Assignment 3
ROS
Divyansh Agarwal

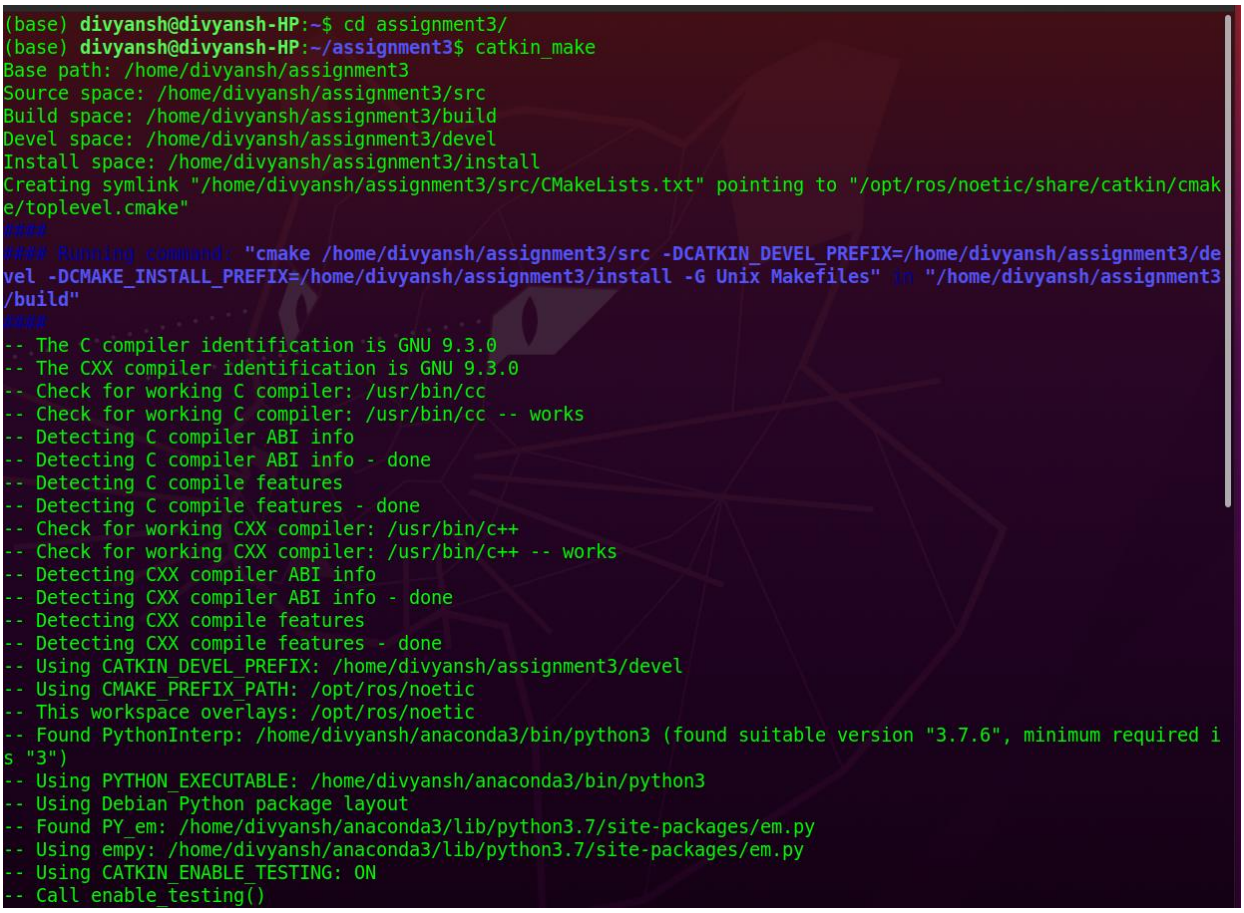
1. Create new ROS Workspace and inside the workspace create the ROS Package

