1부 10장_ROS2 노드와 데이터 통신

요약

- ROS2 프로그램 구성 요소
  - 노드
  - 메시지: 노드 간 주고 받는 데이터
  - 메시지 통신: 데이터 통신
    - 토픽, 서비스, 액션, 파라미터

노드와 메시지 통신

- 노드는 최소 단위의 실행 가능한 프로세스를 가리키는 용어이다.
- 각 노드의 역할을 목적에 맞추어 세분화시켜 노드 간의 의존성은 줄이고 독립성을 높여 재사용성을 높여야 한다.
- 노드와 노드 사이에 입출력 데이터를 서로 주고받을 수 있도록 설계해야 한다.
  - 주고 받는 데이터를 메시지 (Message)라고 한다.
    - Integer, Floating point, Boolean, String 데이터
    - 메시지 안에 메시지를 품고 있는 데이터 구조 또는 배열로도 사용할 수 있다.
  - 주고 받는 방식을 메시지 통신 (Message Communication)이라고 한다.
    - 토픽 (Topic), 서비스 (Service), 액션 (Action), 파라미터 (Parameter)

노드 실행

- 두 개의 터미널 창에 아래 명령어를 하나씩 실행 시킨다.

```
ros2 run turtlesim turtlesim_node
```

```
ros2 run turtlesim turtle_teleop_key
```

- rqt_grpah

```
rqt_graph
```

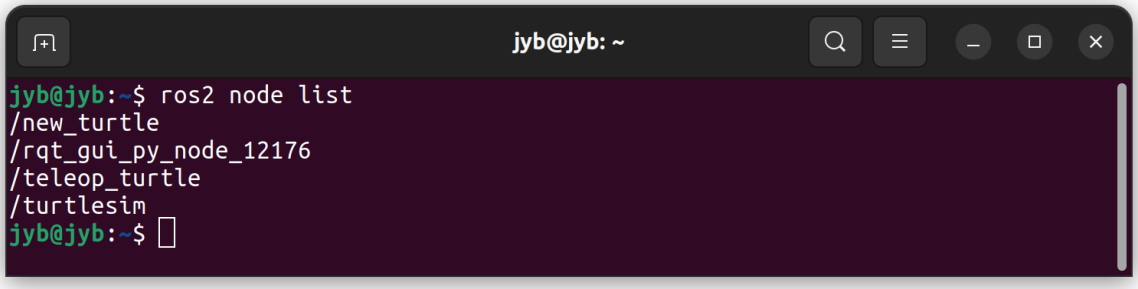
노드 목록

- 아래 명령어로 실행 중인 노드를 확인할 수 있다.
  - turtlesim_node, turtle_teleop_key, rqt_graph 노드가 나와야 한다.
  - rqt_graph는 rqt 복수 개를 실행시킬 수 있어 노드 이름 뒤에 프로세스 아이디가 붙는다.

- 동일 노드를 복수 개 실행시키려면 “ros2 run turtlesim turtlesim_node”을 실행시켜도 되지만 그렇게 되면 동일 이름으로 노드가 실행되어 기존 노드와 구분이 어려워진다. 동일 노드를 중복해서 실행시킬 때에는 노드 이름을 변경하여 실행시키자.

```
ros2 run turtlesim turtlesim_node __node:=new_turtle
```

- 새로운 노드를 추가하여 노드 리스트가 변경되었다.

