

자율주행 데브코스

ROS 기초 실습

(주)자이트론

허성민

smher@xytron.co.kr



Contents

ROS 설치

ROS 환경 설정

ROS 예제 프로그램 구동 실습



ROS 기초 실습

노트북에 ROS 설치하기

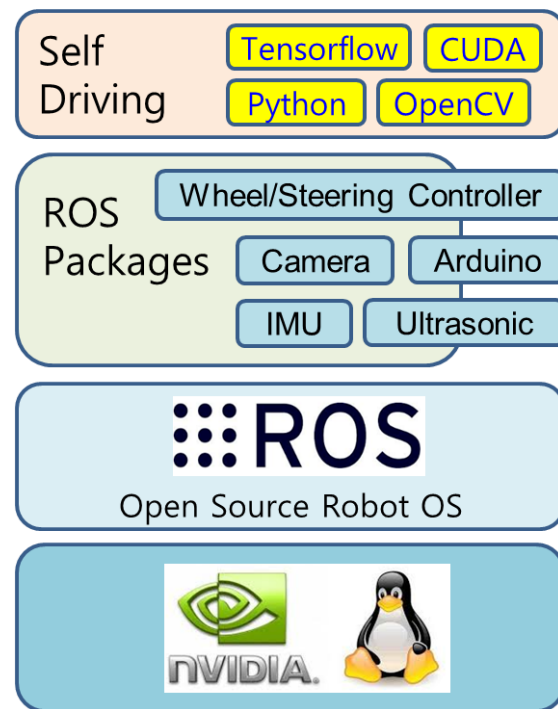


ROS 설치

- 실습 기자재인 모형차에는 TX2 보드가 탑재되어 있고,
 - 이 보드에는 이미 Ubuntu 가 설치되어 있으며
 - 그 위에는 ROS kinetic 이 설치되어 있음



프로세서 모듈
NVIDIA TX2



ROS 설치

- 준비물
 - 리눅스 Ubuntu 16.04가 설치된 노트북 또는 PC
 - 파일 다운로드를 위한 인터넷 환경



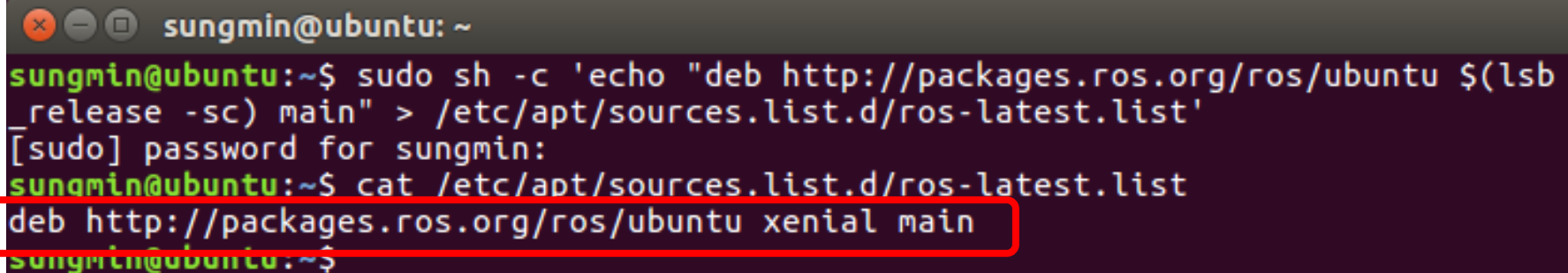
ROS



ROS 설치 과정 (1)

- ROS를 제공하는 Software Repository 등록

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



```
sungmin@ubuntu: ~  
sungmin@ubuntu:~$ sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'  
[sudo] password for sungmin:  
sungmin@ubuntu:~$ cat /etc/apt/sources.list.d/ros-latest.list  
deb http://packages.ros.org/ros/ubuntu xenial main  
sungmin@ubuntu:~$
```

ROS 설치 과정 (2)

- apt key를 셋업

```
$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654
```

```
sungmin@ubuntu: ~  
sungmin@ubuntu:~$ sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' -  
-recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654  
Executing: /tmp/tmp.hDBcBW80gt/gpg.1.sh --keyserver  
hkp://keyserver.ubuntu.com:80  
--recv-key  
C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654  
gpg: requesting key AB17C654 from hkp server keyserver.ubuntu.com  
gpg: key AB17C654: public key "Open Robotics <info@osrfoundation.org>" imported  
gpg: Total number processed: 1  
gpg: imported: 1 (RSA: 1)  
sungmin@ubuntu:~$
```



ROS 설치 과정 (3-1)

- 패키지 설치

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

```
sungmin@ubuntu: ~  
sungmin@ubuntu:~$ sudo apt-get update  
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]  
Get:2 http://packages.ros.org/ros/ubuntu xenial InRelease [4,678 B]  
Hit:3 http://us.archive.ubuntu.com/ubuntu xenial InRelease  
Get:4 http://packages.ros.org/ros/ubuntu xenial/main amd64 Packages [830 kB]  
Get:5 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]  
Get:6 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]  
Get:7 http://packages.ros.org/ros/ubuntu xenial/main i386 Packages [602 kB]  
Fetched 1,762 kB in 18s (95.5 kB/s)  
Reading package lists... Done  
sungmin@ubuntu:~$
```



ROS 설치 과정 (3-2)

- 패키지 설치

```
$ sudo apt-get install ros-kinetic-desktop-full
```

```
sungmin@ubuntu: ~  
The following packages will be upgraded:  
  bzip2 cpp-5 g++-5 gcc-5 gcc-5-base gir1.2-gdkpixbuf-2.0 libasan2 libatomic1  
  libbz2-1.0 libcc1-0 libcilkrts5 libcomerr2 libcurl3 libcurl3-gnutls  
  libexpat1 libfreetype6 libgcc-5-dev libgdk-pixbuf2.0-0  
  libgdk-pixbuf2.0-common libglib2.0-0 libglib2.0-bin libgomp1 libicu55  
  libitm1 libjpeg-turbo8 libldap-2.4-2 liblsan0 libmpx0 libpython2.7  
  libpython2.7-minimal libpython2.7-stdlib libqt5core5a libqt5dbus5 libqt5gui5  
  libqt5network5 libqt5opengl5 libqt5printsupport5 libqt5sql5 libqt5test5  
  libqt5widgets5 libqt5xml5 libquadmamath0 libsqlite3-0 libssl1.0.0  
  libstdc++-5-dev libstdc++6 libtiff5 libtsan0 libubsan0 libudev1 libuuid1  
  libxml2 python2.7 python2.7-minimal udev zlib1g  
56 upgraded, 868 newly installed, 0 to remove and 268 not upgraded.  
Need to get 550 MB of archives.  
After this operation, 2.507 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

y 입력



ROS 설치 과정 (3-3)

- 패키지 설치

```
sungmin@ubuntu: ~  
Setting up ros-kinetic-vision-opencv (1.12.8-0xenial-20191214-092047+0000) ...  
Setting up ros-kinetic-perception (1.3.2-0xenial-20200218-165054+0000) ...  
Setting up dh-strip-nondeterminism (0.015-1) ...  
Setting up odbcinst (2.3.1-4.1) ...  
Setting up ruby2.3 (2.3.1-2~ubuntu16.04.14) ...  
Setting up ruby (1:2.3.0+1) ...  
Setting up gazebo7 (7.0.0+dfsg-2) ...  
Setting up ros-kinetic-gazebo-dev (2.5.19-1xenial-20191214-003634+0000) ...  
Setting up ros-kinetic-gazebo-plugins (2.5.19-1xenial-20200109-025835+0000) ...  
Setting up ros-kinetic-gazebo-ros (2.5.19-1xenial-20200109-034212+0000) ...  
Setting up ros-kinetic-gazebo-ros-pkgs (2.5.19-1xenial-20200109-045116+0000) ...  
Setting up ros-kinetic-simulators (1.3.2-0xenial-20200312-222953+0000) ...  
Setting up ros-kinetic-desktop-full (1.3.2-0xenial-20200312-223631+0000) ...  
Processing triggers for desktop-file-utils (0.22-1ubuntu5.2) ...  
Processing triggers for bamfdaemon (0.5.3~bZR0+16.04.20180209-0ubuntu1) ...  
Rebuilding /usr/share/applications/bamf-2.index...  
Processing triggers for shared-mime-info (1.5-2ubuntu0.2) ...  
Processing triggers for fontconfig (2.11.94-0ubuntu1.1) ...  
Processing triggers for initramfs-tools (0.122ubuntu8.14) ...  
update-initramfs: Generating /boot/initrd.img-4.15.0-45-generic  
Processing triggers for libc-bin (2.23-0ubuntu11) ...  
Processing triggers for systemd (229-4ubuntu21.16) ...  
Processing triggers for ureadahead (0.100.0-19) ...  
sungmin@ubuntu:~$
```



ROS 설치 과정 (4)

- rosdep 초기화

```
$ sudo rosdep init
```

```
$ rosdep update
```

```
sungmin@ubuntu: ~  
sungmin@ubuntu:~$ sudo rosdep init  
Wrote /etc/ros/rosdep/sources.list.d/20-default.list  
Recommended: please run  
  
    rosdep update  
  
sungmin@ubuntu:~$ rosdep update  
reading in sources list data from /etc/ros/rosdep/sources.list.d  
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.y  
aml  
  
.  
.  
.  
  
