-bin(binary)
-boot(boot loader related)
-etc(Literally, etc…)
-dev(device)
-home(home folder, every users' folder will be here)
-lib, lib32, lib64, libx32(libraries for 32bit &64bit system)
-media(sdd, hdd, usb, …)
-opt: softwares and add-on packages that are not part of the default installation
…

Study more by yourself!

0. Ubuntu – compress & decompress

- For ‘zip files’
  zip (name of zip file) (path of the files you want to compress)
  ex) zip capstonedesign.zip ~/capstone_design

  If you want to compress a folder with following folders, use ‘-r’ option
  ex) zip –r capstonedesign.zip ~/capstone_design

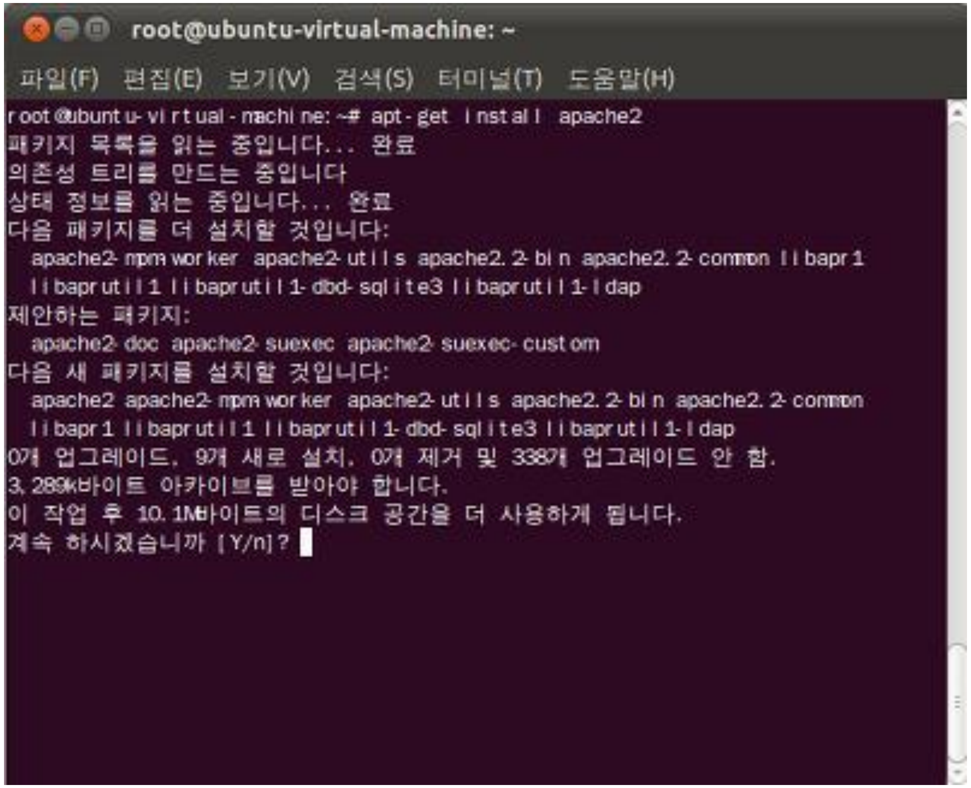
- For ‘tar.~~ files’
  tar –xvf (path of the compressed files)
  ex) tar –xvf opencv.tar.gz

0. Ubuntu – About Package Installation

- Open-source based OS → Applications are free!
- Almost every programs and libraries can be installed via internet.
- The process of installation in Ubuntu is like below.

Install ROS!

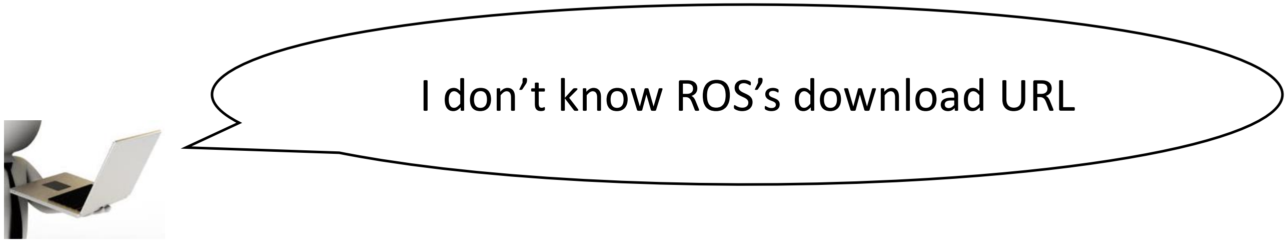
I know ROS's download URL

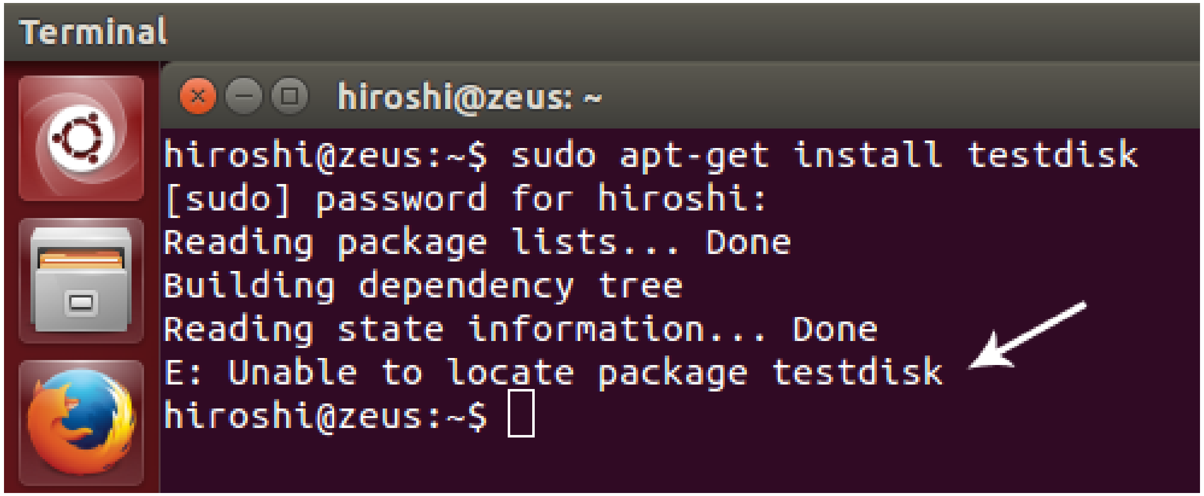

root@ubuntu-virtual-machine: ~
파일(F) 편집(E) 보기(V) 검색(S) 터미널(T) 도움말(H)

```
root@ubuntu-virtual-machine:~# apt-get install apache2
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
다음 패키지를 더 설치할 것입니다:
  apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
제안하는 패키지:
  apache2-doc apache2-suexec apache2-suexec-custom
다음 새 패키지를 설치할 것입니다:
  apache2 apache2-mpm-worker apache2-utils apache2.2-bin apache2.2-common
  libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
0개 업그레이드, 9개 새로 설치, 0개 제거 및 338개 업그레이드 안 함.
3,289k바이트 아카이브를 받아야 합니다.
이 작업 후 10.1M바이트의 디스크 공간을 더 사용하게 됩니다.
계속 하시겠습니까 [Y/n]? 
```

0. Ubuntu – About Package Installation

- Open-source based OS → Applications are free!
- Almost every programs and libraries can be installed via internet.
- The process of installation in Ubuntu is like below.



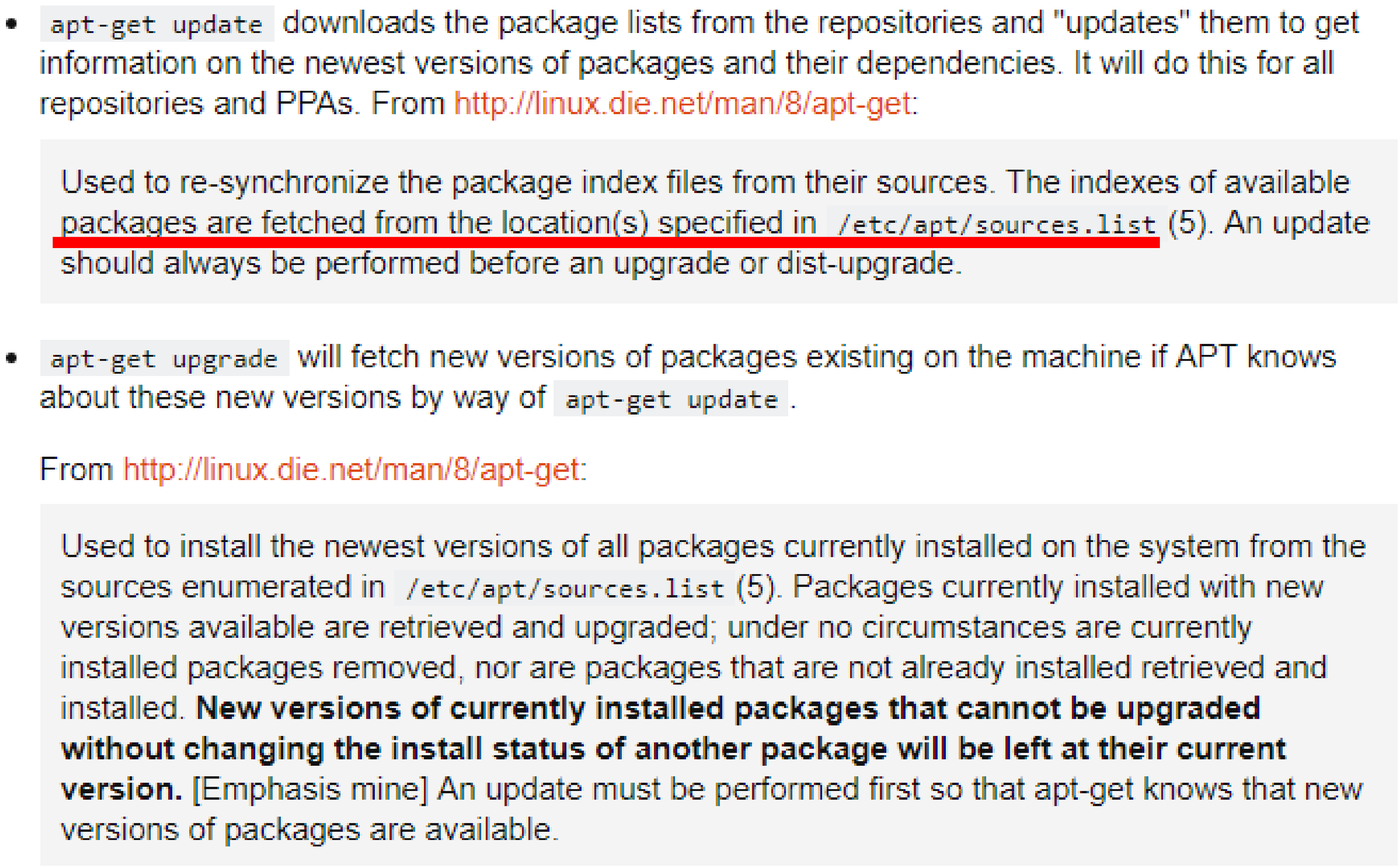


```
hiroshi@zeus:~$ sudo apt-get install testdisk
[sudo] password for hiroshi:
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package testdisk
hiroshi@zeus:~$ 
```

Then, how can we make a computer know packages’ download URL?

0. Ubuntu – About Package Installation

- Open-source based OS → Applications are free!
- Almost every programs and libraries are installed via internet.
- The process of installation in Ubuntu is like below.



- `apt-get update` downloads the package lists from the repositories and "updates" them to get information on the newest versions of packages and their dependencies. It will do this for all repositories and PPAs. From http://linux.die.net/man/8/apt-get:

  > Used to re-synchronize the package index files from their sources. The indexes of available packages are fetched from the location(s) specified in `/etc/apt/sources.list` (5). An update should always be performed before an upgrade or dist-upgrade.

- `apt-get upgrade` will fetch new versions of packages existing on the machine if APT knows about these new versions by way of `apt-get update`.

  From http://linux.die.net/man/8/apt-get:

  > Used to install the newest versions of all packages currently installed on the system from the sources enumerated in `/etc/apt/sources.list` (5). Packages currently installed with new versions available are retrieved and upgraded; under no circumstances are currently installed packages removed, nor are packages that are not already installed retrieved and installed. **New versions of currently installed packages that cannot be upgraded without changing the install status of another package will be left at their current version.** [Emphasis mine] An update must be performed first so that apt-get knows that new versions of packages are available.

0. Ubuntu – About Package Installation

- What if /etc/apt/sources.list is broken?

  ➔ sudo apt-get update cannot output the packages' list.
  ➔ you can't install packages not listed in your computer.

- I edited /etc/apt/sources.list. However, still it is not working
  ➔ just editing /etc/apt/sources.list doesn't update the package list. When you modify /etc/apt/sources.list file, you should 'sudo apt-get update' to apply it

- If you want to check the list is refreshed, utilized 'tab'
  sudo apt-get install ros-kinetic-(push tab here!)
  then the list available will be appeared.

- I modified /etc/apt/sources.list & I can find the package I want to install when I input 'tab'
  ➔ There are two possibility
  1. Your internet status(if you are not connected, you cannot download)
  2. Repository server might be broken(it is completely not your fault….)

0. Ubuntu – About Package Installation

- There are many repositories(they are also called mirror site)
- Why there are many repositories??

- Ubuntu distributors update their own repository(it is the starting point of update because they are provider!)

  ➔ If you are geographically close to the Ubuntu distributors server, then you can update and download packages rapidly.

  ➔ However, in most of users are not close the their server.

  ➔ To resolve it, many mirror site(copy&pasted server) are managed.

  For example,

  kr.archive.ubuntu.com by KAIST SPARCS

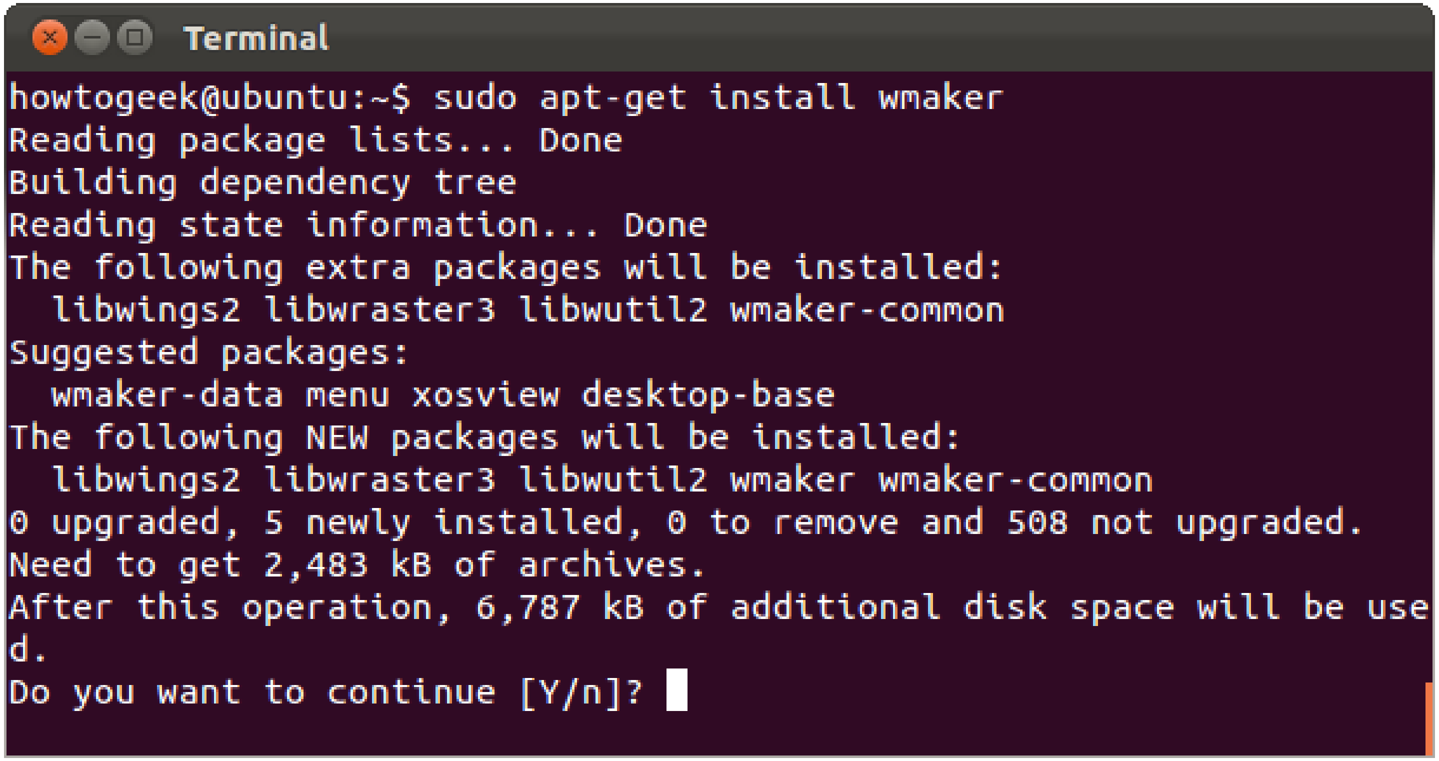
- If this mirror site is in trouble and your ubuntu is set to be connected to this mirror site, then apt-get install will not work.

