

ECE 4703

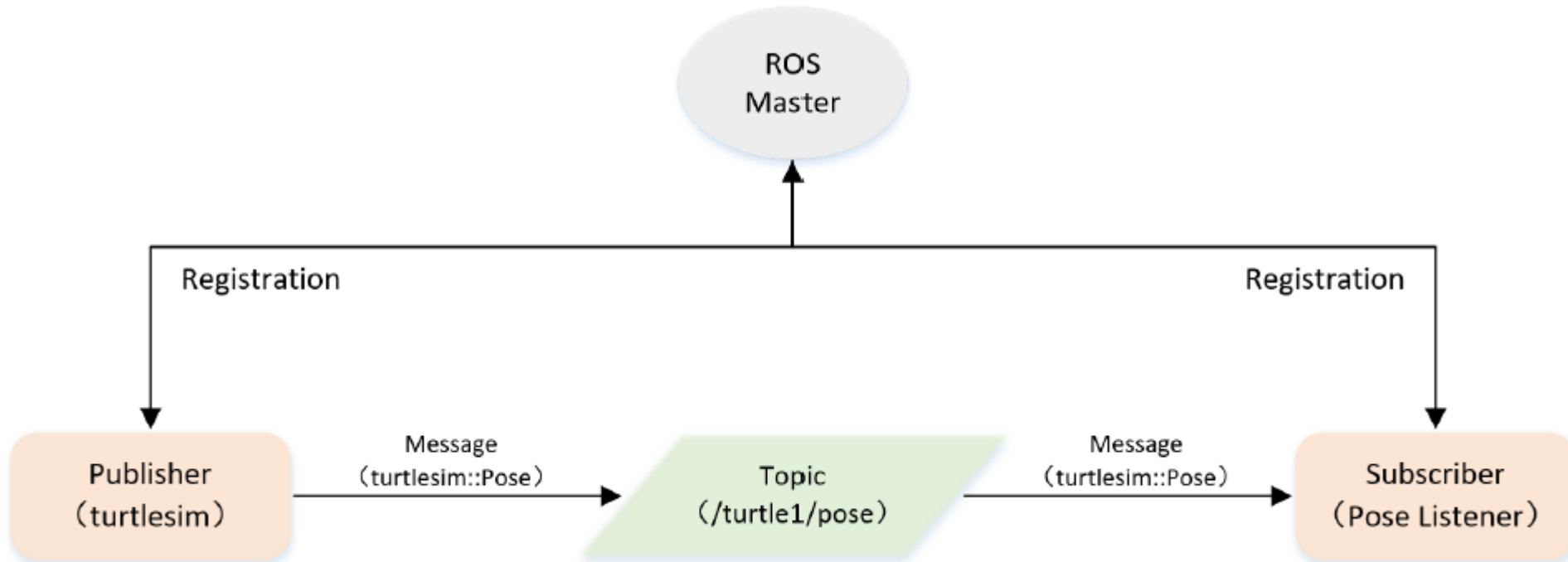
Mobile Autonomous Robots

Jenny Zhen Yu
zhenyu@cpp.edu


Department of Electrical and Computer Engineering
California State Polytechnic University, Pomona

Lecture 6: Subscriber Programming

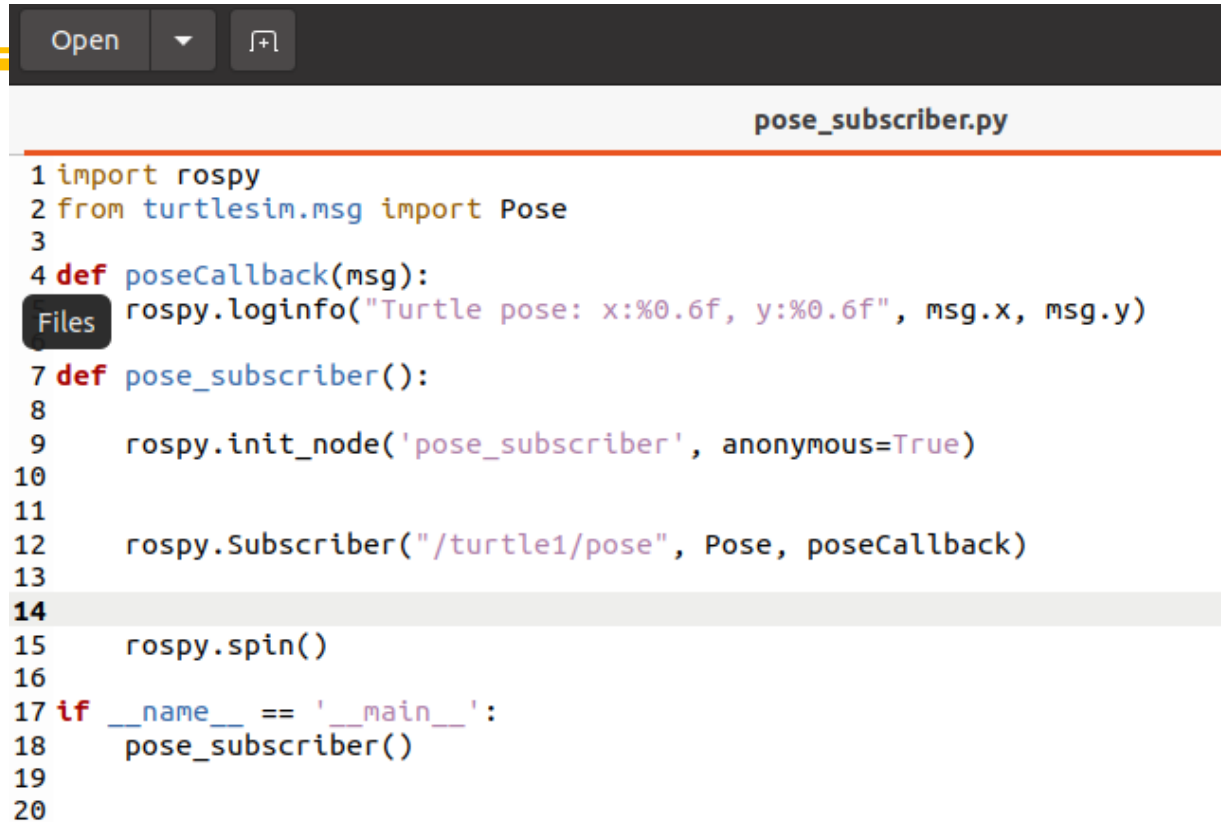
Topic




C++ Code

```
Open ▼   
1 #include <ros/ros.h>  
2 #include "turtlesim/Pose.h"  
3  
4  
5 void poseCallback(const turtlesim::Pose::ConstPtr& msg)  
6 {  
7  
8     ROS_INFO("Turtle pose: x:%0.6f, y:%0.6f", msg->x, msg->y);  
9 }  
10  
11 int main(int argc, char **argv)  
12 {  
13  
14     ros::init(argc, argv, "pose_subscriber");  
15  
16  
17     ros::NodeHandle n;  
18  
19  
20     ros::Subscriber pose_sub = n.subscribe("/turtle1/pose", 10, poseCallback);  
21  
22  
23     ros::spin();  
24  
25     return 0;  
26 }
```

Python Code



```
Open ▼ 
pose_subscriber.py
1 import rospy
2 from turtlesim.msg import Pose
3
4 def poseCallback(msg):
5     rospy.loginfo("Turtle pose: x:%0.6f, y:%0.6f", msg.x, msg.y)
6
7 def pose_subscriber():
8
9     rospy.init_node('pose_subscriber', anonymous=True)
10
11
12     rospy.Subscriber("/turtle1/pose", Pose, poseCallback)
13
14
15     rospy.spin()
16
17 if __name__ == '__main__':
18     pose_subscriber()
19
20
```