Skip end-of-life distro "indigo"  
Skip end-of-life distro "jade"  
Add distro "kinetic"  
Skip end-of-life distro "lunar"  
Add distro "melodic"  
Add distro "noetic"  
updated cache in /home/sungmin/.ros/rosdep/sources.cache  
sungmin@ubuntu:~$
```

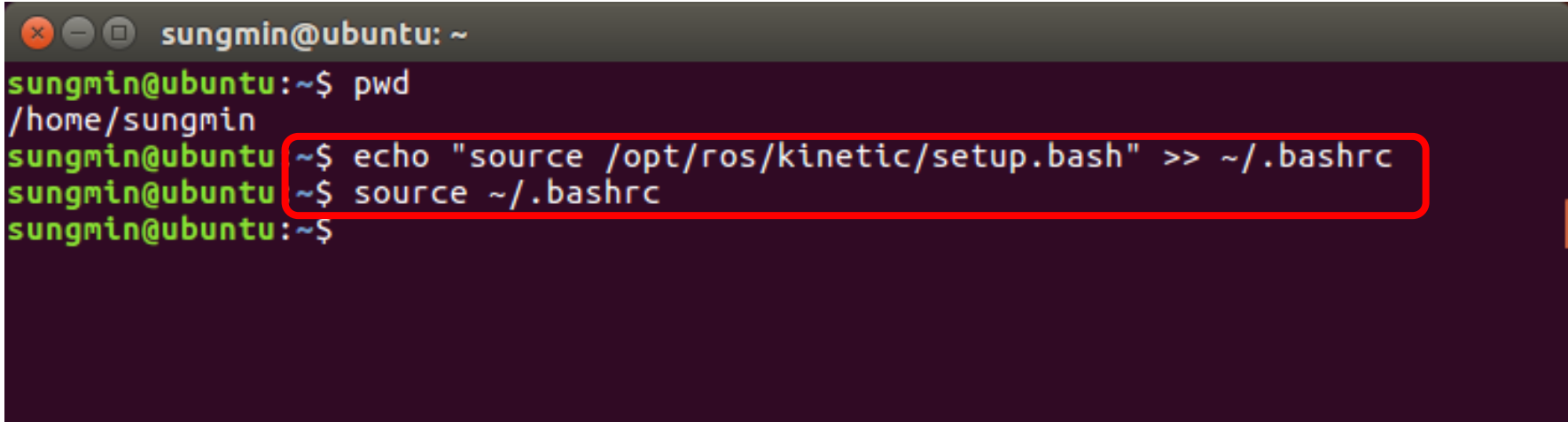


ROS 설치 과정 (5)

- 쉘 환경 설정

```
$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
```

```
$ source ~/.bashrc
```

A terminal window titled 'sungmin@ubuntu: ~' with standard Ubuntu window controls. The terminal shows the following commands and output:

```
sungmin@ubuntu:~$ pwd
/home/sungmin
sungmin@ubuntu ~$ echo "source /opt/ros/kinetic/setup.bash" >> ~/.bashrc
sungmin@ubuntu ~$ source ~/.bashrc
sungmin@ubuntu:~$
```

The two lines of the command sequence are enclosed in a red rectangular box.

ROS 설치 과정 (6-1)

- 추가로 필요한 도구 등 설치

```
$ sudo apt install python-rosinstall python-rosinstall-generator python-wstool build-essential
```

```
sungmin@ubuntu: ~  
bzip2 git git-core git-man liberror-perl libjs-excanvas libserf-1-1 libsvn1  
mercurial mercurial-common python-bzrlib python-configobj python-dbus  
python-gi python-gpgme python-httplib2 python-keyring python-launchpadlib  
python-lazr.restfulclient python-lazr.uri python-oauth python-secretstorage  
python-simplejson python-vcstools python-wadllib subversion  
Suggested packages:  
bzip2-doc bzrtools python-bzrlib.tests git-daemon-run | git-daemon-sysvinit  
git-doc git-el git-email git-gui gitk gitweb git-arch git-cvs git-mediawiki  
git-svn qct kdiff3 | kdiff3-qt | kompare | meld | tkcvs | mgdiff  
python-mysqldb python-bzrlib-dbg python-kerberos python-pycurl  
python-configobj-doc python-dbus-doc python-dbus-dbg python-gi-cairo  
python-fs python-gdata python-kde4 python-keyczar python-testresources  
python-secretstorage-doc db5.3-util subversion-tools  
The following NEW packages will be installed:  
bzip2 git git-core git-man liberror-perl libjs-excanvas libserf-1-1 libsvn1  
mercurial mercurial-common python-bzrlib python-configobj python-dbus  
python-gi python-gpgme python-httplib2 python-keyring python-launchpadlib  
python-lazr.restfulclient python-lazr.uri python-oauth python-rosinstall  
python-rosinstall-generator python-secretstorage python-simplejson  
python-vcstools python-wadllib python-wstool subversion  
0 upgraded, 29 newly installed, 0 to remove and 270 not upgraded.  
Need to get 9,539 kB of archives.  
After this operation, 54.0 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```



ROS 설치 과정 (6-2)

- 추가로 필요한 도구 등 설치

```
sungmin@ubuntu: ~  
Creating config file /etc/mercurial/hgrc.d/hgext.rc with new version  
Setting up python-dbus (1.2.0-3) ...  
Remove stale byte-compiled files...  
Setting up python-gi (3.20.0-0ubuntu1) ...  
Setting up python-gpgme (0.3-1.1) ...  
Setting up python-httplib2 (0.9.1+dfsg-1) ...  
Setting up python-keyring (7.3-1ubuntu1) ...  
Setting up python-lazr.uri (1.0.3-2build1) ...  
Setting up python-simplejson (3.8.1-1ubuntu2) ...  
Setting up python-wadllib (1.3.2-3ubuntu0.16.04.1) ...  
Setting up python-oauth (1.0.1-5) ...  
Setting up python-lazr.restfulclient (0.13.4-5ubuntu2) ...  
Setting up python-launchpadlib (1.10.3-3ubuntu0.1) ...  
Setting up subversion (1.9.3-2ubuntu1.3) ...  
Setting up python-vcstools (0.1.42-1) ...  
Setting up python-wstool (0.1.17-1) ...  
Setting up python-rosinstall (0.7.8-1) ...  
Setting up python-rosinstall-generator (0.1.18-1) ...  
Setting up python-secretstorage (2.1.3-1) ...  
Processing triggers for libc-bin (2.23-0ubuntu11) ...  
sungmin@ubuntu:~$
```



ROS 설치 확인 (7)

- 다음 명령을 실행했을 때 아래 이미지와 같은 화면이 나오면 OK

```
$ roscore
```

```
roscore http://ubuntu:11311/
sungmin@ubuntu:~$ roscore
... logging to /home/sungmin/.ros/log/9d8e9cd8-6f58-11ea-8bd6-000c29b83574/ros-launch-ubuntu-38891.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://ubuntu:39115/
ros_comm version 1.12.14

SUMMARY
=====

PARAMETERS
* /rostdistro: kinetic
* /rosversion: 1.12.14

NODES

auto-starting new master
process[master]: started with pid [38902]
ROS_MASTER_URI=http://ubuntu:11311/
```

(위와는 다른 터미널에서 아래 실행)
\$ rosnodetool list

```
sungmin@ubuntu: ~
sungmin@ubuntu:~$ rosnodetool list
/roscout
sungmin@ubuntu:~$
```

(주의) 나중에 여기에서 Ctrl+C 입력해야 roscore가 종료된다.

실습 (ROS설치)

- 다같이 자신의 리눅스 컴퓨터에 ROS를 설치하시기 바랍니다.