  ➔ change repository(or mirror site)

You already saw this in KakaoTalk Group Chat….

0. Ubuntu – About Package Installation

- Open-source based OS → Applications are free!
- Free applications are in repository.
- Command ‘apt-get’ access the repository and do something related with applications
  apt-get install (package name)
  apt-get remove (package name)
  apt-get update : repository update
  apt-get upgrade : upgrade packages
  … (there are more)



```
howtogeek@ubuntu:~$ sudo apt-get install wmaker
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libwings2 libwraster3 libwutil2 wmaker-common
Suggested packages:
  wmaker-data menu xosview desktop-base
The following NEW packages will be installed:
  libwings2 libwraster3 libwutil2 wmaker wmaker-common
0 upgraded, 5 newly installed, 0 to remove and 508 not upgraded.
Need to get 2,483 kB of archives.
After this operation, 6,787 kB of additional disk space will be use
d.
Do you want to continue [Y/n]?
```

0. Ubuntu – About Package Installation

- Be careful when execute sudo apt-get upgrade

- cd /usr/lib & ls
- You can see libraries with version number.

- sudo apt-get upgrade those libraries to the latest version

Q: The latest, isn't it good?
A: not actually…

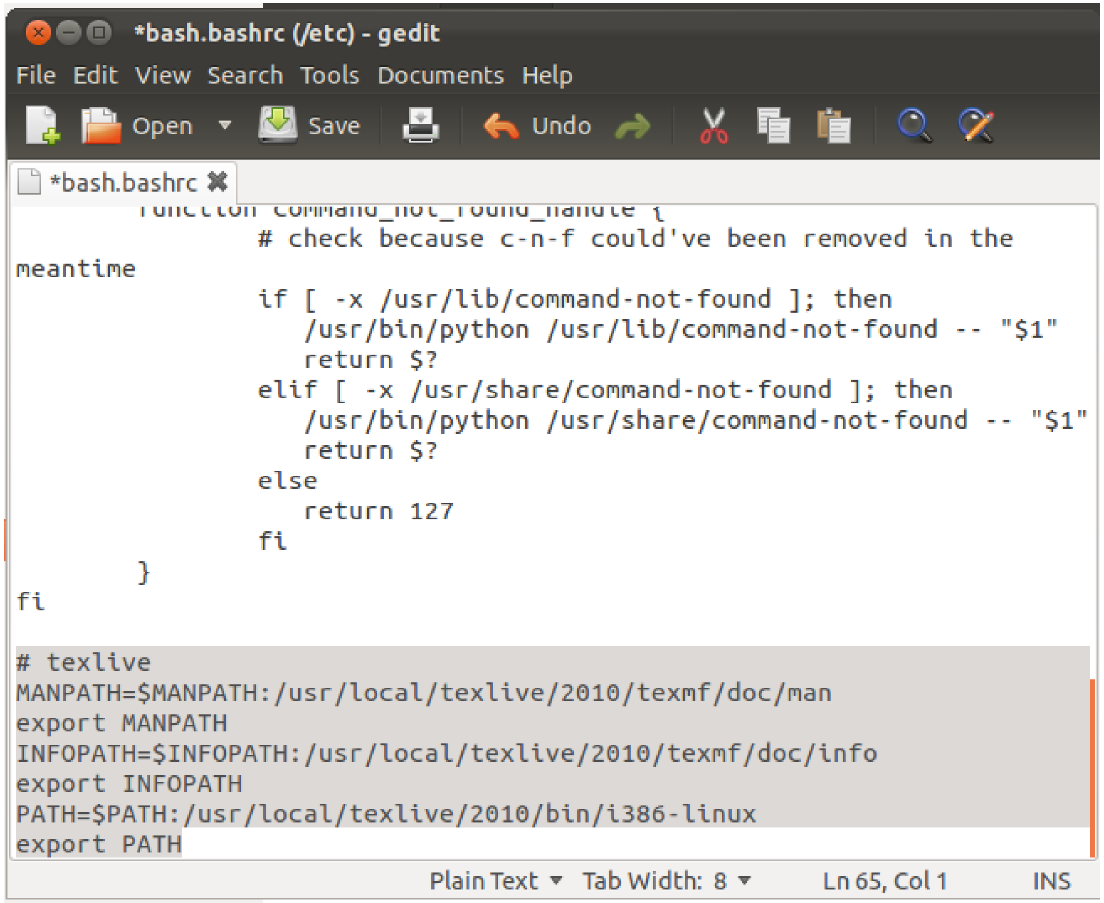
- When you make a program with a library,
  your code & library files will be linked. In addition, executable file will find the libraries used in linking process.

  Let's say there is a program with opencv2.4. The program has no problem in execution.
  → someday, you entered 'sudo apt-get upgrade'
  → opencv is upgraded to version 2.5
  → you run the program again, but it will say like that,

  Error: No such files or libraries "libopencv2.4"

0. Ubuntu – About bashrc

- a script that is executed whenever a new terminal session is started
- The file is in your home folder
- You can’t find the ‘.bashrc’ file by inputting the command ‘ls’ because it is hidden
- Do it yourself) gedit .bashrc   or nano .bashrc
- .(filename) means the file is in hidden mode



```
        function command_not_found_handle {
                # check because c-n-f could've been removed in the
meantime
                if [ -x /usr/lib/command-not-found ]; then
                   /usr/bin/python /usr/lib/command-not-found -- "$1"
                   return $?
                elif [ -x /usr/share/command-not-found ]; then
                   /usr/bin/python /usr/share/command-not-found -- "$1"
                   return $?
                else
                   return 127
                fi
        }
fi

# texlive
MANPATH=$MANPATH:/usr/local/texlive/2010/texmf/doc/man
export MANPATH
INFOPATH=$INFOPATH:/usr/local/texlive/2010/texmf/doc/info
export INFOPATH
PATH=$PATH:/usr/local/texlive/2010/bin/i386-linux
export PATH
```

bashrc is frequently used
to set environment variables.
(very important!!!!!)

0. Ubuntu – About ssh(Secure shell)

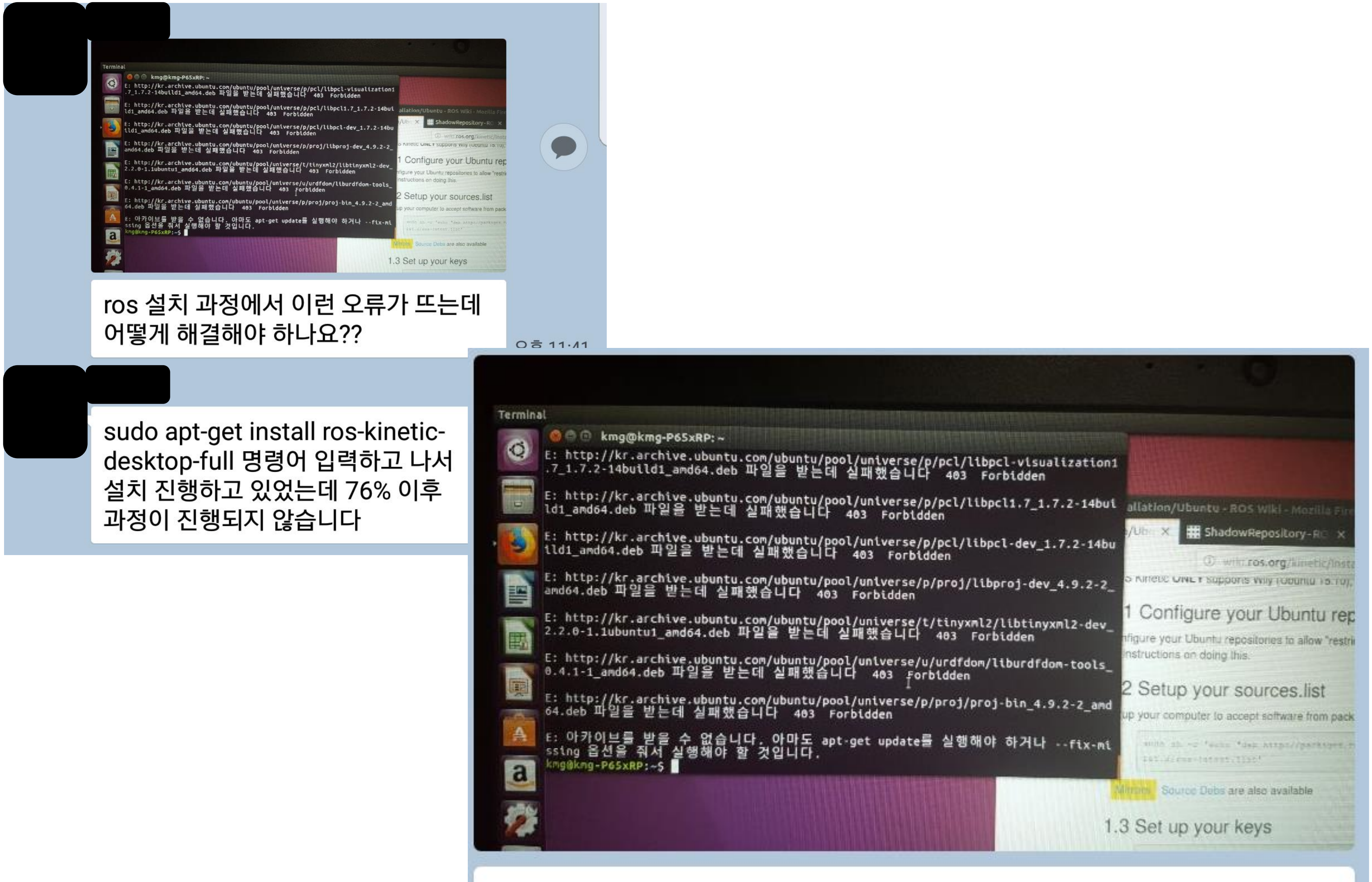
- Tele-communication
- It must be convenient for accessing NUC PC if you can be familiar with this.

- ssh username@hosname

ex) ssh seojh@hostname → no gui supported.
ssh -X seojh@hostname → gui supported. You can use qtcreator, gedit, and so on.

0. Revisit Questions…

- Guess what is the problem & solution.



[redacted]

ros 설치 과정에서 이런 오류가 뜨는데 어떻게 해결해야 하나요??

오후 11:41

[redacted]

sudo apt-get install ros-kinetic-desktop-full 명령어 입력하고 나서 설치 진행하고 있었는데 76% 이후 과정이 진행되지 않습니다

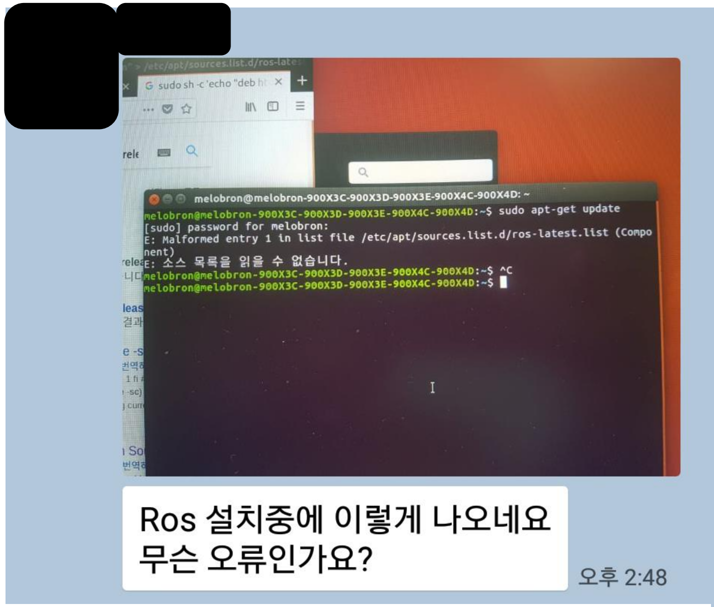
```
kmg@kmg-P65xRP: ~
E: http://kr.archive.ubuntu.com/ubuntu/pool/universe/p/pcl/libpcl-visualization1
.7_1.7.2-14build1_amd64.deb 파일을 받는데 실패했습니다  403  Forbidden

E: http://kr.archive.ubuntu.com/ubuntu/pool/universe/p/pcl/libpcl1.7_1.7.2-14bui
ld1_amd64.deb 파일을 받는데 실패했습니다  403  Forbidden

E: http://kr.archive.ubuntu.com/ubuntu/pool/universe/p/pcl/libpcl-dev_1.7.2-14bu
ild1_amd64.deb 파일을 받는데 실패했습니다  403  Forbidden

E: http://kr.archive.ubuntu.com/ubuntu/pool/universe/p/proj/libproj-dev_4.9.2-2_
amd64.deb 파일을 받는데 실패했습니다  403  Forbidden

E: http://kr.archive.ubuntu.com/ubuntu/pool/universe/t/tinyxml2/libtinyxml2-dev_
2.2.0-1.1ubuntu1_amd64.deb 파일을 받는데 실패했습니다  403  Forbidden

E: http://kr.archive.ubuntu.com/ubuntu/pool/universe/u/urdfdom/liburdfdom-tools_
0.4.1-1_amd64.deb 파일을 받는데 실패했습니다  403  Forbidden

E: http://kr.archive.ubuntu.com/ubuntu/pool/universe/p/proj/proj-bin_4.9.2-2_amd
64.deb 파일을 받는데 실패했습니다  403  Forbidden

E: 아카이브를 받을 수 없습니다. 아마도 apt-get update를 실행해야 하거나 --fix-mi
ssing 옵션을 줘서 실행해야 할 것입니다.
kmg@kmg-P65xRP:~$
```

Terminal
allation/Ubuntu - ROS Wiki - Mozilla Fire
ShadowRepository-R
wiki.ros.org/kinetic/Insta
1 Configure your Ubuntu rep
2 Setup your sources.list
Mirrors
Source Debs are also available
1.3 Set up your keys

0. Revisit Questions…

- Guess what is the problem & solution.

