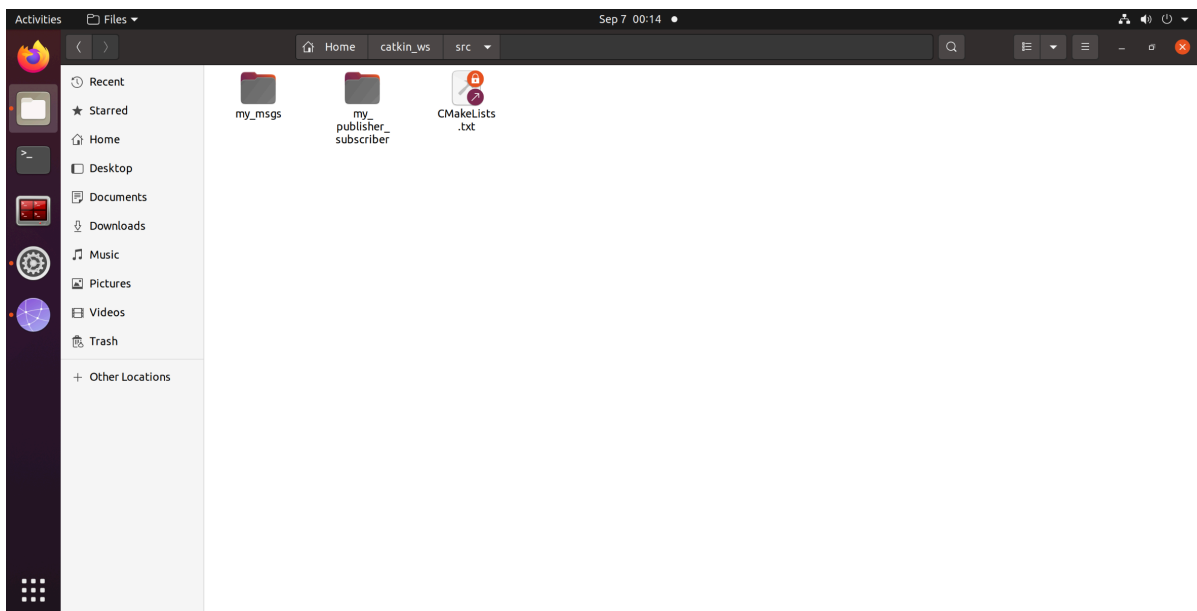
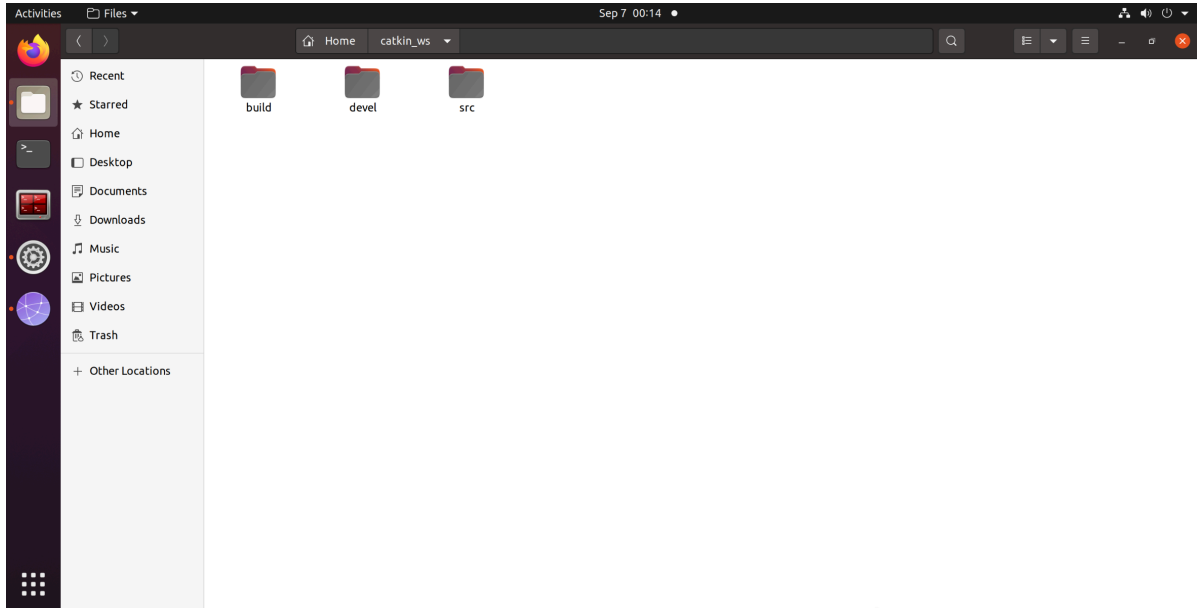
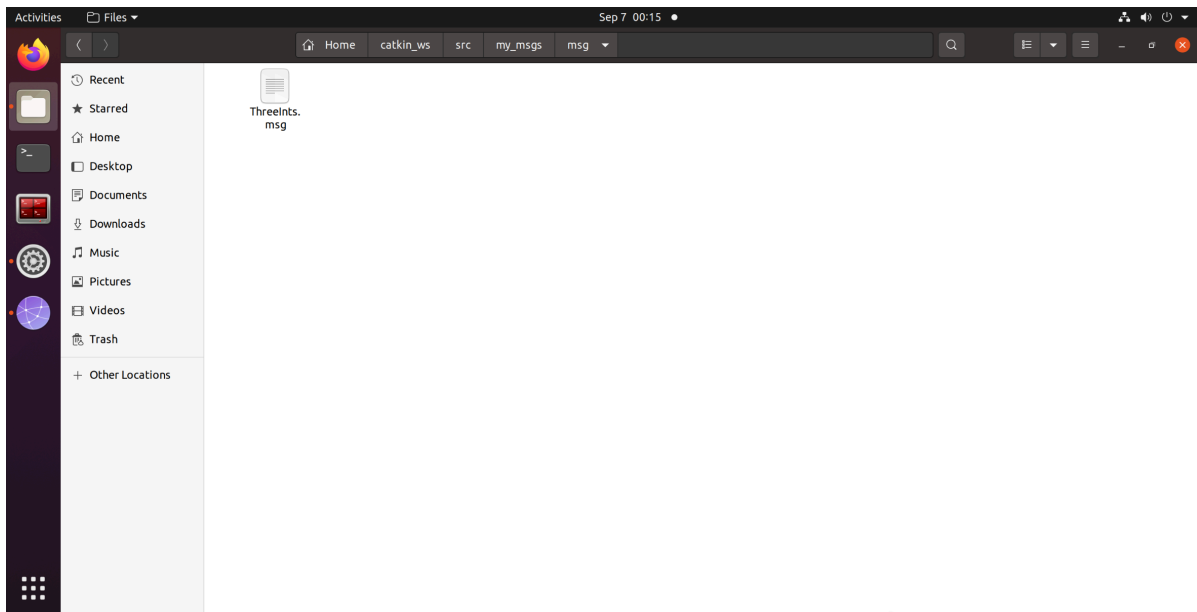


