ROS 2 스터디 3주차

가천대학교 구본우

9장 패키지 설치와 노드 실행

Turtlesim이란?

- ROS 학습용 패키지인만큼 패키지, 노드, 토픽, 서비스, 액션, 파라미터에 대한 기본적인 학습과 함께 CLI 툴, rqt 툴도 체험해볼 수 있다.

- 실제 로봇이 없더라도 컴퓨터 화면상에서 쉽게 ROS의 기본 개념을 설명하고자 turtlesim이라는 프로그램을 제작하게 되었다.

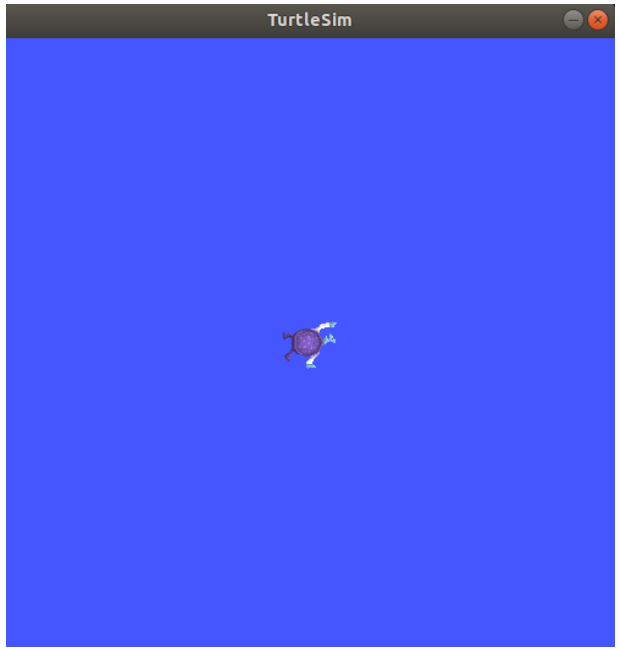
- 개발자, 학생 등의 사이에서 가장 많이 사용되고 있는 ROS의 표준 플랫폼으로 자리 매김하였다.

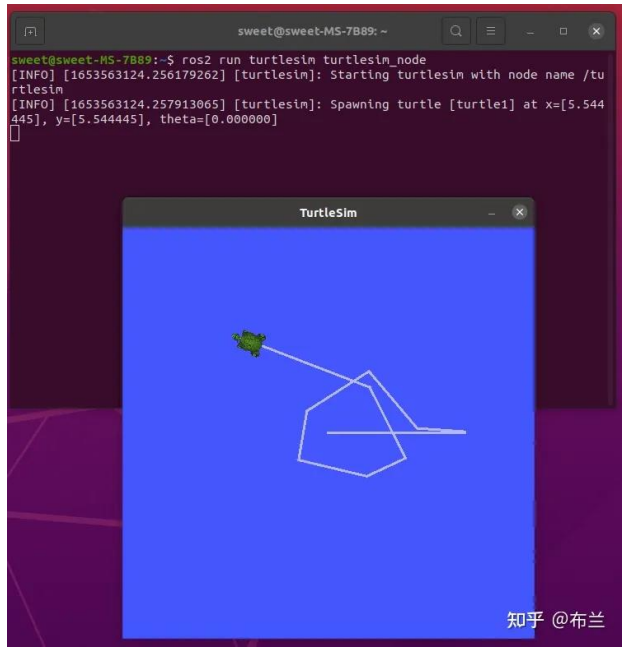
```
$ sudo apt update
$ sudo apt install ros-humble-turtlesim
```

Turtlesim 패키지와 노드 실행

- 노드(Node): 최소 단위의 실행 가능한 프로세스
- 패키지(Package): 하나 이상의 노드 또는 노드 실행을 위한 정보 등을 묶어 놓은 것