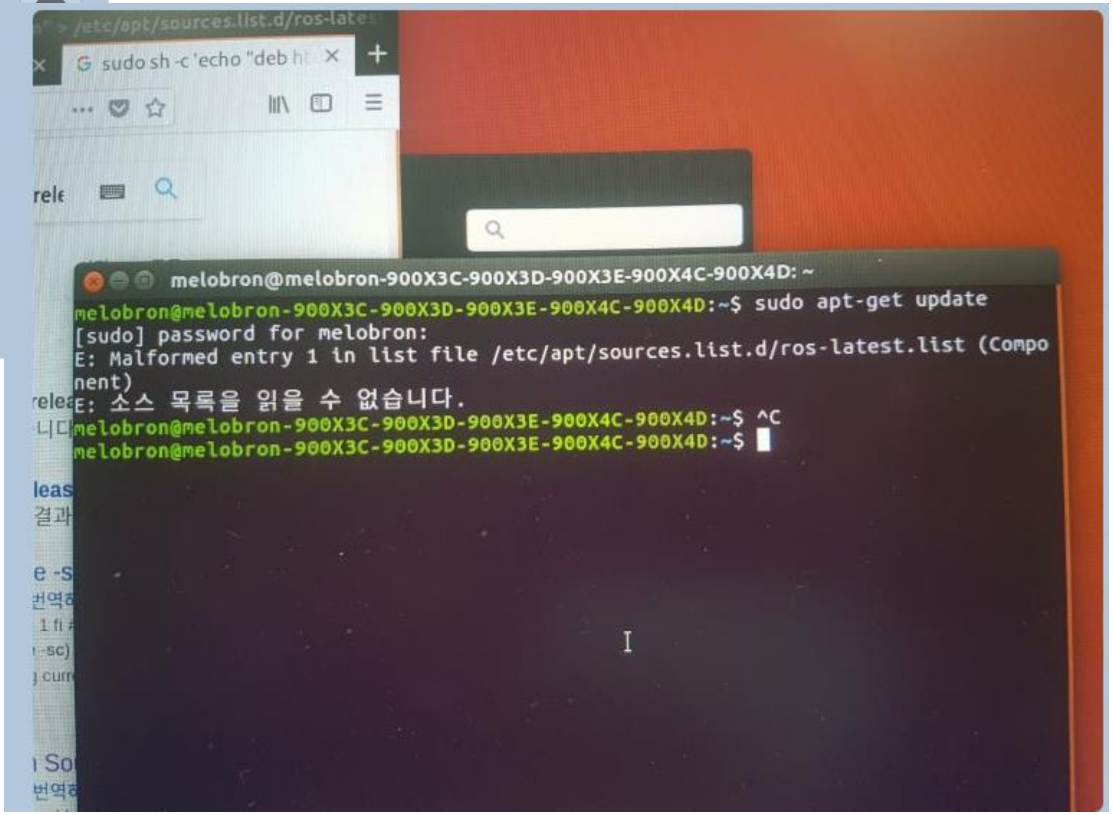
[redacted] [redacted]



```
melobron@melobron-900X3C-900X3D-900X3E-900X4C-900X4D: ~
melobron@melobron-900X3C-900X3D-900X3E-900X4C-900X4D:~$ sudo apt-get update
[sudo] password for melobron:
E: Malformed entry 1 in list file /etc/apt/sources.list.d/ros-latest.list (Compo
nent)
E: 소스 목록을 읽을 수 없습니다.
melobron@melobron-900X3C-900X3D-900X3E-900X4C-900X4D:~$ ^C
melobron@melobron-900X3C-900X3D-900X3E-900X4C-900X4D:~$ 
```

Ros 설치중에 이렇게 나오네요
무슨 오류인가요?

오후 2:48

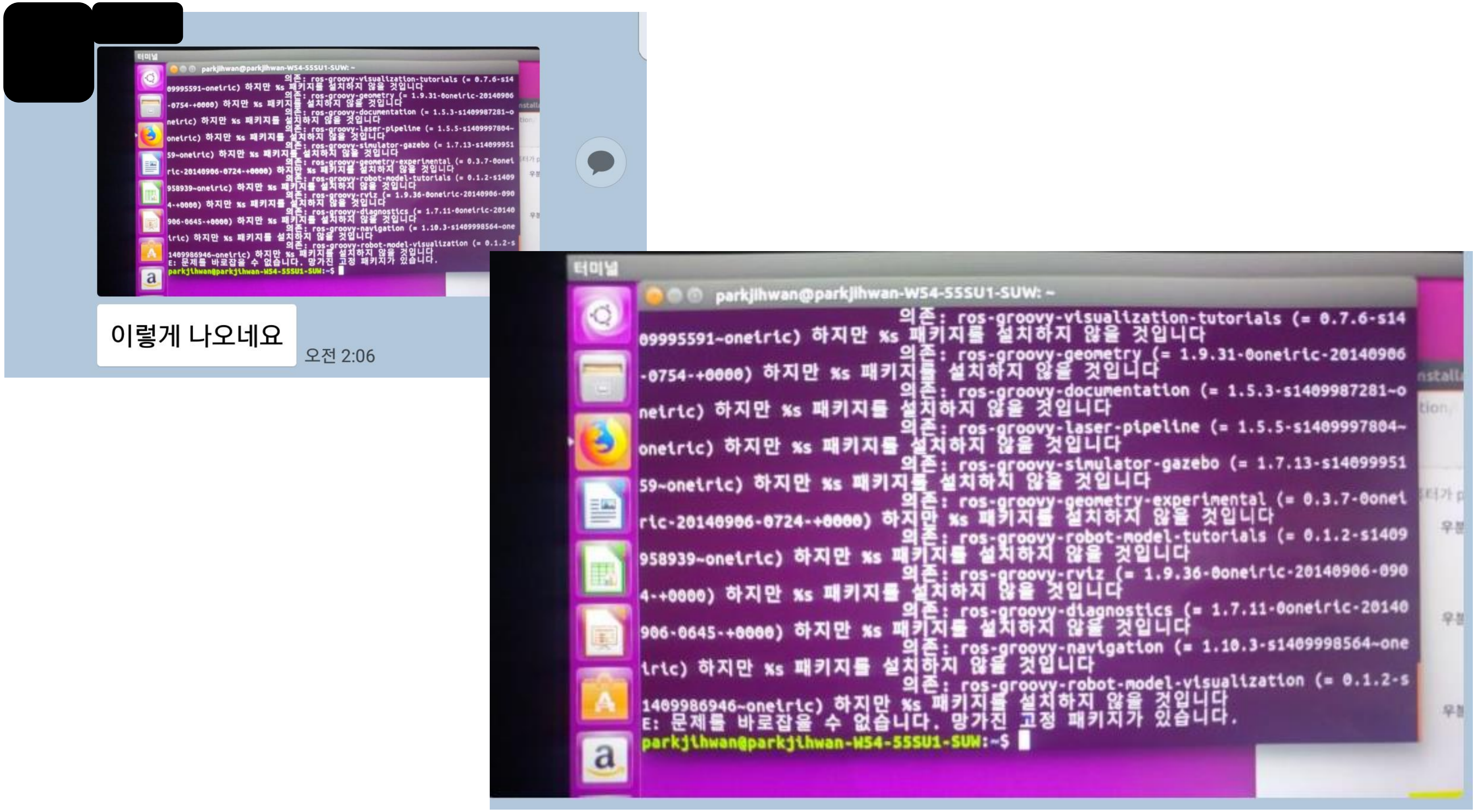


```
melobron@melobron-900X3C-900X3D-900X3E-900X4C-900X4D: ~
melobron@melobron-900X3C-900X3D-900X3E-900X4C-900X4D:~$ sudo apt-get update
[sudo] password for melobron:
E: Malformed entry 1 in list file /etc/apt/sources.list.d/ros-latest.list (Compo
nent)
E: 소스 목록을 읽을 수 없습니다.
melobron@melobron-900X3C-900X3D-900X3E-900X4C-900X4D:~$ ^C
melobron@melobron-900X3C-900X3D-900X3E-900X4C-900X4D:~$ 
```

- Hint. Read a error message carefully…

0. Revisit Questions…

- Guess what is the problem & solution.



[redacted]

```
parkjihwan@parkjihwan-W54-55SU1-SUW: ~
                          의존: ros-groovy-visualization-tutorials (= 0.7.6-s14
09995591~oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                          의존: ros-groovy-geometry (= 1.9.31-0oneiric-20140906
-0754-+0000) 하지만 %s 패키지를 설치하지 않을 것입니다
                          의존: ros-groovy-documentation (= 1.5.3-s1409987281~o
neiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                          의존: ros-groovy-laser-pipeline (= 1.5.5-s1409997804~
oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                          의존: ros-groovy-simulator-gazebo (= 1.7.13-s14099951
59~oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                          의존: ros-groovy-geometry-experimental (= 0.3.7-0onei
ric-20140906-0724-+0000) 하지만 %s 패키지를 설치하지 않을 것입니다
                          의존: ros-groovy-robot-model-tutorials (= 0.1.2-s1409
958939~oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                          의존: ros-groovy-rviz (= 1.9.36-0oneiric-20140906-090
4-+0000) 하지만 %s 패키지를 설치하지 않을 것입니다
                          의존: ros-groovy-diagnostics (= 1.7.11-0oneiric-20140
906-0645-+0000) 하지만 %s 패키지를 설치하지 않을 것입니다
                          의존: ros-groovy-navigation (= 1.10.3-s1409998564~one
iric) 하지만 %s 패키지를 설치하지 않을 것입니다
                          의존: ros-groovy-robot-model-visualization (= 0.1.2-s
1409986946~oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
E: 문제를 바로잡을 수 없습니다. 망가진 고정 패키지가 있습니다.
parkjihwan@parkjihwan-W54-55SU1-SUW:~$
```

이렇게 나오네요

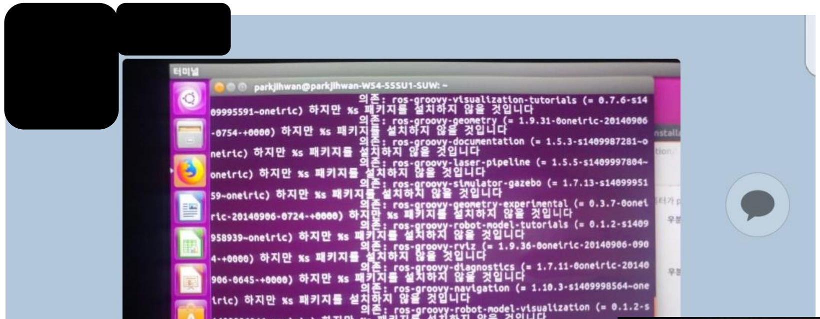
오전 2:06

- Hint. Why this command handles old-version ros?
  (maybe, the link of ROS packages is indicating old URL)

0. Revisit Questions…

- Guess what is the problem & solution.

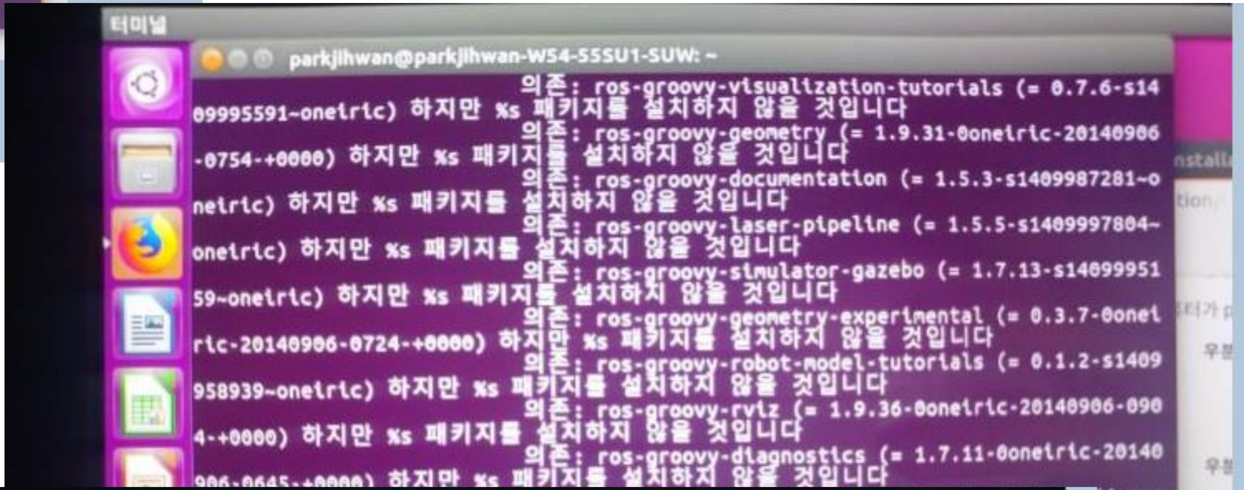
[redacted]



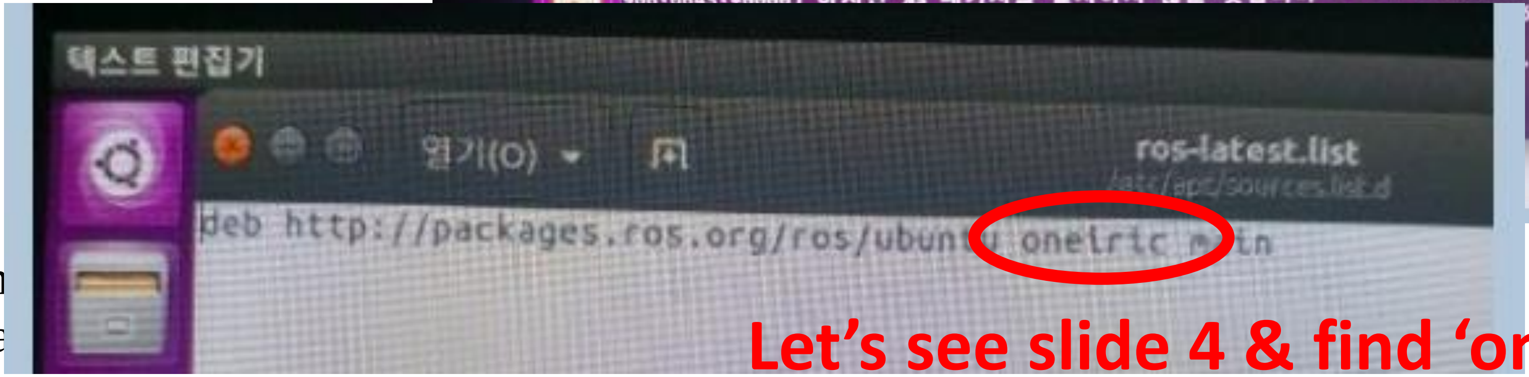
```
parkjihwan@parkjihwan-W54-55SU1-SUW: ~
                              의존: ros-groovy-visualization-tutorials (= 0.7.6-s14
09995591~oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-geometry (= 1.9.31-0oneiric-20140906
-0754-+0000) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-documentation (= 1.5.3-s1409987281~o
neiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-laser-pipeline (= 1.5.5-s1409997804~
oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-simulator-gazebo (= 1.7.13-s14099951
59~oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-geometry-experimental (= 0.3.7-0onei
ric-20140906-0724-+0000) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-robot-model-tutorials (= 0.1.2-s1409
958939~oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-rviz (= 1.9.36-0oneiric-20140906-090
4-+0000) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-diagnostics (= 1.7.11-0oneiric-20140
906-0645-+0000) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-navigation (= 1.10.3-s1409998564~one
iric) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-robot-model-visualization (= 0.1.2-s
1409986946~oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
E: 문제를 바로잡을 수 없습니다. 망가진 고정 패키지가 있습니다.
parkjihwan@parkjihwan-W54-55SU1-SUW:~$
```

이렇게 나오네요

오전 2:06



```
parkjihwan@parkjihwan-W54-55SU1-SUW: ~
                              의존: ros-groovy-visualization-tutorials (= 0.7.6-s14
09995591~oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-geometry (= 1.9.31-0oneiric-20140906
-0754-+0000) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-documentation (= 1.5.3-s1409987281~o
neiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-laser-pipeline (= 1.5.5-s1409997804~
oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-simulator-gazebo (= 1.7.13-s14099951
59~oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-geometry-experimental (= 0.3.7-0onei
ric-20140906-0724-+0000) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-robot-model-tutorials (= 0.1.2-s1409
958939~oneiric) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-rviz (= 1.9.36-0oneiric-20140906-090
4-+0000) 하지만 %s 패키지를 설치하지 않을 것입니다
                              의존: ros-groovy-diagnostics (= 1.7.11-0oneiric-20140
906-0645-+0000) 하지만 %s 패키지를 설치하지 않을 것입니다
```



```
deb http://packages.ros.org/ros/ubuntu oneiric main
```

- Hin
  (ma

Let's see slide 4 & find 'oneiric'

0. Revisit Questions…

- You will do these things again and again and again and again and again…..

