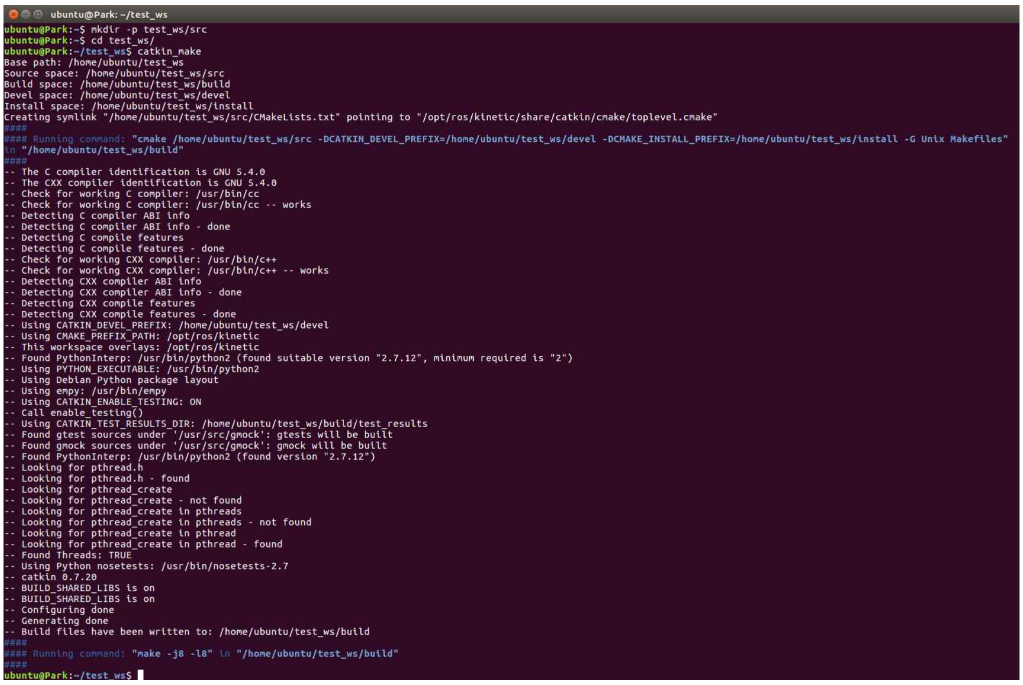
ROS Package, Node, Topic

학과 : 산업인공지능학과
학번 : 2020254016
이름 : 박 민 우

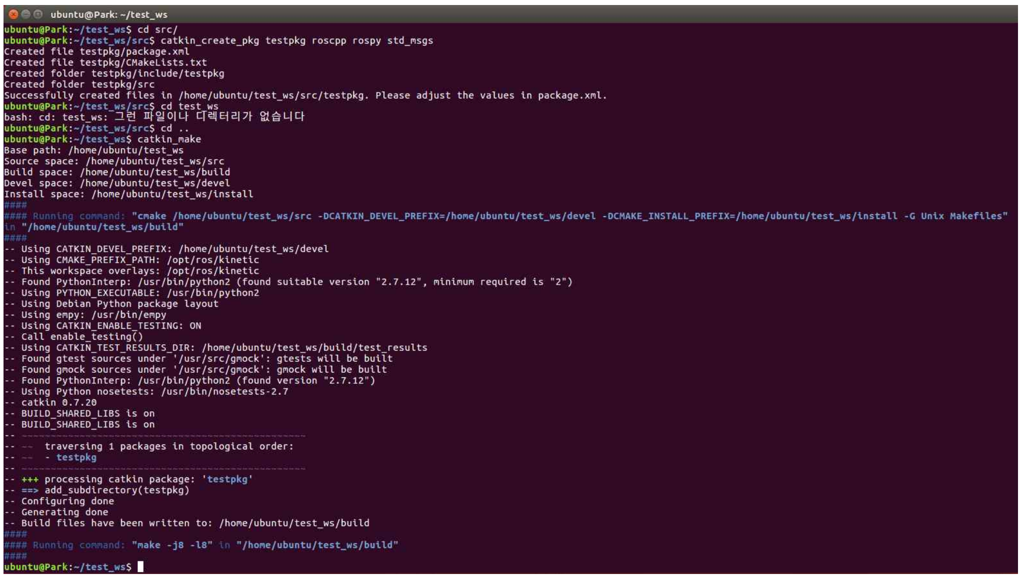
1. ROS Package

1) Workspace 생성



```
ubuntu@Park:~$ mkdir -p test_ws/src
ubuntu@Park:~$ cd test_ws/
ubuntu@Park:~/test_ws$ catkin_make
Base path: /home/ubuntu/test_ws
Source space: /home/ubuntu/test_ws/src
Build space: /home/ubuntu/test_ws/build
Devel space: /home/ubuntu/test_ws/devel
Install space: /home/ubuntu/test_ws/install
Creating symlink "/home/ubuntu/test_ws/src/CMakeLists.txt" pointing to "/opt/ros/kinetic/share/catkin/cmake/toplevel.cmake"
####
#### Running command: "cmake /home/ubuntu/test_ws/src -DCATKIN_DEVEL_PREFIX=/home/ubuntu/test_ws/devel -DCMAKE_INSTALL_PREFIX=/home/ubuntu/test_ws/install -G Unix Makefiles"
in "/home/ubuntu/test_ws/build"
####
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /home/ubuntu/test_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/kinetic
-- This workspace overlays: /opt/ros/kinetic
-- Found PythonInterp: /usr/bin/python2 (found suitable version "2.7.12", minimum required is "2")
-- Using PYTHON_EXECUTABLE: /usr/bin/python2
-- Using Debian Python package layout
-- Using empy: /usr/bin/empy
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/ubuntu/test_ws/build/test_results
-- Found gtest sources under '/usr/src/gmock': gtests will be built
-- Found gmock sources under '/usr/src/gmock': gmock will be built
-- Found PythonInterp: /usr/bin/python2 (found version "2.7.12")
-- Looking for pthread.h
-- Looking for pthread.h - found
-- Looking for pthread_create
-- Looking for pthread_create - not found
-- Looking for pthread_create in pthreads
-- Looking for pthread_create in pthreads - not found
-- Looking for pthread_create in pthread
-- Looking for pthread_create in pthread - found
-- 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/ubuntu/test_ws/build
####
#### Running command: "make -j8 -l8" in "/home/ubuntu/test_ws/build"
####
ubuntu@Park:~/test_ws$ 
```