Commands:

mkdir –p ~/assignment3/src

cd assignment3/

catkin_make



```
(base) divyansh@divyansh-HP:~/assignment3$ vim ~/.bashrc
(base) divyansh@divyansh-HP:~/assignment3$ source ~/.bashrc
(base) divyansh@divyansh-HP:~/assignment3$ cd src
(base) divyansh@divyansh-HP:~/assignment3/src$ catkin_create_pkg threenode rospy std_msgs roscpp
Created file threenode/package.xml
Created file threenode/CMakeLists.txt
Created folder threenode/include/threenode
Created folder threenode/src
Successfully created files in /home/divyansh/assignment3/src/threenode. Please adjust the values in package.xml
```

```
    . /etc/bash_completion
  fi
fi
#source /opt/ros/noetic/setup.bash
#source /home/divyansh/prac_rosext/devel/setup.bash
source /home/divyansh/assignment3/devel/setup.bash

# >>> conda initialize >>>
```

```
(base) divyansh@divyansh-HP:~/assignment3/src$ catkin_create_pkg for_rostopics rospy roscpp std_msgs
Created file for_rostopics/package.xml
Created file for_rostopics/CMakeLists.txt
Created folder for_rostopics/include/for_rostopics
Created folder for_rostopics/src
Successfully created files in /home/divyansh/assignment3/src/for_rostopics. Please adjust the values in package.xml.
(base) divyansh@divyansh-HP:~/assignment3/src$ catkin_create_pkg threenode rospy roscpp std_msgs
Created file threenode/package.xml
Created file threenode/CMakeLists.txt
Created folder threenode/include/threenode
Created folder threenode/src
Successfully created files in /home/divyansh/assignment3/src/threenode. Please adjust the values in package.xml.
```

```
Base path: /home/divyansh/assignment3
Source space: /home/divyansh/assignment3/src
Build space: /home/divyansh/assignment3/build
Devel space: /home/divyansh/assignment3/devel
Install space: /home/divyansh/assignment3/install
####
#### Running command: "cmake /home/divyansh/assignment3/src -DCATKIN_DEVEL_PREFIX=/home/divyansh/assignment3/devel -DCMAKE_INSTALL_PREFIX=/home/divyansh/assignment3/install -G Unix Makefiles" in "/home/divyansh/assignment3/build"
####
-- Using CATKIN_DEVEL_PREFIX: /home/divyansh/assignment3/devel
-- Using CMAKE_PREFIX_PATH: /home/divyansh/assignment3/devel;/opt/ros/noetic
-- This workspace overlays: /home/divyansh/assignment3/devel;/opt/ros/noetic
-- Found PythonInterp: /home/divyansh/anaconda3/bin/python3 (found suitable version "3.7.6", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /home/divyansh/anaconda3/bin/python3
-- Using Debian Python package layout
-- Using empy: /home/divyansh/anaconda3/lib/python3.7/site-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /home/divyansh/assignment3/build/test_results
-- Forcing gtest/gmock from source, though one was otherwise available.
-- Found gtest sources under '/usr/src/googletest': gtests will be built
-- Found gmock sources under '/usr/src/googletest': gmock will be built
-- Found PythonInterp: /home/divyansh/anaconda3/bin/python3 (found version "3.7.6")
-- Using Python nosetests: /usr/bin/nosetests3
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- ~~  traversing 2 packages in topological order:
-- ~~  - for_rostopics
-- ~~  - threenode
-- +++ processing catkin package: 'for_rostopics'
-- ==> add_subdirectory(for_rostopics)
-- +++ processing catkin package: 'threenode'
-- ==> add_subdirectory(threenode)
-- Configuring done
-- Generating done
-- Build files have been written to: /home/divyansh/assignment3/build
####
#### Running command: "make -j8 -l8" in "/home/divyansh/assignment3/build"
```