- Think about the reason why we share questions and answer…
  (Let's follow a question rule from now on: Q.~~~)

- I hope that students' questions are answered by students at the end of semester(or at the end of this year…)

- You can help each others.
  Also, you can be a TA next year…

0. Revisit Questions…

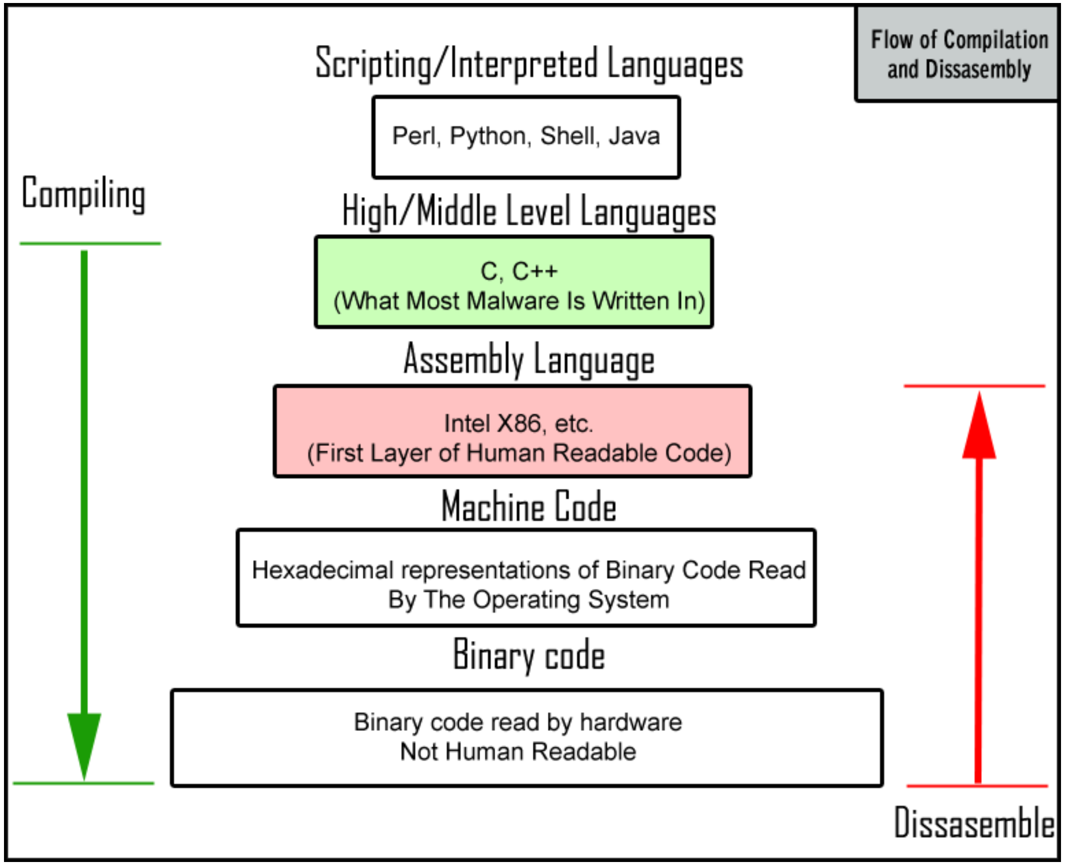
- The more you try and fail, the deeper you will understand.

- Just try something. The worst case that can happen to you is just re-installation of Ubuntu(not the end of the world or F grade for the course)

- Regardless of how much you messed up ubuntu in your laptop, TAs will help you and resolve it clear.

Additional Topic. Concepts of compiler, g++, makefile & cmake

**This material is for students who are not familiar with c++



Flow of Compilation and Dissasembly

Scripting/Interpreted Languages

Perl, Python, Shell, Java

High/Middle Level Languages

C, C++
(What Most Malware Is Written In)

Assembly Language

Intel X86, etc.
(First Layer of Human Readable Code)

Machine Code

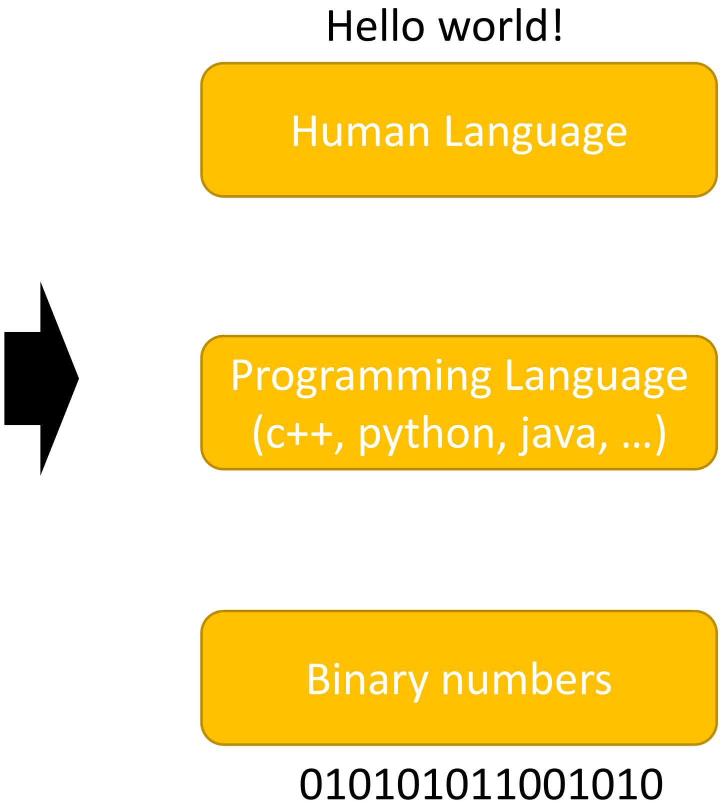
Hexadecimal representations of Binary Code Read By The Operating System

Binary code

Binary code read by hardware
Not Human Readable

Compiling

Dissasemble



Hello world!

Human Language

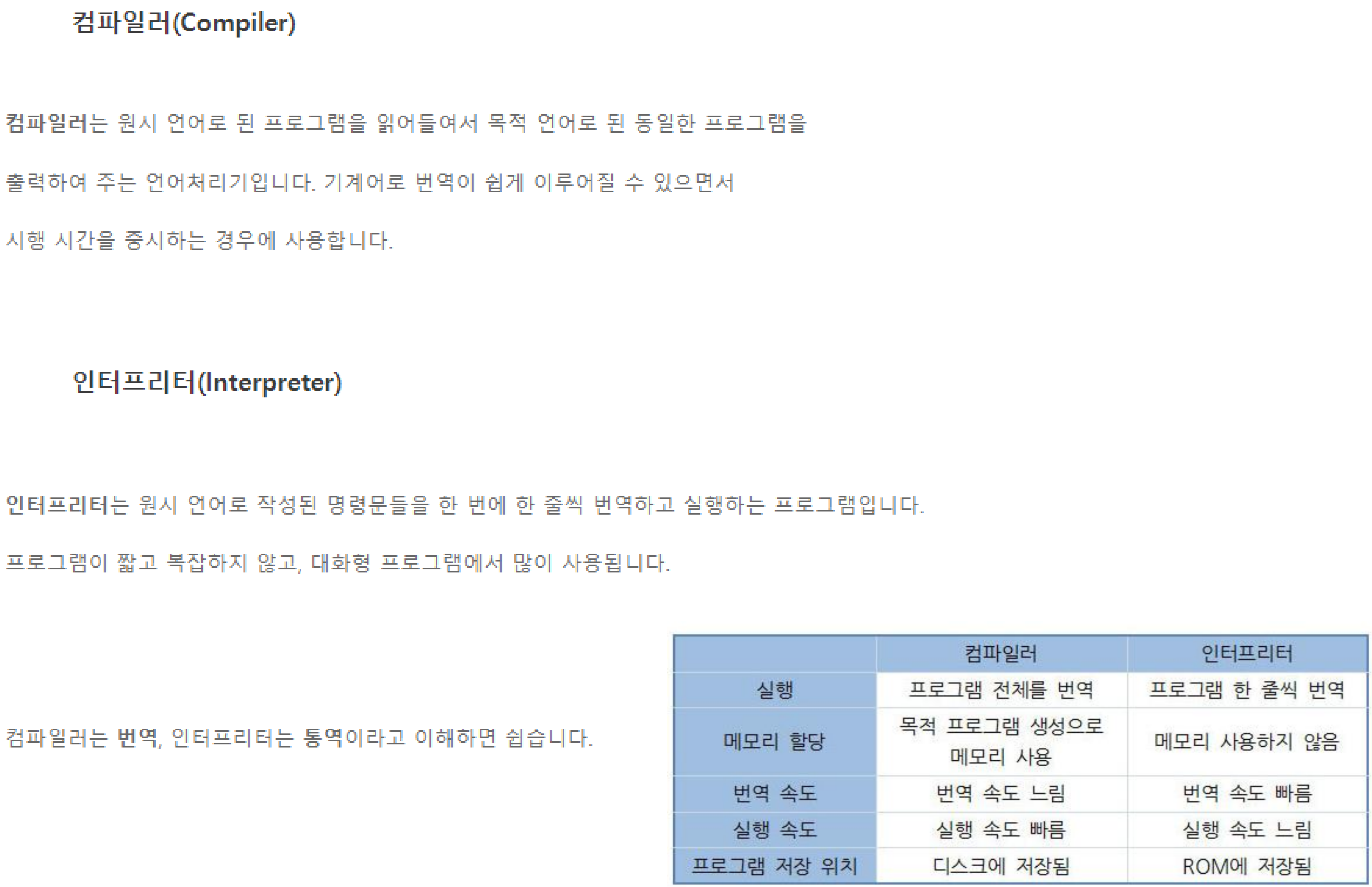
Programming Language
(c++, python, java, …)

Binary numbers

010101011001010

Compiling: Process of converting programming language to machine-understandable language(binary numbers)

Additional Topic. Concepts of compiler, g++, makefile & cmake



컴파일러(Compiler)

컴파일러는 원시 언어로 된 프로그램을 읽어들여서 목적 언어로 된 동일한 프로그램을

출력하여 주는 언어처리기입니다. 기계어로 번역이 쉽게 이루어질 수 있으면서

시행 시간을 중시하는 경우에 사용합니다.

인터프리터(Interpreter)

인터프리터는 원시 언어로 작성된 명령문들을 한 번에 한 줄씩 번역하고 실행하는 프로그램입니다.

프로그램이 짧고 복잡하지 않고, 대화형 프로그램에서 많이 사용됩니다.

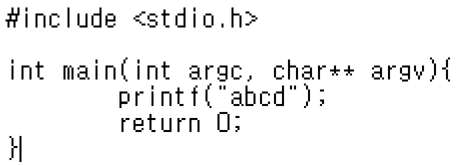
컴파일러는 **번역**, 인터프리터는 **통역**이라고 이해하면 쉽습니다.

	컴파일러	인터프리터
실행	프로그램 전체를 번역	프로그램 한 줄씩 번역
메모리 할당	목적 프로그램 생성으로 메모리 사용	메모리 사용하지 않음
번역 속도	번역 속도 느림	번역 속도 빠름
실행 속도	실행 속도 빠름	실행 속도 느림
프로그램 저장 위치	디스크에 저장됨	ROM에 저장됨

Which one is better language for your mobile platform?

Additional Topic. Concepts of compiler, g++, makefile & cmake

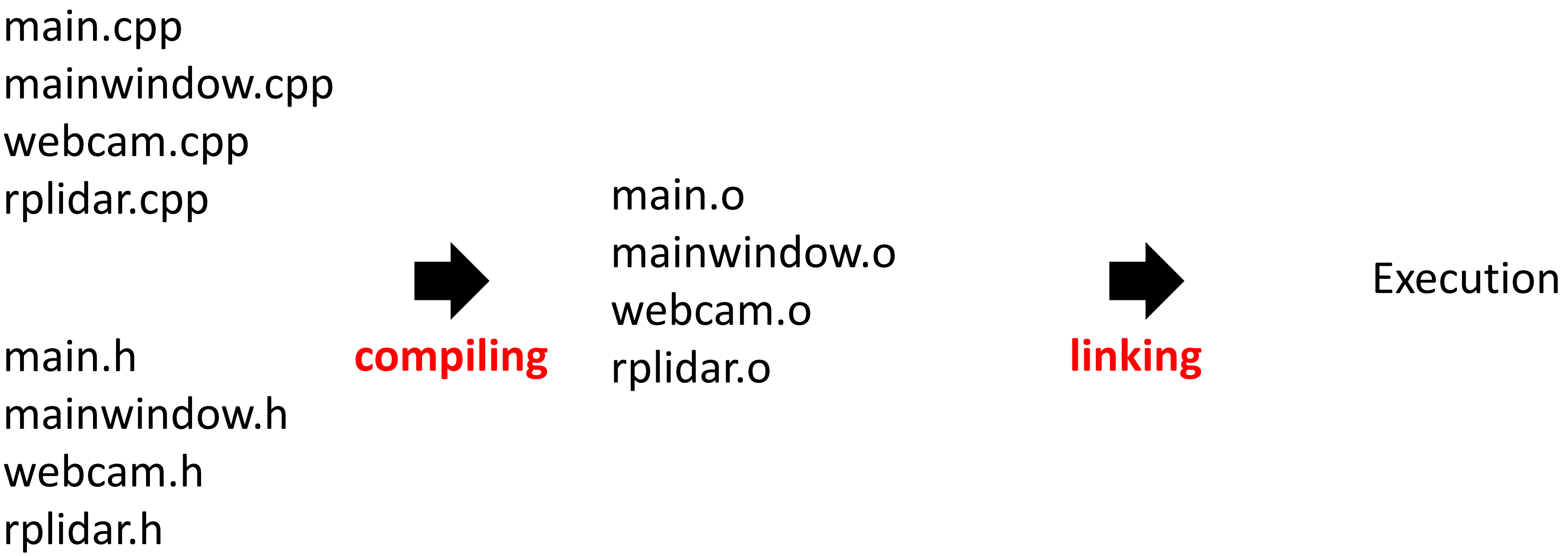
In the early stage of compiling,



```
#include <stdio.h>

int main(int argc, char** argv){
        printf("abcd");
        return 0;
}
```

Execution

However, programs become huge & write a code in a single file becomes inefficient. Programmers started to divided them into pieces. (You leant how to create object in python!)



main.cpp
mainwindow.cpp
webcam.cpp
rplidar.cpp

main.h
mainwindow.h
webcam.h
rplidar.h

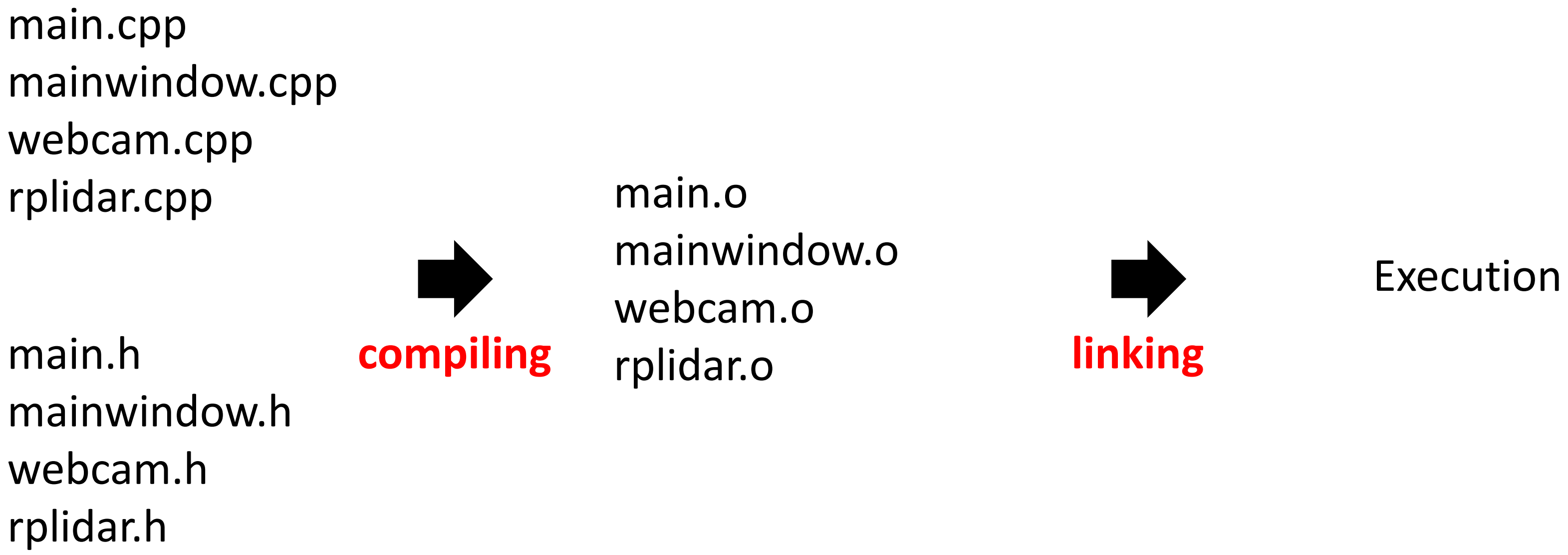
compiling

main.o
mainwindow.o
webcam.o
rplidar.o

linking

Execution

Additional Topic. Concepts of compiler, g++, makefile & cmake



main.cpp
mainwindow.cpp
webcam.cpp
rplidar.cpp

main.h
mainwindow.h
webcam.h
rplidar.h

compiling

main.o
mainwindow.o
webcam.o
rplidar.o

linking

Execution

If the errors are appeared in compiling process
→ you should check your source file(.cpp&.h files)