2) package 생성 후 catkin_make로 빌드

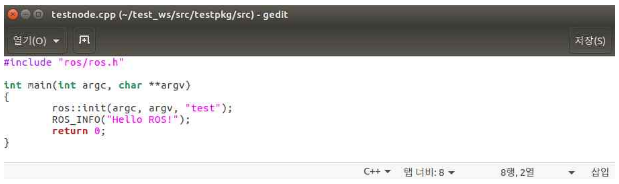


```
ubuntu@Park:~/test_ws$ cd src/
ubuntu@Park:~/test_ws/src$ catkin_create_pkg testpkg roscpp rospy std_msgs
Created file testpkg/package.xml
Created file testpkg/CMakeLists.txt
Created folder testpkg/include/testpkg
Created folder testpkg/src
Successfully created files in /home/ubuntu/test_ws/src/testpkg. Please adjust the values in package.xml.
ubuntu@Park:~/test_ws/src$ cd test_ws
bash: cd: test_ws: 그런 파일이나 디렉터리가 없습니다
ubuntu@Park:~/test_ws/src$ cd ..
ubuntu@Park:~/test_ws$ catkin_make
Base path: /home/ubuntu/test_ws
Source space: /home/ubuntu/test_ws/src
Build space: /home/ubuntu/test_ws/build
Devel space: /home/ubuntu/test_ws/devel
Install space: /home/ubuntu/test_ws/install
####
#### Running command: "cmake /home/ubuntu/test_ws/src -DCATKIN_DEVEL_PREFIX=/home/ubuntu/test_ws/devel -DCMAKE_INSTALL_PREFIX=/home/ubuntu/test_ws/install -G Unix Makefiles"
in "/home/ubuntu/test_ws/build"
####
-- Using CATKIN_DEVEL_PREFIX: /home/ubuntu/test_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/kinetic
-- This workspace overlays: /opt/ros/kinetic
-- Found PythonInterp: /usr/bin/python2 (found suitable version "2.7.12", minimum required is "2")
-- Using PYTHON_EXECUTABLE: /usr/bin/python2
-- Using Debian Python package layout
-- Using empy: /usr/bin/empy
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/ubuntu/test_ws/build/test_results
-- Found gtest sources under '/usr/src/gmock': gtests will be built
-- Found gmock sources under '/usr/src/gmock': gmock will be built
-- Found PythonInterp: /usr/bin/python2 (found version "2.7.12")
-- Using Python nosetests: /usr/bin/nosetests-2.7
-- catkin 0.7.20
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-- ~~  traversing 1 packages in topological order:
-- ~~  - testpkg
-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-- +++ processing catkin package: 'testpkg'
-- ==> add_subdirectory(testpkg)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/ubuntu/test_ws/build
####
#### Running command: "make -j8 -l8" in "/home/ubuntu/test_ws/build"
####
ubuntu@Park:~/test_ws$ 
```

2. ROS Node

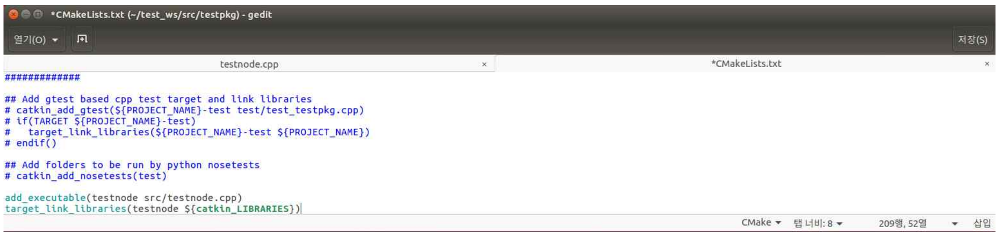
1) Node 추가

- testpkg/src/testnode.cpp에 내용 작성



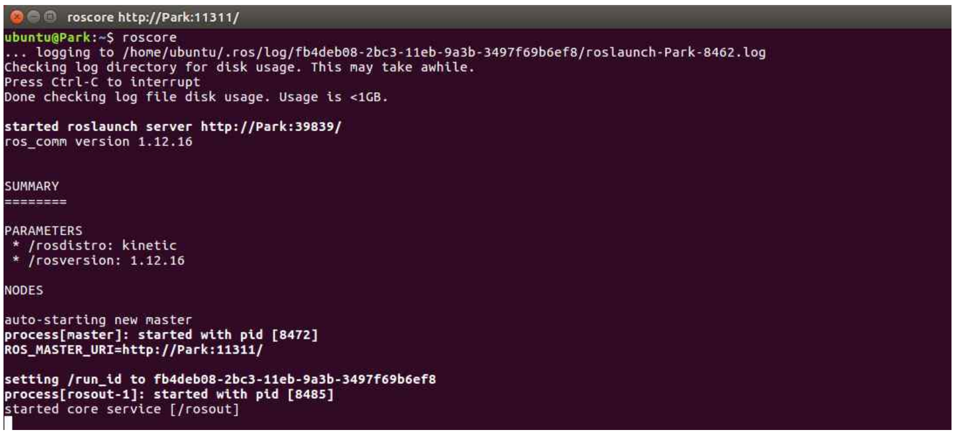
```
#include "ros/ros.h"

int main(int argc, char **argv)
{
        ros::init(argc, argv, "test");
        ROS_INFO("Hello ROS!");
        return 0;
}
```