我选择在catkin的工作环境中实现话题通讯，并在其中的src文件夹中创建my_msgs包

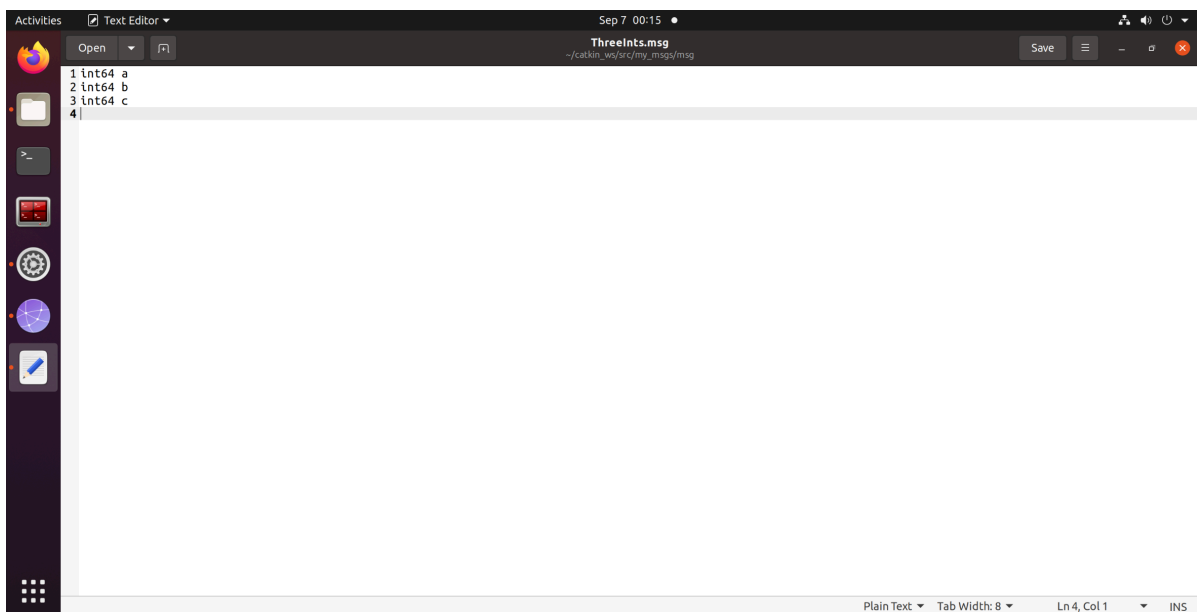


我在尝试话题通讯的过程中，先是将publisher和subscriber文件放在my_msgs中，但是在编译过程中报错，在运行roslaunch时，无法找到这两个文件，所以我选择另设了一个包my_publisher_subscriber用于存放publisher_node.cpp文件和subscriber_node.cpp文件，以及publish_subscribe.launch文件。