If the errors are appeared in linking process
→ you should check the options of your packages
(mostly, checking CMakeLists.txt file will be enough)

Let’s see some examples…

Additional Topic. Concepts of compiler, g++, makefile & cmake

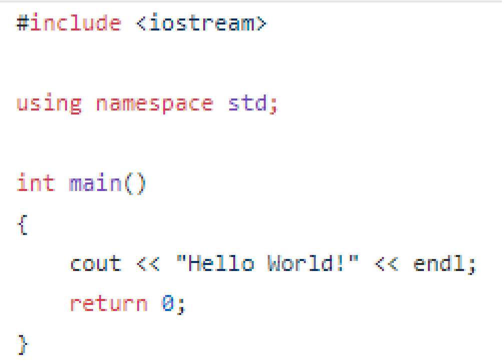
-get the example source codes by typing, git clone https://github.com/kaistmecd/ros.git

-in ros/session_notes/gcc_makefile_cmake folder,

-0. gcc_example   (read README.txt)

-1. makefile_example (read README.txt)

-2. cmake_example
  - Create a project using ‘Qtcreator’
  - select ‘non-qt project ‘ → ‘plain c++ application’ → select ‘cmake‘ as build system.
  - write any code like below.



```
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello World!" << endl;
    return 0;
}
```

-run the program and check the folder ‘build-(your project name)-Desktop-Default
you can find Makefile in that folder!!

Additional Topic. Concepts of compiler, g++, makefile & cmake

In summary!

-gcc: GNU Compiler Collection

*https://en.wikipedia.org/wiki/GNU_Compiler_Collection

-g++: gcc for c++

-by using command 'g++' you can compile any source codes. just make a simple code &
g++ main.cpp

-g++ gets lots of options
→ if the program includes many sources files and libraries, then g++ command will be complex and long.

-to resolve it, Makefile is introduced…

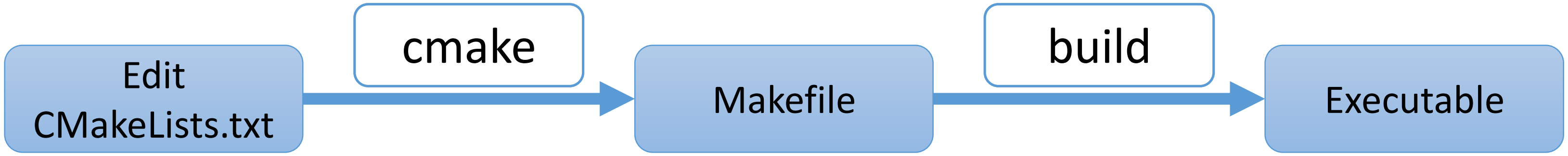
-The problem of Makefile is that it is not compatible in other platform(other OS like windows)

-cmake(Cross-platform Make) supports compatible build process between various platforms.

Additional Topic. Concepts of compiler, g++, makefile & cmake

For ROS students,
when you create a packages, you will use a command ‘catkin_create_pkg’. Then, your packages will be created with CMakeList.txt file.

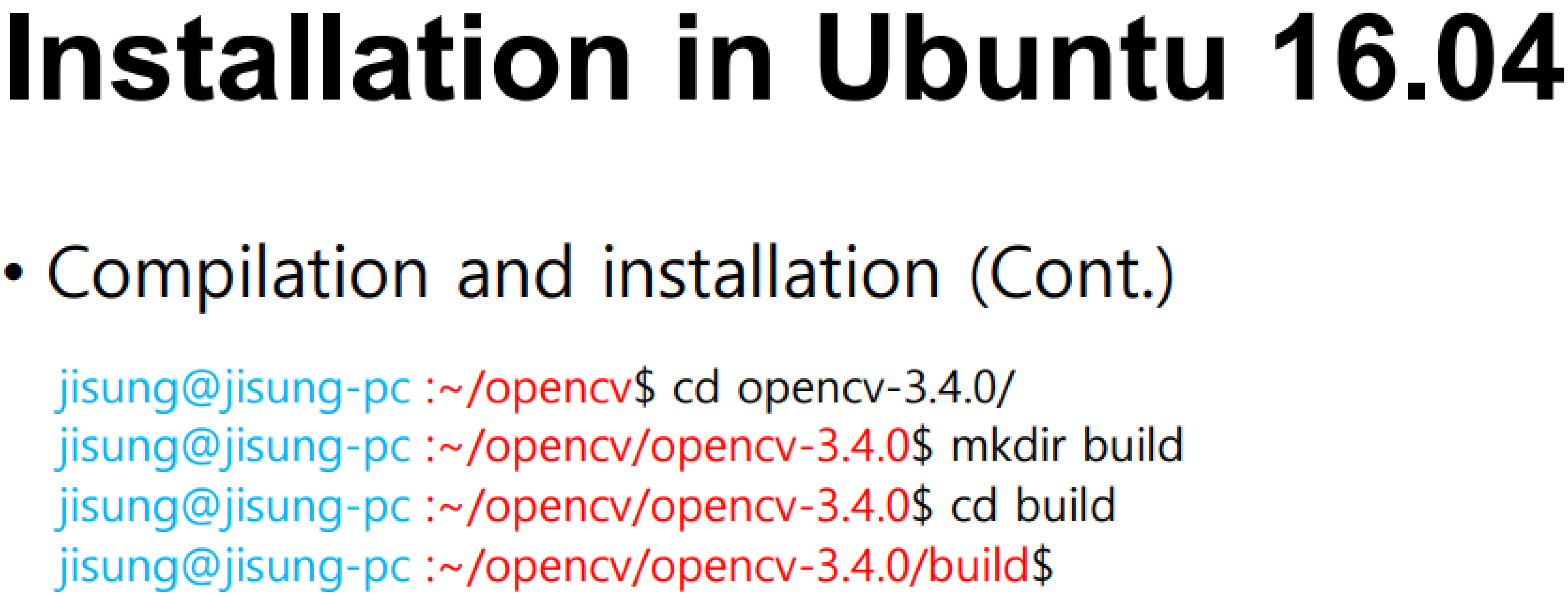
For OpenCV students,
you have an option not to use ‘cmake-based project’. However, your code should be combined with ROS based programs for the project. Therefore, I recommend you to utilize ‘cmake-based project’. (you can do it easily by using qtcreator)



Additional Topic. OpenCV installation

-revisit OpenCV installation process

-see the OpenCV lecture note by TA. Jisung Park



Installation in Ubuntu 16.04

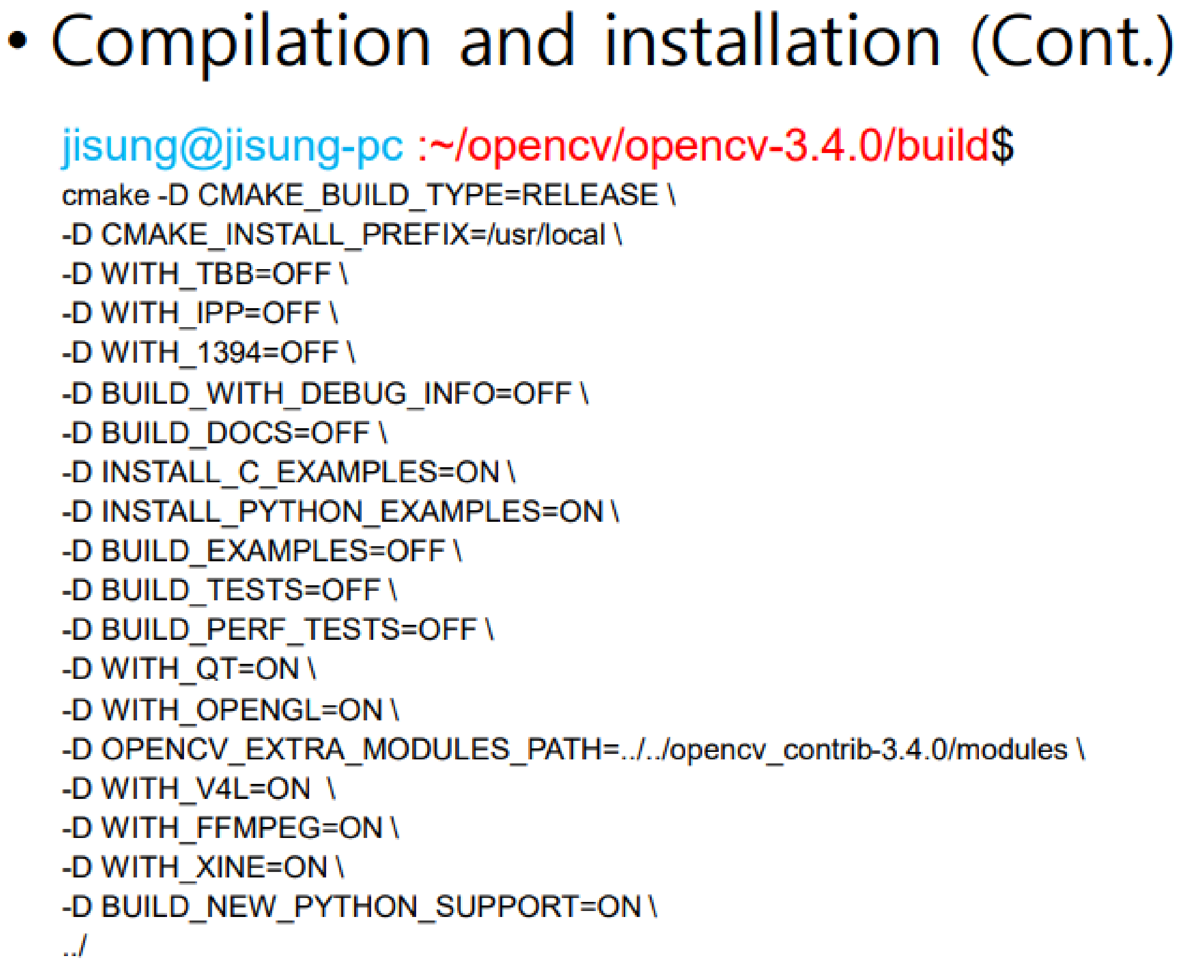
- Compilation and installation (Cont.)

```
jisung@jisung-pc :~/opencv$ cd opencv-3.4.0/
jisung@jisung-pc :~/opencv/opencv-3.4.0$ mkdir build
jisung@jisung-pc :~/opencv/opencv-3.4.0$ cd build
jisung@jisung-pc :~/opencv/opencv-3.4.0/build$
```

Additional Topic. OpenCV installation

-revisit OpenCV installation process

-see the OpenCV lecture note by TA. Jisung Park



- Compilation and installation (Cont.)

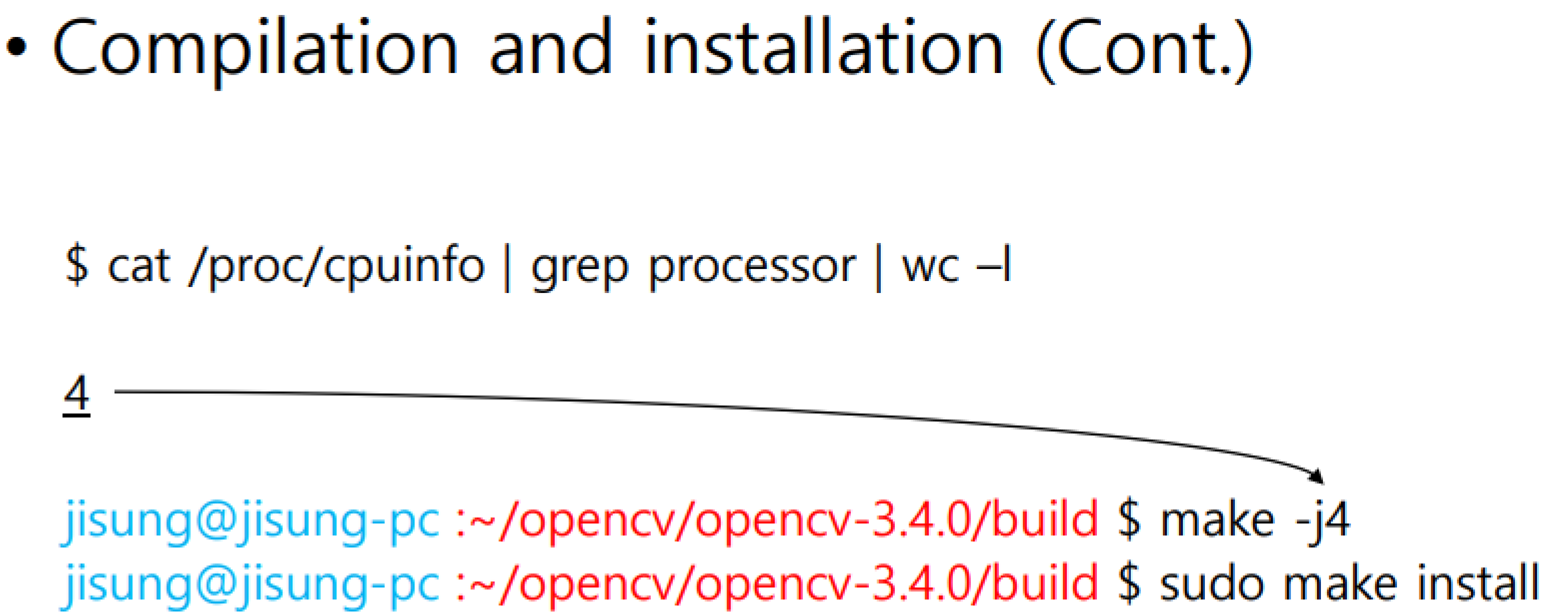
```
jisung@jisung-pc :~/opencv/opencv-3.4.0/build$
cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D WITH_TBB=OFF \
-D WITH_IPP=OFF \
-D WITH_1394=OFF \
-D BUILD_WITH_DEBUG_INFO=OFF \
-D BUILD_DOCS=OFF \
-D INSTALL_C_EXAMPLES=ON \
-D INSTALL_PYTHON_EXAMPLES=ON \
-D BUILD_EXAMPLES=OFF \
-D BUILD_TESTS=OFF \
-D BUILD_PERF_TESTS=OFF \
-D WITH_QT=ON \
-D WITH_OPENGL=ON \
-D OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib-3.4.0/modules \
-D WITH_V4L=ON  \
-D WITH_FFMPEG=ON \
-D WITH_XINE=ON \
-D BUILD_NEW_PYTHON_SUPPORT=ON \
../
```

After you run ‘cmake‘ command, you can find ‘Makefile’ in the same folder!