- testpkg/CMakeList.txt에 내용 추가



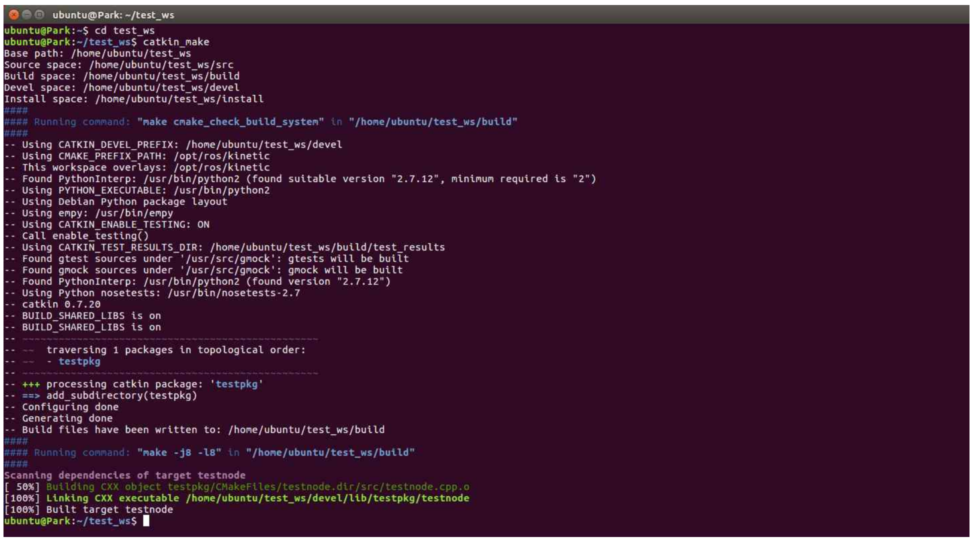
```
#############

## Add gtest based cpp test target and link libraries
# catkin_add_gtest(${PROJECT_NAME}-test test/test_testpkg.cpp)
# if(TARGET ${PROJECT_NAME}-test)
#   target_link_libraries(${PROJECT_NAME}-test ${PROJECT_NAME})
# endif()

## Add folders to be run by python nosetests
# catkin_add_nosetests(test)

add_executable(testnode src/testnode.cpp)
target_link_libraries(testnode ${catkin_LIBRARIES})
```

2) roscore로 master 실행



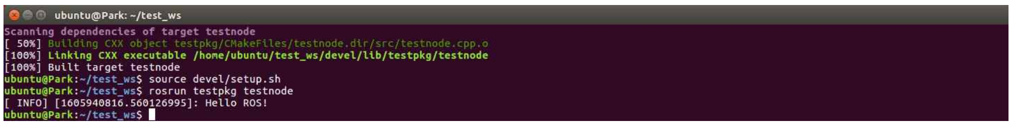
```
ubuntu@Park:~$ roscore
... logging to /home/ubuntu/.ros/log/fb4deb08-2bc3-11eb-9a3b-3497f69b6ef8/roslaunch-Park-8462.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://Park:39839/
ros_comm version 1.12.16


SUMMARY
========

PARAMETERS
 * /rosdistro: kinetic
 * /rosversion: 1.12.16

NODES

auto-starting new master
process[master]: started with pid [8472]
ROS_MASTER_URI=http://Park:11311/

setting /run_id to fb4deb08-2bc3-11eb-9a3b-3497f69b6ef8
process[rosout-1]: started with pid [8485]
started core service [/rosout]
```

3) Workspace에서 빌드



```
ubuntu@Park:~$ cd test_ws
ubuntu@Park:~/test_ws$ catkin_make
Base path: /home/ubuntu/test_ws
Source space: /home/ubuntu/test_ws/src
Build space: /home/ubuntu/test_ws/build
Devel space: /home/ubuntu/test_ws/devel
Install space: /home/ubuntu/test_ws/install
####
#### Running command: "make cmake_check_build_system" in "/home/ubuntu/test_ws/build"
####
-- Using CATKIN_DEVEL_PREFIX: /home/ubuntu/test_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/kinetic
-- This workspace overlays: /opt/ros/kinetic
-- Found PythonInterp: /usr/bin/python2 (found suitable version "2.7.12", minimum required is "2")
-- Using PYTHON_EXECUTABLE: /usr/bin/python2
-- Using Debian Python package layout
-- Using empy: /usr/bin/empy
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/ubuntu/test_ws/build/test_results
-- Found gtest sources under '/usr/src/gmock': gtests will be built
-- Found gmock sources under '/usr/src/gmock': gmock will be built
-- Found PythonInterp: /usr/bin/python2 (found version "2.7.12")
-- Using Python nosetests: /usr/bin/nosetests-2.7
-- catkin 0.7.20
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-- ~~  traversing 1 packages in topological order:
-- ~~  - testpkg
-- ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-- +++ processing catkin package: 'testpkg'
-- ==> add_subdirectory(testpkg)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/ubuntu/test_ws/build
####
#### Running command: "make -j8 -l8" in "/home/ubuntu/test_ws/build"
####
Scanning dependencies of target testnode
[ 50%] Building CXX object testpkg/CMakeFiles/testnode.dir/src/testnode.cpp.o
[100%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/testnode
[100%] Built target testnode
ubuntu@Park:~/test_ws$
```

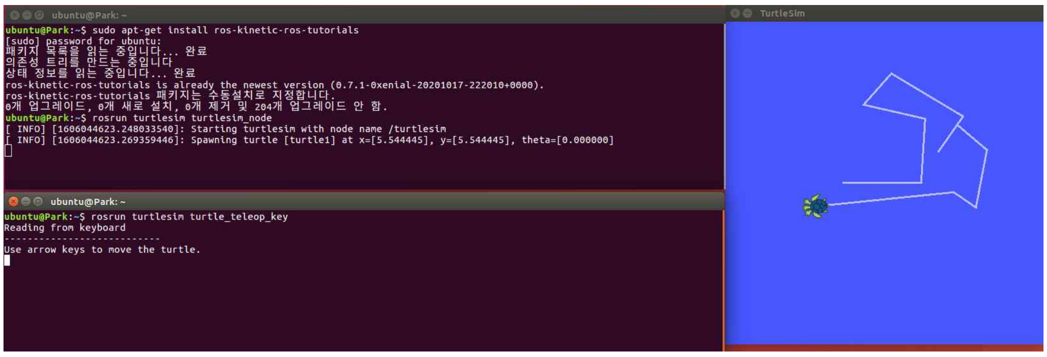
4) 현재 Workspace를 source한 뒤 실행



```
Scanning dependencies of target testnode
[ 50%] Building CXX object testpkg/CMakeFiles/testnode.dir/src/testnode.cpp.o
[100%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/testnode
[100%] Built target testnode
ubuntu@Park:~/test_ws$ source devel/setup.sh
ubuntu@Park:~/test_ws$ rosrun testpkg testnode
[ INFO] [1605940816.560126995]: Hello ROS!
ubuntu@Park:~/test_ws$
```

