



# ROS 3. 노드와 토픽 구독



# Outline



- ▶ 토픽 구독 예제
- ▶ 토픽 구독 노드 실행하기
- ▶ ROS 세부 설명

# 토픽 구독 노드 예제

## ▶ 앞에서 발행되는 토픽을 구독하는 노드

```
#!/usr/bin/env python3
import rospy
from std_msgs.msg import Int32
```

```
def cb_func(msg):
    print(msg.data)
```

```
rospy.init_node('topic_sub')
```

```
sub = rospy.Subscriber('counter', Int32, cb_func)
```

```
rospy.spin()
```

노드 이름

노드 선언

토픽 변수 정의

토픽을 구독하는  
변수로 정의

구독하는 토픽 이름

콜백 함수 이름

# 토픽 구독 노드 예제

## ▶ 앞에서 발행되는 토픽을 구독하는 노드

```
#!/usr/bin/env python3
import rospy
from std_msgs.msg import Int32

def cb_func(msg):
    print(msg.data)

rospy.init_node('topic_sub')

sub = rospy.Subscriber('counter', Int32, cb_func)

rospy.spin()
```

토픽으로 수신한 메시지

메시지 안의 데이터를 출력

토픽을 받았을 때 실행되는 함수

무한 반복하다가 토픽을 받으면 콜백 함수를 실행



# Outline



- ▶ 토픽 구독 예제
- ▶ 토픽 구독 노드 실행하기
- ▶ ROS 세부 설명



# 노드 만들기 – Python



## ▶ 다른 터미널 창 열기

## ▶ 노드 소스 파일 작성

```
user@hostname$ cd ~/my_ws/src/topic_ex_py/scripts  
user@hostname$ gedit topic_subscriber.py
```

## ▶ 실행 파일 지정

```
user@hostname$ chmod +x topic_subscriber.py
```



# 노드 실행



## ▶ 컴파일 및 실행

```
user@hostname$ cd ~/my_ws
```

```
user@hostname$ source devel/setup.bash
```

```
user@hostname$ rosrun topic_ex_py topic_subscriber.py
```

패키지 이름

노드 파일 이름

## ▶ 노드 및 토픽 확인

### ▶ 다른 창 열기

```
user@hostname$ rosnode list
```

```
user@hostname$ rosnode info topic_sub
```

```
user@hostname$ rostopic echo counter -n 5
```



# Outline



- ▶ 토픽 구독 예제
- ▶ 토픽 구독 노드 실행하기
- ▶ ROS 세부 설명



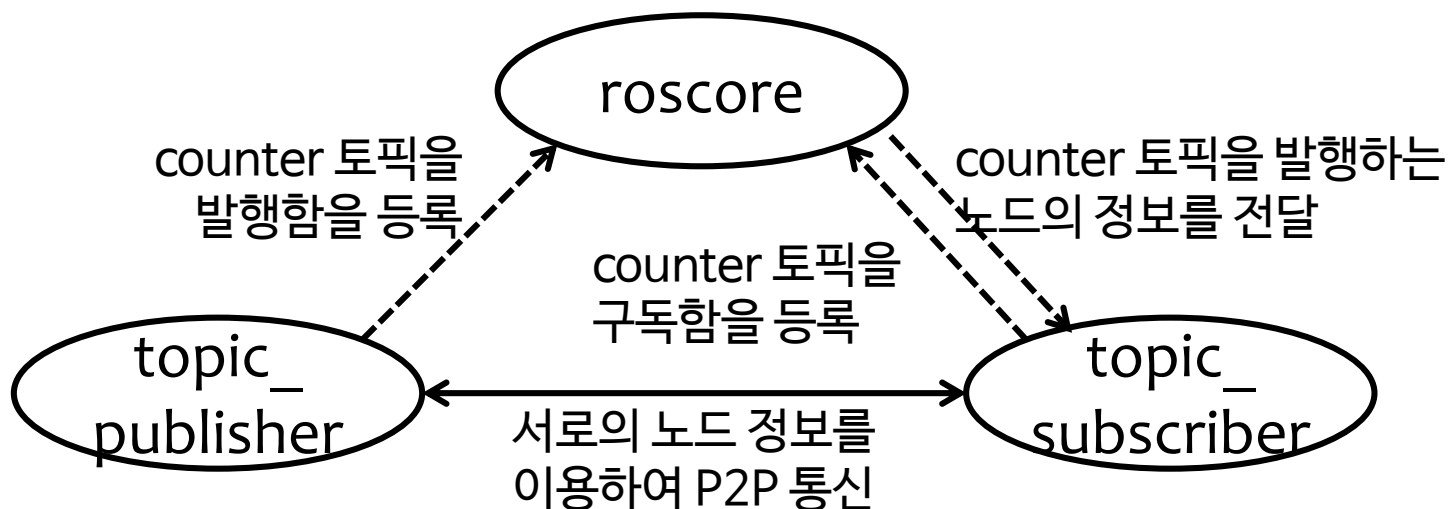


## ▶ roscore

- ▶ ROS 마스터(master): 노드들에게 연결 정보를 제공하여 서로 메시지를 전송할 수 있도록 하는 프로그램
- ▶ roscore는 ROS 마스터 등을 구동하는 명령어임
  - ▶ 이 강의에서는 roscore와 ROS 마스터를 혼용해서 사용함
- ▶ 모든 노드는 시작할 때, 발행하거나 구독하길 원하는 메시지 스트림을 roscore에 등록
- ▶ roscore는 새로운 노드가 나타날 때 같은 메시지 토픽을 발행하거나 구독하는 노드들이 P2P 연결을 형성하는데 필요한 정보를 제공



## ▶ roscore



- ▶ roscore는 노드의 정보를 알려 줄 뿐, 실제로 통신에 관여하지는 않음
- ▶ 노드들은 주기적으로 roscore에 접속하여 상대 노드의 정보를 파악



# 패키지와 캣킨



- ▶ 패키지(package)
  - ▶ ROS 소프트웨어는 코드와 데이터, 문서로 조합된 패키지로 구성됨
  - ▶ 패키지 단위로 공개되거나 공유됨
  - ▶ ROS 생태계에는 수천 가지 패키지가 공개되어 있음
- ▶ 캣킨 (catkin)
  - ▶ catkin\_init\_workspace, catkin\_create\_pkg, catkin\_make, ...
  - ▶ ROS에서 사용하는 빌드 시스템
- ▶ 작업 공간
  - ▶ 소스 코드들을 모아 놓는 공간
  - ▶ 여러 개의 ROS 작업 공간을 가질 수 있지만, 한번에는 하나의 작업 공간에서만 작업해야 함



# 메시지 큐



## ▶ queue size

```
pub = rospy.Publisher('counter', Int32, queue_size=1000)
sub = rospy.Subscriber('counter', Int32, cb_func, queue_size=1000)
```

## ▶ 메시지 큐

- ▶ 토픽 발행: 실제 전송되는 속도보다 더 빨리 발행하는 경우
- ▶ 토픽 구독: 실제 수신되는 속도보다 느리게 토픽을 처리하는 경우
- ▶ 큐의 크기를 넘어서는 경우 오래된 데이터부터 잃어버림

## ▶ 큐의 크기

- ▶ 토픽을 잃어버리면 안 되는 경우: 큐의 크기를 크게
- ▶ 가장 최근에 생성된 토픽이 필요한 경우: 큐의 크기를 작게