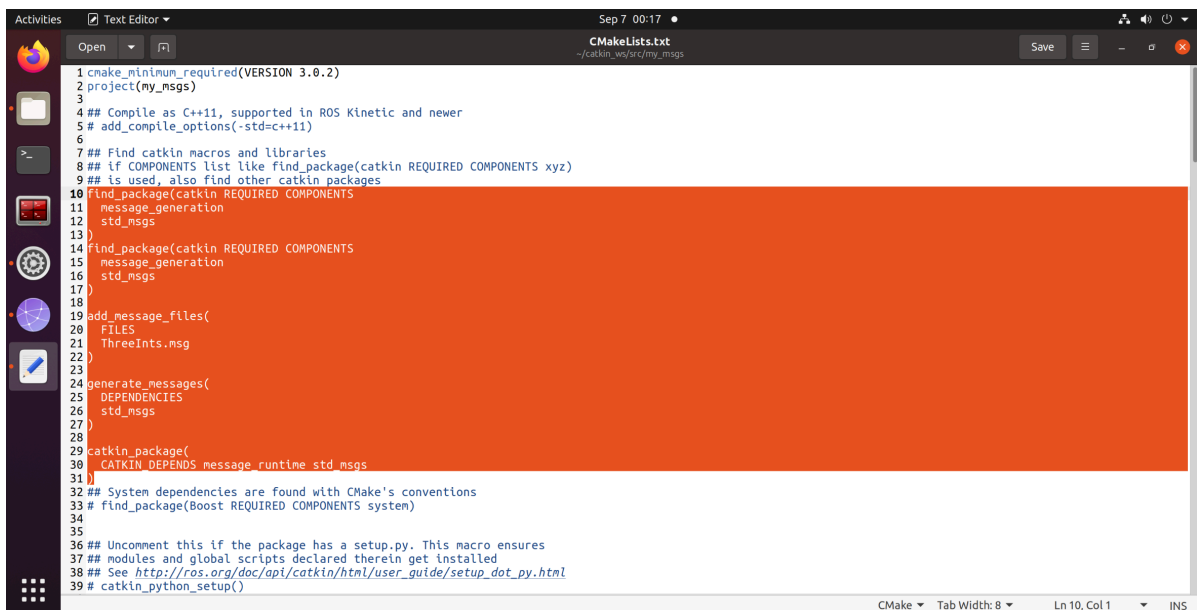
在my_msgs包中的msg文件夹中用touch创建ThreeInts.msg文件，用于定义发送的三个数为int型