ROS 기초 실습

ROS 프로그래밍을 위한 환경 설정



ROS 워크스페이스 Workspace

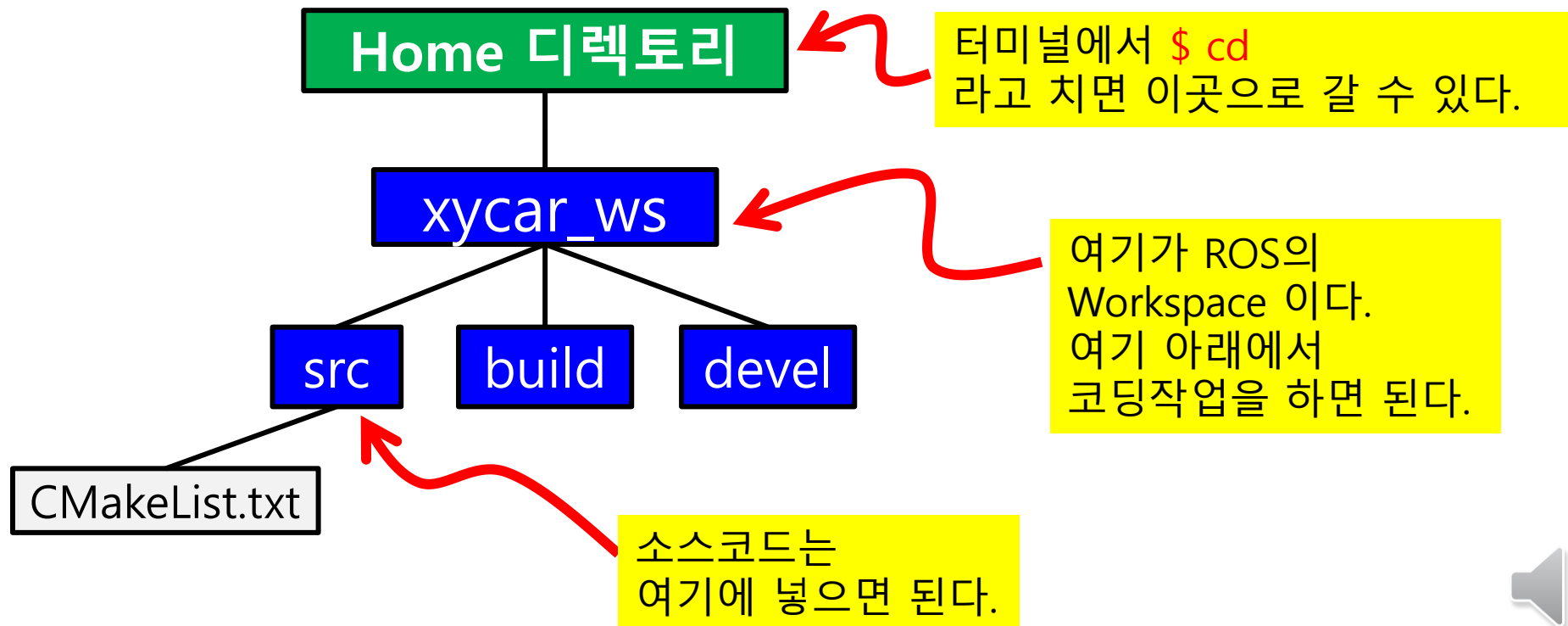
- ROS에서 코딩을 하기 위해서는 Workspace 라는 공간이 필요하다.
 - 다음과 같은 방법으로 'xycar_ws' 라는 이름의 Workspace를 생성한다. (폴더 생성)
 - ▶ `$ cd` → Home 폴더로 이동
 - ▶ `$ mkdir -p ~/xycar_ws/src` → 서브 폴더 생성
 - ▶ `$ cd xycar_ws` → xycar_ws 폴더 아래로 이동
 - ▶ `$ catkin_make` → ROS 코딩 환경 셋업과 정리 (빌드)

```
sungmin@machine: ~/xycar_ws
-- Found Threads: TRUE
-- Using Python nosetests: /usr/bin/nosetests-2.7
-- catkin 0.7.20
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- Configuring done
-- Generating done
-- Build files have been written to: /home/sungmin/xycar_ws/build
####
#### Running command: "make -j2 -l2" in "/home/sungmin/xycar_ws/build"
####
sungmin@machine:~/xycar_ws$
```



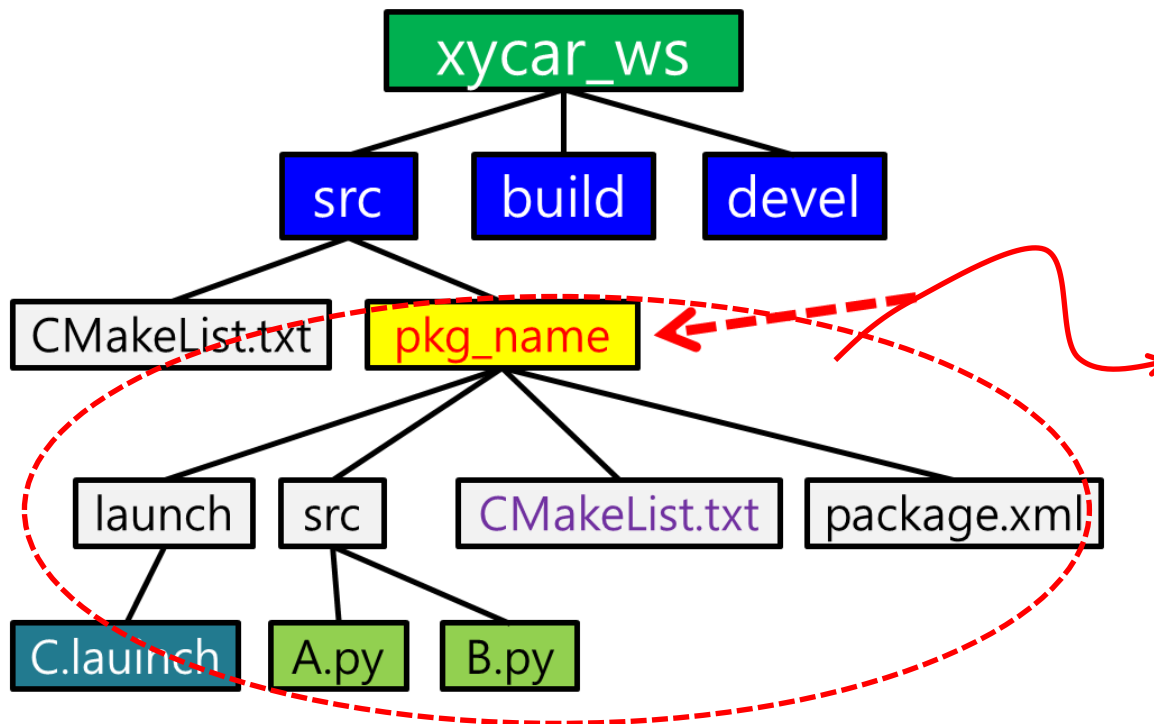
ROS 워크스페이스 Workspace

- 앞서 작업이 끝나면 아래와 같은 구조의 Workspace가 만들어짐
 - Workspace의 이름은 'xycar_ws'
 - ROS 프로그래밍 작업은 Workspace에서 진행함.
 - 소스코드는 /src 폴더 안에 만들면 됨.



빌드 명령 : catkin_make

- ROS의 Workspace에서 새로운 소스코드 파일이나 패키지가 만들어지면
 - catkin_make 명령을 통해 빌드(build) 작업을 진행한다.
 - ROS 프로그래밍 작업과 관련 있는 모든 것들을 깔끔하게 정리해서
 - 최신 상태로 만드는 작업이라고 생각하면 됨



여기에 뭔가 수정이
발생하면
catkin_make 명령으로
빌드한다.