Compiling Code

```
## Declare a C++ executable
## With catkin_make all packages are built within a single CMake context
## The recommended prefix ensures that target names across packages don't collide
# add_executable(${PROJECT_NAME}_node src/learning_topic_node.cpp)

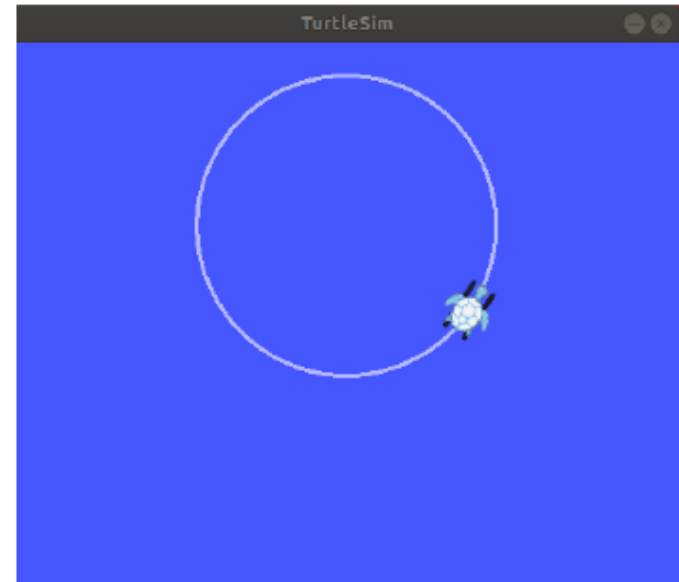
## Specify libraries to link a library or executable target against
# target_link_libraries(${PROJECT_NAME}_node
#   ${catkin_LIBRARIES}
# )

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

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

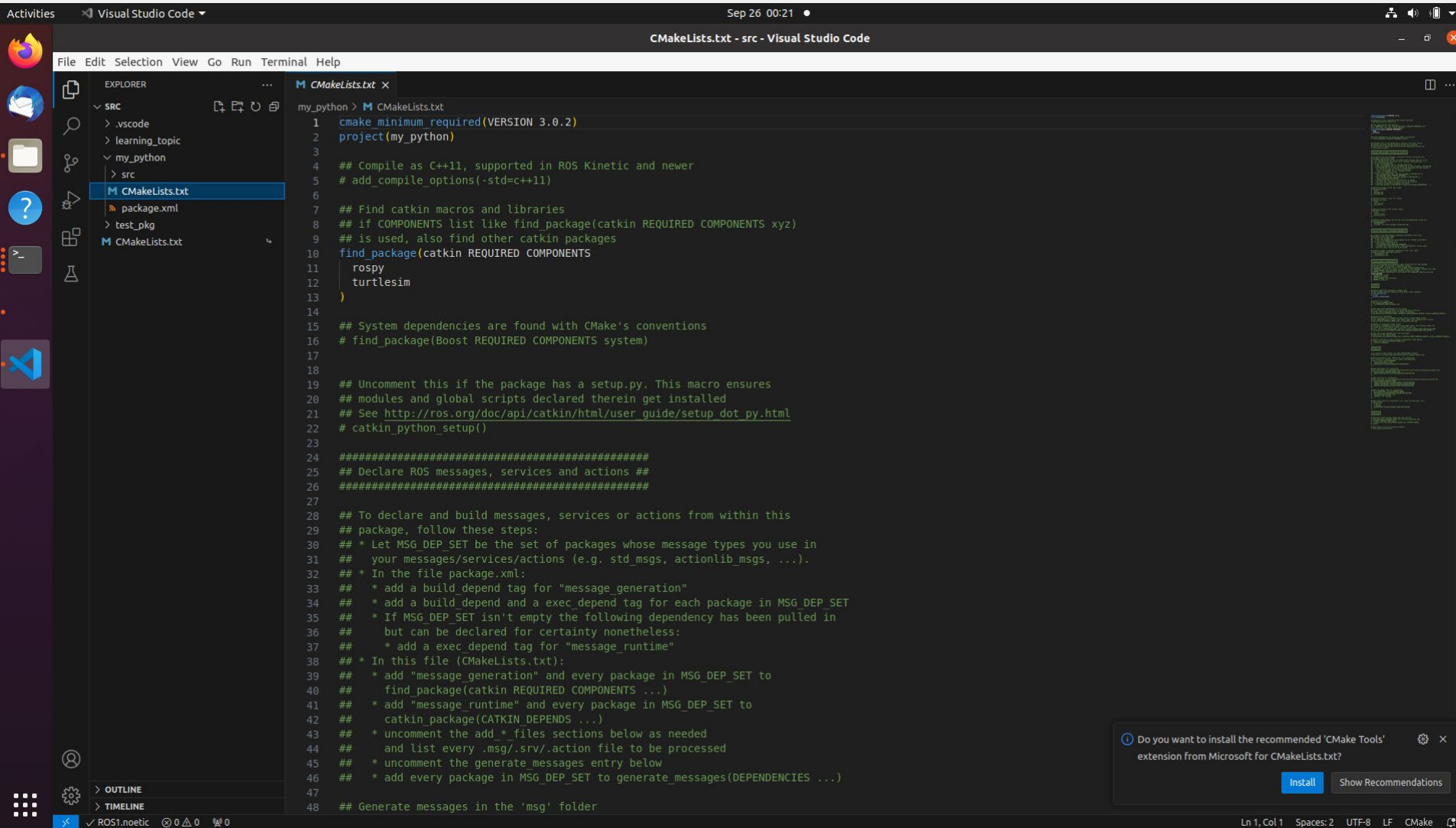
Run Program

```
$ cd ~/catkin_ws
$ catkin_make
$ source devel/setup.bash
$ roscore
$ rosrun turtlesim turtlesim_node
$ rosrun learning_topic velocity_publisher
```

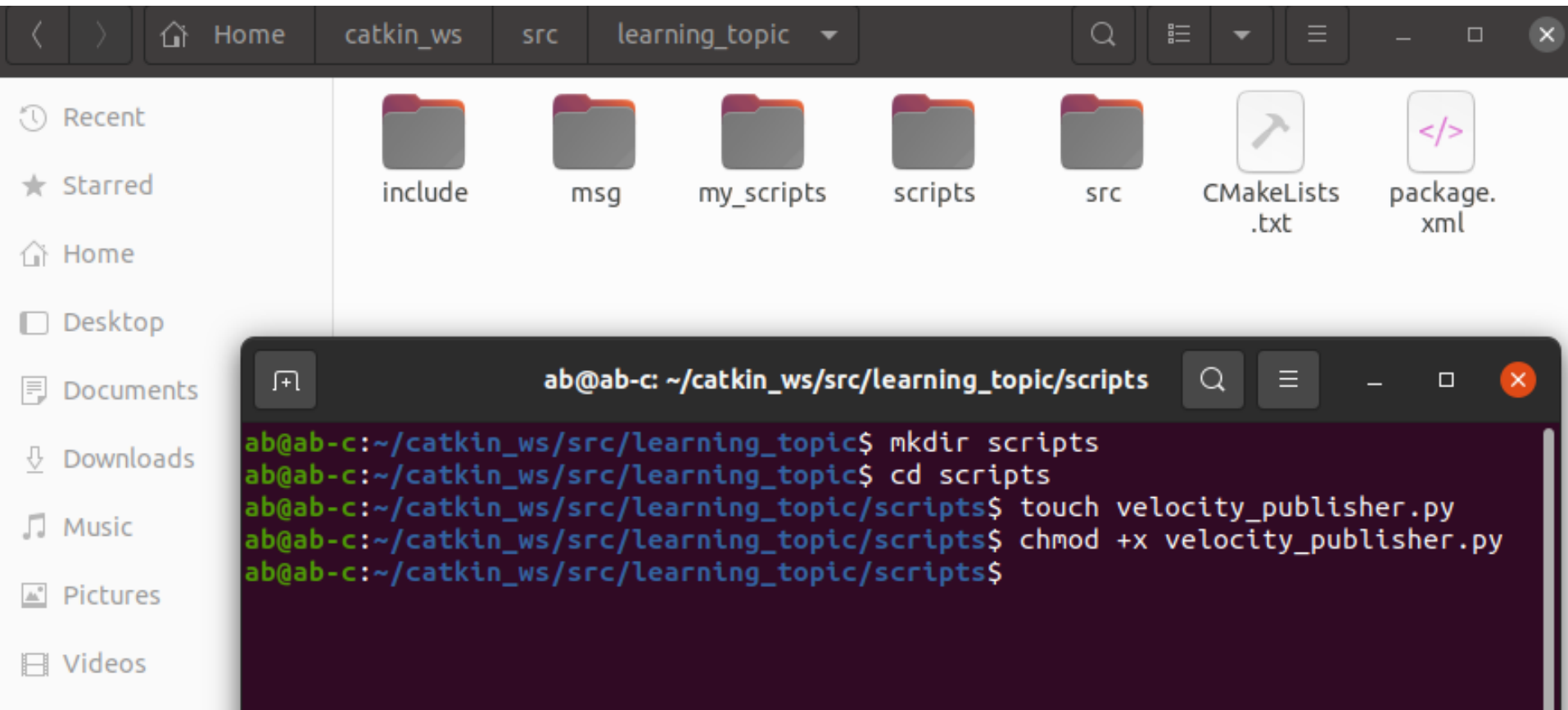


```
hcx@hcx-vpc:~/catkin_ws$ rosrun learning_topic pose_subscriber
[ INFO] [1562211557.322259871]: Turtle pose: x:6.389005, y:10.396028
[ INFO] [1562211557.339097278]: Turtle pose: x:6.381475, y:10.398730
[ INFO] [1562211557.354512018]: Turtle pose: x:6.373938, y:10.401410
[ INFO] [1562211557.370549572]: Turtle pose: x:6.366391, y:10.404065
[ INFO] [1562211557.387085434]: Turtle pose: x:6.358836, y:10.406695
[ INFO] [1562211557.402710847]: Turtle pose: x:6.351273, y:10.409303
[ INFO] [1562211557.418887039]: Turtle pose: x:6.343701, y:10.411885
[ INFO] [1562211557.434469988]: Turtle pose: x:6.336121, y:10.414443
[ INFO] [1562211557.450210135]: Turtle pose: x:6.328533, y:10.416977
[ INFO] [1562211557.465994903]: Turtle pose: x:6.320937, y:10.419487
[ INFO] [1562211557.482173454]: Turtle pose: x:6.313333, y:10.421972
```

Python



Scripts





Recent

Starred

Home

Desktop

Documents

Downloads

Music

Pictures

Videos

Trash

+ Other Locations