3. ROS Topic

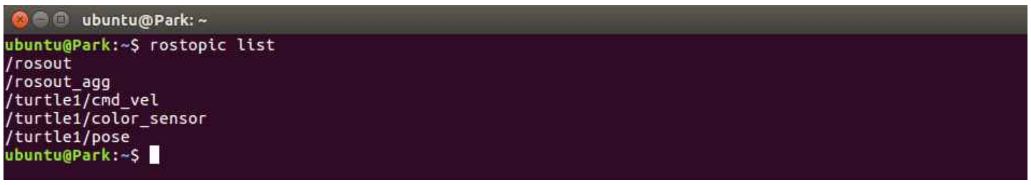
1) turtlesim 설치 및 실행



```
ubuntu@Park:~$ sudo apt-get install ros-kinetic-ros-tutorials
[sudo] password for ubuntu:
패키지 목록을 읽는 중입니다... 완료
의존성 트리를 만드는 중입니다
상태 정보를 읽는 중입니다... 완료
ros-kinetic-ros-tutorials is already the newest version (0.7.1-0xenial-20201017-222010+0000).
ros-kinetic-ros-tutorials 패키지는 수동설치로 지정합니다.
0개 업그레이드, 0개 새로 설치, 0개 제거 및 204개 업그레이드 안 함.
ubuntu@Park:~$ rosrun turtlesim turtlesim_node
[ INFO] [1606044623.248033540]: Starting turtlesim with node name /turtlesim
[ INFO] [1606044623.269359446]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.000000]
```

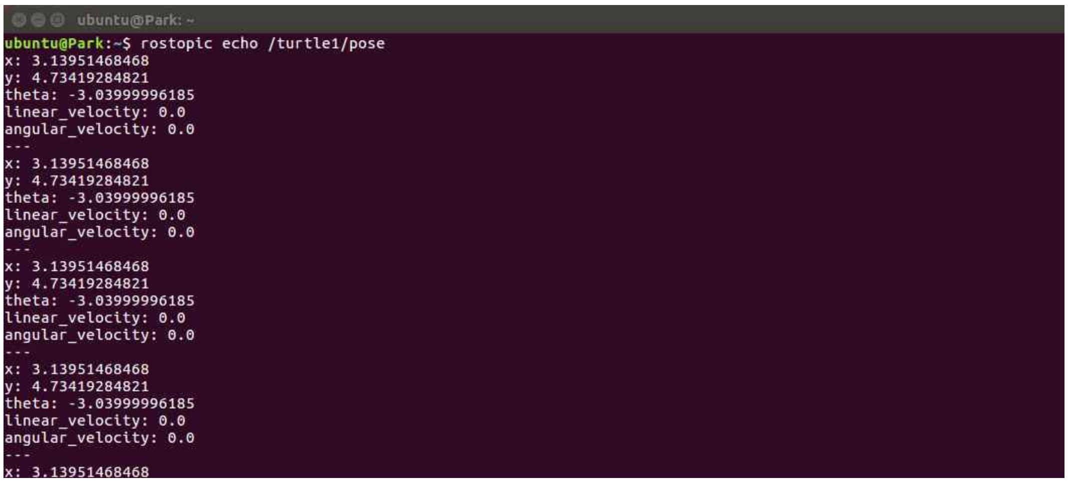
```
ubuntu@Park:~$ rosrun turtlesim turtle_teleop_key
Reading from keyboard
---------------------------
Use arrow keys to move the turtle.
```

2) rostopic list 확인



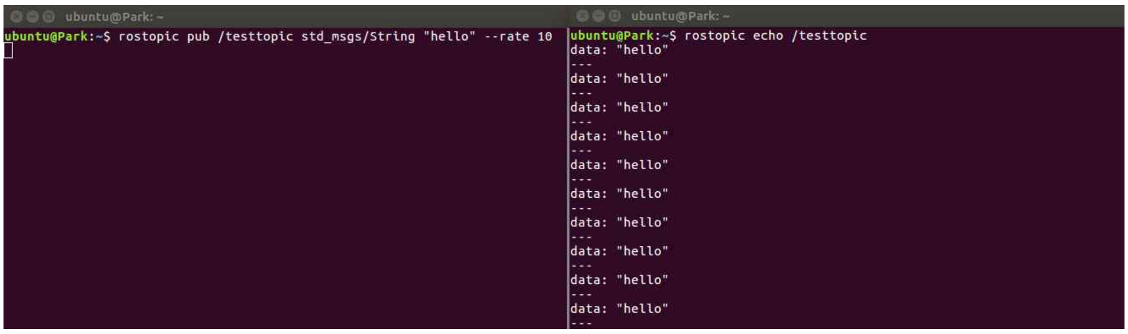
```
ubuntu@Park: ~
ubuntu@Park:~$ rostopic list
/rosout
/rosout_agg
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
ubuntu@Park:~$ 
```

3) rostopic echo



```
ubuntu@Park: ~
ubuntu@Park:~$ rostopic echo /turtle1/pose
x: 3.13951468468
y: 4.73419284821
theta: -3.03999996185
linear_velocity: 0.0
angular_velocity: 0.0
---
x: 3.13951468468
y: 4.73419284821
theta: -3.03999996185
linear_velocity: 0.0
angular_velocity: 0.0
---
x: 3.13951468468
y: 4.73419284821
theta: -3.03999996185
linear_velocity: 0.0
angular_velocity: 0.0
---
x: 3.13951468468
y: 4.73419284821
theta: -3.03999996185
linear_velocity: 0.0
angular_velocity: 0.0
---
x: 3.13951468468
```