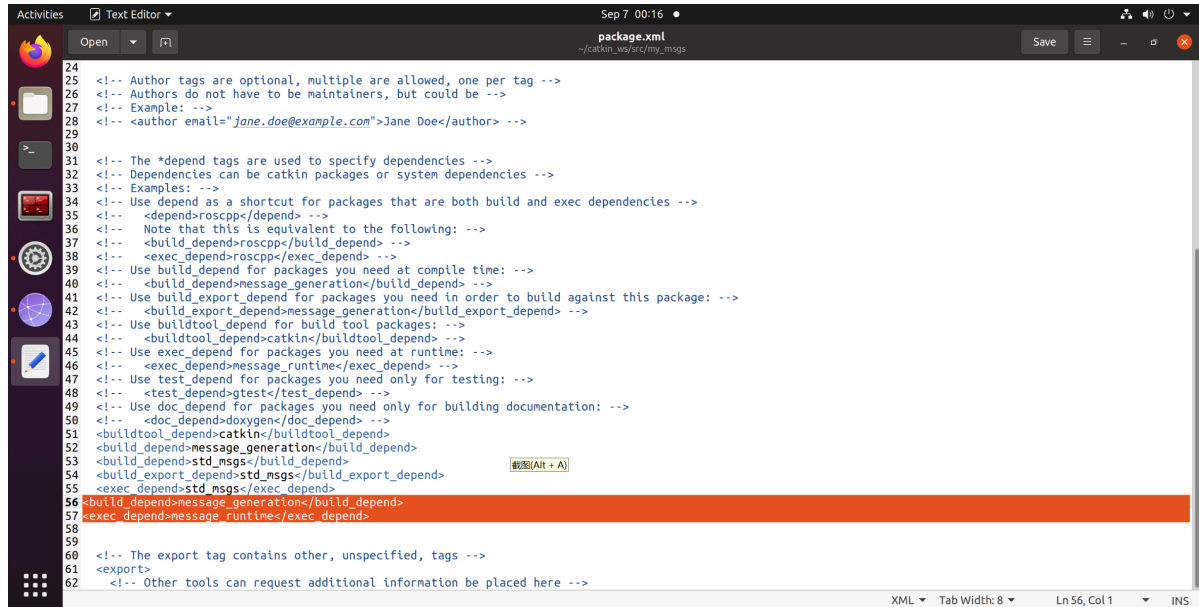
gedit ThreeInts.msg打开输入以下三行代码



并退到my_msgs中，gedit CMakeLists.txt进入添加以下框选的代码，用于添加其对自定义消息的支持



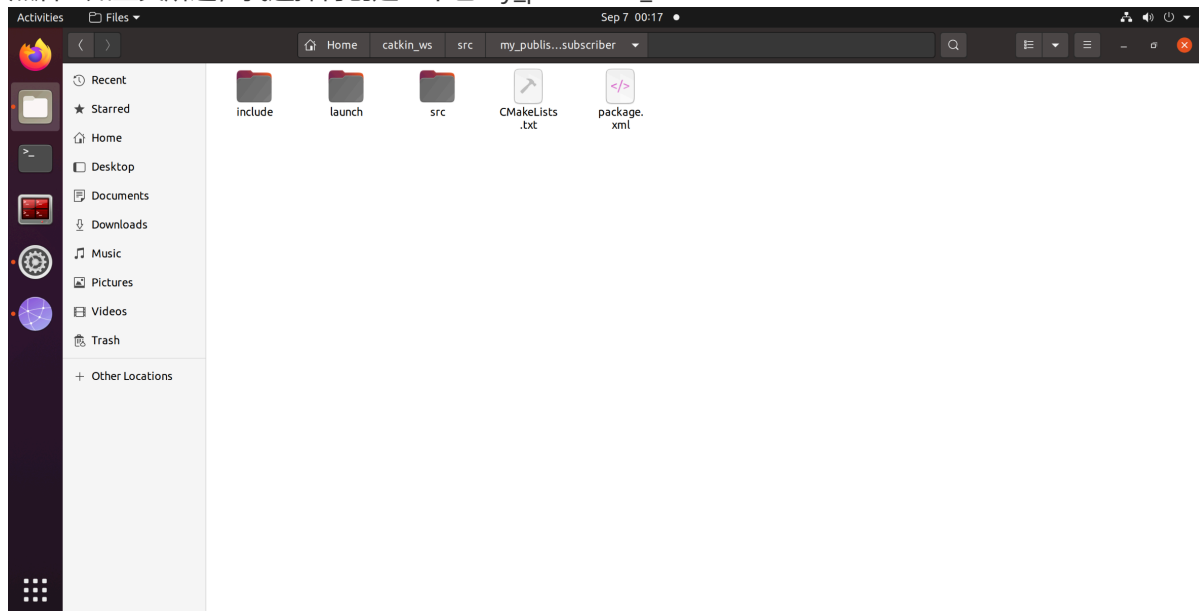
并进入package.xml中添加下列两个依赖



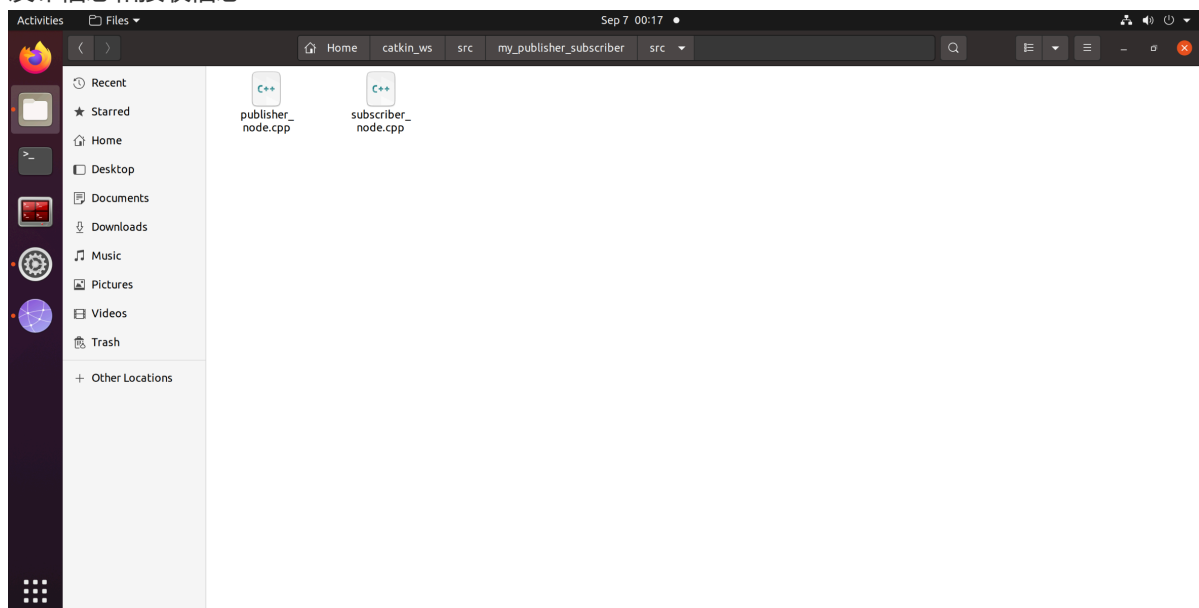
```
24
25 <!-- Author tags are optional, multiple are allowed, one per tag -->
26 <!-- Authors do not have to be maintainers, but could be -->
27 <!-- Example: -->
28 <!-- <author email="jane.doe@example.com">Jane Doe</author> -->
29
30
31 <!-- The *depend tags are used to specify dependencies -->
32 <!-- Dependencies can be catkin packages or system dependencies -->
33 <!-- Examples: -->
34 <!-- Use depend as a shortcut for packages that are both build and exec dependencies -->
35 <!-- <depend>roscpp</depend> -->
36 <!-- Note that this is equivalent to the following: -->
37 <!-- <buildtool_depend>roscpp</buildtool_depend> -->
38 <!-- <exec_depend>roscpp</exec_depend> -->
39 <!-- Use buildtool_depend for packages you need at compile time: -->
40 <!-- <buildtool_depend>message_generation</buildtool_depend> -->
41 <!-- Use build_export_depend for packages you need in order to build against this package: -->
42 <!-- <build_export_depend>message_generation</build_export_depend> -->
43 <!-- Use buildtool_depend for build tool packages: -->
44 <!-- <buildtool_depend>catkin</buildtool_depend> -->
45 <!-- Use exec_depend for packages you need at runtime: -->
46 <!-- <exec_depend>message_runtime</exec_depend> -->
47 <!-- Use test_depend for packages you need only for testing: -->
48 <!-- <test_depend>gtest</test_depend> -->
49 <!-- Use doc_depend for packages you need only for building documentation: -->
50 <!-- <doc_depend>doxygen</doc_depend> -->
51 <buildtool_depend>catkin</buildtool_depend>
52 <buildtool_depend>message_generation</buildtool_depend>
53 <buildtool_depend>std_msgs</buildtool_depend>
54 <build_export_depend>std_msgs</build_export_depend>
55 <exec_depend>std_msgs</exec_depend>
56 <buildtool_depend>message_generation</buildtool_depend>
57 <exec_depend>message_runtime</exec_depend>
58
59
60 <!-- The export tag contains other, unspecified, tags -->
61 <export>
62 <!-- Other tools can request additional information be placed here -->
```