```
ab@ab-c: ~/catkin_ws/src
ab@ab-c:~/catkin_ws/src$ code .
ab@ab-c:~/catkin_ws/src$
```

my_forst_node.cpp - src - Visual Studio Code

File Edit Selection View Go Run Terminal Help

EXPLORER

SRC

> .vscode

> learning_topic

> my_scripts

> test_pkg

CMakeLists.txt

OUTLINE

TIMELINE

my_forst_node.cpp

1 #include <ros/ros.h>

2 #include <geometry_msgs/Twist.h>

3

4 int main(int argc, char **argv)

5 {

6

7 ros::init(argc, argv, "velocity_publisher");

8

9

10 ros::NodeHandle n;

11

12

13 ros::Publisher turtle_vel_pub = n.advertise<geometry_msgs::Twist>

14

15

16 ros::Rate loop_rate(10);

17

18 int count = 0;

19 while (ros::ok())

20 {

21

22 geometry_msgs::Twist vel_msg;

23 vel_msg.linear.x = 0.5;

24 vel_msg.angular.z = 0.2;

25

26

27 turtle_vel_pub.publish(vel_msg);

28 ROS_INFO("Publish turtle velocity command[%0.2f m/s, %0.2f rad/s]",

29 vel_msg.linear.x, vel_msg.angular.z);

30

31

32 loop_rate.sleep();

33 }

34

35 return 0;

Do you want to install the recommended 'C/C++ Extension Pack' extension from Microsoft for the C++ language?

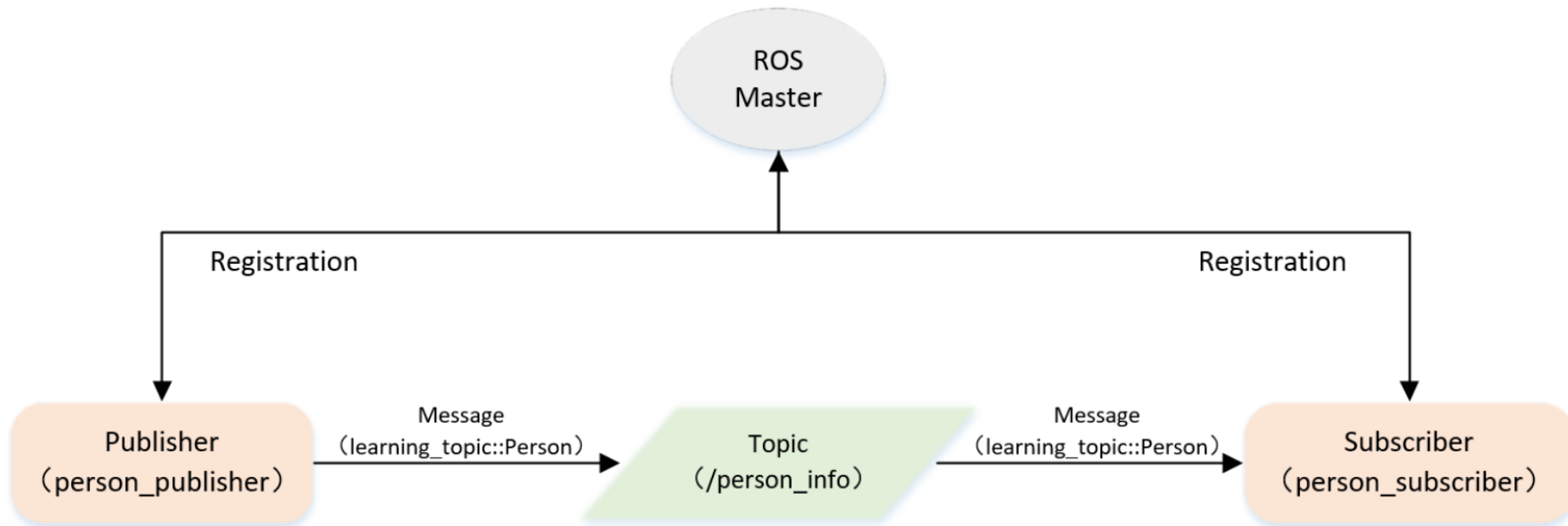
Install

Show Recommendations

Ln 37, Col 1 Tab Size: 4 UTF-8 LF { } C++ ROS

Lecture 7: Topic Message

Topic Model



Topic Model

string name

uint8 sex

uint8 age

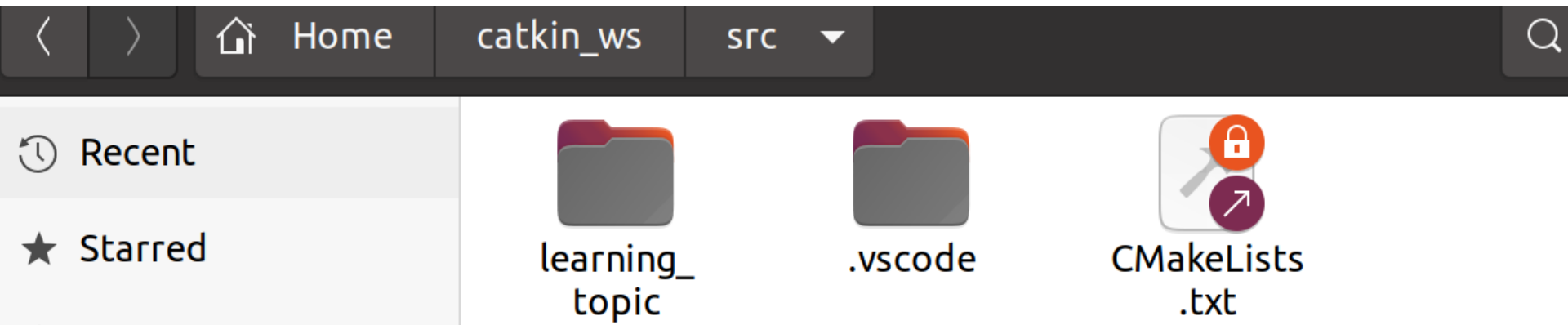
uint8 unknown = 0

uint8 male = 1

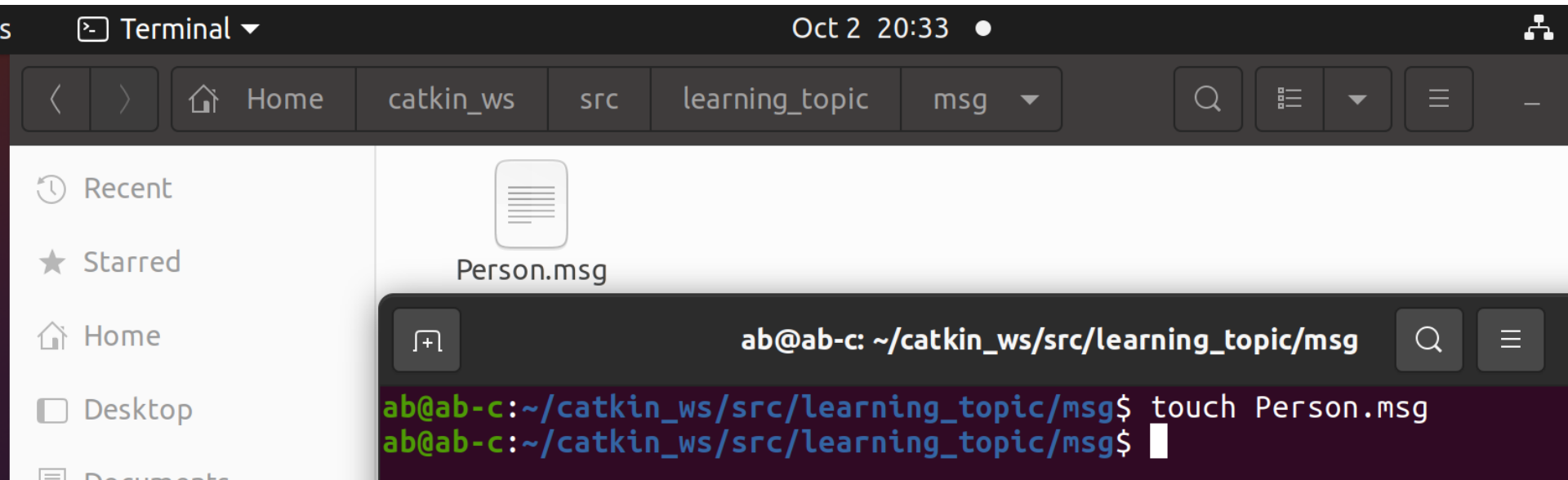
uint8 female =2

Person.msg

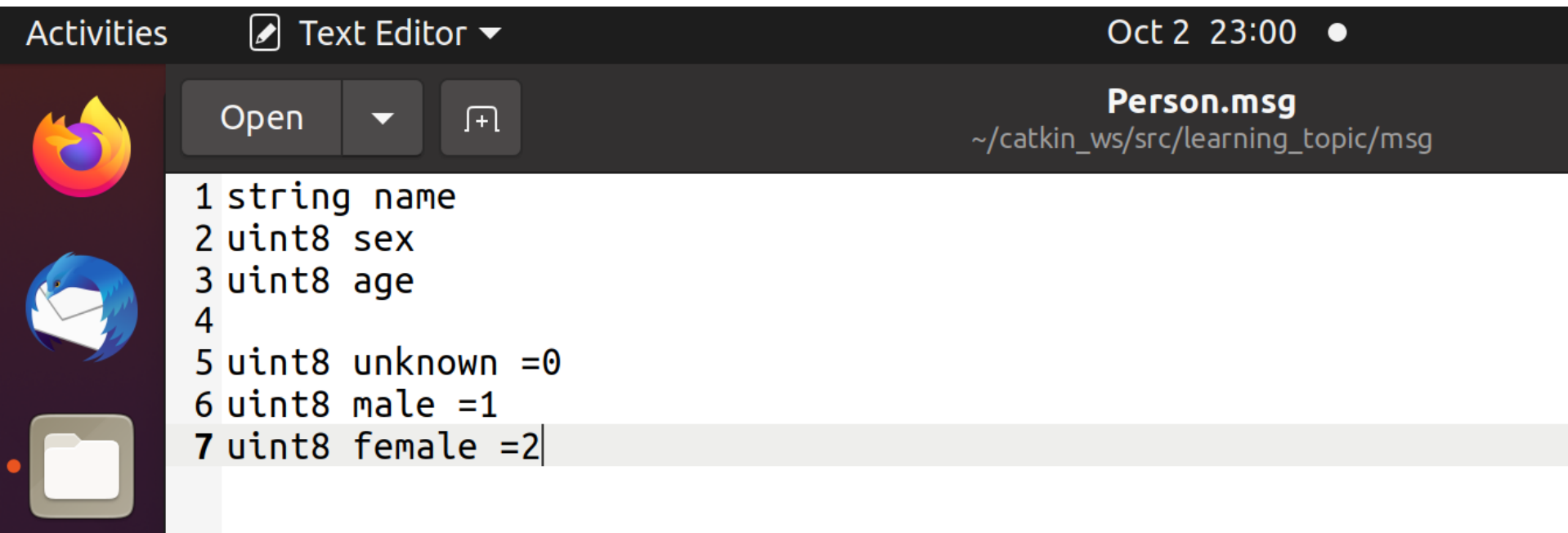
Build msg in learning_topic



Build Person.msg



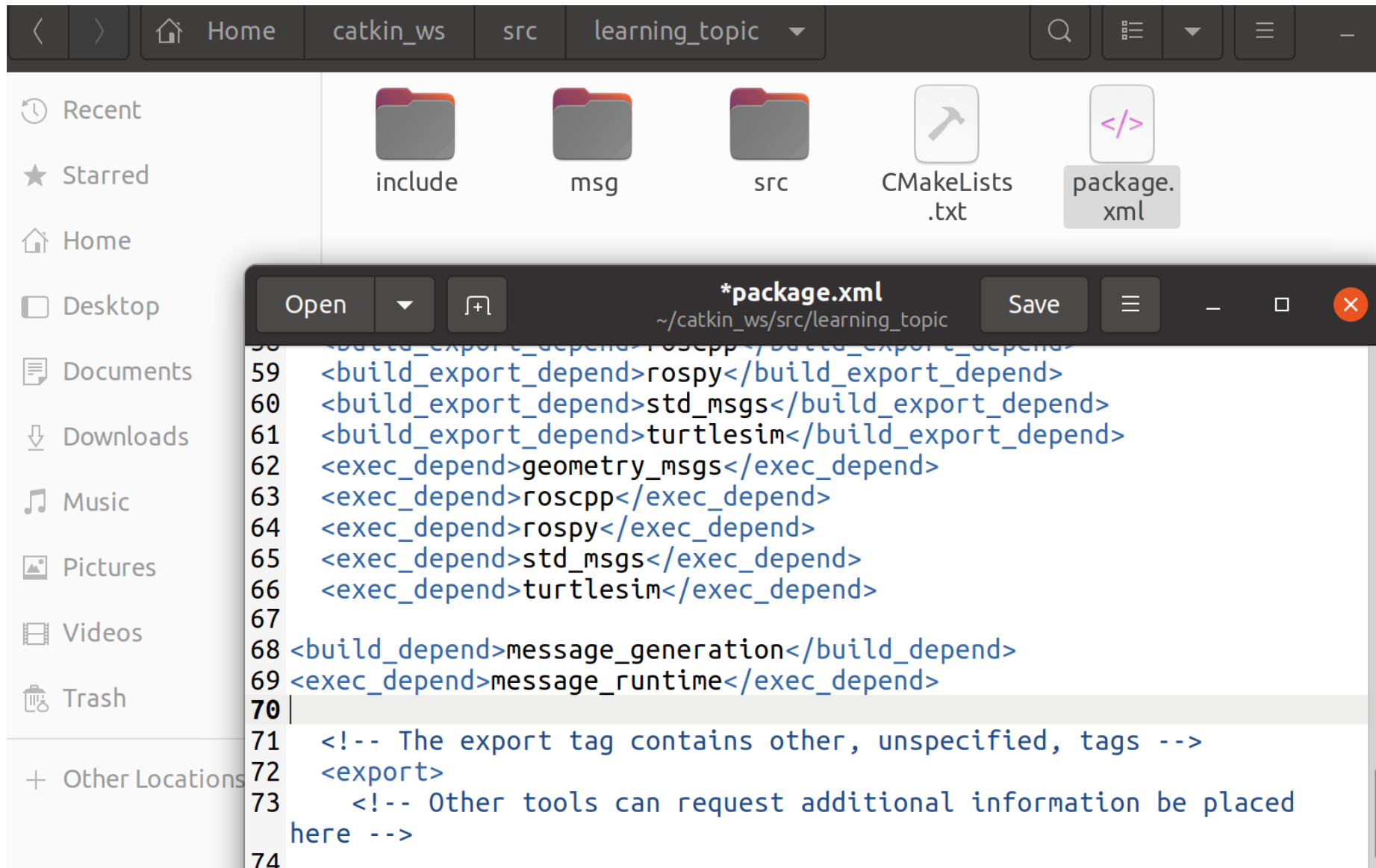
Data Interface Definition Person.msg



The image shows a screenshot of an Ubuntu desktop environment. The top bar displays 'Activities', 'Text Editor', and the date/time 'Oct 2 23:00'. The left sidebar contains icons for Firefox, Mail, and Files. The main window is a text editor titled 'Person.msg' with the path '~/catkin_ws/src/learning_topic/msg'. The editor contains the following text:

```
1 string name
2 uint8 sex
3 uint8 age
4
5 uint8 unknown =0
6 uint8 male =1
7 uint8 female =2|
```


Add Dependence in package.xml



The screenshot shows a file manager window with the following structure:

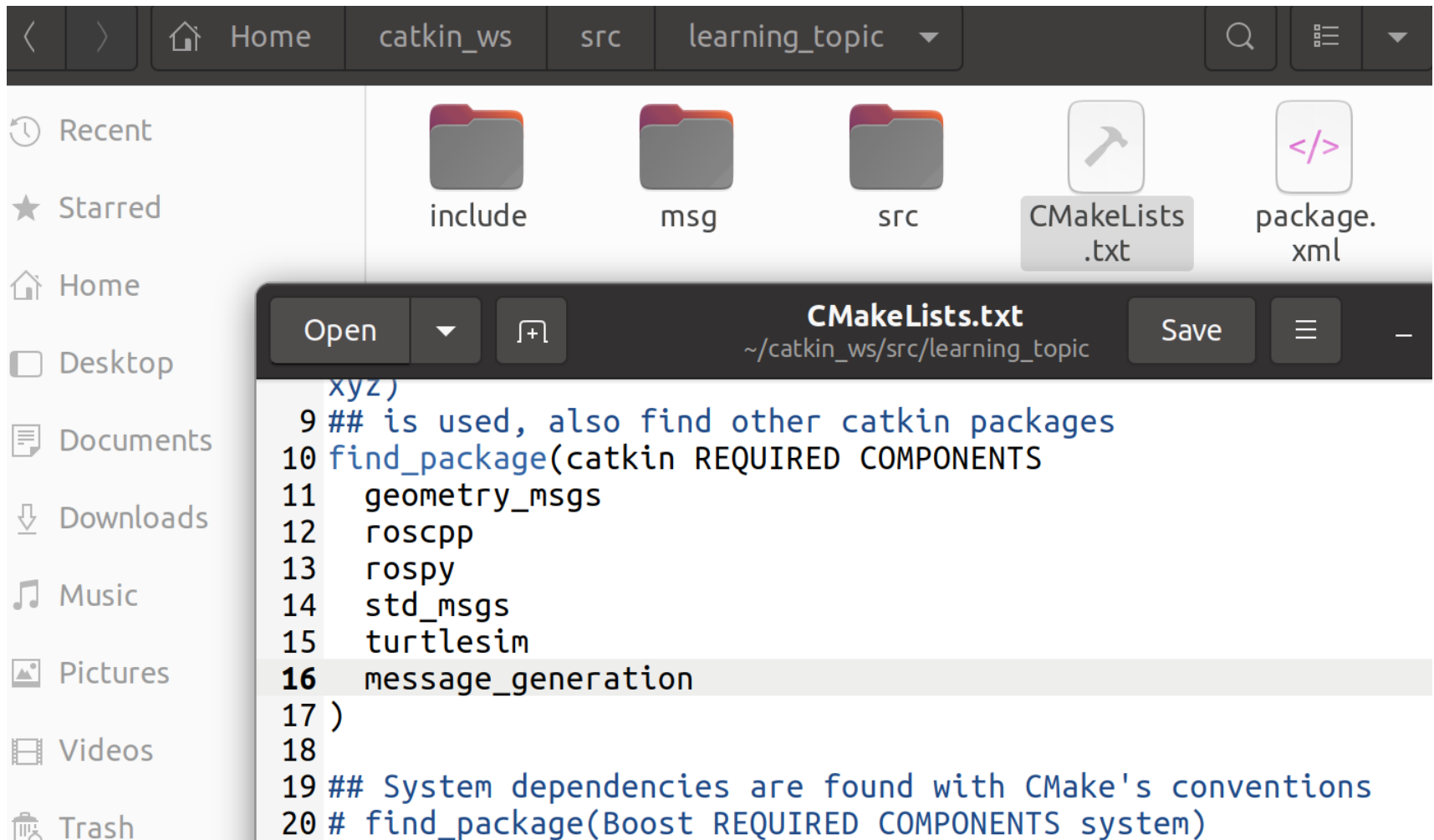
- Top bar: Navigation icons, Home, catkin_ws, src, learning_topic (selected), Search, and menu icons.
- Left sidebar: Recent, Starred, Home, Desktop, Documents, Downloads, Music, Pictures, Videos, Trash, and Other Locations.
- Main area: Displays the contents of the 'learning_topic' directory, including folders 'include', 'msg', and 'src', and files 'CMakeLists.txt' and 'package.xml'.

The 'package.xml' file is open in a text editor, showing the following XML content:

```
58 <build_export_depend>roscpp</build_export_depend>
59 <build_export_depend>rospy</build_export_depend>
60 <build_export_depend>std_msgs</build_export_depend>
61 <build_export_depend>turtlesim</build_export_depend>
62 <exec_depend>geometry_msgs</exec_depend>
63 <exec_depend>roscpp</exec_depend>
64 <exec_depend>rospy</exec_depend>
65 <exec_depend>std_msgs</exec_depend>
66 <exec_depend>turtlesim</exec_depend>
67
68 <build_depend>message_generation</build_depend>
69 <exec_depend>message_runtime</exec_depend>
70
71 <!-- The export tag contains other, unspecified, tags -->
72 <export>
73   <!-- Other tools can request additional information be placed
74   here -->
```

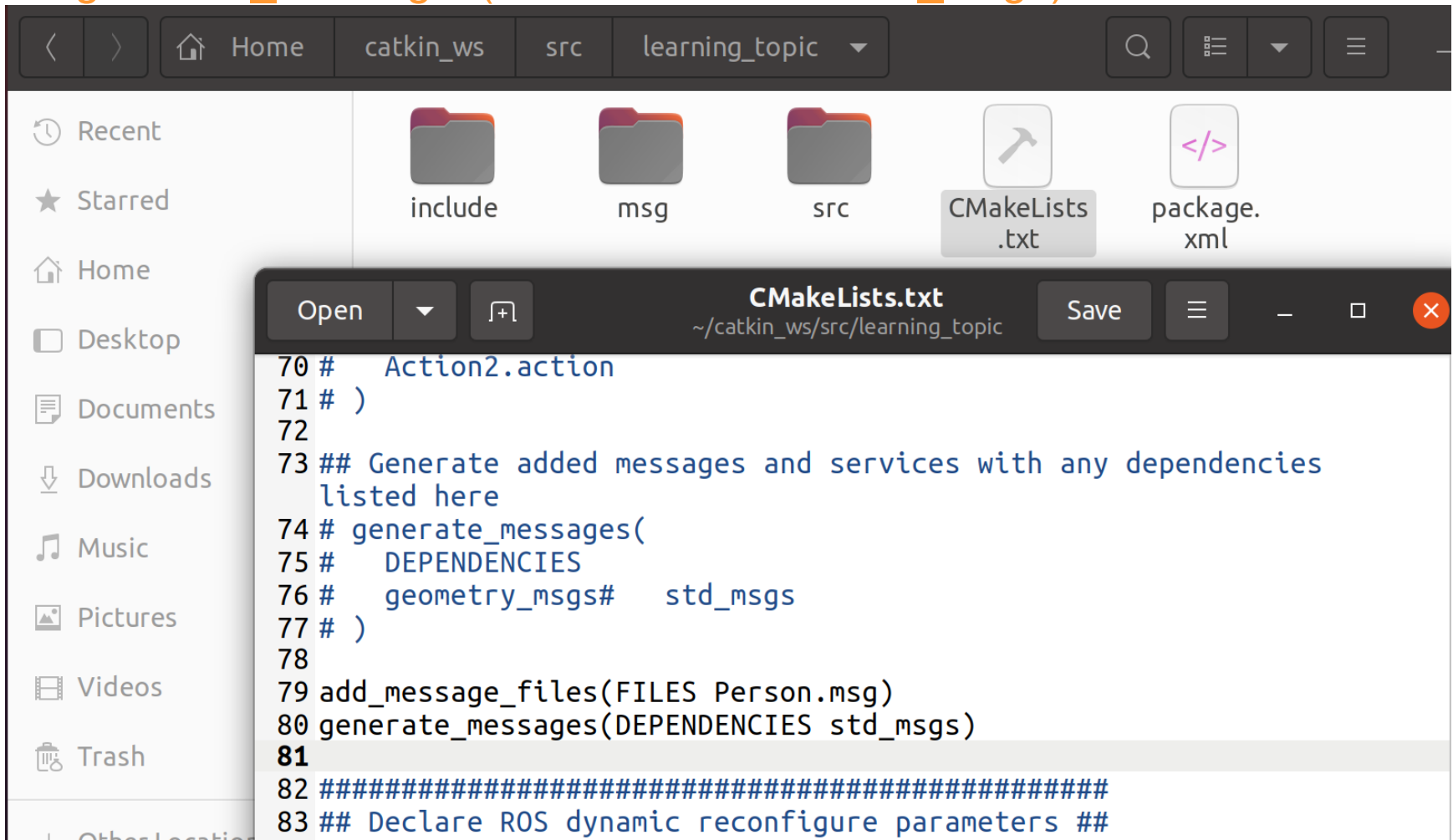
Add Compile Options in CMakeLists.txt

1. message_generation



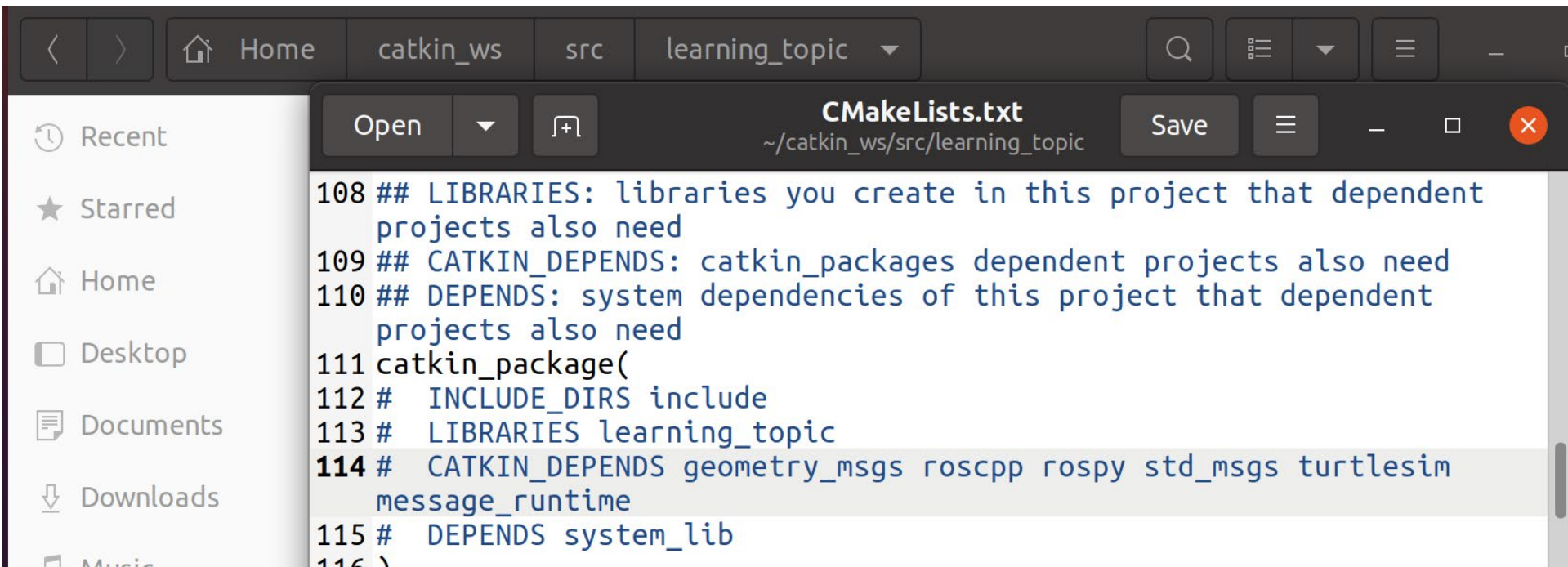
Add Compile Options in CMakeLists.txt

2. `add_message_files(FILES Person.msg)`
`generate_messages(DEPENDENCIES std_msgs)`



Add Compile Options in CMakeLists.txt

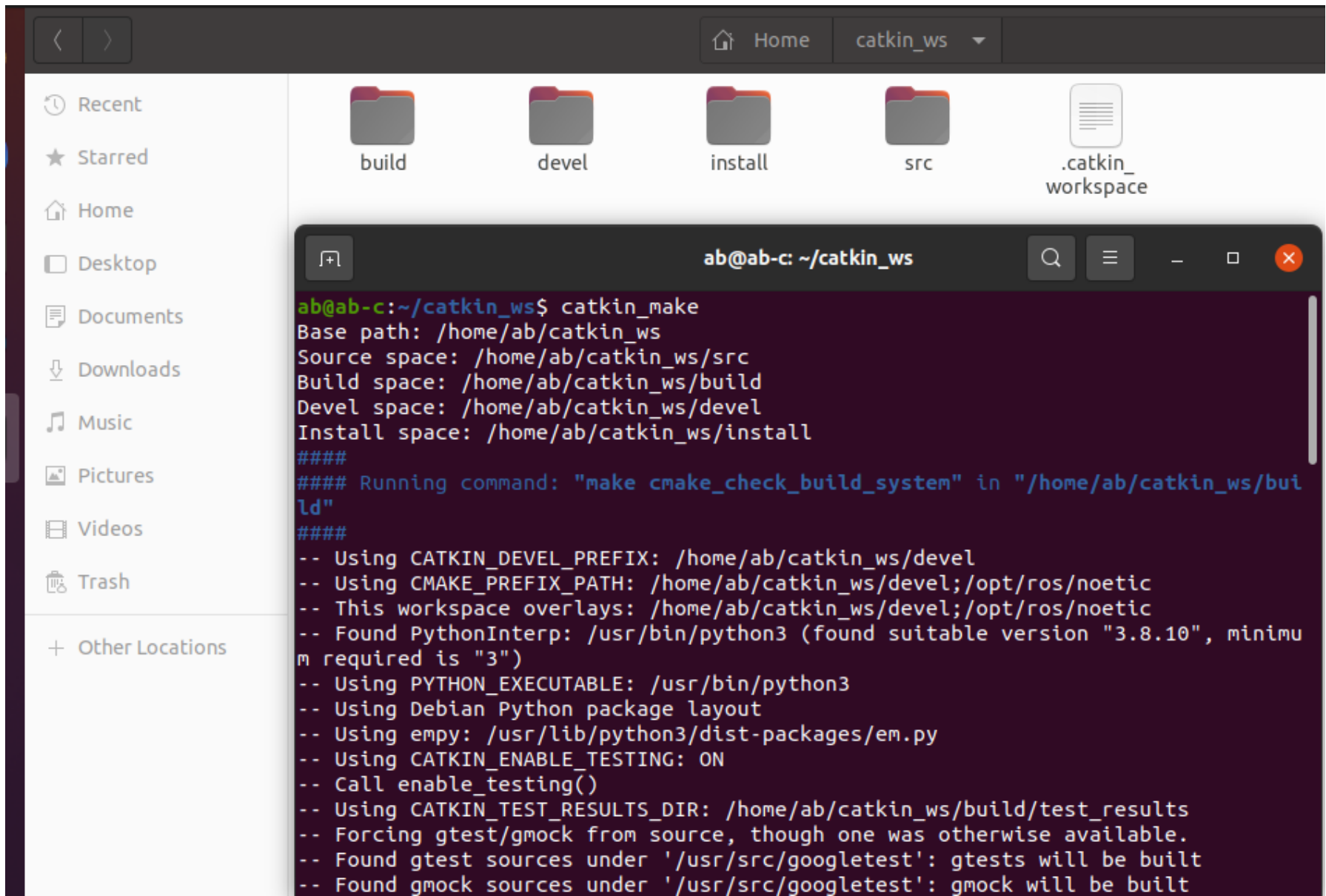
3. message_runtime



The screenshot shows a code editor with a dark theme. The top toolbar includes navigation icons, a search icon, and a dropdown menu showing 'learning_topic'. The file path is '~catkin_ws/src/learning_topic'. The editor displays the CMakeLists.txt file with the following content:

```
108 ## LIBRARIES: libraries you create in this project that dependent
    projects also need
109 ## CATKIN_DEPENDS: catkin_packages dependent projects also need
110 ## DEPENDS: system dependencies of this project that dependent
    projects also need
111 catkin_package(
112 #   INCLUDE_DIRS include
113 #   LIBRARIES learning_topic
114 #   CATKIN_DEPENDS geometry_msgs roscpp rospy std_msgs turtlesim
    message_runtime
115 #   DEPENDS system_lib
116 \
```

catkin_make Compile



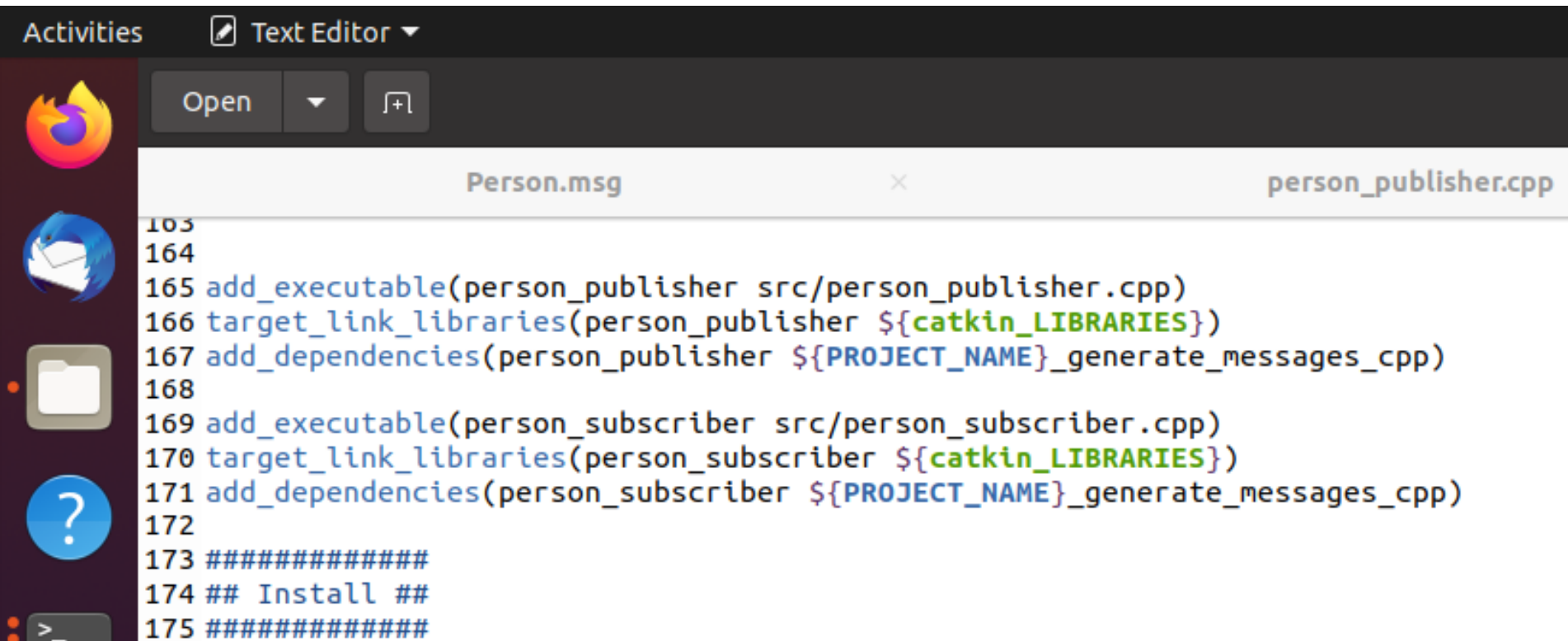
Create Publisher Code C++

Person.msg

×

```
1 #include <ros/ros.h>
2 #include "learning_topic/Person.h"
3
4 int main(int argc, char **argv)
5 {
6
7     ros::init(argc, argv, "person_publisher");
8
9
10    ros::NodeHandle n;
11
12
13    ros::Publisher person_info_pub = n.advertise<learning_topic::Person>("/person_info", 10);
14
15
16    ros::Rate loop_rate(1);
17
18    int count = 0;
19    while (ros::ok())
20    {
21
22        learning_topic::Person person_msg;
23        person_msg.name = "Tom";
24        person_msg.age = 18;
25        person_msg.sex = learning_topic::Person::male;
26
27
28        person_info_pub.publish(person_msg);
29
30        ROS_INFO("Publish Person Info: name:%s age:%d sex:%d",
31                person_msg.name.c_str(), person_msg.age, person_msg.sex);
32
33
34        loop_rate.sleep();
35    }
36
37    return 0;
38 }
```

Compiling Code



The screenshot shows a Linux desktop environment. On the left is a vertical dock with icons for Firefox, a mail client, a file manager, and a help icon. The top bar shows 'Activities' and 'Text Editor'. The text editor window has two tabs: 'Person.msg' and 'person_publisher.cpp'. The 'person_publisher.cpp' tab is active, displaying CMakeLists.txt code. The code includes line numbers 163 through 175. The code defines two executables, 'person_publisher' and 'person_subscriber', and links them to the 'catkin_LIBRARIES'. It also adds dependencies for generating messages from the 'Person.msg' file.