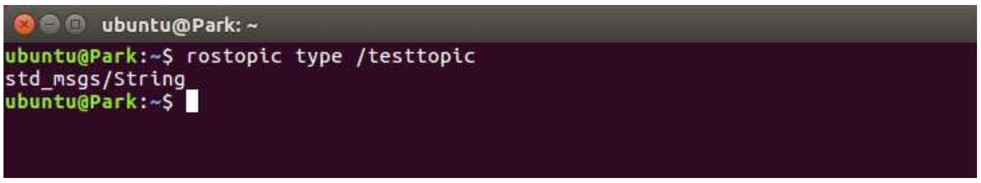
4) topic을 publish



```
ubuntu@Park: ~
ubuntu@Park:~$ rostopic pub /testtopic std_msgs/String "hello" --rate 10
```

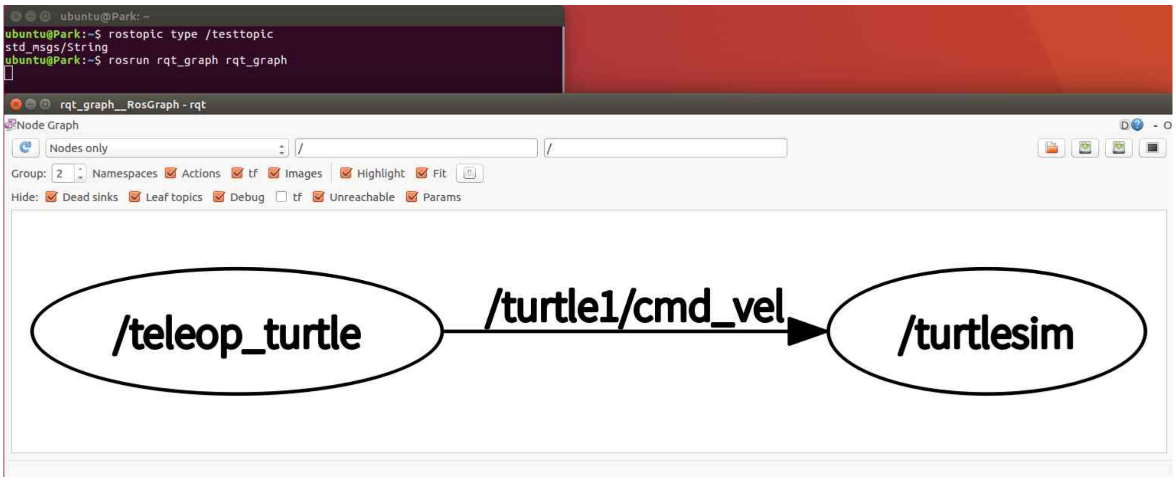
```
ubuntu@Park: ~
ubuntu@Park:~$ rostopic echo /testtopic
data: "hello"
---
data: "hello"
---
data: "hello"
---
data: "hello"
---
data: "hello"
---
data: "hello"
---
data: "hello"
---
data: "hello"
---
data: "hello"
---
data: "hello"
---
```

5) topic의 type 확인



```
ubuntu@Park: ~
ubuntu@Park:~$ rostopic type /testtopic
std_msgs/String
ubuntu@Park:~$ 
```

6) Node와 topic을 그래프로 시각화



```
ubuntu@Park: ~
ubuntu@Park:~$ rostopic type /testtopic
std_msgs/String
ubuntu@Park:~$ rosrun rqt_graph rqt_graph
```