然后在catkin_ws中对其编译，使用命令catkin_make。完成上述后，my_msgs包就已经配置好了。

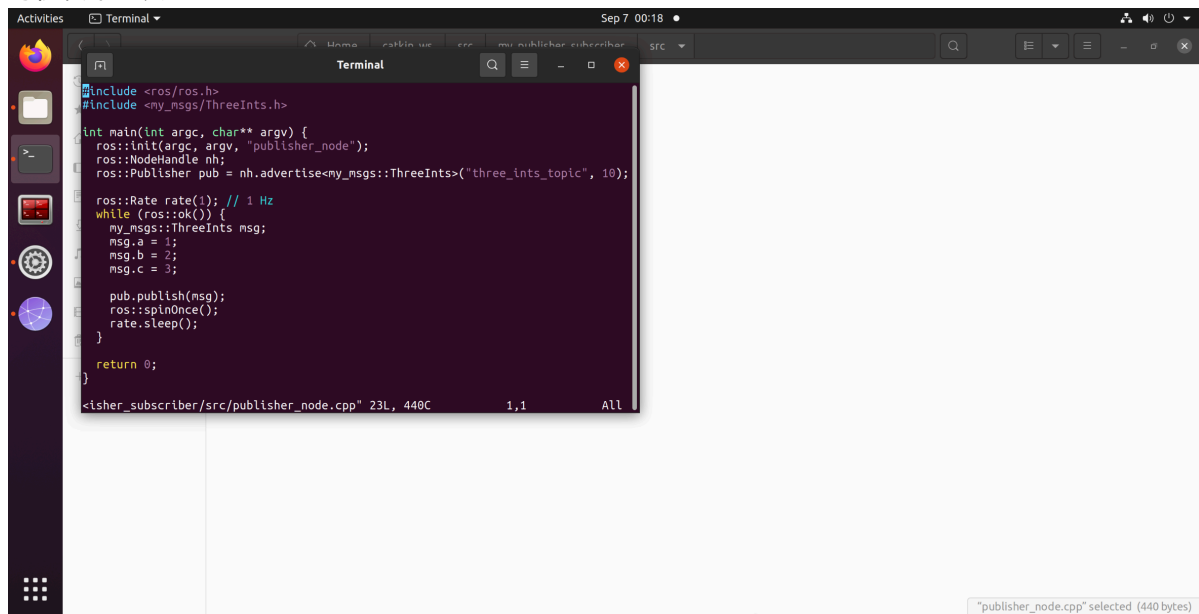
然后正如上文所述，我选择再创建一个包my_publisher_subscriber



并在其src的文件夹中创建了publisher_node.cpp和subscriber_node.cpp两个文件分别用于话题通讯中发布信息和接收信息



publisher_node.cpp中代码实现话题通讯中的发布功能，将此前所定义的int型a,b,c分别定为1，2，3并每秒发布一次



```
#include <ros/ros.h>
#include <my_msgs/ThreeInts.h>

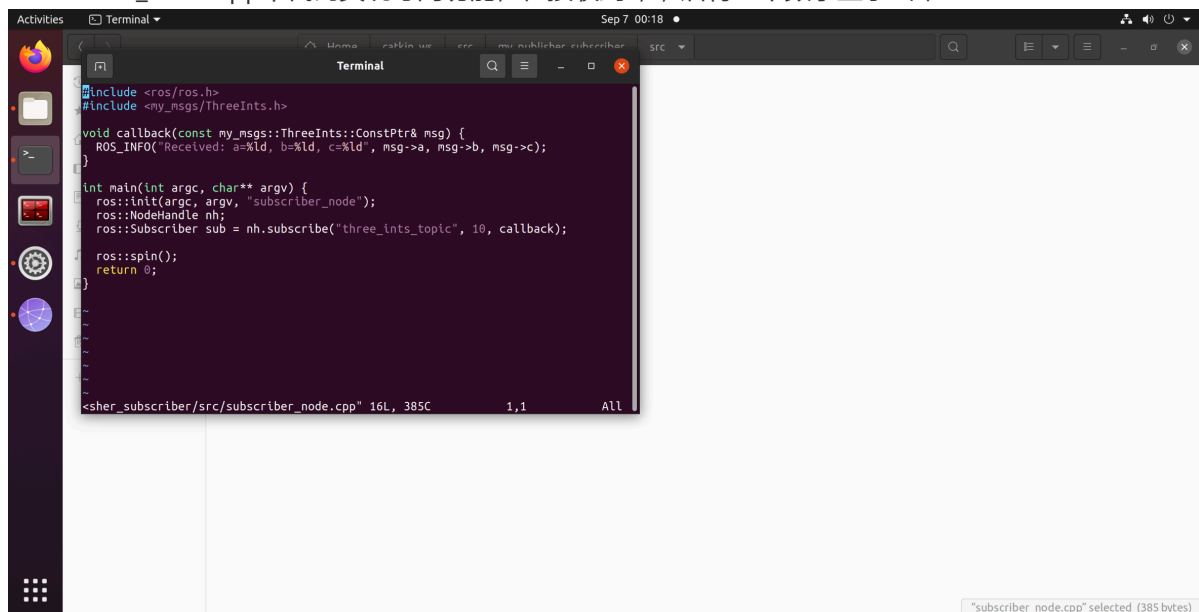
int main(int argc, char** argv) {
    ros::init(argc, argv, "publisher_node");
    ros::NodeHandle nh;
    ros::Publisher pub = nh.advertise<my_msgs::ThreeInts>("three_ints_topic", 10);

    ros::Rate rate(1); // 1 Hz
    while (ros::ok()) {
        my_msgs::ThreeInts msg;
        msg.a = 1;
        msg.b = 2;
        msg.c = 3;

        pub.publish(msg);
        ros::spinOnce();
        rate.sleep();
    }
    return 0;
}
```