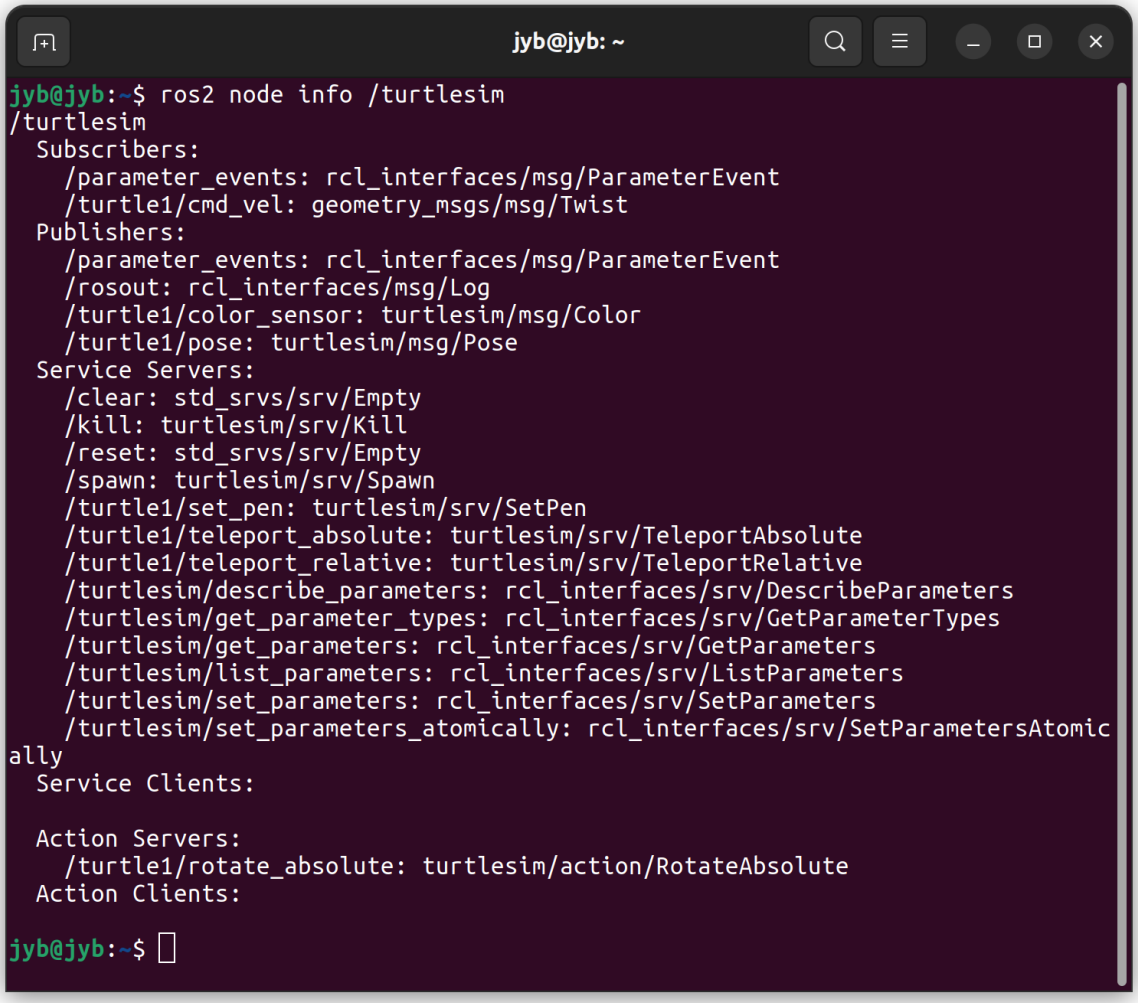


노드 정보

- “ros2 node info” 명령어로 지정된 노드의 Publisher, Subscriber, Service, Action, Parameter 정보를 확인할 수 있다.