ROS 작업환경 설정

- ROS 작업에 필요한 환경변수 설정
 - 홈 디렉토리에 있는 .bashrc 파일을 수정
 - ▶ \$ cd (홈디렉토리로 이동)
 - ▶ \$ sudo gedit ~/.bashrc (아래 내용 추가)
 - ▶ \$ source .bashrc (수정한 내용을 시스템에 반영)

.bashrc 파일의 내용

...

```
alias cm='cd ~/xycar_ws && catkin_make'
```

```
source /opt/ros/kinetic/setup.bash
```

```
source ~/xycar_ws/devel/setup.bash
```

```
export ROS_MASTER_URI=http://localhost:11311
```

```
export ROS_HOSTNAME=localhost
```



ROS 작업환경 설정

- ROS 작업에 필요한 환경변수 설정 확인

- `$ printenv | grep ROS`

→ 이 명령으로 ROS 환경변수가 어떻게 설정되었는지 확인할 수 있음

```
sungmin@machine: ~  
sungmin@machine:~$ printenv | grep ROS  
ROS_ROOT=/opt/ros/kinetic/share/ros  
ROS_PACKAGE_PATH=/home/sungmin/xy-car_ws/src:/home/sungmin/catkin_ws/src:/opt/ros/kinetic/share  
ROS_MASTER_URI=http://localhost:11311  
ROS_PYTHON_VERSION=2  
ROS_VERSION=1  
ROS_HOSTNAME=localhost  
ROSLISP_PACKAGE_DIRECTORIES=/home/sungmin/xy-car_ws/devel/share/common-lisp:/home/sungmin/catkin_ws/devel/share/common-lisp  
ROS_DISTRO=kinetic  
ROS_ETC_DIR=/opt/ros/kinetic/etc/ros  
sungmin@machine:~$
```



실습 (ROS 작업환경 설정)

