ROS2_day1-3

day1-3

hpp 파일들로 틀을 보자면

- talker.hpp

```
#ifndef TALKER_HPP
#define TALKER_HPP

#include <rclcpp/rclcpp.hpp>
#include <std_msgs/msg/string.hpp>
#include <std_msgs/msg/int64.hpp>

class Talker : public rclcpp::Node {
public:
    Talker();
private:
    void publish_message();
    rclcpp::Publisher<std_msgs::msg::String>::SharedPtr publi
    rclcpp::Publisher<std_msgs::msg::Int64>::SharedPtr count_
    rclcpp::TimerBase::SharedPtr timer_;
};

#endif // TALKER_HPP
```

talker 노드
Talker 클래스 : rclcpp::Node 상속받아 ros2노드로서의 기능 제공

publish_message() : 퍼블리셔 역할을 해줄 함수
timer_; : 별도의 스레드 관리를 안하기 위해(간단한 퍼블리시, 서브스크라이브 기능을 하는 패키지)

- listener.hpp

```
#ifndef LISTENER_HPP
#define LISTENER_HPP
```

```
#include <rclcpp/rclcpp.hpp>
#include <std_msgs/msg/string.hpp>
#include <std_msgs/msg/int64.hpp>

class Listener : public rclcpp::Node {
public:
    Listener();
private:
    void string_callback(const std_msgs::msg::String::SharedP
    void int_callback(const std_msgs::msg::Int64::SharedPtr m
    rclcpp::Subscription<std_msgs::msg::String>::SharedPtr st
    rclcpp::Subscription<std_msgs::msg::Int64>::SharedPtr int
};

#endif // LISTENER_HPP
```

listener 노드

퍼블리시 처럼 서브스크립션 변수 선언
콜백함수
: talker노드에서 퍼블리시된 해당 토픽이 수신될 때 트리거 됨

- talker.cpp

```
Talker::Talker() : Node("talker") {
    publisher_ = this->create_publisher<std_msgs::msg::String
    count_publisher_ = this->create_publisher<std_msgs::msg::
    timer_ = this->create_wall_timer(std::chrono::seconds(1),
        std::bind(&Talker::publish_message, this));
}
```

publisher_, count_publisher_ : 퍼블리셔로써 각각 String, Int64를 보낸다, 토픽 명은 각각
"/chatter_cli", "/chatter_count"이다.

timer_ = this→create_wall_timer(std::chrono::seconds(1),
std::bind(&Talker::publish_message, this));
: 주기적으로 트리거 되는 타이머를 생성하는 메서드 호출

std::chrono::seconds(1) : 콜백 간의 간격 지정

std::bind(&Talker::publish_message, this) : 멤버 함수 publish_message()를 현재 인스턴스(this)에 바인딩(특정 함수나 변수를 다른 요소에 연결하는 것을 의미, 멤버함수와 인스턴스를 바인딩하면 해당 함수 호출시 어느 인스턴스에서 실행되어야 하는지 명확이 지정 가능) 하는데 사용, 이를 통해 타이머가 트리거 될 때 publish_message함수가 올바른 객체에서 호출 되도록한다.

create_wall_timer는 호출 가능한 객체를 기대하기 때문에 std::bind 가 필요하며, publish_message()는 호출될 인스턴스가 필요한 멤버 함수이기 때문입니다.

create_wall_timer : 타이머를 생성

std::bind : 멤버 함수를 호출 가능한 객체로 만들어주기 위해 사용, 멤버 함수는 클래스 인스턴스와 함께 호출되어야 하기 때문에, this 포인터를 바인딩하여 publish_message함수가 Talker 클래스의 인스턴스에서 호출

```
std::cin.ignore(std::numeric_limits<std::streamsize>::max(),

    publisher_->publish(msg);
    count_publisher_->publish(count_msg);
```

카운트 값을 읽은 후, std::cin.ignore를 사용하여 입력 버퍼를 지워, 입력에 방해가 되지 않도록

- listener.cpp

```
Listener::Listener() : Node("listener") {
    string_subscription_ = this->create_subscription<std_msgs
    ("/chatter_cli", 10, std::bind(&Listener::string_callback

    int_subscription_ = this->create_subscription<std_msgs::m
    ("/chatter_count", 10, std::bind(&Listener::int_callback,
}

void Listener::string_callback(const std_msgs::msg::String::S
    RCLCPP_INFO(this->get_logger(), "Subscribed: '%s'", msg->
}

void Listener::int_callback(const std_msgs::msg::Int64::Share
```