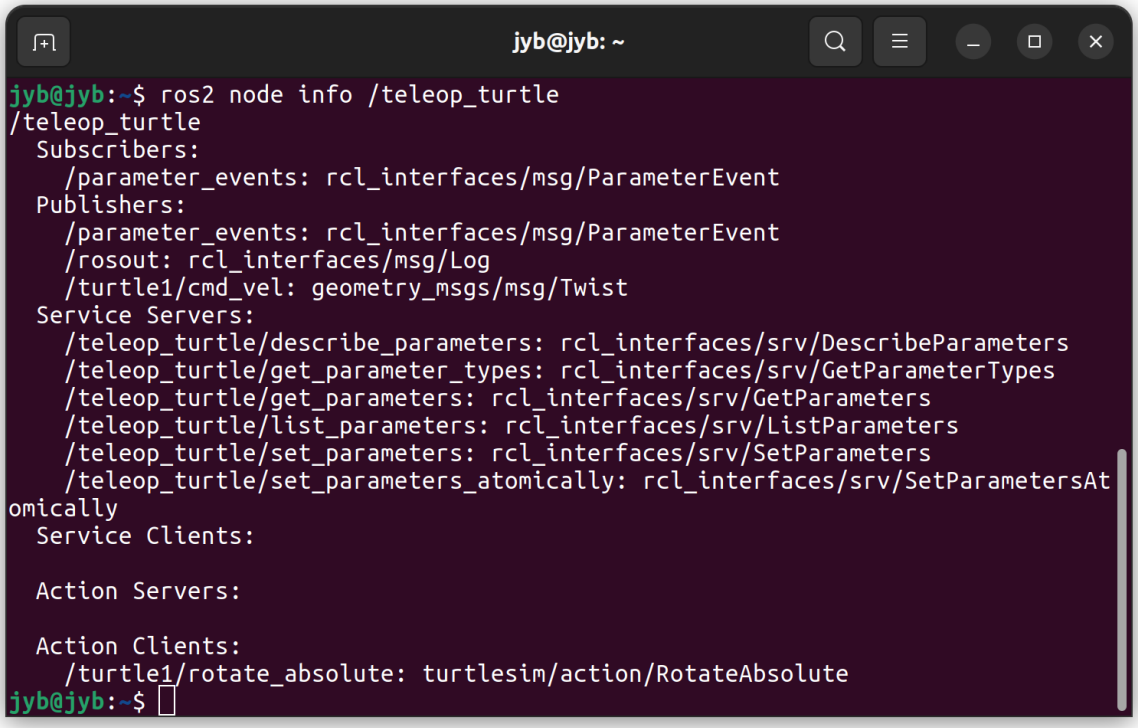
```
ros2 node info /turtlesim
```

```
ros2 node info /teleop_turtle
```


jyb@jyb: ~
jyb@jyb:~$ ros2 node info /turtlesim
/turtlesim
  Subscribers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /turtle1/cmd_vel: geometry_msgs/msg/Twist
  Publishers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /rosout: rcl_interfaces/msg/Log
    /turtle1/color_sensor: turtlesim/msg/Color
    /turtle1/pose: turtlesim/msg/Pose
  Service Servers:
    /clear: std_srvs/srv/Empty
    /kill: turtlesim/srv/Kill
    /reset: std_srvs/srv/Empty
    /spawn: turtlesim/srv/Spawn
    /turtle1/set_pen: turtlesim/srv/SetPen
    /turtle1/teleport_absolute: turtlesim/srv/TeleportAbsolute
    /turtle1/teleport_relative: turtlesim/srv/TeleportRelative
    /turtlesim/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /turtlesim/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /turtlesim/get_parameters: rcl_interfaces/srv/GetParameters
    /turtlesim/list_parameters: rcl_interfaces/srv/ListParameters
    /turtlesim/set_parameters: rcl_interfaces/srv/SetParameters
    /turtlesim/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomic
ally
  Service Clients:

  Action Servers:
    /turtle1/rotate_absolute: turtlesim/action/RotateAbsolute
  Action Clients:

jyb@jyb:~$


jyb@jyb: ~
jyb@jyb:~$ ros2 node info /teleop_turtle
/teleop_turtle
  Subscribers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
  Publishers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /rosout: rcl_interfaces/msg/Log
    /turtle1/cmd_vel: geometry_msgs/msg/Twist
  Service Servers:
    /teleop_turtle/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /teleop_turtle/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /teleop_turtle/get_parameters: rcl_interfaces/srv/GetParameters
    /teleop_turtle/list_parameters: rcl_interfaces/srv/ListParameters
    /teleop_turtle/set_parameters: rcl_interfaces/srv/SetParameters
    /teleop_turtle/set_parameters_atomically: rcl_interfaces/srv/SetParametersAt
omically
  Service Clients:

  Action Servers:

  Action Clients:
    /turtle1/rotate_absolute: turtlesim/action/RotateAbsolute
jyb@jyb:~$