Additional Topic. OpenCV installation

-revisit OpenCV installation process

-see the OpenCV lecture note by TA. Jisung Park



- Compilation and installation (Cont.)

```
$ cat /proc/cpuinfo | grep processor | wc –l

4

jisung@jisung-pc :~/opencv/opencv-3.4.0/build $ make -j4
jisung@jisung-pc :~/opencv/opencv-3.4.0/build $ sudo make install
```

‘make’ command will compile the opencv source codes
‘sudo make install’ command will locate the opencv files into system folders like ‘/usr/local/include’ & ‘/usr/lib’

Break Time.
Any Questions?

1. ROS – Overview

ROS.org

- Robot Operation System
- Introduced by WillowGarage, USA (but they are vanished……)
- Still maintaining

1. ROS – Install

- Google ‘ros install’ or visit ros.org

- http://wiki.ros.org/kinetic/Installation/Ubuntu

- Installation can be done with terminal and commands only.

1. ROS – Install(Explanations of commands)

-
  ```
  sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list‘
  ```

  우분투가 설치된 직후의 repository는 ros 설치파일들이 있는 경로를 가지고 있지 않습니다. (cd /etc/apt/에 들어가서 있는 파일들을 살펴보면 ros관련 경로들은 없는 걸 확인할 수 있음). ROS의 설치파일들이 있는 주소를 repository list에 넣어주는 명령입니다.

-
  ```
  sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116
  ```

  ROS설치파일이 있는 repository link에 접근하기 위한 키를 설정하는 명령(어지간하면 다시 쓸일없는 명령어)

-
  ```
  sudo apt-get update
  ```
  ROS의 설치파일이 있는 주소를 repository list를 담당하는 파일에는 포함시켰지만, 아직 그 주소를 반영하지는 않은 상태이므로, 이를 반영하기 위해 update진행.(명령어 실행 후, 나오는 메세지들을 잘 살펴보면 http://packages.ros.org/ros/ubuntu~~가 추가된걸 확인할 수 있음)

1. ROS – Install(Explanations of commands)

- sudo apt-get install ros-kinetic-desktop-full
  (이제 ROS의 위치를 알고있으니, 다른 프로그램들 설치하듯이 ros 또한 apt-get install 명령으로 설치)

- sudo rosdep init && rosdep update
  (ros를 설치하는 과정에서 설치된 여러 패키지들의 dependency를 자동으로 설정해주는 명령어. 이것도 한 번 실행하고 나면 다시 쓸 일이 거의 없음. Ros설치 과정에서 설치된 패키지들은 이 시점에서 실행하는걸로 dependency가 설정되고, 추후에 작성되는 패키지들은 CMakeLists.txt에서 dependency 처리를 담당함)

- echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
  (ROS관련 명령어를 실행하기 위해서는 ROS관련 패키지들의 위치를 터미널상에서 설정해주어야 사용가능. 매번 터미널 열때마다 이를 반복할 수 없으니, bashrc에 추가해두고 편하게 쓰는 것)

- source ~/.bashrc
  (.bashrc를 수정하는것 자체는 터미널에 영향을 주지 않음. Bashrc를 수정한 후에 터미널을 새로 열거나, 아니면 위의 명령어를 이용하면 현재 터미널에서 .bashrc의 내용을 다시 불러올 수 있음)

1. ROS – Initialize workspace

- mkdir ~/catkin_ws
  (catkin_ws 라는 폴더 생성. 이 곳을 workspace로 활용함.)

- cd catkin_ws (catkin_ws 폴더로 이동)
- mkdir src (src 폴더 생성)
- cd src (or cd ~/catkin_ws/src)
- catkin_init_workspace
  (현재 경로인 ~/catkin_ws/src 를 workspace로 설정)

- cd .. (or ~/catkin_ws)
- catkin_make
  (일반적으로는 ros packag를 작성후 빌드하는 명령어이나,
  지금은 catkin_init_workspace라는 명령어를 통해 생성된 CMakeLists.txt를 빌드하여 ros관련해서 필요한 dependency, environment variables 등을 설정하도록 해줌)

- echo "source ~/catkin_ws/devel/setup.bash" >> ~/.bashrc
  (catkin_make 후에 생성된 setup.bash 파일을 .bashrc파일에 추가함으로써 이후에 생성되는 터미널들이 workspace관련 설정을 활용할 수 있도록 함)
- source ~/.bashrc
  (이후에 생성될 터미널들은 설정이 되겠지만, 여태까지 사용해오던 터미널은 적용되어 있지 않으므로, 다시 .bashrc를 불러와주는 명령어)

1. ROS – revisit shell script

- https://github.com/ROBOTIS-GIT/robotis_tools/blob/master/install_ros_kinetic.sh

- It is just a combined commands for ROS installation & workspace initialization!

1. ROS –Tutorial

- Run ‘roscore’
- Run ‘rosrun rospy_tutorials listener’
- Run ‘rosrun rospy_tutorials talker’
- Run ‘rosrun rqt_graph rqt_graph’
- rostopic echo /talker

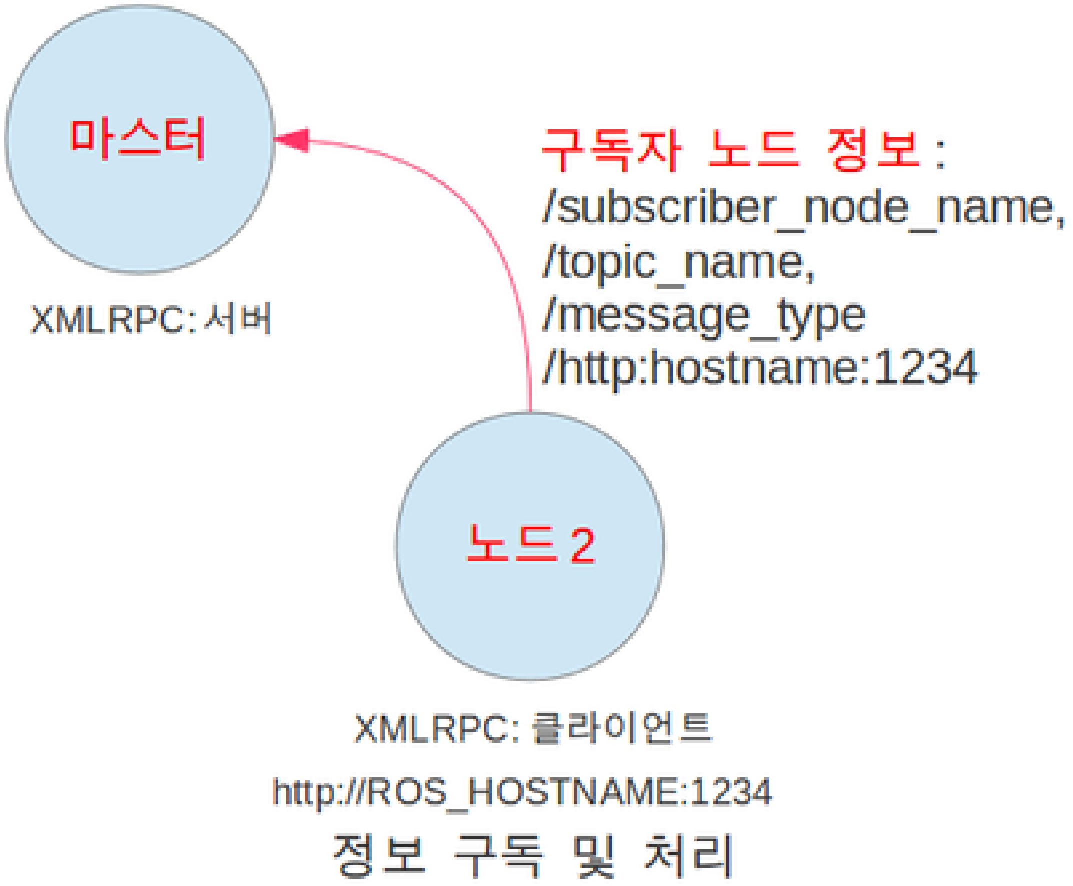
1. ROS –Tutorial(explanation)

