**실습 4 – ROS 타임 슬롯 문제**

문제점 : 주기적 발송에서 타임슬롯을 오버하면 어떻게 되는가?

준비

1) 파이썬 파일 2개

- teacher_int32_job.py

```
import rospy
import timeit
from std_msgs.msg import Int32

rospy.init_node('teacher')
pub = rospy.Publisher('msg_to_students', Int32, queue_size=0)

def do_job(time):
        for i in range(0,time):
                i+=1
                pub.publish(i)
n=int(input('Input :'))
rate = rospy.Rate(5)
count=0

while not rospy.is_shutdown():
        if count==5:
                break
        count+=1
        start=rospy.Time.now().to_sec()
        do_job(n)
        end=rospy.Time.now().to_sec()

        print("---------- "+str(count)+" ----------")
        using_time=end-start
        print("using time : "+ str('%.5f'% using_time))

        wait_time=rospy.Time.now().to_sec()
        rate.sleep()
        print("waiting time : " + str('%.5f'% (rospy.Time.now().to_sec()-
wait_time)))
```

- student_int.py

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

import rospy
from std_msgs.msg import Int32

def callback(msg):
    print msg.data

rospy.init_node('student')

sub = rospy.Subscriber('msg_to_students', Int32, callback)

rospy.spin()
```

-

2) 0.2초씩 다섯 차례 토픽 전송

3) 시간 함수를 이용하여 각 슬롯간 소요시간과 5개의 전체 슬롯의 소요시간 계산 및 출력

• Input=100일 때

```
Input :100
---------- 1 ----------
using time : 0.00792
waiting time : 0.19270
---------- 2 ----------
using time : 0.02323
waiting time : 0.17596
---------- 3 ----------
using time : 0.02424
waiting time : 0.17497
---------- 4 ----------
using time : 0.02698
waiting time : 0.17298
---------- 5 ----------
using time : 0.02395
waiting time : 0.17526
```

• Input=1000일 때

```
Input :1000
---------- 1 ----------
using time : 0.05399
waiting time : 0.14736
---------- 2 ----------
using time : 0.06134
waiting time : 0.13754
---------- 3 ----------
using time : 0.08575
waiting time : 0.11334
---------- 4 ----------
using time : 0.14353
waiting time : 0.05669
---------- 5 ----------
using time : 0.06940
waiting time : 0.12963
```

• Input=5000일 때

```
Input :5000
---------- 1 ----------
using time : 0.22738
waiting time : 0.00009
---------- 2 ----------
using time : 0.31467
waiting time : 0.00005
---------- 3 ----------
using time : 0.11723
waiting time : 0.00006
---------- 4 ----------
using time : 0.11552
waiting time : 0.02455
---------- 5 ----------
using time : 0.18801
waiting time : 0.01162
```

• Input=10000일 때

```
Input :10000
---------- 1 ----------
using time : 0.37511
waiting time : 0.00010
---------- 2 ----------
using time : 0.36530
waiting time : 0.00005
---------- 3 ----------
using time : 0.45636
waiting time : 0.00005
---------- 4 ----------
using time : 0.24630
waiting time : 0.00006
---------- 5 ----------
using time : 0.38079
waiting time : 0.00009
```

다른 방법

```
rate.sleep()

total_start = time.time()

for j in range(0,5):
    start_time = time.time()
    do_job(num)
    end_time = time.time()

    rate.sleep()
    sleep_time= time.time()
    list_append_time ()

 total_end = time.time()
```