```
163
164
165 add_executable(person_publisher src/person_publisher.cpp)
166 target_link_libraries(person_publisher ${catkin_LIBRARIES})
167 add_dependencies(person_publisher ${PROJECT_NAME}_generate_messages_cpp)
168
169 add_executable(person_subscriber src/person_subscriber.cpp)
170 target_link_libraries(person_subscriber ${catkin_LIBRARIES})
171 add_dependencies(person_subscriber ${PROJECT_NAME}_generate_messages_cpp)
172
173 #####
174 ## Install ##
175 #####
```


catkin_make Compile

```
ab@ab-c: ~/catkin_ws
ab@ab-c:~/catkin_ws$ catkin_make
Base path: /home/ab/catkin_ws
Source space: /home/ab/catkin_ws/src
Build space: /home/ab/catkin_ws/build
Devel space: /home/ab/catkin_ws/devel
Install space: /home/ab/catkin_ws/install
####
#### Running command: "make cmake_check_build_system" in "/home/ab/catkin_ws/build"
####
####
#### Running command: "make -j2 -l2" in "/home/ab/catkin_ws/build"
####
[ 0%] Built target std_msgs_generate_messages_cpp
[ 0%] Built target _learning_topic_generate_messages_check_deps_Person
[ 15%] Built target velocity_publisher
[ 15%] Built target std_msgs_generate_messages_eus
[ 15%] Built target std_msgs_generate_messages_py
[ 15%] Built target std_msgs_generate_messages_lisp
[ 15%] Built target std_msgs_generate_messages_nodejs
[ 23%] Built target learning_topic_generate_messages_cpp
[ 38%] Built target learning_topic_generate_messages_eus
ab@ab-c:~$
```


Publisher and Subscriber Compile

```
$ cd ~/catkin_ws
```

```
$ catkin_make
```

```
$ source devel/setup.bash
```

```
$ roscore
```

```
$ rosrun learning_topic person_subscriber
```

```
$ rosrun learning_topic person_publisher
```

Publisher and Subscriber Compile

The image shows a ROS development environment with several terminal windows. The top window displays the CMakeLists.txt file for a learning_topic package. Below it, a roscore terminal window shows the ROS master starting. In the foreground, two terminal windows show the compilation and execution of the publisher and subscriber nodes.

```
CMakeLists.txt
~/catkin_ws/src/learning_topic

Person.msg
103
164
165 add_executable(person_publisher src/person_publisher.cpp)
166 target_link_libraries(person_publisher ${catkin_LIBRARIES})
167 add_dependencies(person_publisher ${PROJECT_NAME})
168
169 add_executable(person_subscriber src/person_subscriber.cpp)
170 target_link_libraries(person_subscriber ${catkin_LIBRARIES})
171 add_dependencies(person_subscriber ${PROJECT_NAME})
172
173 #####
174 ## Install ##
175 #####
176
177 # all install targets should use catkin DESTINATION
178 # See http://ros.org/doc/api/catkin/html/adv\_user\_guide/setting-variables-for-installing.html
179
180 ## Mark executable scripts (Python etc.) for installation
181 ## in contrast to setup.py, you can choose the destination
182 catkin_install_python(PROGRAMS
183   scripts/my_python_script
184   DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION})
185 )
186
187 ## Mark other files for installation
188 ## See http://ros.org/doc/api/catkin/html/adv\_user\_guide/setting-variables-for-installing.html for more details
189 install(DIRECTORY include/${PROJECT_NAME}/
190   DESTINATION ${CATKIN_PACKAGE_INCLUDE_DESTINATION}
191   USE_SOURCE_PERMISSIONS)
192
193 # To ensure that this file will be used by catkin, either:
194 # 1) add a line like below to the file (CMAKE_MODULE_PATH needs to be
195 #    set before this line)
196 # find_package(catkin REQUIRED)
197 #
198 # catkin_package(
199 #   INCLUDE_DIRS include
200 #   LIBRARIES learning_topic
201 #   CATKIN_DEPENDS roscpp std_msgs
202 #   DEPENDS system_lib
203 # )
204
205 # Uncomment this line to build out of source
206 # catkin_add_export_dir(OUTDIR)
207
208 ## Install and export the package
209 install(TARGETS person_publisher person_subscriber
210   ARCHIVE DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION}
211   RUNTIME DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION})
212
213 # To ensure that this file will be used by catkin, either:
214 # 1) add a line like below to the file (CMAKE_MODULE_PATH needs to be
215 #    set before this line)
216 # find_package(catkin REQUIRED)
217 #
218 # catkin_package(
219 #   INCLUDE_DIRS include
220 #   LIBRARIES learning_topic
221 #   CATKIN_DEPENDS roscpp std_msgs
222 #   DEPENDS system_lib
223 # )
224
225 # Uncomment this line to build out of source
226 # catkin_add_export_dir(OUTDIR)
227
228 ## Install and export the package
229 install(TARGETS person_publisher person_subscriber
230   ARCHIVE DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION}
231   RUNTIME DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION})
```

```
roscore http://ab-c:11311/
ab@ab-c:~$ roscore
... logging to /home/ab/.ros/log/011d2230-61b1-11ee-bd77-09718d42e051/roslaunch-
ab-c-2460.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
started roslaunch server http://ab-c:46089/
ros_comm version 1.16.0

SUMMARY
=====
PARAMETERS
* /roscpp: noetic
* /rosversion: 1.16.0

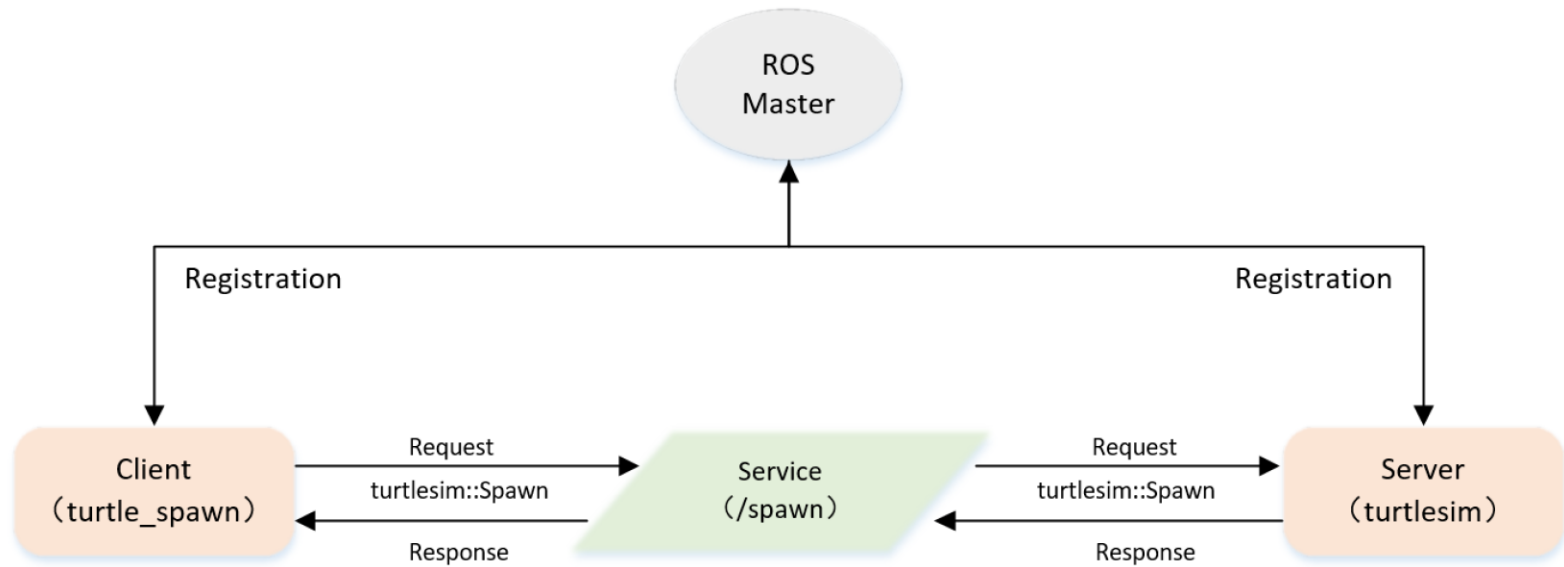
NODES
auto-starting new master
[2470]
```

```
ab@ab-c:~$ rosrun learning_topic person_publisher
[ INFO] [1696312411.912730408]: Publish Person Info: name:Tom age:18 sex:1
[ INFO] [1696312412.915190481]: Publish Person Info: name:Tom age:18 sex:1
[ INFO] [1696312413.962278202]: Publish Person Info: name:Tom age:18 sex:1
[ INFO] [1696312414.915530690]: Publish Person Info: name:Tom age:18 sex:1
[ INFO] [1696312415.917382192]: Publish Person Info: name:Tom age:18 sex:1
[ INFO] [1696312416.920365421]: Publish Person Info: name:Tom age:18 sex:1
[ INFO] [1696312417.918619094]: Publish Person Info: name:Tom age:18 sex:1
[ INFO] [1696312418.921758693]: Publish Person Info: name:Tom age:18 sex:1
[ INFO] [1696312419.918754184]: Publish Person Info: name:Tom age:18 sex:1
[ INFO] [1696312420.915418721]: Publish Person Info: name:Tom age:18 sex:1
[ INFO] [1696312421.915691463]: Publish Person Info: name:Tom age:18 sex:1
[ INFO] [1696312422.933296260]: Publish Person Info: name:Tom age:18 sex:1
[ INFO] [1696312423.929112721]: Publish Person Info: name:Tom age:18 sex:1
[ INFO] [1696312424.929409171]: Publish Person Info: name:Tom age:18 sex:1
ab@ab-c:~$
```