rqt_graph__RosGraph - rqt

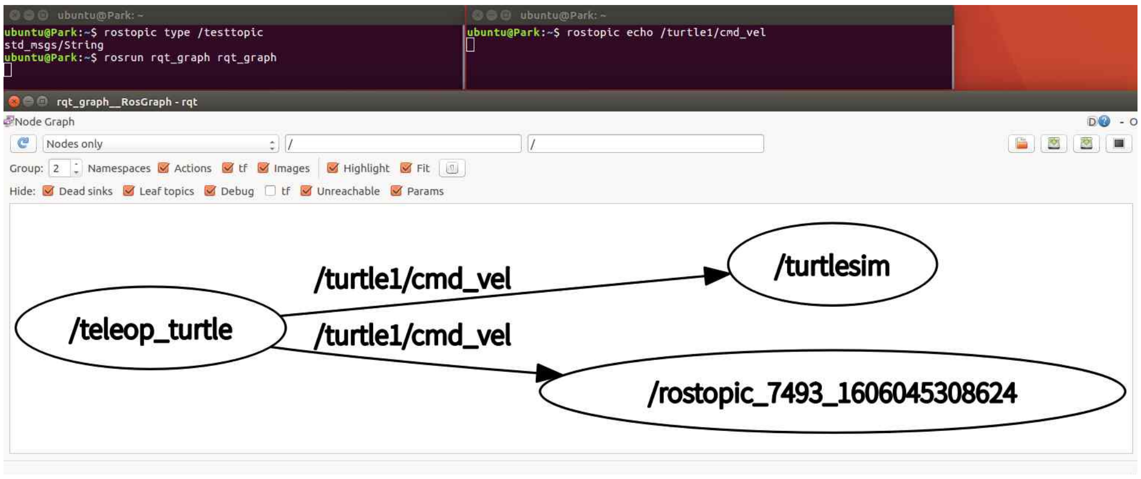
Node Graph

Nodes only | / | /

Group: 2 | Namespaces ✓ | Actions ✓ | tf ✓ | Images ✓ | Highlight ✓ | Fit ✓

Hide: Dead sinks ✓ | Leaf topics ✓ | Debug ✓ | tf ☐ | Unreachable ✓ | Params ✓

7) Topic echo를 실행한 뒤 그래프 확인

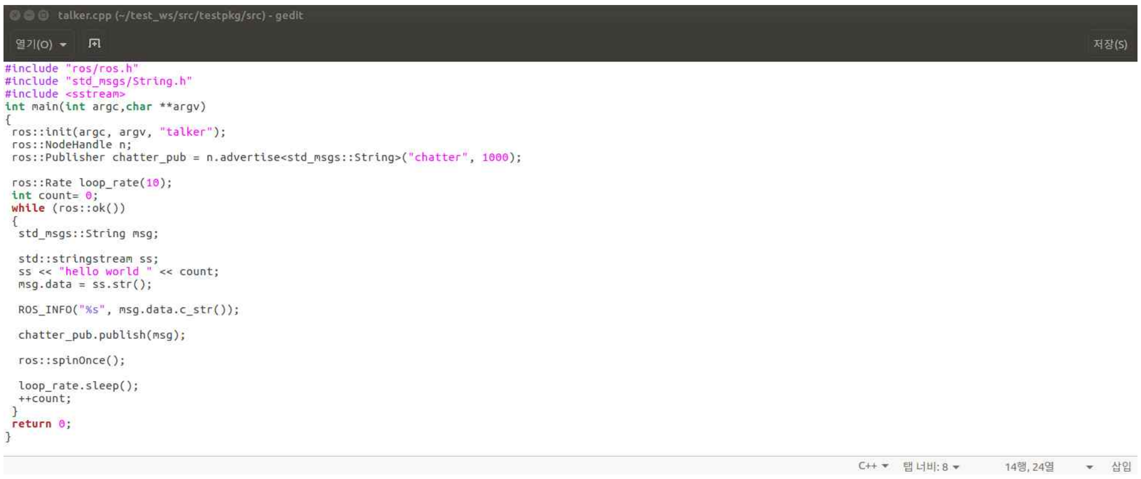


```
ubuntu@Park:~$ rostopic type /testtopic
std_msgs/String
ubuntu@Park:~$ rosrun rqt_graph rqt_graph
```

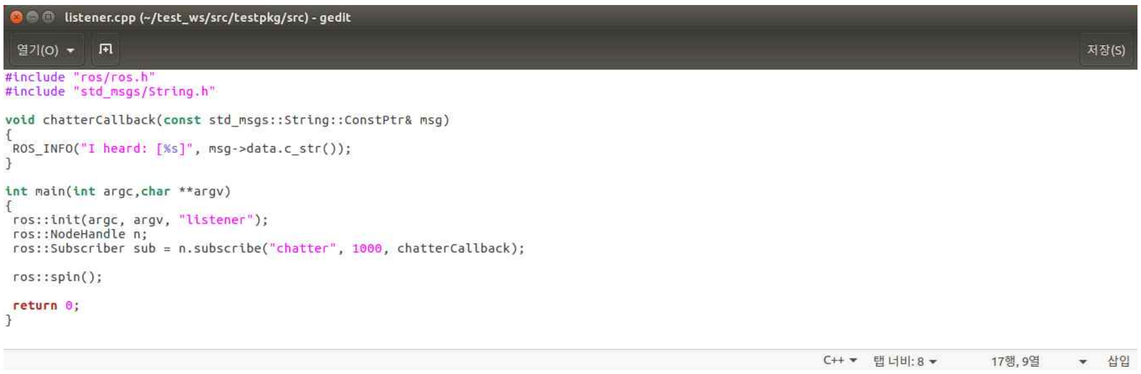
```
ubuntu@Park:~$ rostopic echo /turtle1/cmd_vel
```

8) talker / listener cpp 작성

- testpkg/src에 talker.cpp, listener.cpp 작성

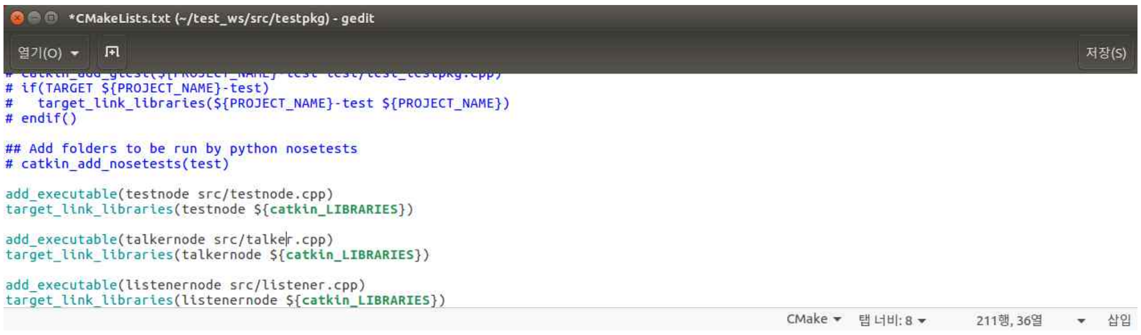


```cpp
#include "ros/ros.h"
#include "std_msgs/String.h"
#include <sstream>
int main(int argc,char **argv)
{
 ros::init(argc, argv, "talker");
 ros::NodeHandle n;
 ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);

 ros::Rate loop_rate(10);
 int count= 0;
 while (ros::ok())
 {
  std_msgs::String msg;

  std::stringstream ss;
  ss << "hello world " << count;
  msg.data = ss.str();

  ROS_INFO("%s", msg.data.c_str());

  chatter_pub.publish(msg);

  ros::spinOnce();

  loop_rate.sleep();
  ++count;
 }
 return 0;
}
```



```cpp
#include "ros/ros.h"
#include "std_msgs/String.h"

void chatterCallback(const std_msgs::String::ConstPtr& msg)
{
 ROS_INFO("I heard: [%s]", msg->data.c_str());
}

int main(int argc,char **argv)
{
 ros::init(argc, argv, "listener");
 ros::NodeHandle n;
 ros::Subscriber sub = n.subscribe("chatter", 1000, chatterCallback);

 ros::spin();

 return 0;
}
```