XMLRPC: 서버

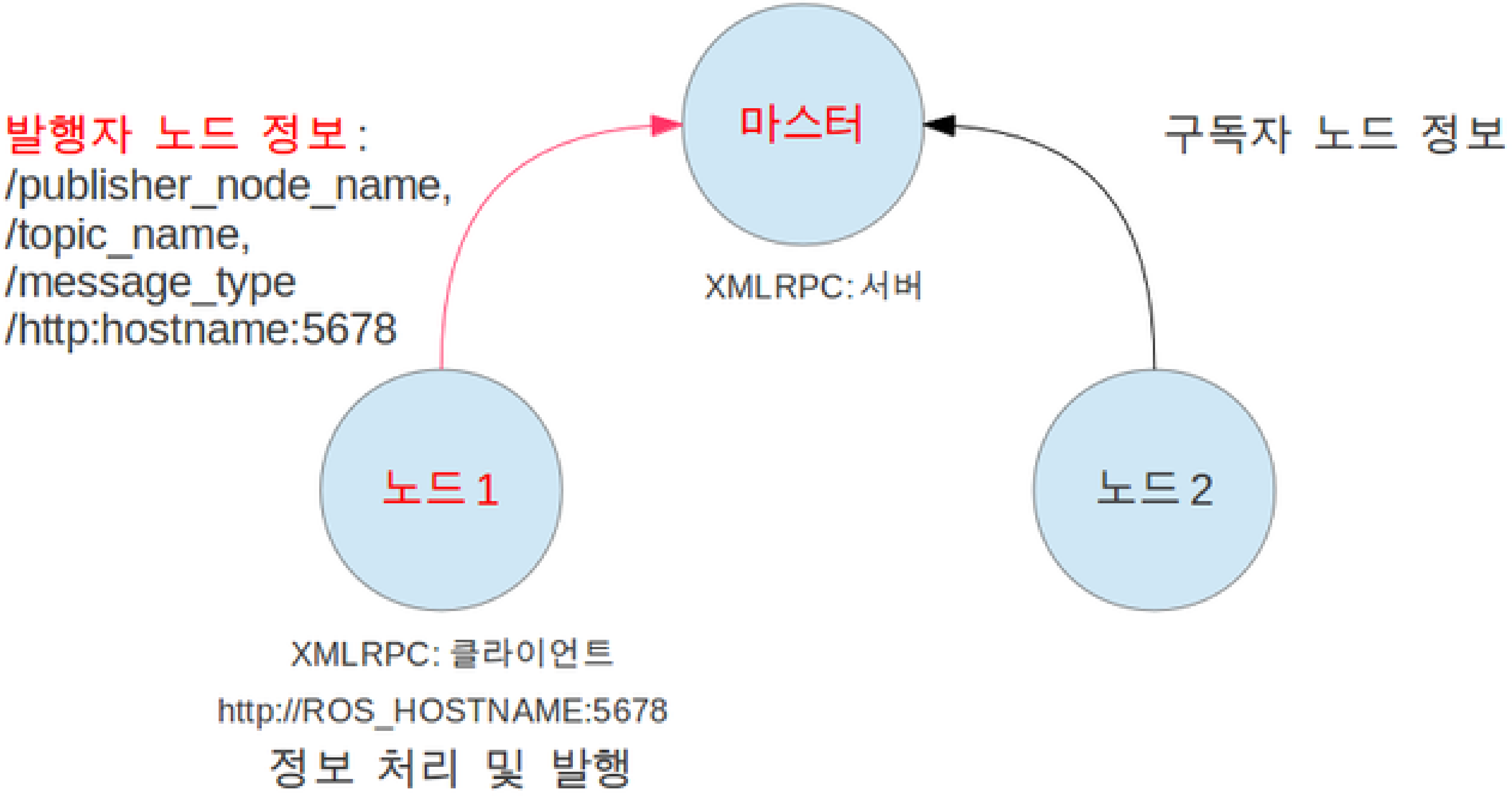
http://ROS_MASTER_URI:11311

노드 정보 관리

1. ROS –Tutorial(explanation)



1. ROS –Tutorial(explanation)



발행자 노드 정보 :
/publisher_node_name,
/topic_name,
/message_type
/http:hostname:5678

마스터

XMLRPC: 서버

구독자 노드 정보

노드 1

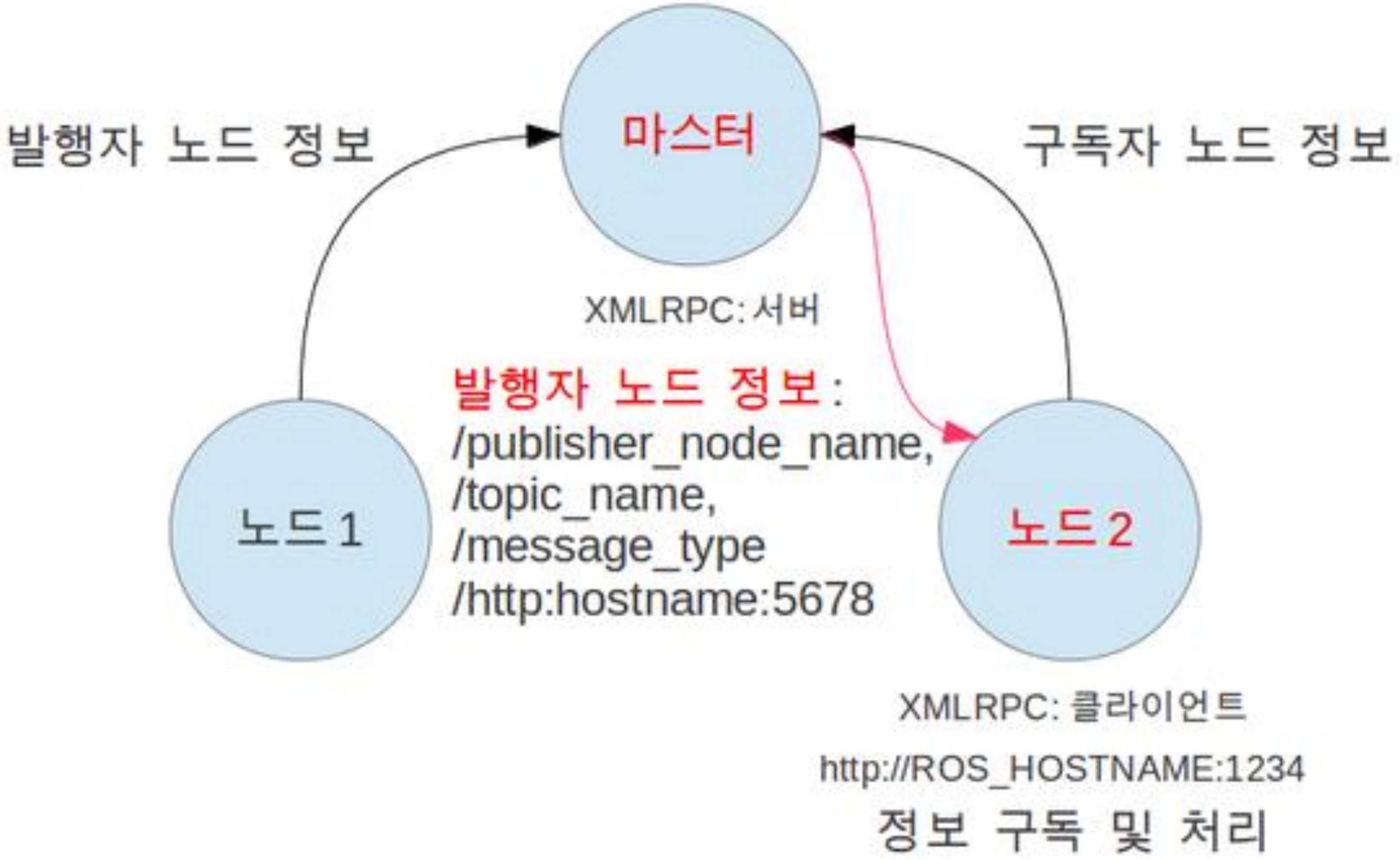
노드 2

XMLRPC: 클라이언트

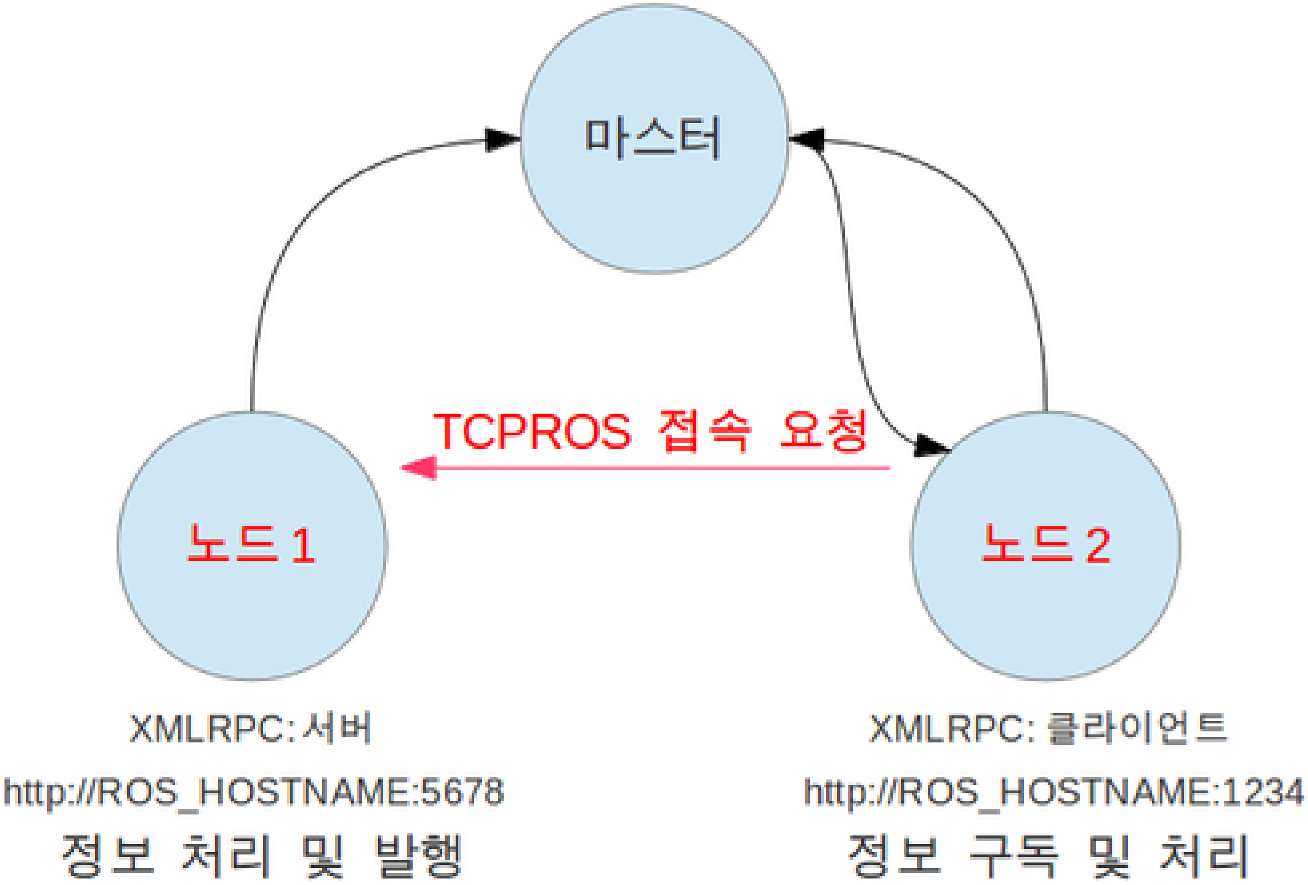
http://ROS_HOSTNAME:5678

정보 처리 및 발행

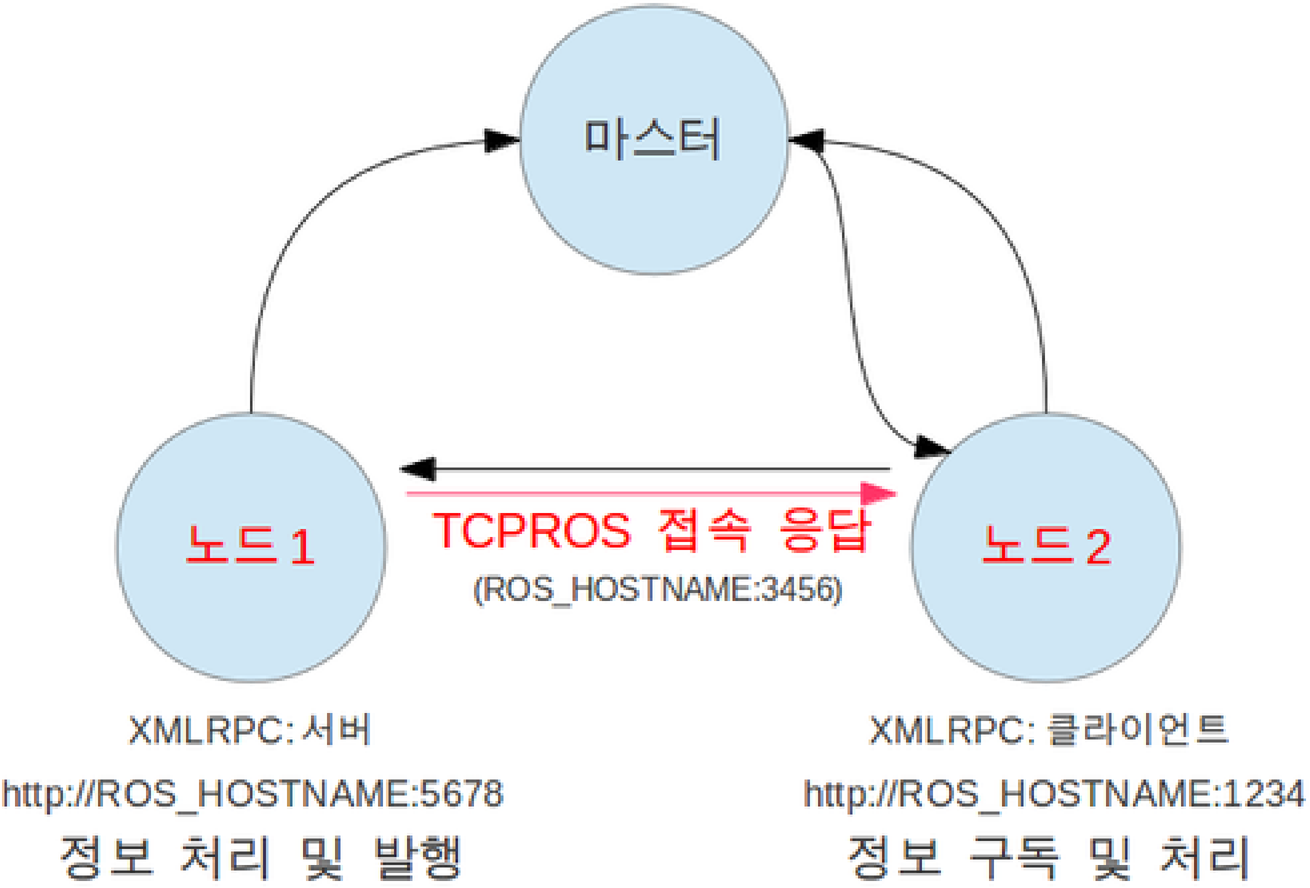
1. ROS –Tutorial(explanation)



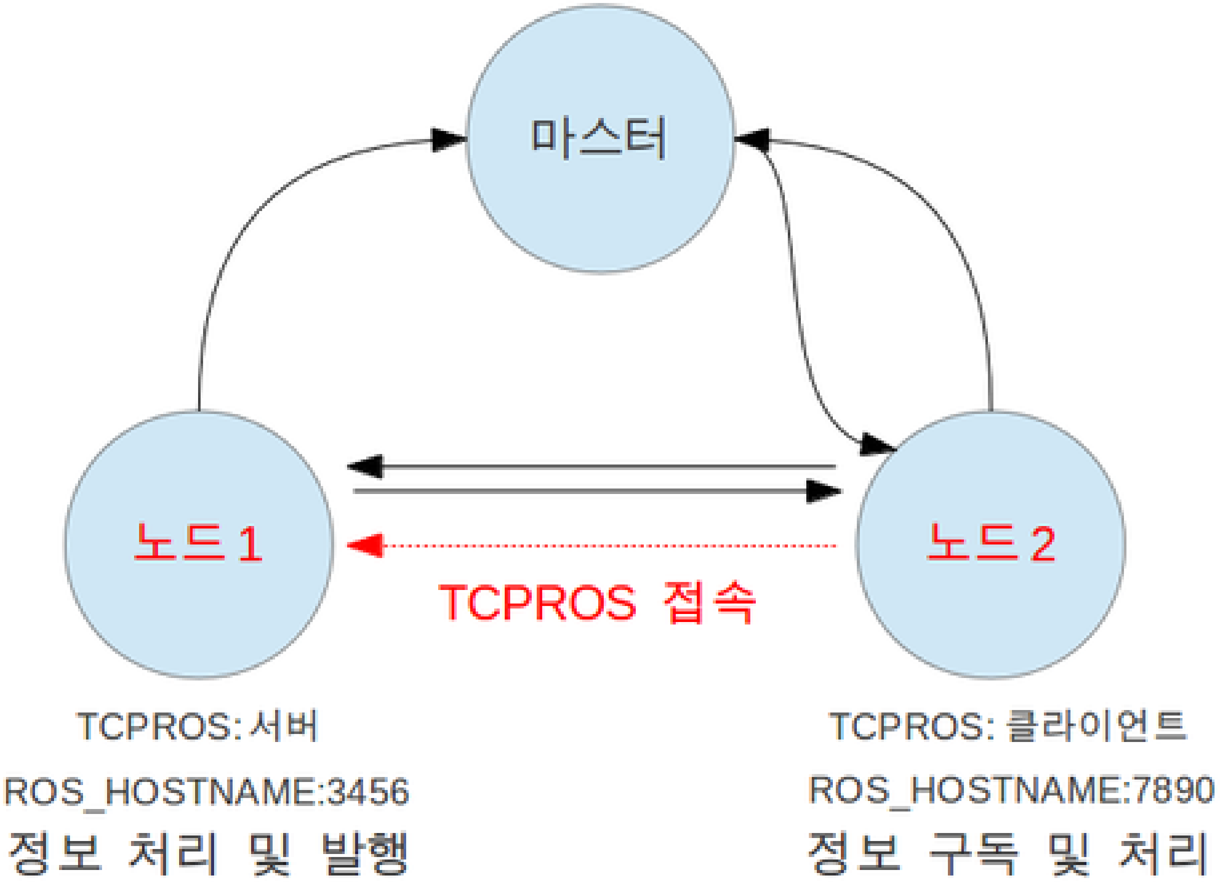
1. ROS –Tutorial(explanation)



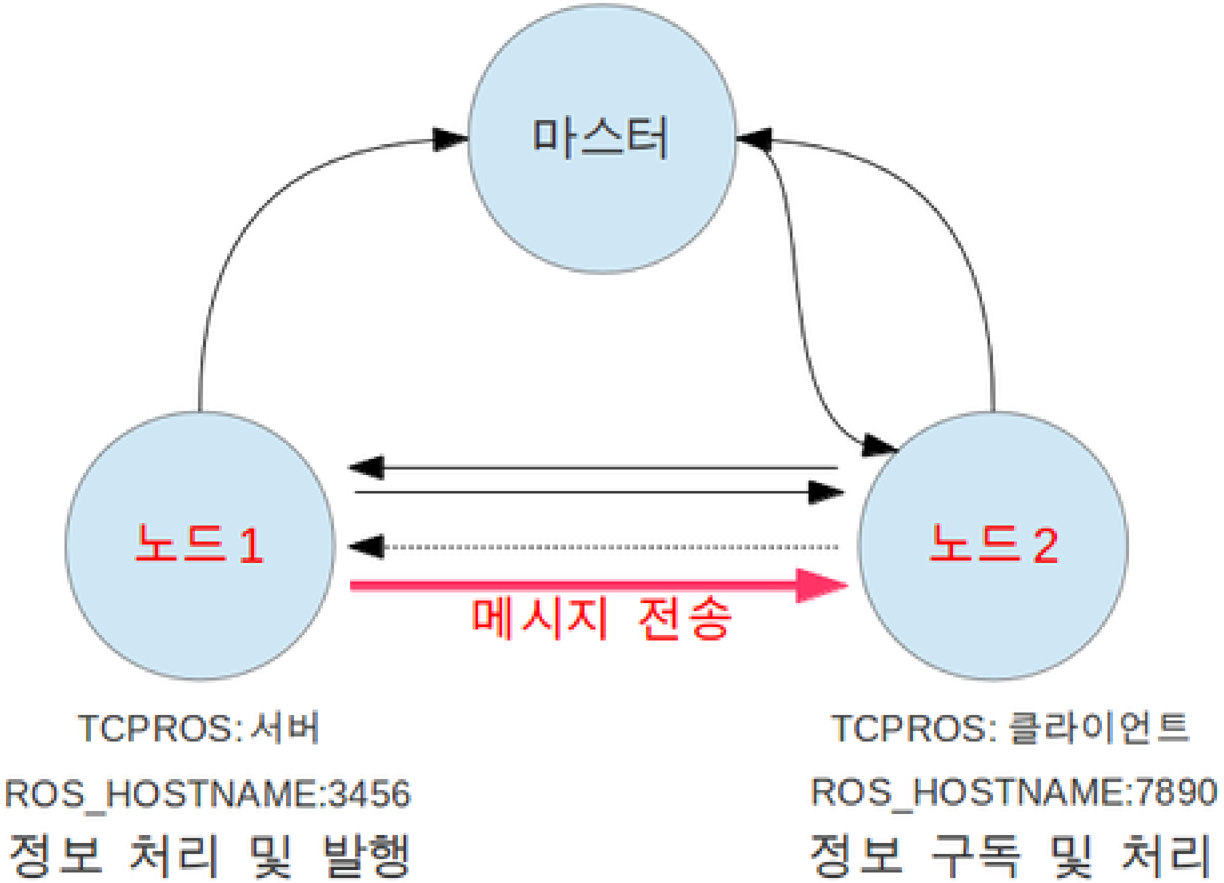
1. ROS –Tutorial(explanation)