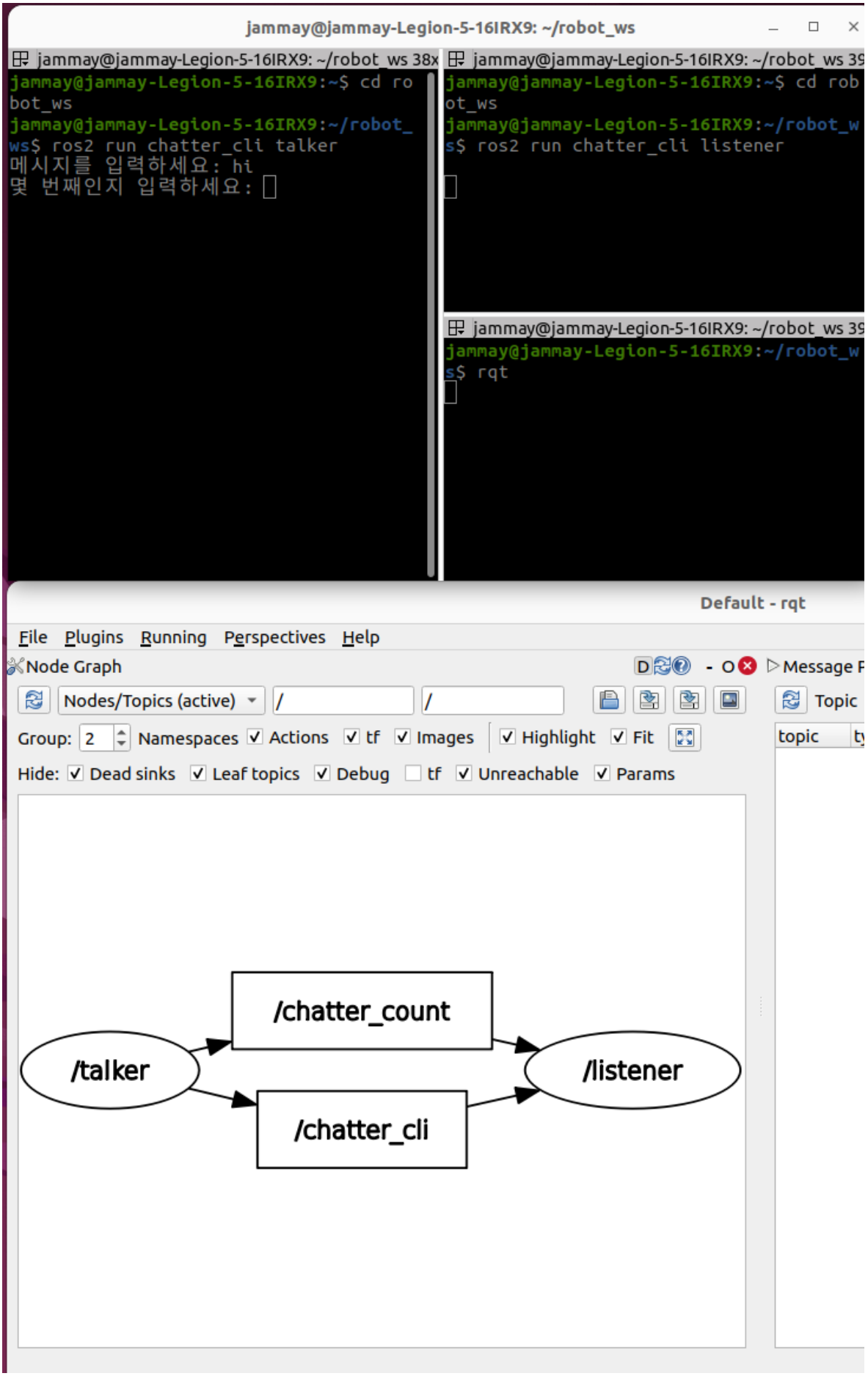
```
    RCLCPP_INFO(this->get_logger(), "Count: '%ld'", msg->data
}
```