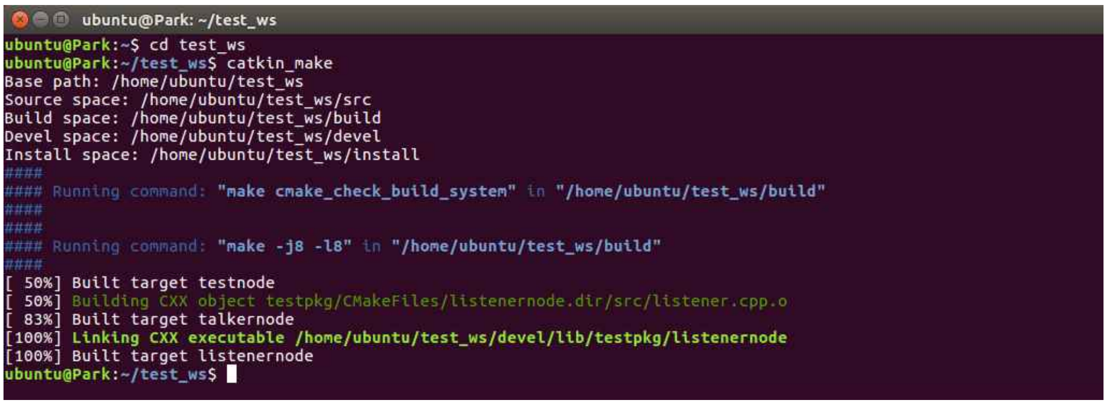
- CMakeList.txt에 add_executable, target_link_libraries 작성



```cmake
# if(TARGET ${PROJECT_NAME}-test)
#   target_link_libraries(${PROJECT_NAME}-test ${PROJECT_NAME})
# endif()

## Add folders to be run by python nosetests
# catkin_add_nosetests(test)

add_executable(testnode src/testnode.cpp)
target_link_libraries(testnode ${catkin_LIBRARIES})

add_executable(talkernode src/talker.cpp)
target_link_libraries(talkernode ${catkin_LIBRARIES})

add_executable(listenernode src/listener.cpp)
target_link_libraries(listenernode ${catkin_LIBRARIES})
```

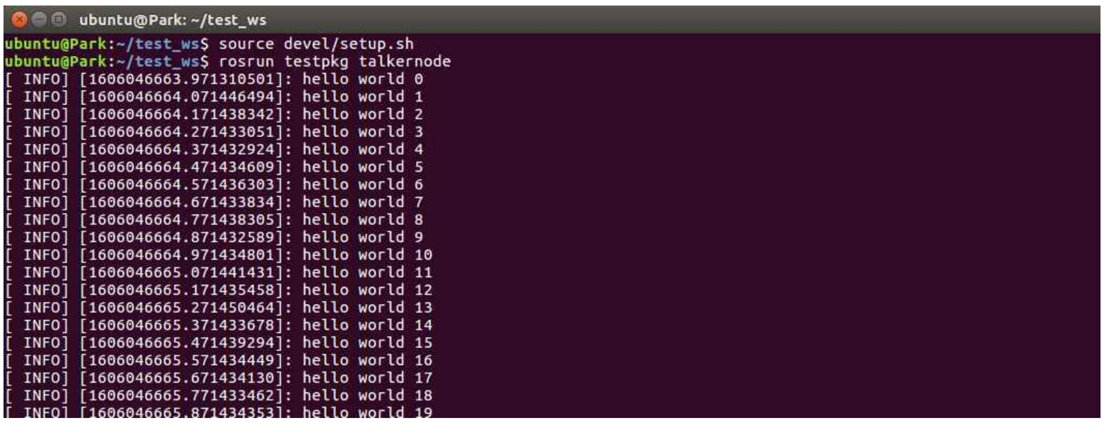
9) catkin_make 빌드 후 실행

- catkin_make 빌드



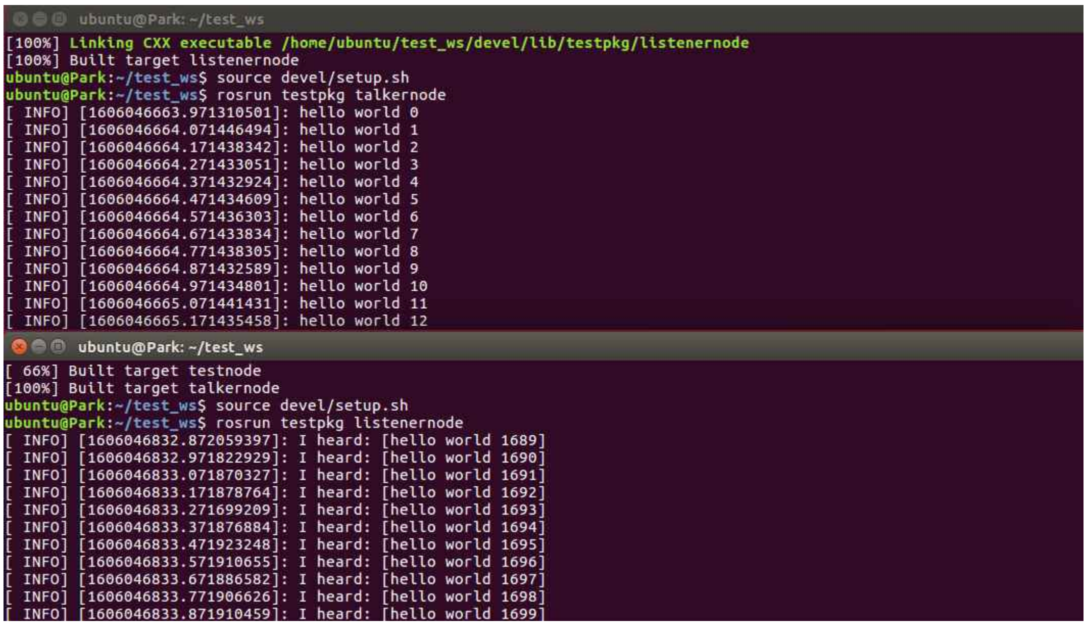
```
ubuntu@Park: ~/test_ws
ubuntu@Park:~$ cd test_ws
ubuntu@Park:~/test_ws$ catkin_make
Base path: /home/ubuntu/test_ws
Source space: /home/ubuntu/test_ws/src
Build space: /home/ubuntu/test_ws/build
Devel space: /home/ubuntu/test_ws/devel
Install space: /home/ubuntu/test_ws/install
####
#### Running command: "make cmake_check_build_system" in "/home/ubuntu/test_ws/build"
####
####
#### Running command: "make -j8 -l8" in "/home/ubuntu/test_ws/build"
####
[ 50%] Built target testnode
[ 50%] Building CXX object testpkg/CMakeFiles/listenernode.dir/src/listener.cpp.o
[ 83%] Built target talkernode
[100%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/listenernode
[100%] Built target listenernode
ubuntu@Park:~/test_ws$
```