- 다 같이 ROS 작업에 필요한 환경을 설정합니다



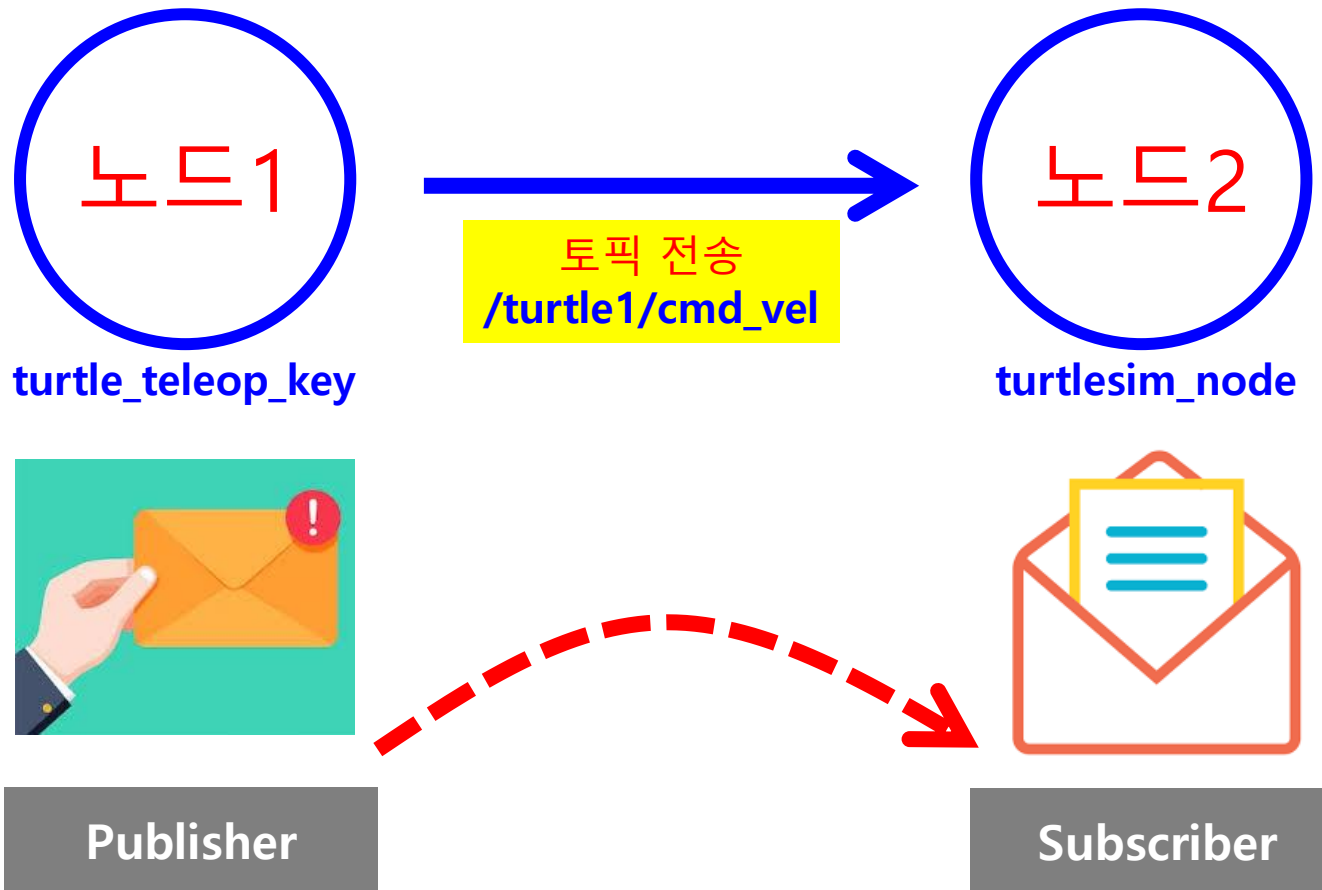


ROS의 기초

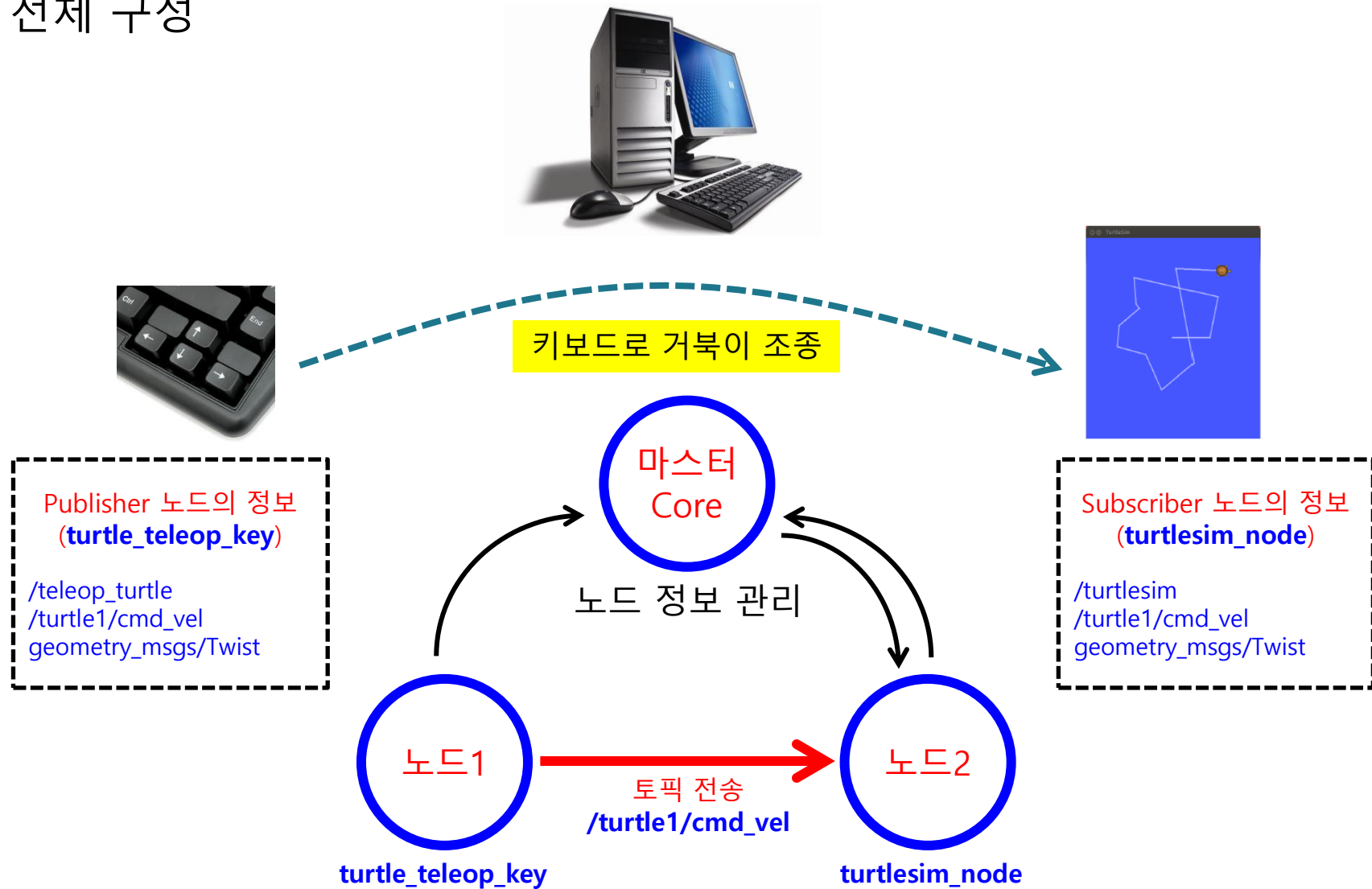
ROS 예제 프로그램 구동 실습



- 전체 구성



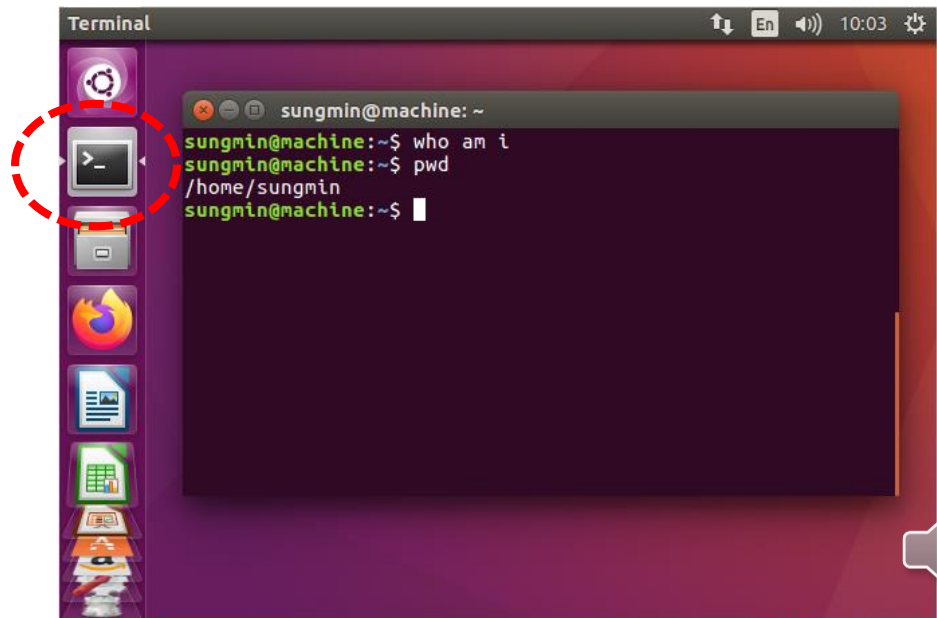
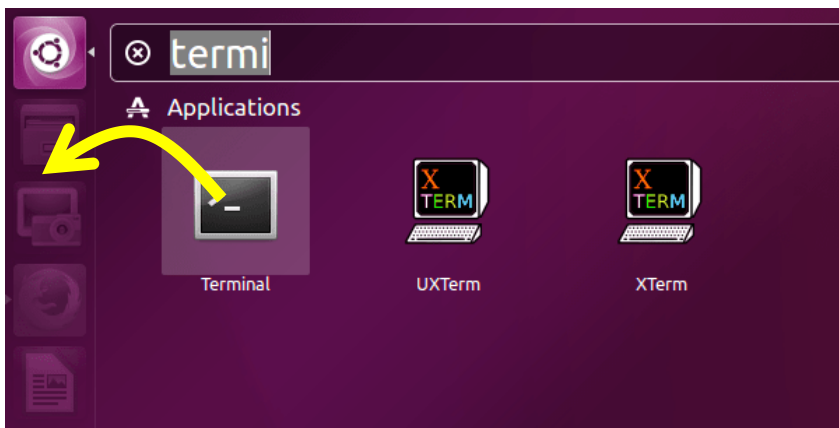
- 전체 구성



우분투 터미널 사용법

- 터미널

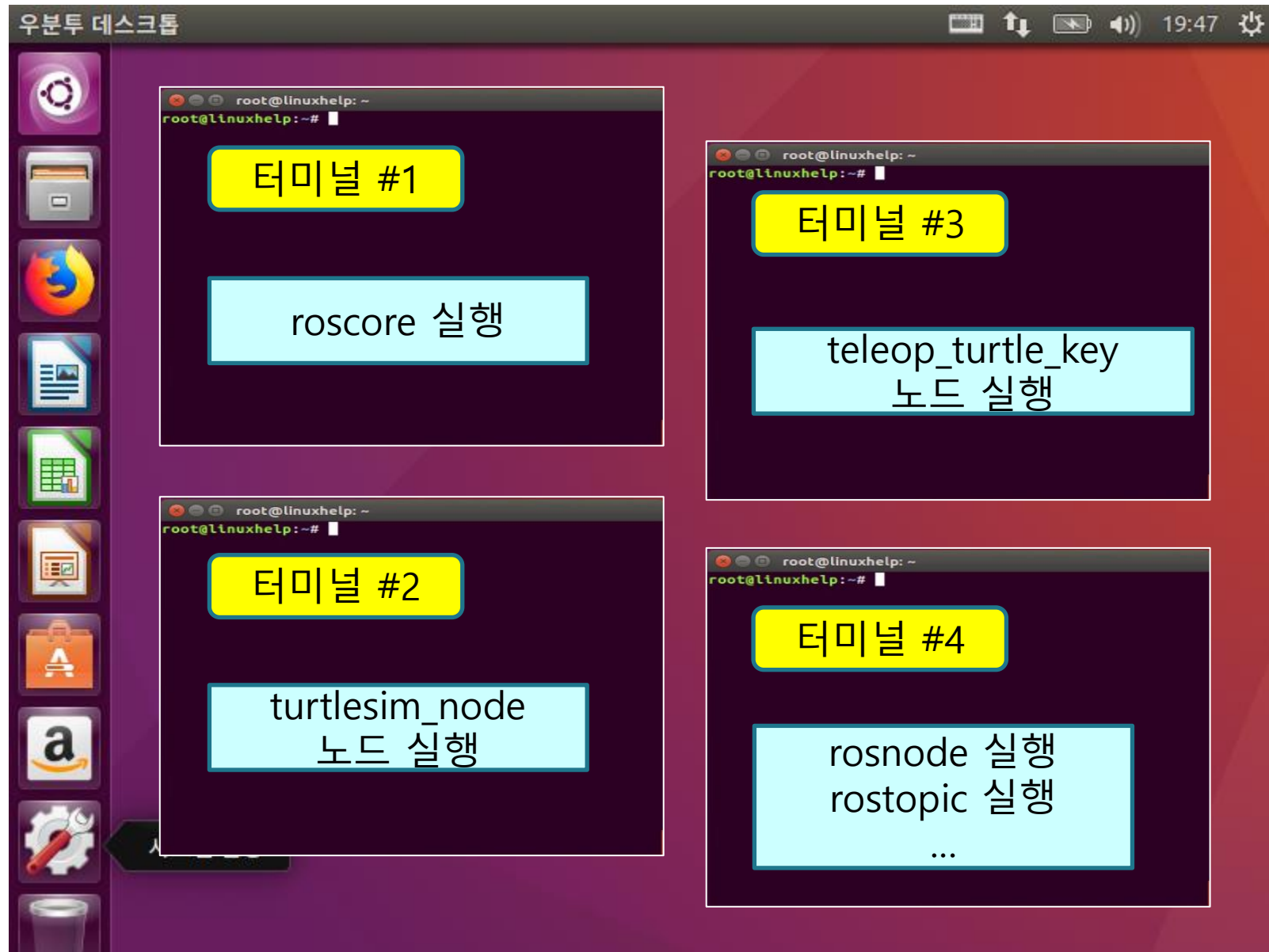
- 명령 실행을 위한 윈도우 창 – 터미널 창 각각이 별개의 컴퓨터라고 생각해도 됨.
 - ▶ 맨 위에 있는 아이콘 클릭 후 termi 입력하면 Terminal 아이콘이 나옴.
 - ▶ 이걸 마우스를 끌어다가 왼쪽 아이콘 리스트에 삽입함.
 - ▶ 이후에 터미널 아이콘 클릭하면 터미널을 하나 열 수 있음.
 - ▶ 마우스의 오른쪽 버튼을 클릭하면 새로운 터미널을 추가로 열 수 있음.