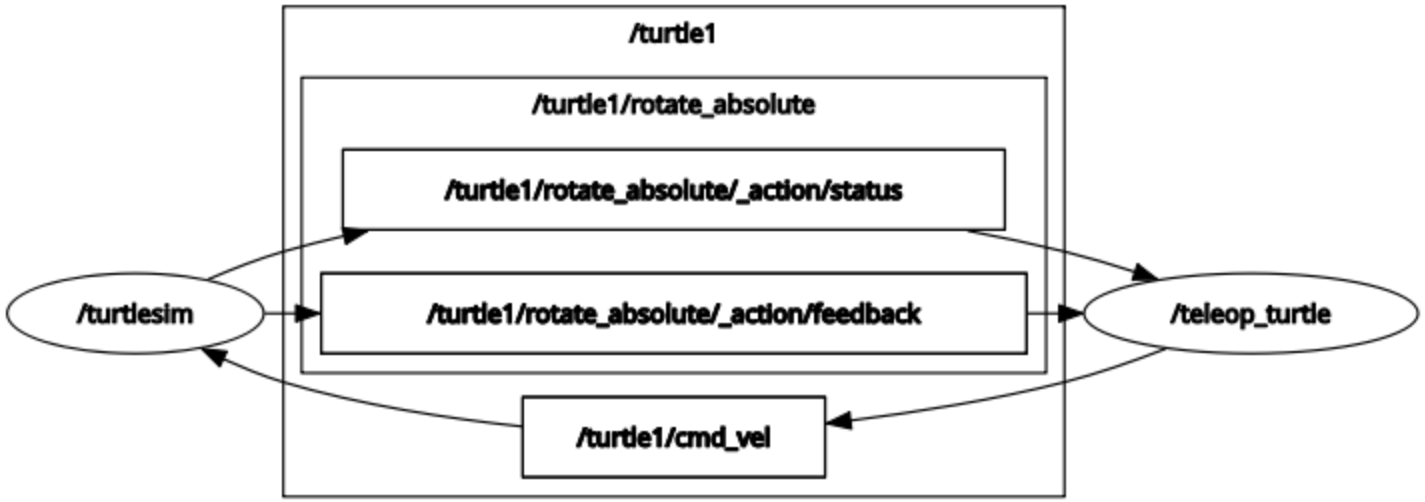
```
$ ros2 run turtlesim turtlesim_node
$ ros2 run turtlesim turtle_teleop_key
```