- Terminal 1



```
ubuntu@Park: ~/test_ws
ubuntu@Park:~/test_ws$ source devel/setup.sh
ubuntu@Park:~/test_ws$ rosrun testpkg talkernode
[ INFO] [1606046663.971310501]: hello world 0
[ INFO] [1606046664.071446494]: hello world 1
[ INFO] [1606046664.171438342]: hello world 2
[ INFO] [1606046664.271433051]: hello world 3
[ INFO] [1606046664.371432924]: hello world 4
[ INFO] [1606046664.471434609]: hello world 5
[ INFO] [1606046664.571436303]: hello world 6
[ INFO] [1606046664.671433834]: hello world 7
[ INFO] [1606046664.771438305]: hello world 8
[ INFO] [1606046664.871432589]: hello world 9
[ INFO] [1606046664.971434801]: hello world 10
[ INFO] [1606046665.071441431]: hello world 11
[ INFO] [1606046665.171435458]: hello world 12
[ INFO] [1606046665.271450464]: hello world 13
[ INFO] [1606046665.371433678]: hello world 14
[ INFO] [1606046665.471439294]: hello world 15
[ INFO] [1606046665.571434449]: hello world 16
[ INFO] [1606046665.671434130]: hello world 17
[ INFO] [1606046665.771433462]: hello world 18
[ INFO] [1606046665.871434353]: hello world 19
```

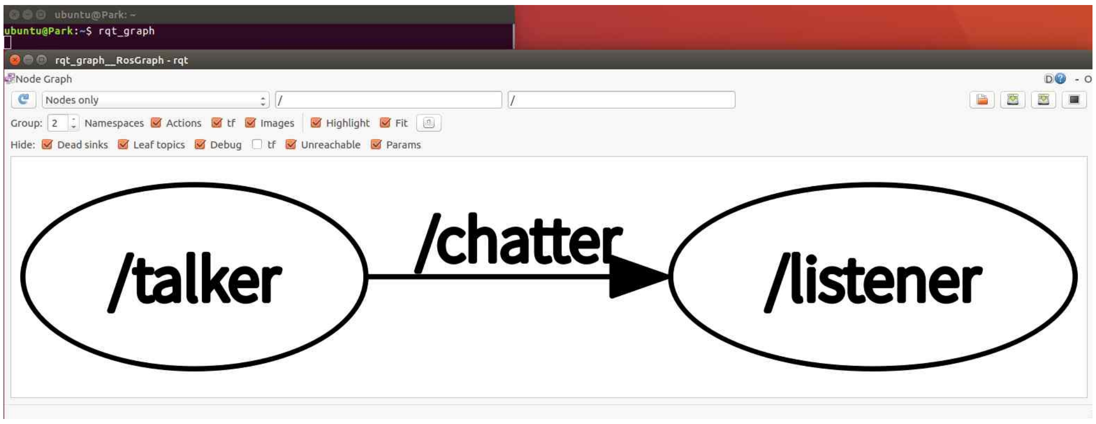
- Terminal 2



```
ubuntu@Park: ~/test_ws
[100%] Linking CXX executable /home/ubuntu/test_ws/devel/lib/testpkg/listenernode
[100%] Built target listenernode
ubuntu@Park:~/test_ws$ source devel/setup.sh
ubuntu@Park:~/test_ws$ rosrun testpkg talkernode
[ INFO] [1606046663.971310501]: hello world 0
[ INFO] [1606046664.071446494]: hello world 1
[ INFO] [1606046664.171438342]: hello world 2
[ INFO] [1606046664.271433051]: hello world 3
[ INFO] [1606046664.371432924]: hello world 4
[ INFO] [1606046664.471434609]: hello world 5
[ INFO] [1606046664.571436303]: hello world 6
[ INFO] [1606046664.671433834]: hello world 7
[ INFO] [1606046664.771438305]: hello world 8
[ INFO] [1606046664.871432589]: hello world 9
[ INFO] [1606046664.971434801]: hello world 10
[ INFO] [1606046665.071441431]: hello world 11
[ INFO] [1606046665.171435458]: hello world 12
```

```
ubuntu@Park: ~/test_ws
[ 66%] Built target testnode
[100%] Built target talkernode
ubuntu@Park:~/test_ws$ source devel/setup.sh
ubuntu@Park:~/test_ws$ rosrun testpkg listenernode
[ INFO] [1606046832.872059397]: I heard: [hello world 1689]
[ INFO] [1606046832.971822929]: I heard: [hello world 1690]
[ INFO] [1606046833.071870327]: I heard: [hello world 1691]
[ INFO] [1606046833.171878764]: I heard: [hello world 1692]
[ INFO] [1606046833.271699209]: I heard: [hello world 1693]
[ INFO] [1606046833.371876884]: I heard: [hello world 1694]
[ INFO] [1606046833.471923248]: I heard: [hello world 1695]
[ INFO] [1606046833.571910655]: I heard: [hello world 1696]
[ INFO] [1606046833.671886582]: I heard: [hello world 1697]
[ INFO] [1606046833.771906626]: I heard: [hello world 1698]
[ INFO] [1606046833.871910459]: I heard: [hello world 1699]
```

- 실행 결과 확인



```
ubuntu@Park: ~
ubuntu@Park:~$ rqt_graph
```

rqt_graph__RosGraph - rqt

Node Graph

Nodes only | / | /

Group: 2 [✓ Namespaces] [✓ Actions] [✓ tf] [✓ Images] [✓ Highlight] [✓ Fit]

Hide: [✓ Dead sinks] [✓ Leaf topics] [✓ Debug] tf [✓ Unreachable] [✓ Params]