

ROS 프로그래밍1

Youngjoon Han
young@ssu.ac.kr

Contents

1. Simple Publisher and Subscriber

1. Writing the Publisher Node
2. 수업과제 : class method publish

1. Simple Publisher and Subscriber

- Beginner Tutorials 패키지로 디렉토리 변경

```
## roscd beginner_tutorials
```

- src 폴더 생성

```
## mkdir -p src
```

- 소스코드 작성

```
## vim talker.cpp
```

1. Simple Publisher and Subscriber

- 코드 작성

```
1 talker.cpp
2 #include "ros/ros.h"
3 #include "beginner_tutorials/simple_msg.h"
4 #include <sstream>
5
6 int main(int argc, char **argv)
7 {
8     ros::init(argc, argv, "talker");
9     ros::NodeHandle n;
10    ros::Publisher chatter_pub = n.advertise<beginner_tutorials::simple_msg>("chatter",
11    1000);
12
13    ros::Rate loop_rate(10);
14    int count = 0;
15    while (ros::ok())
16    {
17        beginner_tutorials::simple_msg msg;
18
19        //gen msg
20        std::stringstream ss;
21        ss << "hello world " << count;
22        msg.data = ss.str();
23
24        //load id from param
25        std::string id;
26        n.param("/chatter_id", id, std::string("no id registered"));
27        msg.id = id;
28
29        ROS_INFO("[%s] : %s ", msg.id.c_str(), msg.data.c_str());
30
31        chatter_pub.publish(msg);
32
33        ros::spinOnce();
34
35        loop_rate.sleep();
36        ++count;
37    }
38
39    return 0;
40 }
```

1. Simple Publisher and Subscriber

- msg 파일 생성

```
## roscd beginner_tutorials  
## mkdir msg  
## vim simple_msg.msg
```

```
1  string id  
2  string data
```

- Subscriber 코드 작성

```
## (src 폴더에서) vim listener.cpp
```

1. Simple Publisher and Subscriber

- subscriber 코드

```
1  #include "ros/ros.h"
2  #include "beginner_tutorials/simple_msg.h"
3  void chatterCallback(const beginner_tutorials::simple_msgConstPtr& ptr)
4  {
5      ROS_INFO("I heard from %s: [%s]", ptr->id.c_str(), ptr->data.c_str());
6  }
7
8  int main(int argc, char **argv)
9  {
10     ros::init(argc, argv, "listener");
11     ros::NodeHandle n;
12
13     ros::Subscriber sub = n.subscribe("chatter", 1000, chatterCallback);
14
15     ros::spin();
16
17     return 0;
18 }
```

1. Simple Publisher and Subscriber

- CMakeLists.txt 수정
 - msg 생성 설정

```
1 cmake_minimum_required(VERSION 2.8.3)
2 project(beginner_tutorials)
3
4 find_package(catkin REQUIRED)
5
6 find_package(catkin REQUIRED COMPONENTS
7   message_generation
8   roscpp
9 )
10
11 add_message_files(FILES
12   simple_msg.msg
13 )
14
15 generate_messages(DEPENDENCIES
16   std_msgs
17 )
```

1. Simple Publisher and Subscriber

- CMakeLists.txt 수정
 - 런타임 의존성 설정

```
catkin_package(  
  CATKIN_DEPENDS  
  message_runtime  
)  
  
include_directories(  
  include ${catkin_INCLUDE_DIRS}  
)
```

- 노드 등록 및 링커 설정

```
31 add_dependencies(talker ${${PROJECT_NAME}_EXPORTED_TARGETS} ${catkin_EXPORTED_TARGETS})  
32 target_link_libraries(talker  
33   ${catkin_LIBRARIES}  
34 )  
35 add_executable(listener  
36   src/listener.cpp  
37 )  
38 add_dependencies(listener ${${PROJECT_NAME}_EXPORTED_TARGETS} ${catkin_EXPORTED_TARGETS})  
39 target_link_libraries(listener  
40   ${catkin_LIBRARIES}  
41 )
```