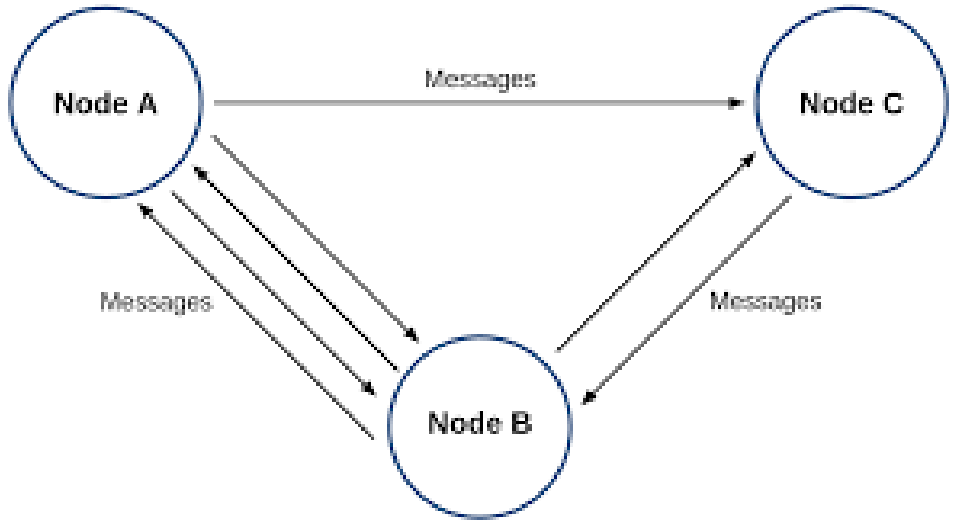
rqt_graph로 보는 노드와 토픽의 그래프 뷰

- 현재 개발환경에서의 모든 노드와 토픽, 액션을 그래프로 확인할 수 있다.
- 동그라미가 노드(Node)이고, 네모는 토픽(Topic) 또는 액션(Action)이다.
- 서비스(Service)는 필요할 때 순간적으로 사용되는 형식이라 표시가 안 되지만, 액션은 토픽과 유사한 퍼블리쉬, 서브스크라이브 통신 방식의 응용되기에 표시되고 있다.



10장 ROS 2 노드와 데이터 통신

- 역할을 목적에 맞추어 세분화시켜 각 노드들 간의 의존성은 줄이고 독립성을 높여 다른 목적의 작업에서도 일부 노드를 재사용할 수 있도록 한다.



노드 실행 & 목록 & 정보

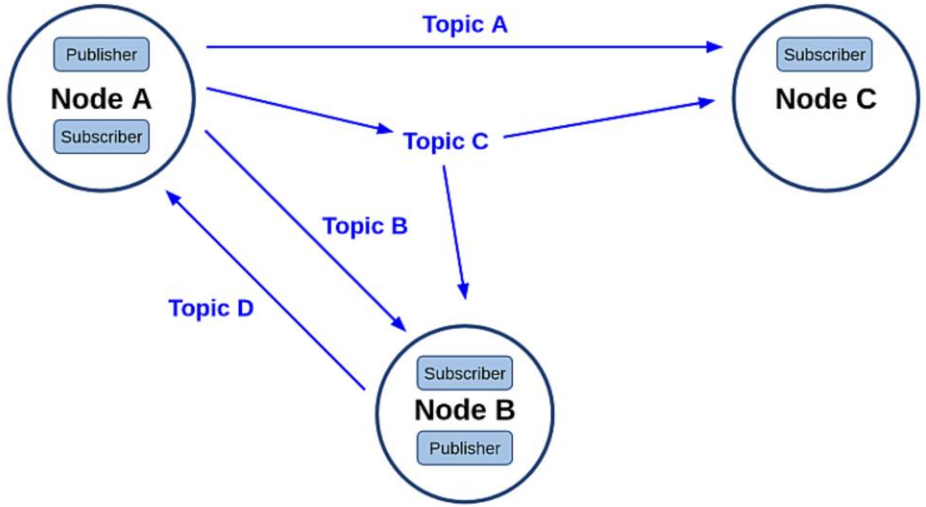
- 하나의 노드 실행: ros2 run, 하나 이상의 노드 실행: ros2 launch
- 또는 rqt, rqt_graph, rviz2와 같이 지정된 실행 명령어도 가능

- 노드 목록 확인 $ ros2 node list

- 노드 정보 확인 $ ros2 node info [node 이름]
- 지정된 노드의 Publishers, Subscriber, Service, Action, Parameter 정보를 확인할 수 있다.

11장 ROS 2 토픽

- 토픽(Topic)은 비동기식 단방향 메시지 송수신 방식으로 msg 인터페이스 형태의 메시지를 퍼블리쉬하는 Publisher와 메시지를 서브스크라이브하는 Subscriber 간의 통신이다.
- 1:1통신을 기본으로 하지만, 1:N, N:N 통신도 가능하다
- 가장 널리 사용되는 방법이다 (비동기성, 연속성)



[그림 2: 다자간 통신]

- 토픽 목록 확인
  - 현재 개발환경에서 동작 중인 모든 노드들의 토픽 정보를 볼 수 있다.