publisher_node.cpp selected (440 bytes)

subscriber_node.cpp中代码实现订阅功能，在接收到a,b,c后将三个数字显示出来



```
#include <ros/ros.h>
#include <my_msgs/ThreeInts.h>

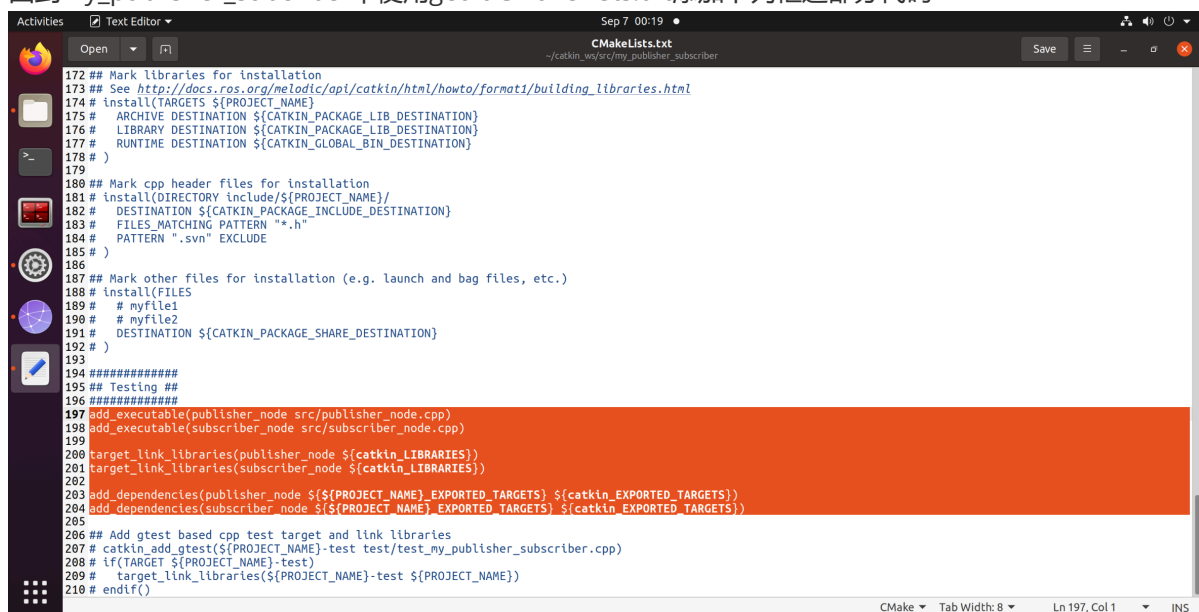
void callback(const my_msgs::ThreeInts::ConstPtr& msg) {
    ROS_INFO("Received: a=%ld, b=%ld, c=%ld", msg->a, msg->b, msg->c);
}

int main(int argc, char** argv) {
    ros::init(argc, argv, "subscriber_node");
    ros::NodeHandle nh;
    ros::Subscriber sub = nh.subscribe("three_ints_topic", 10, callback);

    ros::spin();
    return 0;
}
```

subscriber_node.cpp selected (385 bytes)

