ÉCOLE NATIONALE SUPÉRIEURE DE TECHNIQUES AVANCÉES

ROB311

Homework 1

Diego Bernardes Rafael Pincer

Palaiseau, France 2024

1 Step by Step

1.1 setting up

After setting up the workspace by following the tutorials and examples, I had already defined my "ros_workspace". Remember that everything was done in the Ubuntu20.04 terminal running on Windows 11 Pro.

```
$ cd ~/ros_workspace/src
```

Inside the workspace, I created the ROS package "homework_1".

```
$ catkin_create_pkg homework_1 std_msgs rospy roscpp
```

In it, I created the "scripts" folder where I saved the code for the 4 nodes.

```
$ mkdir scripts
```

```
$ touch scripts/node_A.py scripts/node_B.py scripts/node_C.py
```

1.2 About the code

It was requested to create a logic similar to 'Chinese whispers,' where one node would receive a message from the previous node and forward it. From there, to define it a bit further, the idea was that node A would pass the message 'test' to node B, which would add 'B' to the end of the message, and so on. When the message returned to node A for the third time, the program would be terminated.

This leaves the code for node A as follows :

```python
#!/usr/bin/env python3

import rospy
from std_msgs.msg import String

# Global counter to track how many times the message comes back to node A
return_count = 0

def callback(msg):
    global return_count
    rospy.loginfo("Node A received: %s", msg.data)

    if return_count < 2:  # Message hasn't come back 3 times yet
        return_count += 1
        rospy.loginfo("Returning message for the %d time", return_count)
```

```
        pub.publish(msg.data + "A")  # Add "A" and publish again
    else:
        rospy.loginfo("Message received for the third time. Stopping...")
        rospy.signal_shutdown("Third reception completed")  # Stop after 3rd time

# Initialize node A
rospy.init_node('node_A')

# Publishes on the topic 'topic_B'
pub = rospy.Publisher('topic_B', String, queue_size=10)

# Subscribes to the topic 'topic_A'
sub = rospy.Subscriber('topic_A', String, callback)

# Publish initial message
rospy.sleep(1)  # Wait for everything to initialize
pub.publish("teste")

# Keeps the node running
rospy.spin()
```

The code for the other nodes doesn't include the cycle-counting part, only the message-receiving and message-publishing task.

```
#!/usr/bin/env python3

import rospy
from std_msgs.msg import String

def callback(msg):
    rospy.loginfo("Node B received: %s", msg.data)
    pub.publish(msg.data + "B")  # Adds "B" to the message

# Initialize node B
rospy.init_node('node_B')

# Publishes on the topic 'topic_C'
pub = rospy.Publisher('topic_C', String, queue_size=10)

# Subscribes to the topic 'topic_B'
sub = rospy.Subscriber('topic_B', String, callback)

# Keeps the node running
```

```
20 rospy.spin()
```

1.3 Code execution

To run the created program, 5 terminals were opened : 1 for the roscore and the others for running the nodes.

To run the roscore :

```
$ cd ros_workspace/
$ roscore
```

To run the nodes, it is done as follows for node A, but replacing the ending for each node :

```
$ rosrun homework_1 node_A.py
```

As a result, the following is seen :

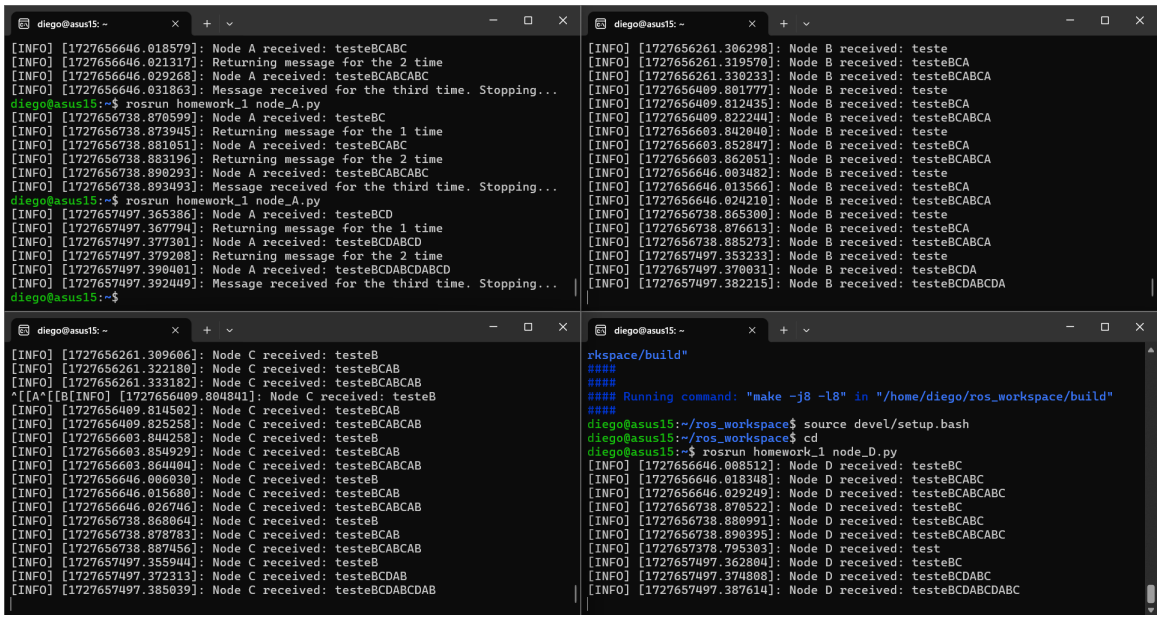

FIGURE 1 – Code execution.

1.4 Git

Once completed, all the material was published on GitHub and can be accessed at the following link :

`https://github.com/diegopincer/ROB311_2025/tree/HW1`

The content relating to this homework assignment is in branch HW1.