(실습) 우분투 터미널 실습



터미널을 여러 개 열어서 작업하기



ROS Core 실행

- (1) 마스터(roscore)의 실행

- (터미널1 열어서)
- `$ roscore`

```
SUMMARY
=====

PARAMETERS
* /roscpp: kinetic
* /rosversion: 1.12.14

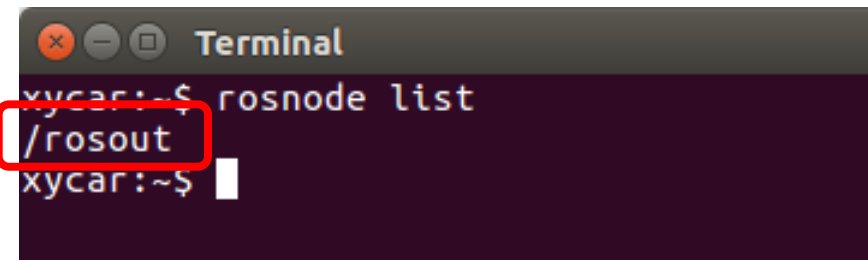
NODES

auto-starting new master
process[master]: started with pid [17150]
ROS_MASTER_URI=http://kmusw-ThinkPad-T450:11311/

setting /run_id to d3de1266-c957-11e9-9641-5ce0c558eccc
process[rosout-1]: started with pid [17163]
started core service [/rosout]
```

- (2) ROS node의 확인

- (터미널4 열어서)
- `$ roscpp list`
- 어떤 node가 동작중인지 볼 수 있음



```
Terminal
xycar:~$ roscpp list
/roscpp
xycar:~$
```

ROS 노드 실행 - 토픽을 받아서 거북이를 이동시킴

- (터미널2 하나 열어서)
 - `$ rosrun turtlesim turtlesim_node`

```
Terminal
xycar:~$ rosrun turtlesim turtlesim_node
[ INFO] [1594178446.611221465]: Starting turtlesim with node name /turtlesim
[ INFO] [1594178446.630118554]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
```

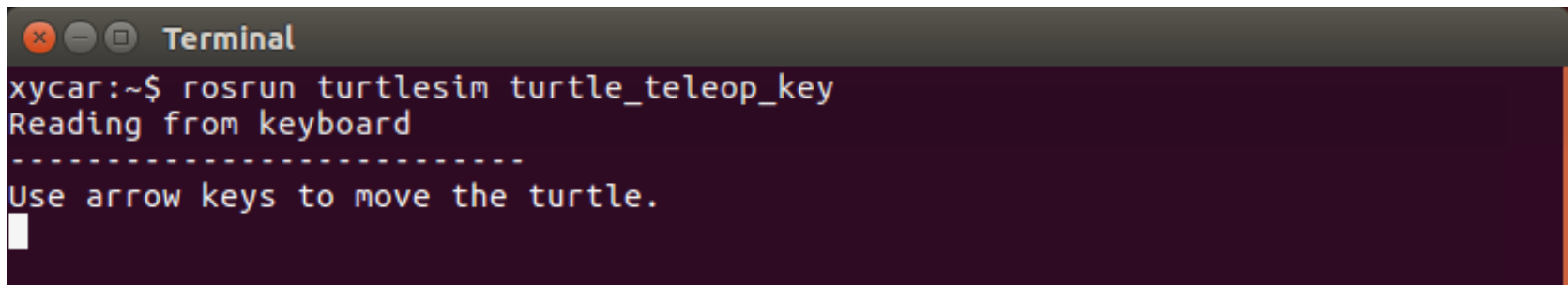
- (터미널4에서)
 - `$ roscall turtlesim turtlesim_node`

```
Terminal
xycar:~$ roscall turtlesim turtlesim_node
/roscall
xycar:~$ roscall turtlesim turtlesim_node
/roscall
/turtlesim
```



ROS 노드 실행 - 사용자 입력에 맞춰 토픽을 발행함

- roscore와 turtlesim_node가 실행중인 상태에서, 터미널3을 열어
 - `$ rosrun turtlesim turtle_teleop_key`



```
Terminal
xycar:~$ rosrun turtlesim turtle_teleop_key
Reading from keyboard
-----
Use arrow keys to move the turtle.
█
```

- 터미널3 안에서 키보드 방향키를 누르면 (상/하/좌/우 화살표 키)
 - 키보드 입력값을 토픽에 담아서 보내는 일을 함
 - turtle이 그에 따라 움직임을 확인



노드 사이에서 토픽(topic) 주고 받기

- (터미널4을 열어)
 - \$ rosnodet list (어떤 노드들이 돌고 있는지 살펴보자)

```
Terminal
xycar:~$ rosnodet list
/rosout
xycar:~$ rosnodet list
/rosout
/turtlesim
xycar:~$ rosnodet list
/rosout
/teleop_turtle
/turtlesim
```



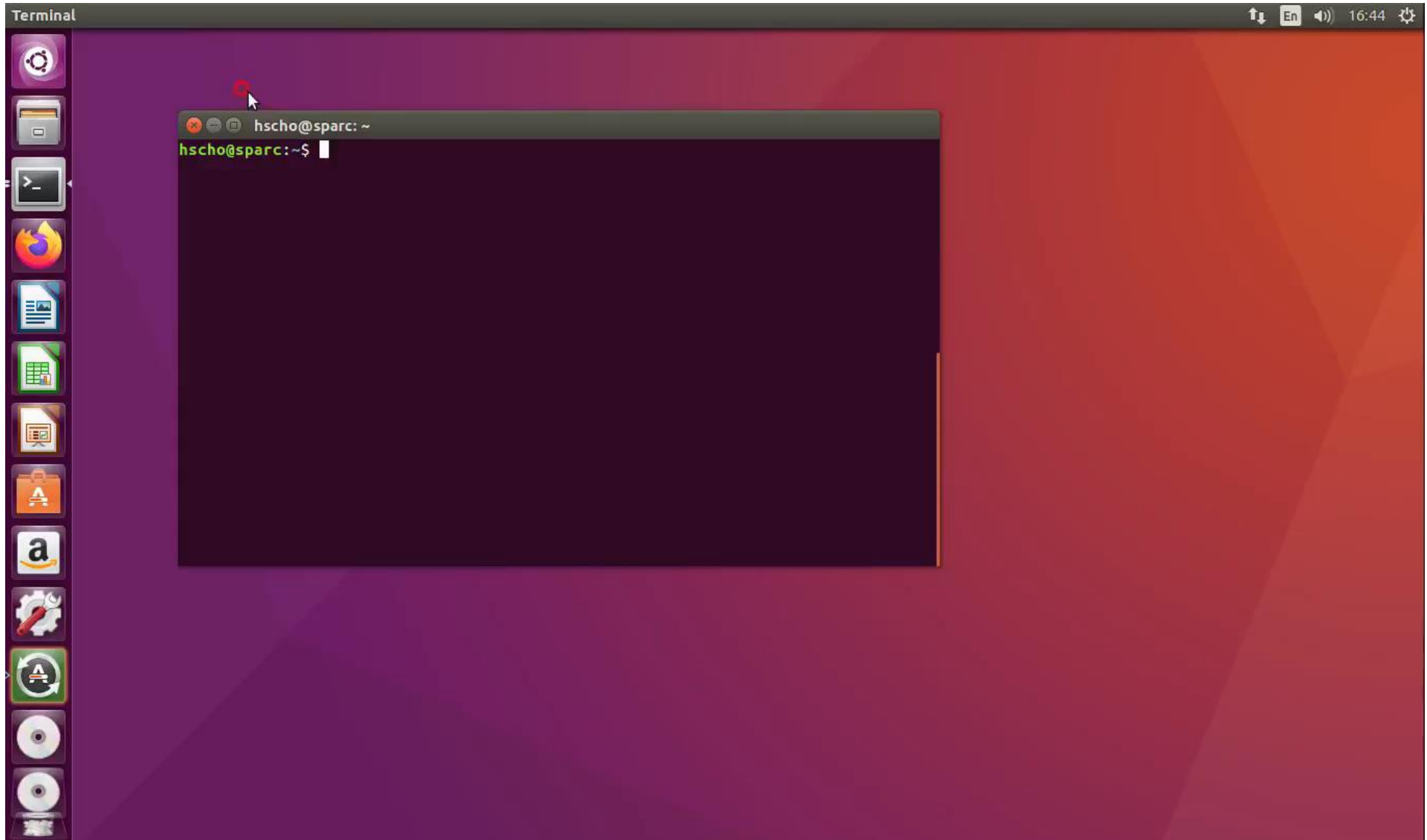
노드 /teleop_turtle는
토픽을 발행한다.
(사용자의 키보드 입력에 따라)



노드 /turtlesim은
위 토픽을 구독하여
키보드 입력값을 알아내고
이에 따라 turtle을 이동시킴



실습 (노드 만들기)