回到my_publisher_subscriber中使用gedit CMakeLists.txt添加下列框选部分代码

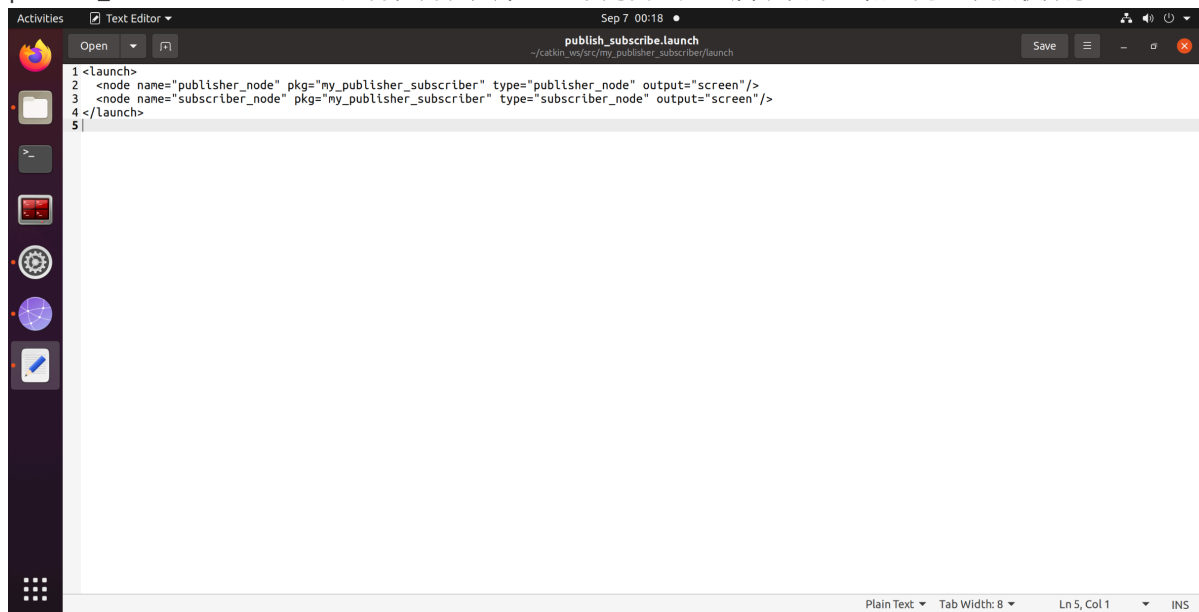


```
172 ## Mark libraries for installation
173 # See http://docs.ros.org/melodic/api/catkin/html/howto/format1/building_libraries.html
174 # install(TARGETS ${PROJECT_NAME}
175 #   ARCHIVE DESTINATION ${CATKIN_PACKAGE_LIB_DESTINATION}
176 #   LIBRARY DESTINATION ${CATKIN_PACKAGE_LIB_DESTINATION}
177 #   RUNTIME DESTINATION ${CATKIN_GLOBAL_BIN_DESTINATION}
178 # )
179
180 ## Mark cpp header files for installation
181 # install(DIRECTORY include/${PROJECT_NAME}/
182 #   DESTINATION ${CATKIN_PACKAGE_INCLUDE_DESTINATION}
183 #   FILES_MATCHING PATTERN "*.h"
184 #   PATTERN ".svn" EXCLUDE
185 # )
186
187 ## Mark other files for installation (e.g. launch and bag files, etc.)
188 # install(FILES
189 #   myfile1
190 #   myfile2
191 #   DESTINATION ${CATKIN_PACKAGE_SHARE_DESTINATION}
192 # )
193
194 #####
195 ## Testing ##
196 #####
197 add_executable(publisher_node src/publisher_node.cpp)
198 add_executable(subscriber_node src/subscriber_node.cpp)
199
200 target_link_libraries(publisher_node ${catkin_LIBRARIES})
201 target_link_libraries(subscriber_node ${catkin_LIBRARIES})
202
203 add_dependencies(publisher_node ${${PROJECT_NAME}_EXPORTED_TARGETS} ${catkin_EXPORTED_TARGETS})
204 add_dependencies(subscriber_node ${${PROJECT_NAME}_EXPORTED_TARGETS} ${catkin_EXPORTED_TARGETS})
205
206 ## Add gtest based cpp test target and link libraries
207 # catkin_add_gtest(${PROJECT_NAME}-test test/test_my_publisher_subscriber.cpp)
208 # if(TARGET ${PROJECT_NAME}-test)
209 #   target_link_libraries(${PROJECT_NAME}-test ${PROJECT_NAME})
210 # endif()
```