```
ab@ab-c:~$ rosrun learning_topic person_subscriber
[ INFO] [1696312415.917559865]: Subscribe Person Info: name:Tom age:18 sex:1
[ INFO] [1696312416.922716047]: Subscribe Person Info: name:Tom age:18 sex:1
[ INFO] [1696312417.919766313]: Subscribe Person Info: name:Tom age:18 sex:1
[ INFO] [1696312418.922685545]: Subscribe Person Info: name:Tom age:18 sex:1
[ INFO] [1696312419.920485106]: Subscribe Person Info: name:Tom age:18 sex:1
[ INFO] [1696312420.916208760]: Subscribe Person Info: name:Tom age:18 sex:1
[ INFO] [1696312421.916781120]: Subscribe Person Info: name:Tom age:18 sex:1
[ INFO] [1696312422.933875723]: Subscribe Person Info: name:Tom age:18 sex:1
[ INFO] [1696312423.930730183]: Subscribe Person Info: name:Tom age:18 sex:1
[ INFO] [1696312424.937896625]: Subscribe Person Info: name:Tom age:18 sex:1
ab@ab-c:~$
```

Lecture 8: Client Program

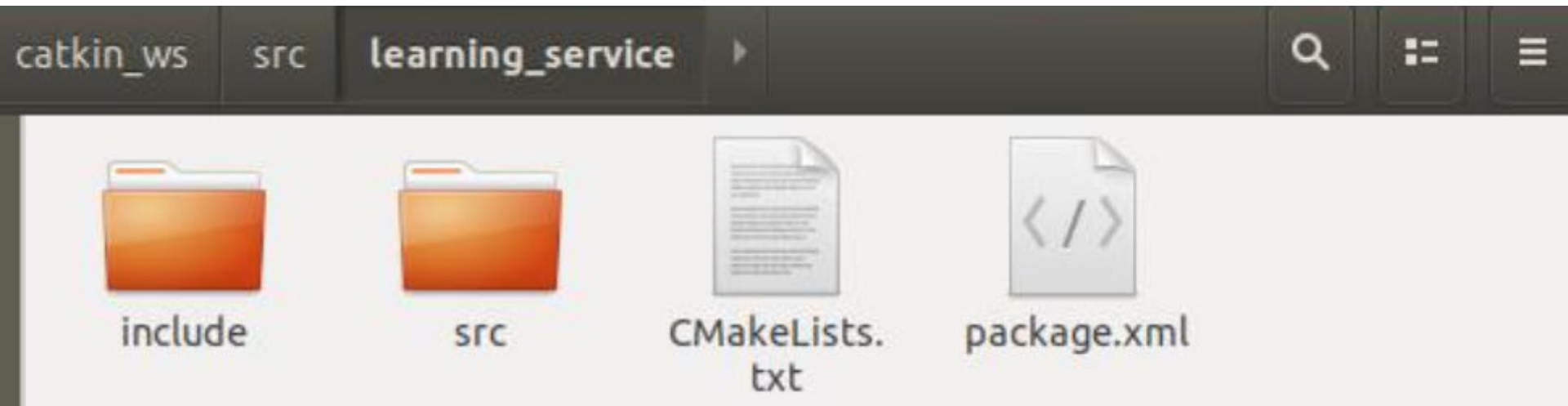
Publisher and Subscriber Compile



Build Package

```
$ cd ~/catkin_ws/src
```

```
$ catkin_create_pkg learning_service roscpp rospy std_msgs geometry_msgs turtlesim
```



C++ Code

```
Open ▼ [icon]
1 #include <ros/ros.h>
2 #include <turtlesim/Spawn.h>
3
4 int main(int argc, char** argv)
5 {
6
7     ros::init(argc, argv, "turtle_spawn");
8
9
10    ros::NodeHandle node;
11
12
13    ros::service::waitForService("/spawn");
14    ros::ServiceClient add_turtle = node.serviceClient<turtlesim::Spawn>("/spawn");
15
16
17    turtlesim::Spawn srv;
18    srv.request.x = 2.0;
19    srv.request.y = 2.0;
20    srv.request.name = "turtle2";
21
22
23    ROS_INFO("Call service to spawn turtle[x:%0.6f, y:%0.6f, name:%s]",
24            srv.request.x, srv.request.y, srv.request.name.c_str());
25
26    add_turtle.call(srv);
27
28
29    ROS_INFO("Spawn turtle successfully [name:%s]", srv.response.name.c_str());
30
31    return 0;
32 };
```

Compiling Code

```
## Declare a C++ executable
## With catkin_make all packages are built within a single CMake context
## The recommended prefix ensures that target names across packages don't collide
# add_executable(${PROJECT_NAME}_node src/learning_service_node.cpp)

## Specify libraries to link a library or executable target against
# target_link_libraries(${PROJECT_NAME}_node
#   ${catkin_LIBRARIES}
# )

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

Compiling Code

```
$ cd ~/catkin_ws
```

```
$ catkin_make
```

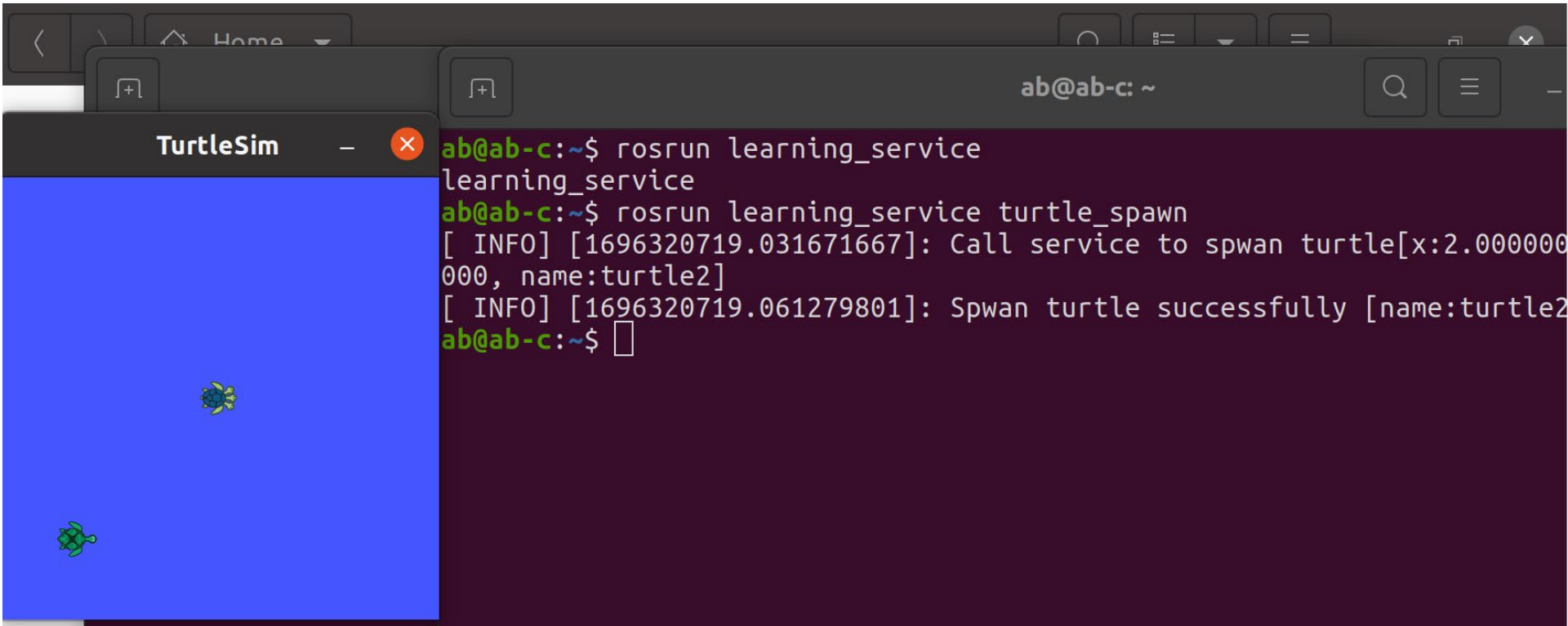
```
$ source devel/setup.bash
```

```
$ roscore
```

```
$ rosrun turtlesim turtlesim_node
```

```
$ rosrun learning_service turtle_spawn
```


Compiling Code



Python Code

```
Open ▼ 
1 import sys
2 import rospy
3 from turtlesim.srv import Spawn
4
5 def turtle_spawn():
6
7     rospy.init_node('turtle_spawn')
8
9
10    rospy.wait_for_service('/spawn')
11    try:
12        add_turtle = rospy.ServiceProxy('/spawn', Spawn)
13
14
15        response = add_turtle(2.0, 2.0, 0.0, "turtle2")
16        return response.name
17    except rospy.ServiceException, e:
18        print "Service call failed: %s"%e
19
20 if __name__ == "__main__":
21
22     print "Spwan turtle successfully [name:%s]" %(turtle_spawn())
23
24
```

Reference



Free

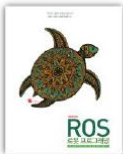


Download link



Language:

English, chinese, Japanese, Korean



“ROS Robot Programming”

A Handbook is written by TurtleBot3 Developers

Reference

- ❑ **R. Siegwart, I. R. Nourbakhsh, D. Scaramuzza. Introduction to Autonomous Mobile Robots. MIT Press, 2nd Edition, 2011, ISBN-10: 0262015358.**
- ❑ **Y. Pyo, H. Cho, R. Jung, and T. Lim, ROS Robot Programming, ROBOTIS Co., Ltd., 2017, ISBN 979-11-962307-1-5**
- ❑ **J. O’Kane, A Gentle Introduction to ROS, CreateSpace Independent Publishing Platform, 2013, ISBN-13: 978-1492143239**