1. ROS –Tutorial(explanation)

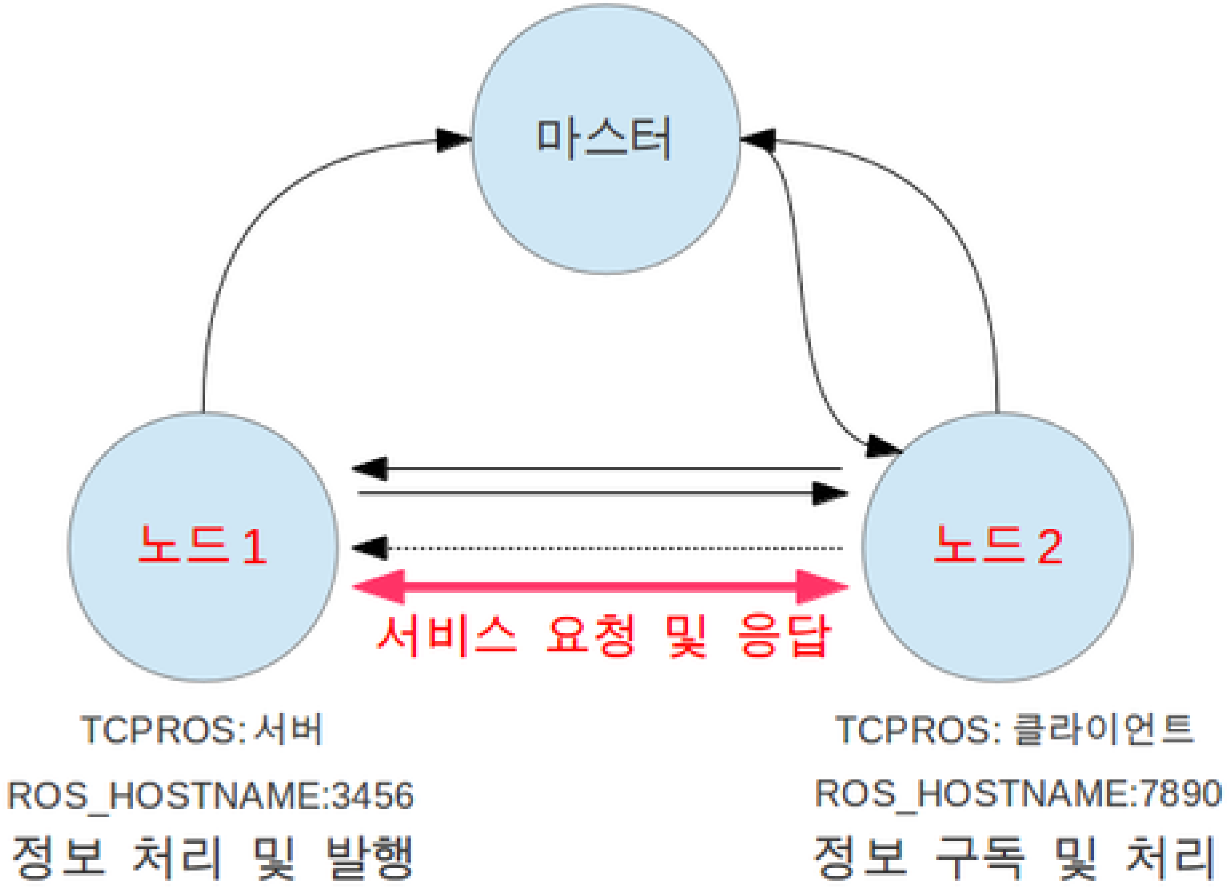


1. ROS –Tutorial(explanation)



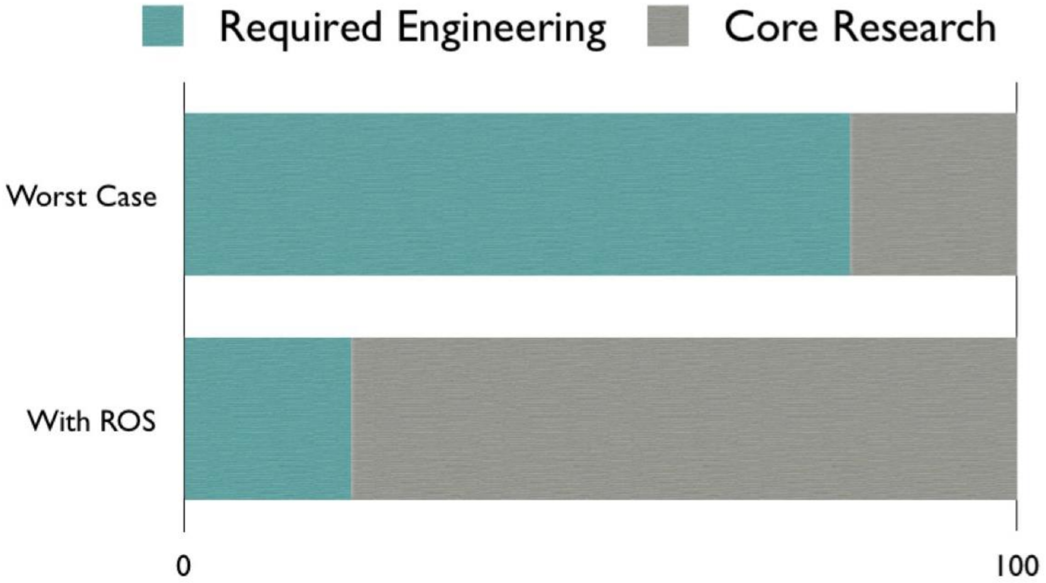
Message: asynchronous transfer mode(비동기 전송)

1. ROS –Tutorial(explanation)



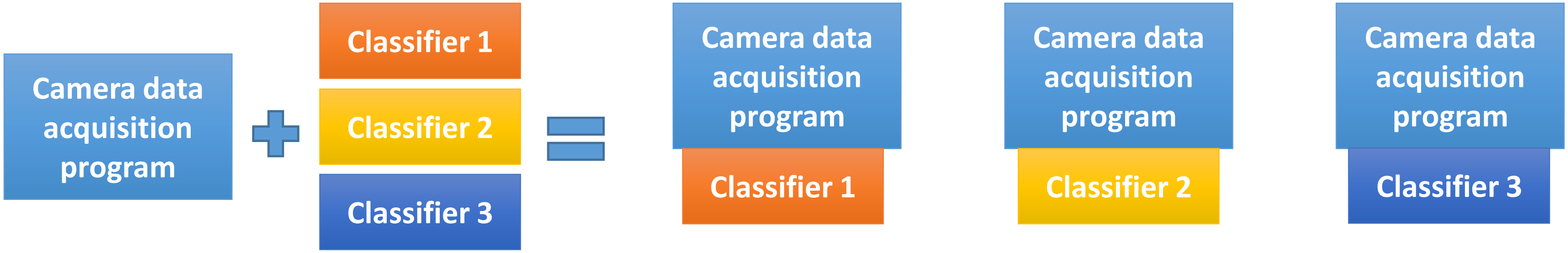
Service : synchronous transfer mode(동기 전송)

1. ROS –Why ROS?



- Now, you might understand what this figure means…
- Single function for single nodes → Modularization
- Ex) Once program a node for acquiring Camera data, then this node can be used for various applications

Without ROS



Camera data acquisition program

Classifier 1

Classifier 2

Classifier 3

Camera data acquisition program

Classifier 1

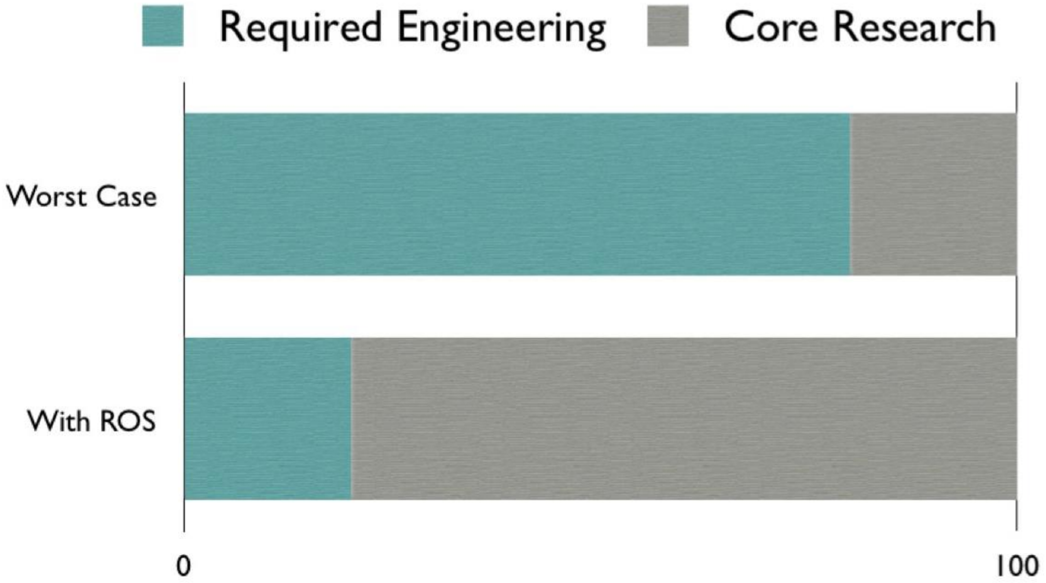
Camera data acquisition program

Classifier 2

Camera data acquisition program

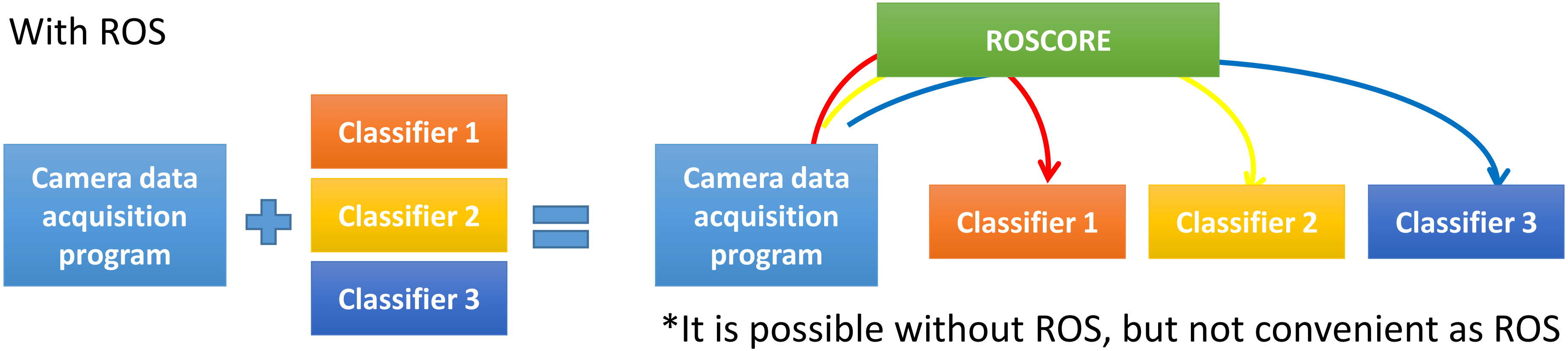
Classifier 3

1. ROS –Why ROS?



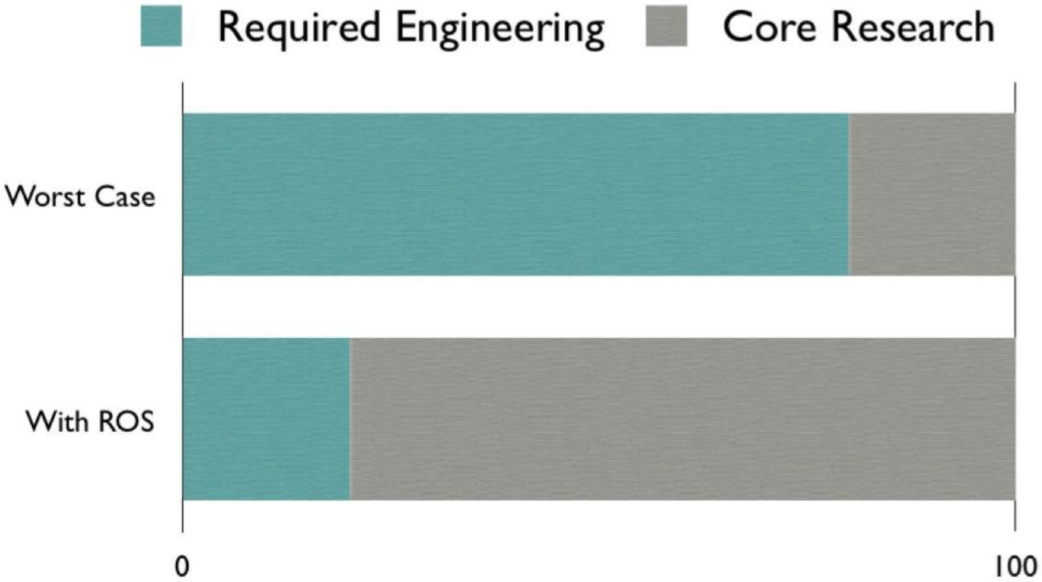
- Now, you might understand what this figure means…
- Single function for single nodes → Modularization
- Ex) Program a node for acquiring Camera data, then this node can be used for various applications

With ROS

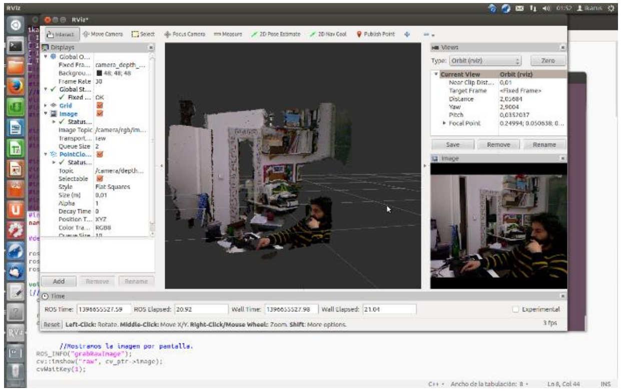


*It is possible without ROS, but not convenient as ROS

1. ROS –Why ROS?



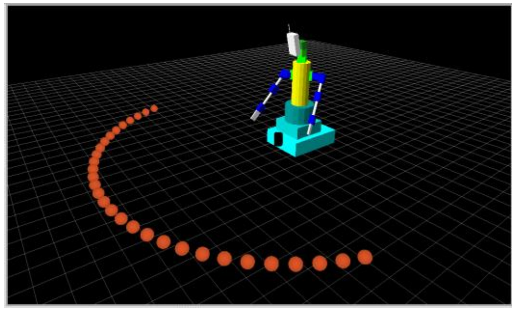
- Also, various libraries are provided.
- Ex) pointcloud library, opencv, etc …
- In addition, data visualization, data management, simulation tools are provided

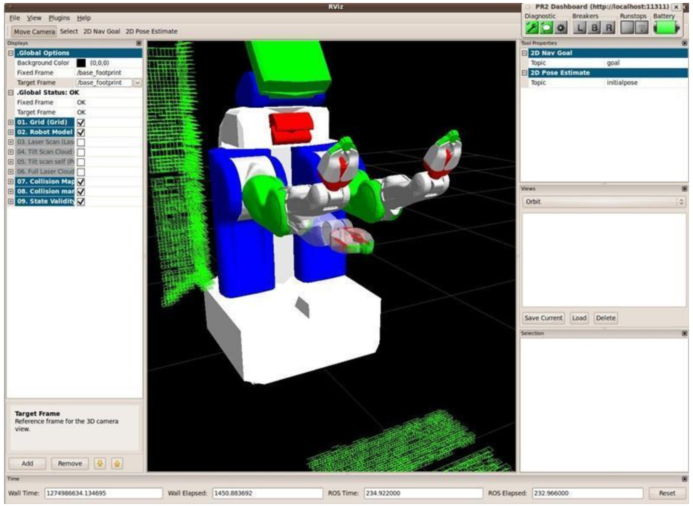


ROS Time: 1396655527.59 ROS Elapsed: 20.92 Wall Time: 1396655527.98 Wall Elapsed: 21.04

Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel: Zoom. Shift: More options.

```
    //Mostramos la imagen por pantalla.
ROS_INFO("grabRawImage");
cv::imshow("raw", cv_ptr->image);
cvWaitKey(1);
```





Target Frame
Reference frame for the 3D camera view.

Wall Time: 1274986634.134695 Wall Elapsed: 1450.883692 ROS Time: 234.922000 ROS Elapsed: 232.966000

1. ROS –Useful commands & tools

- Useful ROS commands

- roscore, rosrun, roslaunch
- roscd, rosls
- rostopic, rosnode, rosservice, rosparam, rosmsg, rossrv, rosbag

- Useful ROS tools

- rqt_plot
- rqt_graph
- rqt_bag

- Study by yourself

2. ROS – Create Nodes

1. Initialize your workspace (after initialization, you don’t have to do this again)

```
-mkdir catkin_ws
-cd catkin_ws
-mkdir src
-cd src
-catkin_init_workspace
-cd ..    (or cd ~/catkin_ws)
-catkin_make
```

2. Create a ros package

```
-cd ~/catkin_ws/src
-catkin_create_pkg (pkg_name) (dependencies)
```

for example,   ‘catkin_create_pkg ros_tutorial roscpp std_msgs’

3. Check the contents in the folder of the ros package created

2. ROS – Create Nodes

4. Make a source file in /src folder
   - gedit main.cpp
   - (follow step by step)

5. Modify CMakeLists.txt

6. Go to catkin_ws and build it!
   -cd ~/catkin_ws
   -catkin_make

7. Update the list of ros packages in your computer
   -rospack profile

8. Execute
   rosrun (pkgname) (nodename)

   *source codes and video will be uploaded in https://github.com/kaistmecd/

ROS Session Finished.
Any Questions?

Finished