토픽을 조사해 보자

- (터미널4을 열어)
 - `$ rostopic list`
 - ▶ 어떤 토픽이 날아다니는지 살펴볼 수 있음

```
xycar:~$ rostopic list
/rosout
/rosout_agg
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
```

- `$ rostopic echo /turtle1/cmd_vel` (어떤 값인지?)
 - ▶ 토픽에 담긴 메시지의 내용을 볼 수 있음

키보드 입력으로
turtle을 움직여 보면



```
linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
linear:
  x: 0.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: -2.0
---
```



노드와 토픽 관계를 시각화

- (터미널4을 열어)
 - \$ rqt_graph



Publisher node

Subscriber node



토픽을 좀 더 자세히 살펴보면

- 토픽을 찾아 보자
 - `$ rostopic list -v`

```
Terminal
xycar:~$ rostopic list -v

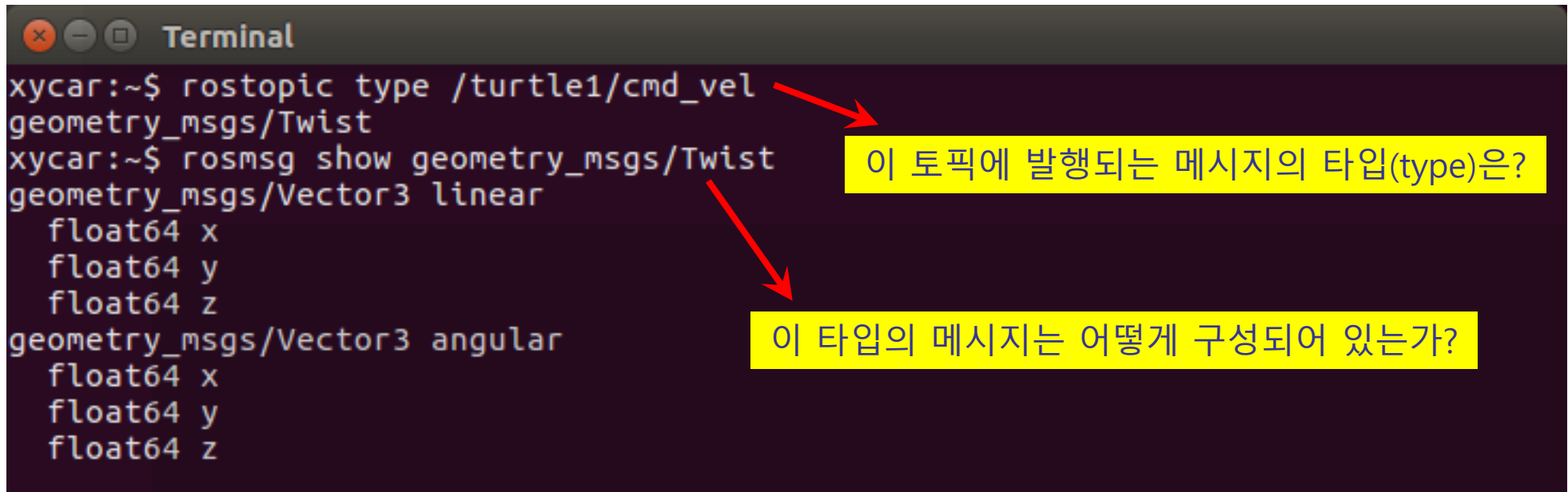
Published topics:
* /turtle1/color_sensor [turtlesim/Color] 1 publisher
* /turtle1/cmd_vel [geometry_msgs/Twist] 1 publisher
* /rosout [rosgraph_msgs/Log] 2 publishers
* /rosout_agg [rosgraph_msgs/Log] 1 publisher
* /turtle1/pose [turtlesim/Pose] 1 publisher

Subscribed topics:
* /turtle1/cmd_vel [geometry_msgs/Twist] 1 subscriber
* /rosout [rosgraph_msgs/Log] 1 subscriber
```



토픽을 좀 더 자세히 살펴보면

- 메시지의 타입과 구성을 살펴보자
 - `$ rostopic type /turtle1/cmd_vel`
 - `$ rosmmsg show geometry_msgs/Twist`



```
Terminal
xycar:~$ rostopic type /turtle1/cmd_vel
geometry_msgs/Twist
xycar:~$ rosmmsg show geometry_msgs/Twist
geometry_msgs/Vector3 linear
  float64 x
  float64 y
  float64 z
geometry_msgs/Vector3 angular
  float64 x
  float64 y
  float64 z
```

이 토픽에 발행되는 메시지의 타입(type)은?

이 타입의 메시지는 어떻게 구성되어 있는가?

ROS 토픽의 메시지가 궁금할 때...

- <https://wiki.ros.org/>

The screenshot shows the ROS.org search results page. The search bar at the top right contains the text 'Twist' and a 'Submit' button. Below the search bar, the 'Search Results' section shows 'About 9,600 results (0.25 seconds)'. The first result is 'geometry_msgs/Twist Documentation' with a link to 'docs.ros.org/melodic/api/geometry_msgs/html/msg/Twist.html'. A red arrow points from this result to a larger, detailed view of the 'geometry_msgs/Twist Message' on the right.

geometry_msgs/Twist Message

File: `geometry_msgs/Twist.msg`

Raw Message Definition

```
# This expresses velocity in free space broken into its linear and angular parts.  
Vector3 linear  
Vector3 angular
```

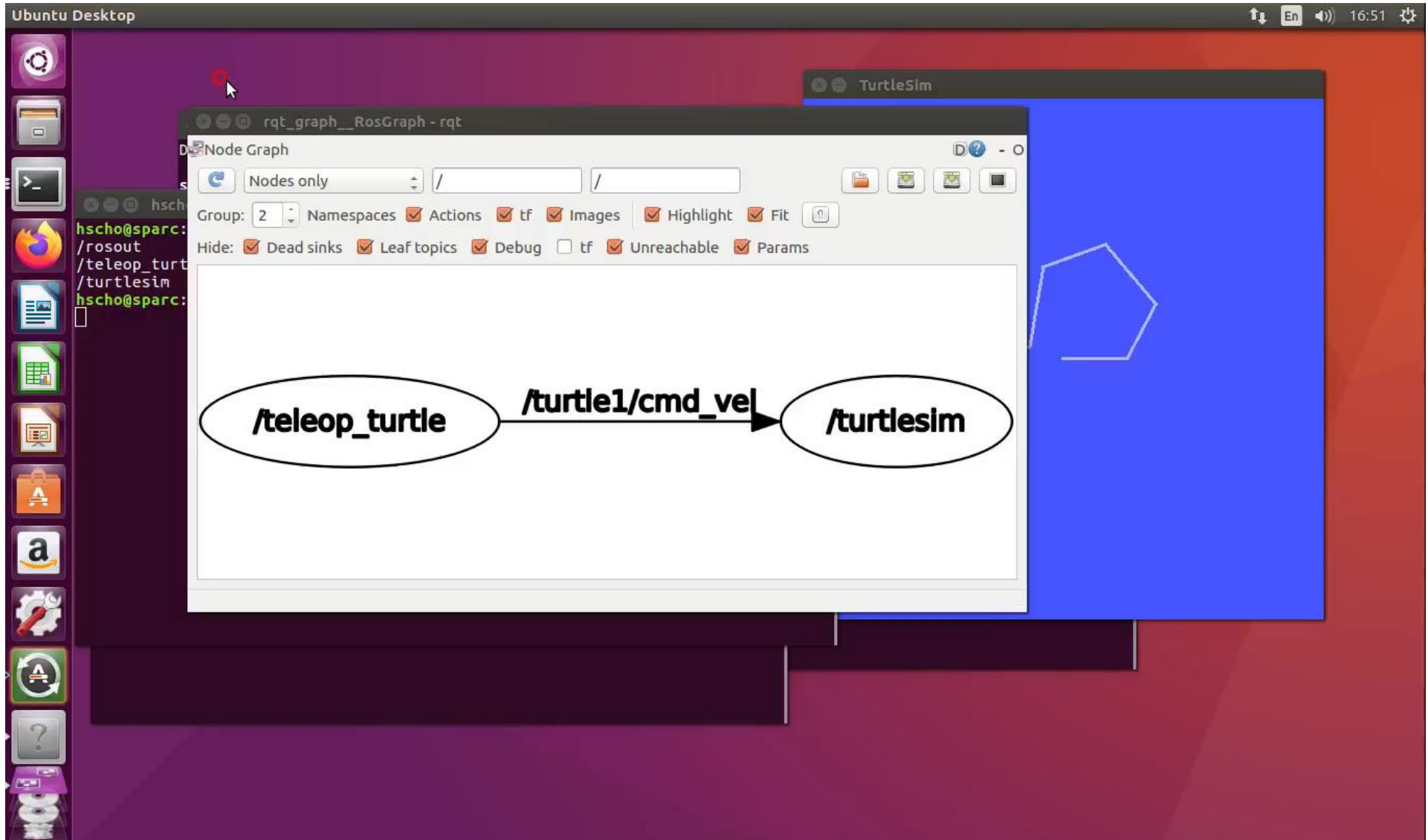
Compact Message Definition

```
geometry_msgs/Vector3 linear  
geometry_msgs/Vector3 angular
```

autogenerated on Sat, 15 Aug 2020 03:19:46



실습 (토픽 살펴보기)



토픽을 직접 발행해 보자

- 터미널에서 아래와 같이 입력한다.