CMake Tab Width: 8 Ln 197, Col 1 INS

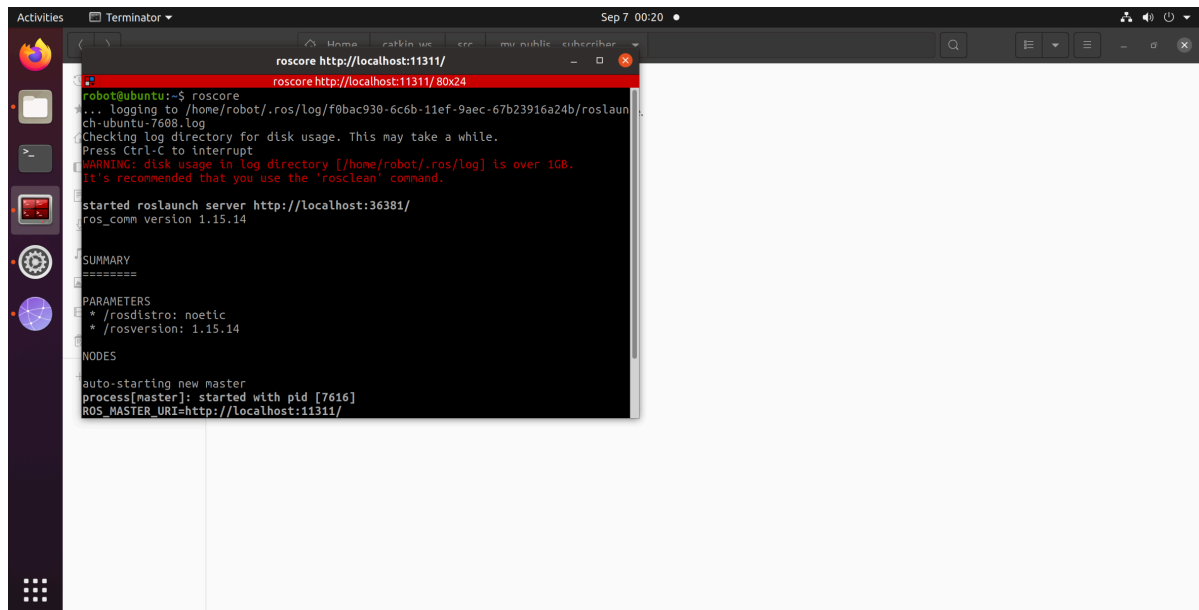
再对工作环境进行编译。

然后仍在my_publisher_subscriber 包中创建一个launch文件夹并在其中创建一个名为publish_subscribe.launch的文件，并在其中写入下列代码用于启动发布器和订阅器以接收数字



```
1<launch>
2  <node name="publisher_node" pkg="my_publisher_subscriber" type="publisher_node" output="screen"/>
3  <node name="subscriber_node" pkg="my_publisher_subscriber" type="subscriber_node" output="screen"/>
4</launch>
5
```

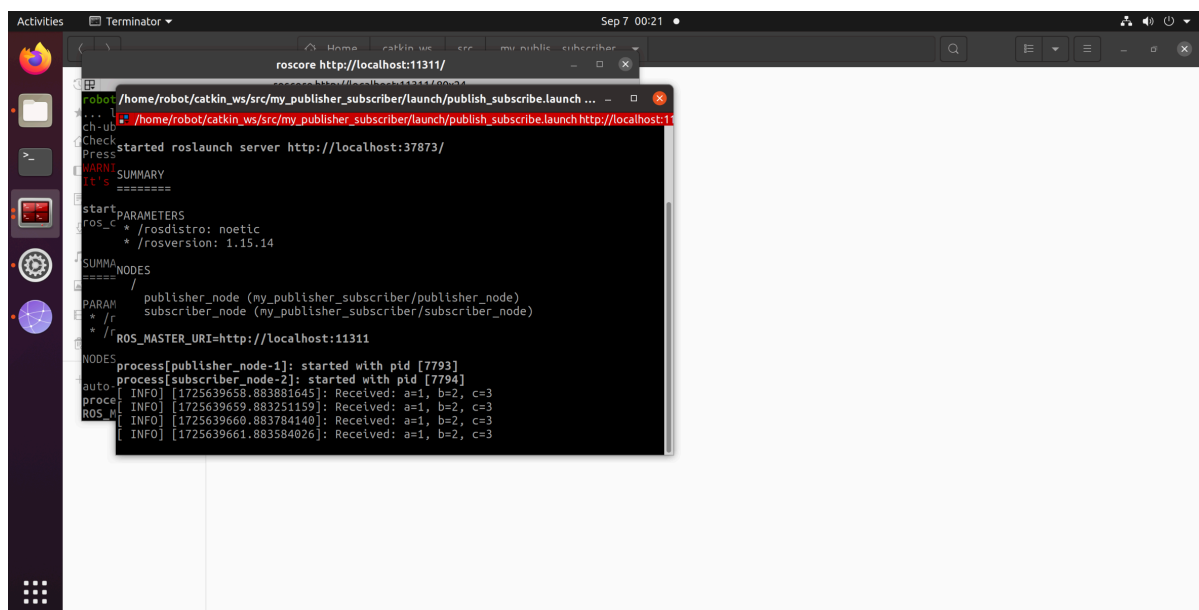
最后输入roscore启动ros,并再打开一个新的终端输入roslaunch my_publisher_subscriber publish_subscribe.launch启动发布器和订阅器



```
robot@ubuntu:~$ roscore
... logging to /home/robot/.ros/log/f0bac930-6c6b-11ef-9aec-67b23916a24b/roslaunch-ubuntu-7608.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
WARNING: disk usage in log directory [/home/robot/.ros/log] is over 1GB.
It's recommended that you use the 'rosclean' command.
started roslaunch server http://localhost:36381/
ros_comm version 1.15.14

SUMMARY
=====
PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.15.14

NODES
auto-starting new master
process[master]: started with pid [7616]
ROS_MASTER_URI=http://localhost:11311/
```



```
robot@ubuntu:~$ roslaunch my_publisher_subscriber publish_subscribe.launch
... logging to /home/robot/.ros/log/f0bac930-6c6b-11ef-9aec-67b23916a24b/roslaunch-ubuntu-7608.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
WARNING: disk usage in log directory [/home/robot/.ros/log] is over 1GB.
It's recommended that you use the 'rosclean' command.
started roslaunch server http://localhost:37873/

SUMMARY
=====
PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.15.14

SUMMARY
=====
PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.15.14

NODES
auto-starting new master
process[master]: started with pid [7616]
ROS_MASTER_URI=http://localhost:11311/

process[publisher_node-1]: started with pid [7793]
process[subscriber_node-2]: started with pid [7794]
INFO [1725639658.883881645]: Received: a=1, b=2, c=3
INFO [1725639659.883251159]: Received: a=1, b=2, c=3
INFO [1725639660.883784140]: Received: a=1, b=2, c=3
INFO [1725639661.883584026]: Received: a=1, b=2, c=3
```

成功接收到了发布的a=1,b=2,c=3，实现话题通讯