string_callback 함수
: /chatter_cli 토픽에서 문자열 메세지를 수진했을 때 호출, 메세지 출력

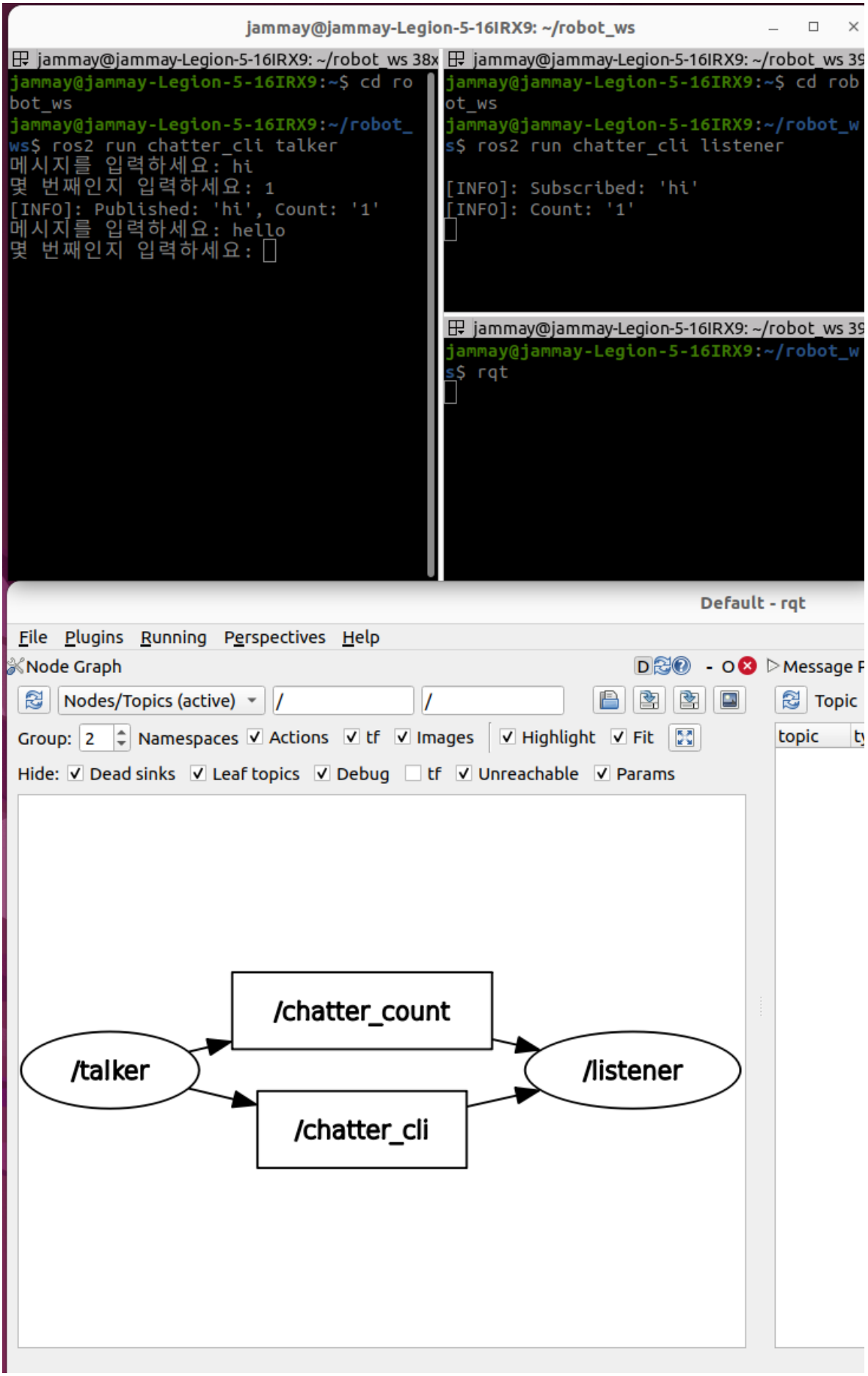
int_callback 함수

: /chatter_count 토픽에서 정수 메세지 수신 시 호출 ⇒ 메세지 출력

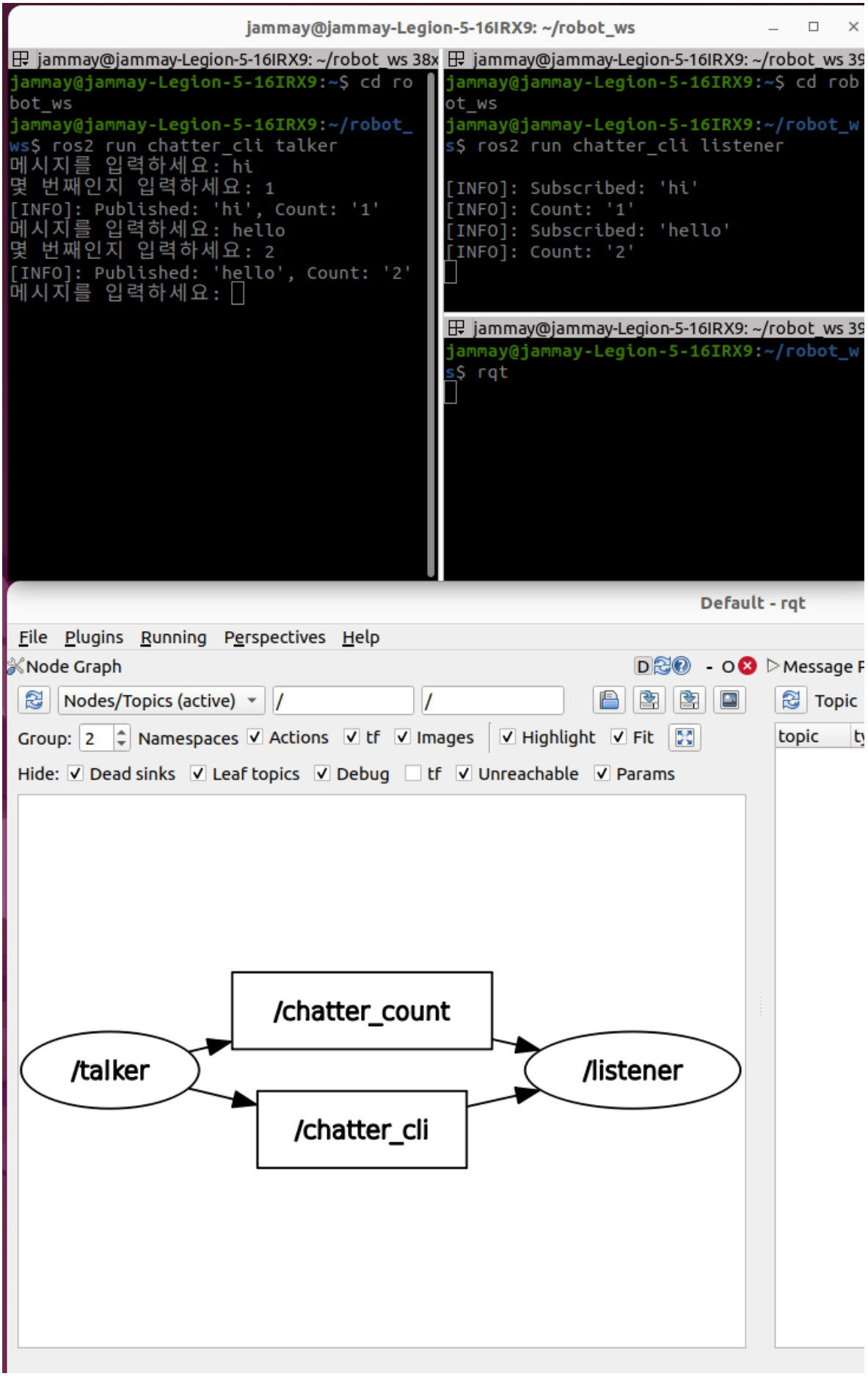
- 첫번째 string만 입력시


jammay@jammay-Legion-5-16IRX9: ~/robot_ws
jammay@jammay-Legion-5-16IRX9: ~/robot_ws 38x
jammay@jammay-Legion-5-16IRX9:~$ cd ro
bot_ws
jammay@jammay-Legion-5-16IRX9:~/robot_
ws$ ros2 run chatter_cli talker
메시지를 입력하세요: hi
몇 번째인지 입력하세요:
jammay@jammay-Legion-5-16IRX9: ~/robot_ws 39
jammay@jammay-Legion-5-16IRX9:~$ cd rob
ot_ws
jammay@jammay-Legion-5-16IRX9:~/robot_w
s$ ros2 run chatter_cli listener
jammay@jammay-Legion-5-16IRX9: ~/robot_ws 39
jammay@jammay-Legion-5-16IRX9:~/robot_w
s$ rqt
Default - rqt
File Plugins Running Perspectives Help
Node Graph
Nodes/Topics (active)
Group: 2 Namespaces Actions tf Images Highlight Fit
Hide: Dead sinks Leaf topics Debug tf Unreachable Params
Topic
topic
/chatter_count
/talker
/listener
/chatter_cli

