Lab 1 Part 1b

What is a ROS_DOMAIN_ID?

ROS2 uses Data Distribution Service (DDS) as a communication protocol. ROS_DOMAIN_ID is a kind of logical network in DDS that allows nodes on the same domain to freely communicate with each other without disturbing or being disturbed by other devices.

What is a node?

Nodes are blocks of program that communicate with each other. Each node usually has a single purpose. Nodes communicate with each other by sending and receiving data from each other via topics or services.

What is a topic?

According to the lecture, "Topics are named buses over which nodes exchange messages." Normally, once a node publishes data to a topic and this piece of data will be received by all subscribers of this topic. Topic is how the data is moved and distributed among nodes.

What is a message?

Messages are chunks of information exchanged in between nodes through topics.

What is a subscriber? Write the syntax to create a subscriber that subscribes to the topic amazing_int, which takes message of type UInt64, and uses the callback function magic_fun, in C++ or Python.

A subscriber is a node that receives messages via a topic.

Python Syntax:

```
self.amazing_int_subscriber_ = self.create_subscription(String, "/some
directory/amazing_int", self.magic_fun, )
```

Code for the whole subscriber node:

```
import rclpy
from rclpy.node import Node

# import Topic
from numpy.uintc import amazing_int

# import messages to receive through Topic
from std_msgs.msg import String

class AmazingIntSubscriberNode(Node):
    def __init__(self):
        super().__init__("amazing_int_subscriber")
        self.amazing_int_subscriber_ = self.create_subscription(String, "/some
directory/amazing_int", self.magic_fun, )
        self.subscription
    # call back function called magic_fun(), with arbitary args and contents
    def magic_fun(self, msg: amazing_int):
        self.get_logger().info('msg: ')

def main(args=None):
    rclpy.init(args=args)
    node = AmazingIntSubscriberNode()
    rclpy.spin(node)
    rclpy.shutdwon()
```

What is a publisher? Write the syntax to create a publisher that publishes to the topic amazing_bool, which takes message of type Bool, in Python.

A publisher is a node that sends messages via a topic.

Python Syntax:

```
self.publisher_=self.create_publisher(String, '/some directory/amazing_bool', )
```

Code for the whole publisher node:

```
import rclpy
from rclpy.node import Node

# import Topic
from numpy.bool_ import amazing_bool

# import messsage to publish through Topic
from std_msgs.msg import String

class AmazingBoolPulisherNode(Node):
    def __init__(self):
        super().__init__("amazing_bool_publisher")
        self.publisher_=self.create_publisher(String, '/some
directory/amazing_bool', )
        # timer_period = 0.5

        # no timer callback

def main(args=None):
    rclpy.init(args=args)
    amazing_bool_publisher = AmazingBoolPulisherNode()
    rclpy.spin(amazing_bool_publisher)
    rclpy.shutdown()
```

Can a node have multiple subscribers? Can a node have multiple publishers?

Yes, a node can have multiple subscribers and publishers. What's more, a node can be both a publisher and a subscriber at the same time. A node can publish and subscribe to many different topics.