1. Simple Publisher and Subscriber

- 노드 빌드

```
## cd ~/catkin_ws && catkin_make
```

- talker 노드 실행

```
## rosrun beginner_tutorials talker
```

- listener 노드 실행

```
## rosrun beginner_tutorials listener
```

1. Simple Publisher and Subscriber

- topic msg 확인

```
## rostopic echo /chatter
```

또는 Subscriber를 실행중인 터미널에서 확인

```
[ INFO] [1539048175.689426464]: I heard : [hello world 290]
[ INFO] [1539048175.789193618]: I heard : [hello world 291]
[ INFO] [1539048175.890143926]: I heard : [hello world 292]
[ INFO] [1539048175.989726526]: I heard : [hello world 293]
[ INFO] [1539048176.089349194]: I heard : [hello world 294]
[ INFO] [1539048176.189582726]: I heard : [hello world 295]
[ INFO] [1539048176.289788178]: I heard : [hello world 296]
```

- chatter_id 파라미터 값 수정하기

```
## rosparam set chatter_id sample
```

1. Simple Publisher and Subscriber

아래와 같이 노드를 작성하라.

1. ros_programming1 패키지를 생성한다.
2. Publisher(ros_programming_pub, 주기는 10hz)
 1. 다음과 같은 메시지를 정의한다.

```
1  int32 limit
2  int32 randint
```

2. “/limit”라는 파라미터의 값을 읽는다.(default는 10), 파라미터 값 등록은 터미널을 이용
3. 0 ~ (limit - 1)까지의 값을 무작위로 생성한다.
4. msg의 limit와 randint의 값을 저장해
“ros_practice”라는 이름으로 publish한다. (큐 크기는 1000)

1. Simple Publisher and Subscriber

아래와 같이 노드를 작성하라.

1. Subscriber(ros_programming_sub)

1. “ros_practice” 메시지를 subscribe한 후 정보를 출력한다.
2. subscriber의 기능을 추가한다. (pub 수정 금지)
 - subscriber가 켜진 후 몇 번째 메시지인지 추가한다. (sequence 출력 기능 추가)
 - subscriber한 숫자의 누적 합을 함께 출력한다.
 - 지난 메시지 수신으로 부터 몇 초가 지났는지 함께 출력한다(ros::Time tutorial 또는 API 참

1. Simple Publisher and Subscriber

실행예시 - 순서대로 publisher 메시지 /subscriber 터미널 캡처

```
limit: 20
randint: 11Mbc
---
limit: 20
randint: 10
---
limit: 20
randint: 8
```

```
[ INFO] [1539053268.434045727]: randint, limit : 7 20
[ INFO] [1539053268.434093154]: seq : 1432
[ INFO] [1539053268.434121120]: sum : 13386
[ INFO] [1539053268.434149833]: gap between last msg : 0.098148
[ INFO] [1539053268.535922326]: randint, limit : 10 20
[ INFO] [1539053268.536000160]: seq : 1433
[ INFO] [1539053268.536027684]: sum : 13396
[ INFO] [1539053268.536073199]: gap between last msg : 0.101856
[ INFO] [1539053268.635981581]: randint, limit : 7 20
[ INFO] [1539053268.636059025]: seq : 1434
[ INFO] [1539053268.636085695]: sum : 13403
[ INFO] [1539053268.636132028]: gap between last msg : 0.100059
[ INFO] [1539053268.735997956]: randint, limit : 11 20
[ INFO] [1539053268.736073987]: seq : 1435
[ INFO] [1539053268.736099857]: sum : 13414
[ INFO] [1539053268.736146041]: gap between last msg : 0.100016
[ INFO] [1539053268.835936192]: randint, limit : 9 20
[ INFO] [1539053268.836011570]: seq : 1436
[ INFO] [1539053268.836038331]: sum : 13423
[ INFO] [1539053268.836083979]: gap between last msg : 0.099938
```