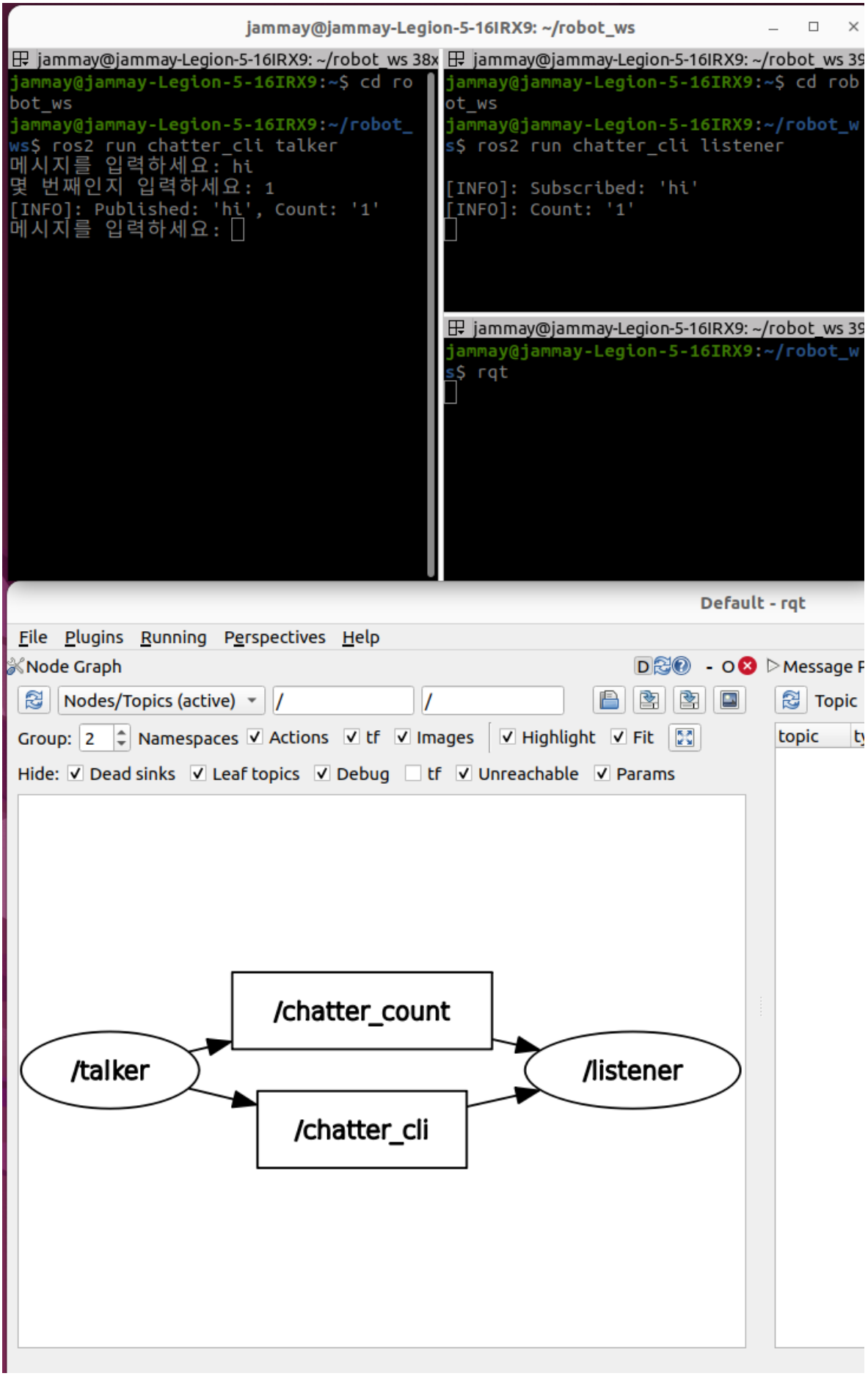
- 첫번째 int 입력시 ⇒ listen 출력


jammay@jammay-Legion-5-16IRX9: ~/robot_ws
jammay@jammay-Legion-5-16IRX9: ~/robot_ws 38x
jammay@jammay-Legion-5-16IRX9:~$ cd robot_ws
jammay@jammay-Legion-5-16IRX9:~/robot_ws$ ros2 run chatter_cli talker
메시지를 입력하세요: hi
몇 번째인지 입력하세요: 1
[INFO]: Published: 'hi', Count: '1'
메시지를 입력하세요: hello
몇 번째인지 입력하세요:
jammay@jammay-Legion-5-16IRX9: ~/robot_ws 39
jammay@jammay-Legion-5-16IRX9:~$ cd robot_ws
jammay@jammay-Legion-5-16IRX9:~/robot_ws$ ros2 run chatter_cli listener
[INFO]: Subscribed: 'hi'
[INFO]: Count: '1'
jammay@jammay-Legion-5-16IRX9: ~/robot_ws 39
jammay@jammay-Legion-5-16IRX9:~/robot_ws$ rqt
Default - rqt
File Plugins Running Perspectives Help
Node Graph
Nodes/Topics (active)
Group: 2 Namespaces Actions tf Images Highlight Fit
Hide: Dead sinks Leaf topics Debug tf Unreachable Params
Topic
topic
/chatter_count
/talker
/listener
/chatter_cli

- 두번째 string 입력


jammay@jammay-Legion-5-16IRX9: ~/robot_ws
jammay@jammay-Legion-5-16IRX9: ~/robot_ws 38x
jammay@jammay-Legion-5-16IRX9:~$ cd robot_ws
jammay@jammay-Legion-5-16IRX9:~/robot_ws$ ros2 run chatter_cli talker
메시지를 입력하세요: hi
몇 번째인지 입력하세요: 1
[INFO]: Published: 'hi', Count: '1'
메시지를 입력하세요: hello
몇 번째인지 입력하세요: 2
[INFO]: Published: 'hello', Count: '2'
메시지를 입력하세요:
jammay@jammay-Legion-5-16IRX9: ~/robot_ws 39
jammay@jammay-Legion-5-16IRX9:~$ cd robot_ws