  $ ros2 topic list –t

- 토픽 정보 확인
  - rqt_graph를 이용해 확인하는 방법 외에도 ROS2 CLI를 이용

  $ ros2 topic info /turtle1/cmd_vel

- 토픽 내용 확인
  - 특정 토픽의 메시지 내용을 실시간으로 표시하는 명령

  $ ros2 topic echo /turtle1/cmd_vel

- 토픽 대역폭 확인

  - 메시지의 대역폭, 즉 송수신 받는 토픽 메시지의 크기 확인

  $ ros2 topic bw /turtle1/cmd_vel

- 토픽 주기 확인 (전송 주기)

  $ ros2 topic hz /turtlesim/cmd_vel

- 토픽 지연 시간 확인

  - 토픽은 RMW 및 네트워크 장비를 거치기 때문에 latenc가 반드시 존재한다.

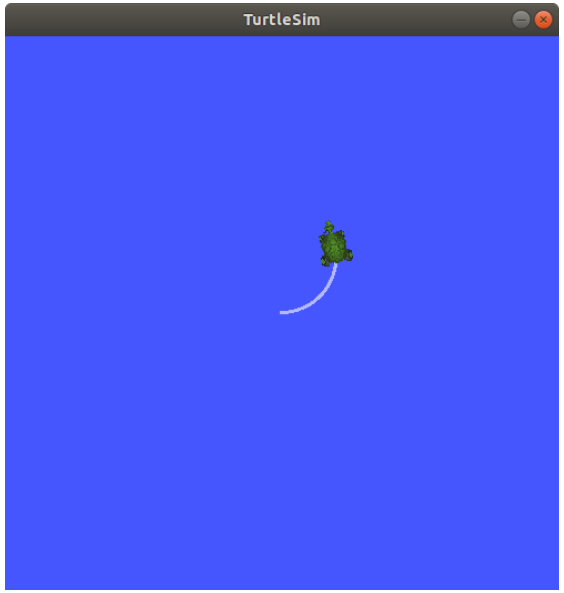
  $ ros2 topic delay /TOPIC_NAME

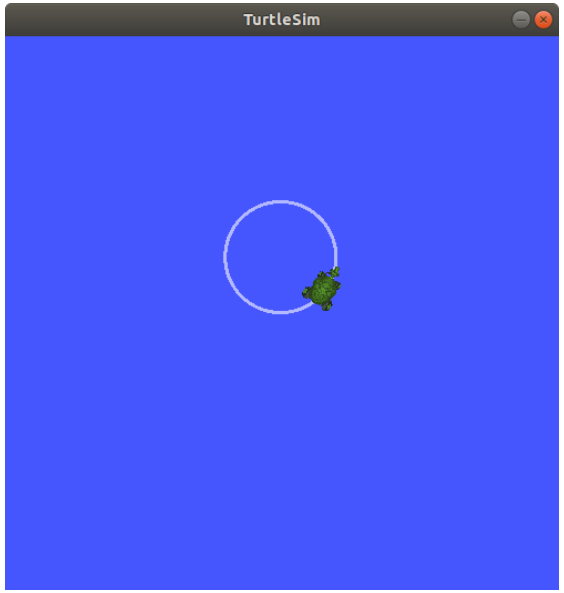
- 토픽 퍼블리시

  - ros2 topic pub <topic_name> <msg_tupe> “<args>”

  $ ros2 topic pub –once /turtle1/cmd_vel geometry_msgs/msg/Twist “{linear: {x: 2.0, y: 0.0, z:0.0}, angular: {x: 0.0, y: 0.0, z: 1.8}}”

  $ ros2 topic pub –rate 1 /turtle1/cmd_vel geometry_msgs/msg/Twist “{linear: {x: 2.0, y: 0.0, z:0.0}, angular: {x: 0.0, y: 0.0, z: 1.8}}”





• Bag 기록 (ros2 bag record)

- 퍼블리시되는 토픽을 파일 형태로 저장하고 필요할 때 저장된 토픽을 다시 불러와 동일한 주기로 재생할 수 있는 기능

- 예를 들어 SLAM 알고리즘을 개발할 때 매번 데이터를 취득하기 어려운 상황에 이용

• Bag 정보 (ros2 bag info)

