실습 5 – 노드의 순차 실행

문제점 : 협업해야 하는 노드를 순서대로 기동시 킬 수 있는가? (node1->node2->node3...)

준비

1) First-five.py 파일, 런치 파일 1개

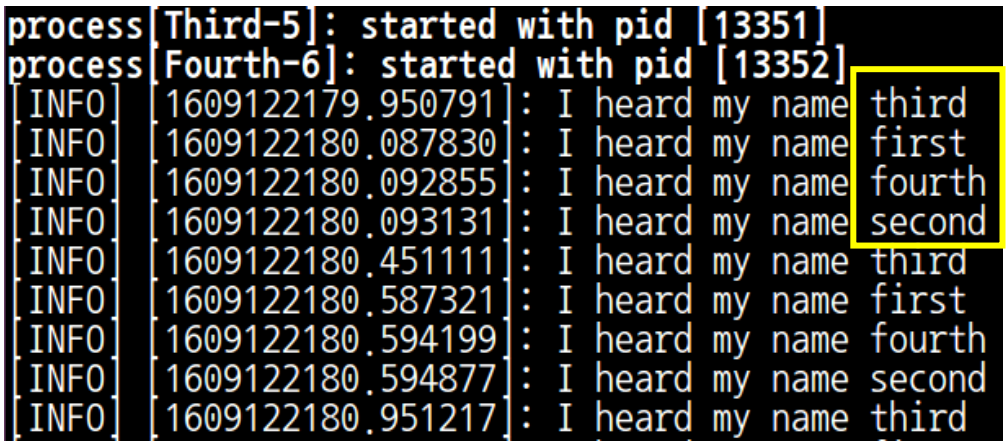
- first-five.py

```
#!/usr/bin/env python
import rospy
from std_msgs.msg import String
import time

rospy.init_node('First')
pub=rospy.Publisher('msg_to_receiver', String, queue_size=10)

rate=rospy.Rate(1)
m_data="first"
while not rospy.is_shutdown():
        # first 부터 fourth까지 0.2씩 차이 나도록 설정
        time.sleep(0.2)
        pub.publish(m_data)

        rate.sleep()
```

- receiver.py

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

import rospy
from std_msgs.msg import String


def callback(msg):
        rospy.loginfo("I heard my name "+ msg.data)


rospy.init_node('Receiver')
sub=rospy.Subscriber('msg_to_receiver', String, callback)

rospy.spin()
```

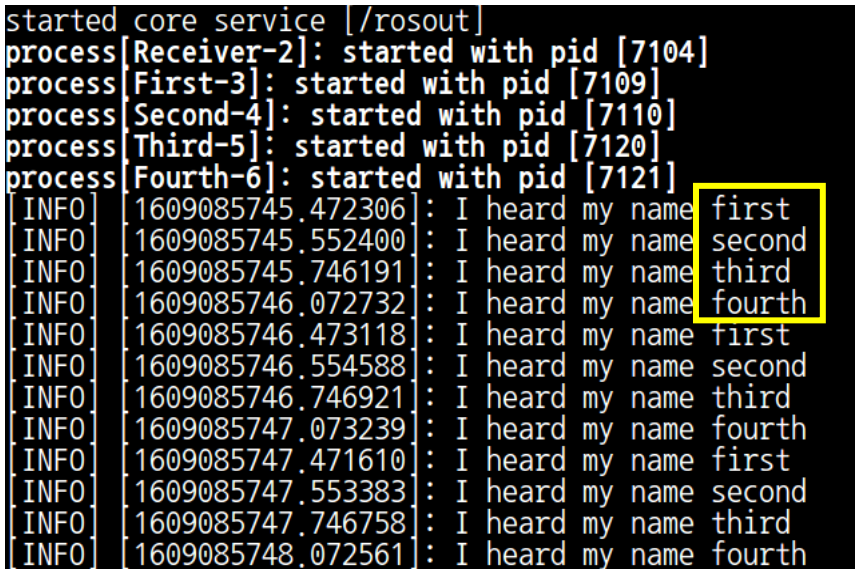
2) First-five.py 순서대로 receiver에 메시지 전송 및 순서확인



3) 해결 방법

rospy.Rata(1)의 값이 바뀌어도 순서대로 출력할 수 있도록 시간 배분을 설정

(위에서 작성한 코드 실행 결과)



- 정확성 높은 방법

Receiver가 보내라는 사인을 주기전까지 각 노드들을 기다리게 함

-receiver.py

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

import rospy
from std_msgs.msg import String

name = "receiver"
pub_topic = "start_ctl"
sub_topic = "msg_to_receiver"

def callback(data):
    rospy.loginfo("I heard %s", data.data)

rospy.init_node(name)
rospy.Subscriber(sub_topic, String, callback)
pub = rospy.Publisher(pub_topic, String, queue_size=1)

rate = rospy.Rate(10)
hello_str = String()


 rospy.sleep(1)

sq = ["first", "second", "third", "fourth"]
pub_msg = String()

for i in sq:
    pub_msg.data = i+":go"
    pub.publish(pub_msg)
    rospy.sleep(3)

rospy.spin()
```

-first-five.py

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

import rospy
from std_msgs.msg import String

name = "first"  # second, third, fourth.py도 이 부분만 수정하여 저장
pub_topic = "msg_to_receiver"
sub_topic = "start_ctl"

OK = None

def ctl_callback(data):
    global OK
    OK = str(data.data)

rospy.init_node(name)
rospy.Subscriber(sub_topic, String, ctl_callback)


while True:
    if OK == None:
        continue
    d = OK.split(":")
    if (len(d) == 2) and (d[0] == name) and (d[1] == "go"):
        break

pub = rospy.Publisher(pub_topic, String, queue_size=1)

rate = rospy.Rate(2)
hello_str = String()

while not rospy.is_shutdown():
    hello_str.data = "my name is " + name
    pub.publish(hello_str)
    rate.sleep()
```