jammay@jammay-Legion-5-16IRX9:~/robot_ws$ ros2 run chatter_cli listener
[INFO]: Subscribed: 'hi'
[INFO]: Count: '1'
[INFO]: Subscribed: 'hello'
[INFO]: Count: '2'
jammay@jammay-Legion-5-16IRX9: ~/robot_ws 39
jammay@jammay-Legion-5-16IRX9:~/robot_ws$ rqt
Default - rqt
File Plugins Running Perspectives Help
Node Graph
Nodes/Topics (active)
Group: 2 Namespaces Actions tf Images Highlight Fit
Hide: Dead sinks Leaf topics Debug tf Unreachable Params
Topic
topic
/chatter_count
/talker
/listener
/chatter_cli

- 두번째 int 입력 ⇒ listen 출력


jammay@jammay-Legion-5-16IRX9: ~/robot_ws
jammay@jammay-Legion-5-16IRX9: ~/robot_ws 38x
jammay@jammay-Legion-5-16IRX9:~$ cd robot_ws
jammay@jammay-Legion-5-16IRX9:~/robot_ws$ ros2 run chatter_cli talker
메시지를 입력하세요: hi
몇 번째인지 입력하세요: 1
[INFO]: Published: 'hi', Count: '1'
메시지를 입력하세요:
jammay@jammay-Legion-5-16IRX9: ~/robot_ws 39
jammay@jammay-Legion-5-16IRX9:~$ cd robot_ws
jammay@jammay-Legion-5-16IRX9:~/robot_ws$ ros2 run chatter_cli listener
[INFO]: Subscribed: 'hi'
[INFO]: Count: '1'
jammay@jammay-Legion-5-16IRX9: ~/robot_ws 39
jammay@jammay-Legion-5-16IRX9:~/robot_ws$ rqt
Default - rqt
File Plugins Running Perspectives Help
Node Graph
Nodes/Topics (active)
Group: 2 Namespaces Actions tf Images Highlight Fit
Hide: Dead sinks Leaf topics Debug tf Unreachable Params
Topic
topic
/chatter_count
/talker
/listener
/chatter_cli