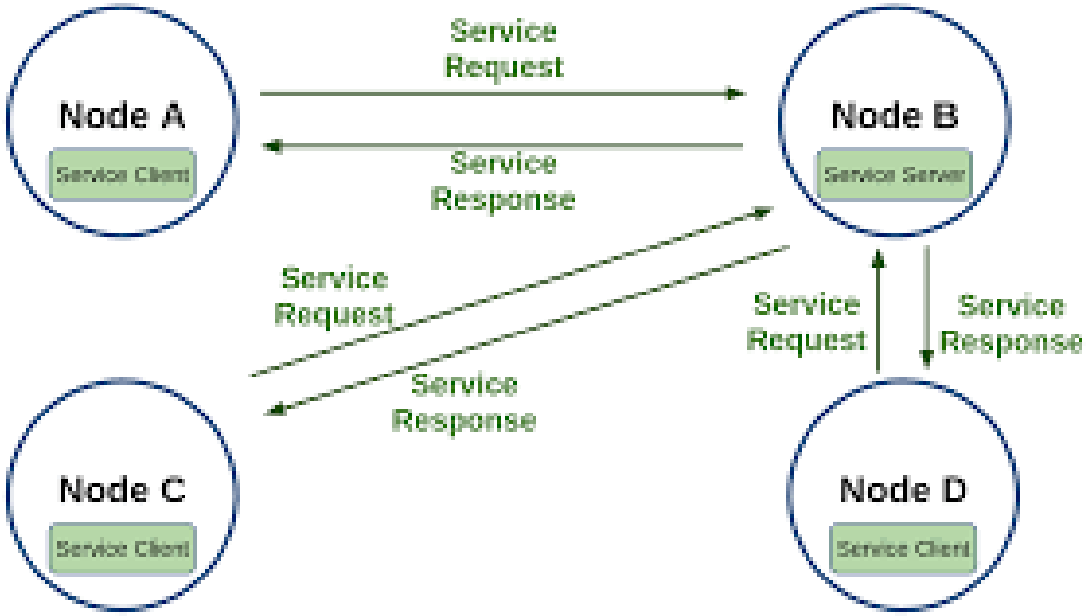
```
$ ros2 bag info rosbag2_2021_05_11-20_48_45/
```

• Bag 재생 (ros2 bag play)

```
$ ros2 bag play rosbag2_2021_05_11-20_48_45/
```

12장 ROS 2 서비스

- 서비스(Service)는 동기식 양방향 메시지 송수신 방식이다.
- 특정 요청을 하는 클라이언트 단과 요청 받은 일을 수행한 후에 결과값을 전달하는 서버 단과의 통신이다.



- 서비스 목록 확인

  - 현재 개발환경에서 동작 중인 모든 노드들의 서비스 정보를 볼 수 있다.

  $ ros2 service list

- 서비스 형태 확인

  - 해당 명령어 뒤에 특정 서비스명을 입력하고 실행하면 어떤 형태의 서비스인지 출력해준다.

  $ ros2 service type [서비스 이름]

  $ ros2 service list –t

  - 서비스 목록과 형태를 함께 볼 수 있다.

- 서비스 찾기

 - 특정 형태를 입력하면 해당 형태의 서비스를 사용하는 서비스 명을 확인할 수 있다.

 $ ros2 service find std_srvs/srv/Empty

- 서비스 요청

 $ ros2 service call <service_name> <service_type> “<arguments>”

turtlesim 내의 서비스

- clear: 이동 궤적 지우기
- kill: 거북이 죽이기
- reset: 다시 거북이 살리기
- set_pen: 궤적의 색, 크기 지정하기
- spawn: 지정한 위치 및 자세로 지정한 이름의 거북이 추가하기