Catkin_create_pkg threenode rospy roscpp std_msgs

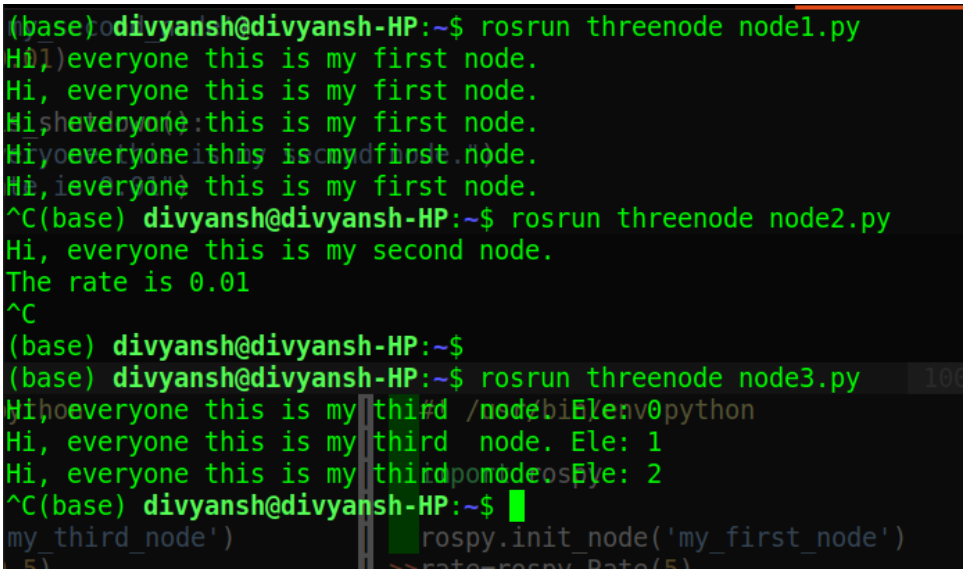
Catkin_create_pkg for_rostopics rospy roscpp std_msgs

2. Inside the created package create Three ROS Nodes using Python and that nodes should be active until and unless if we manually stop the process.

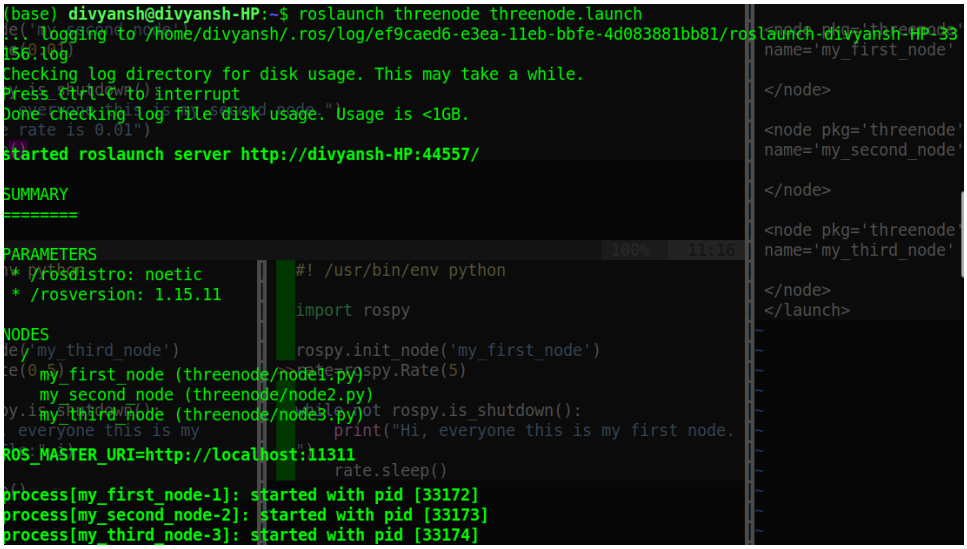
```
#! /usr/bin/env python


import rospy


rospy.init_node('my_first_node')
rate=rospy.Rate(5)


while not rospy.is_shutdown():
print("Hi, everyone this is my first node.")
rate.sleep()
```