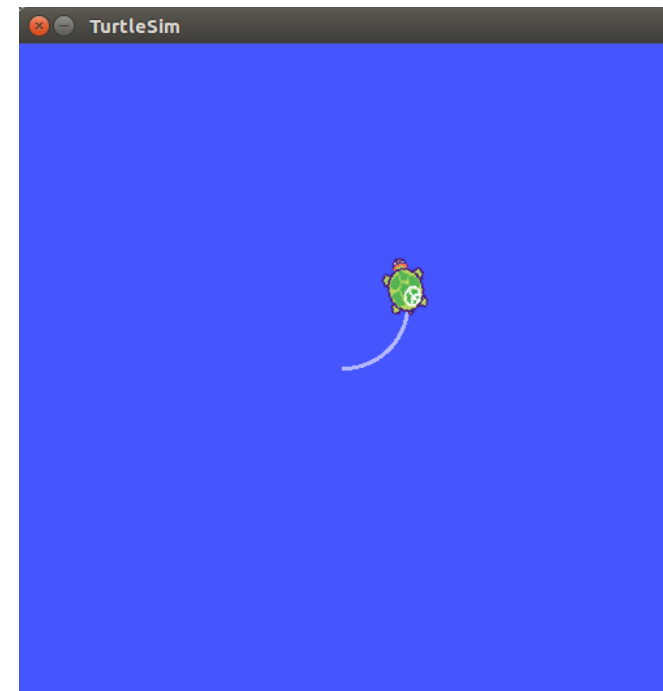
```
$ rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]'
```

발행은
한 번만

토픽

메시지
타입

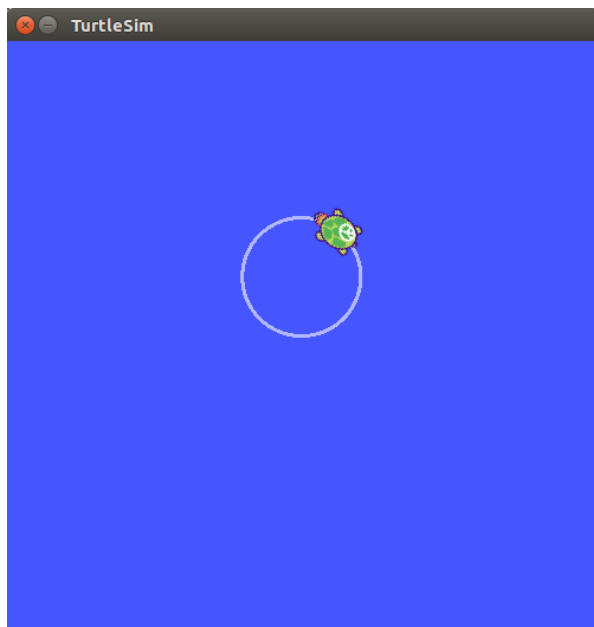
메시지 내용



주기적으로 반복 발행되는 메시지

- 터미널에서 아래와 같이 입력한다.

```
$ rostopic pub /turtle1/cmd_vel geometry_msgs/Twist -r 1 -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]'
```



발행 주기는 1Hz → 1초에 한 번씩

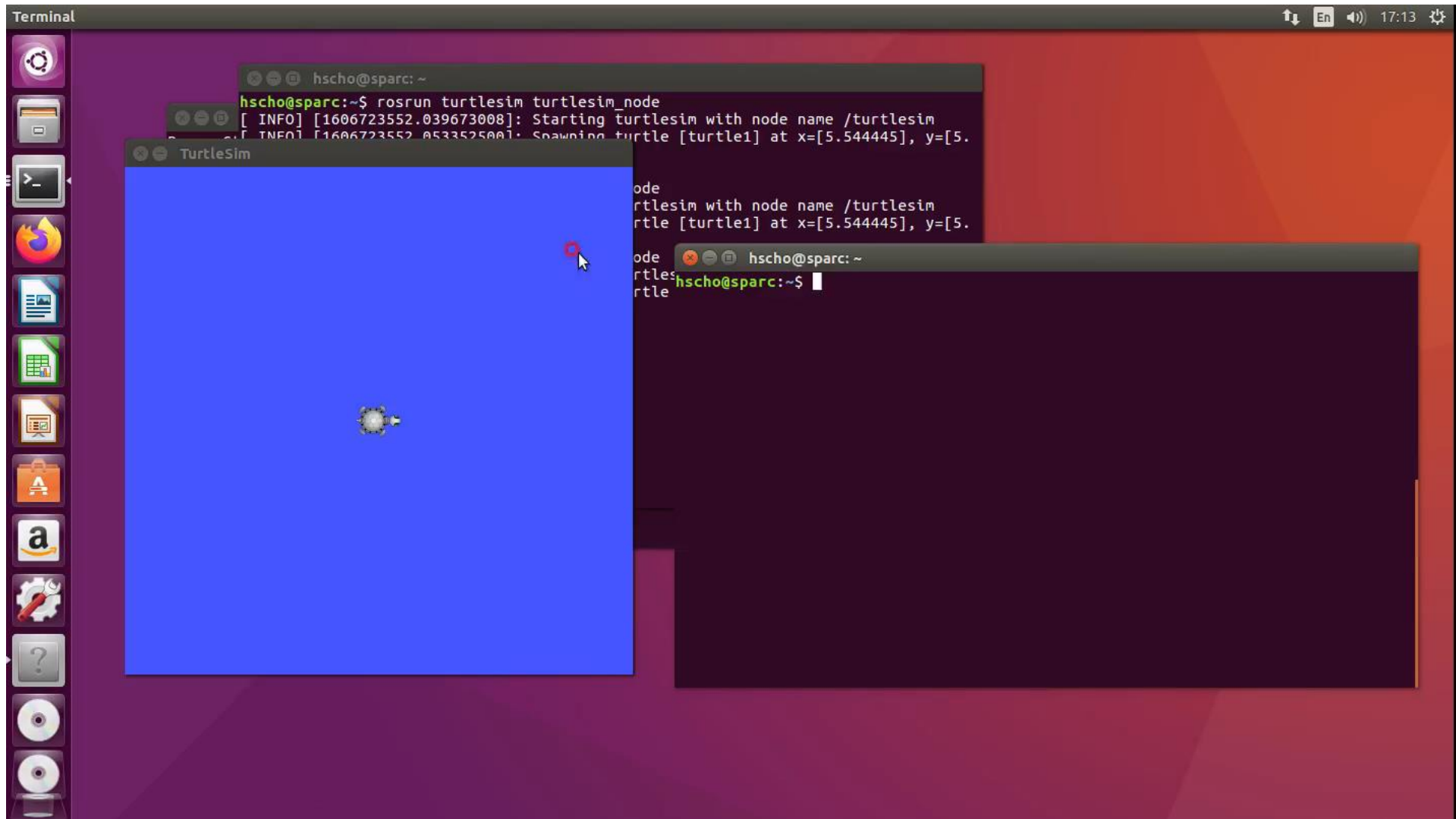
이 창에서 Ctrl-C 눌러서
프로그램 종료하면
turtle의 움직임도 멈춘다
(메시지 발행이 중단되므로)

A screenshot of a terminal window with the title 'Terminal'. It shows the command: `xycar:~$ rostopic pub /turtle1/cmd_vel geometry_msgs/Twist -r 1 -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]'` followed by a cursor. A red arrow points from the yellow text box above to the terminal window.

```
xycar:~$ rostopic pub /turtle1/cmd_vel geometry_msgs/Twist -r 1 -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, 1.8]'
```



실습 (토픽을 직접 발행)





Q&A

감사합니다.



(주)자이트론

허성민

smher@xytron.co.kr