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


import rospy


rospy.init_node('my_second_node')
rate=rospy.Rate(0.01)


while not rospy.is_shutdown():
print("Hi, everyone this is my second node.")
```

```
print("The rate is 0.01")

rate.sleep()


#! /usr/bin/env python



import rospy



rospy.init_node('my_third_node')

rate=rospy.Rate(0.5)

i=0

while not rospy.is_shutdown():

print("Hi, everyone this is my third node. Ele:",i)

i+=1

rate.sleep()
```



3. Create ROS launch file for launching three nodes at a time.



Launch file in launch folder

<launch>

<!-- format is node with pkg name type (the node file name) then the name() of node-->

<node pkg='threenode' type='node1.py' name='my_first_node' output='screen' >

</node>

<node pkg='threenode' type='node2.py' name='my_second_node' output='screen'>

```
</node>


<node pkg='threenode' type='node3.py' name='my_third_node' output='screen' >


</node>
</launch>
```

4. Create a ROS Topic to publisher node and subscriber node to communicate the String data via ros topic (Create new ros topic as well).

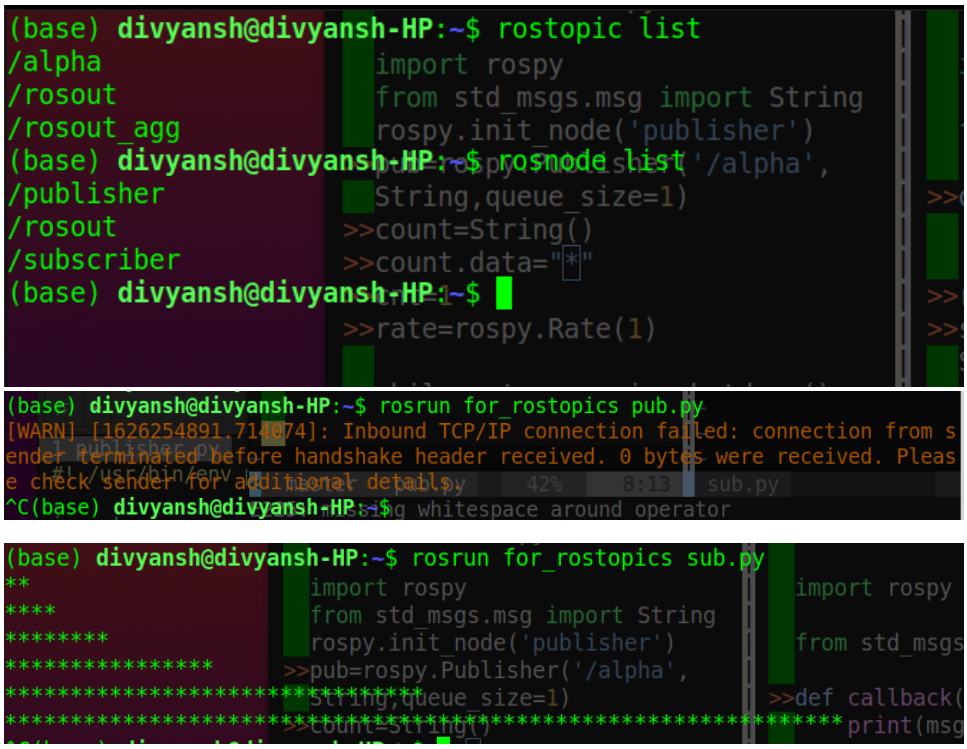
Publsher:

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


import rospy
from std_msgs.msg import String
rospy.init_node('publisher')
pub=rospy.Publisher('/alpha',String,queue_size=1)
count=String()
count.data="*"
cnt=1
rate=rospy.Rate(1)


while not rospy.is_shutdown():
pub.publish(count)
count.data+=count.data
```

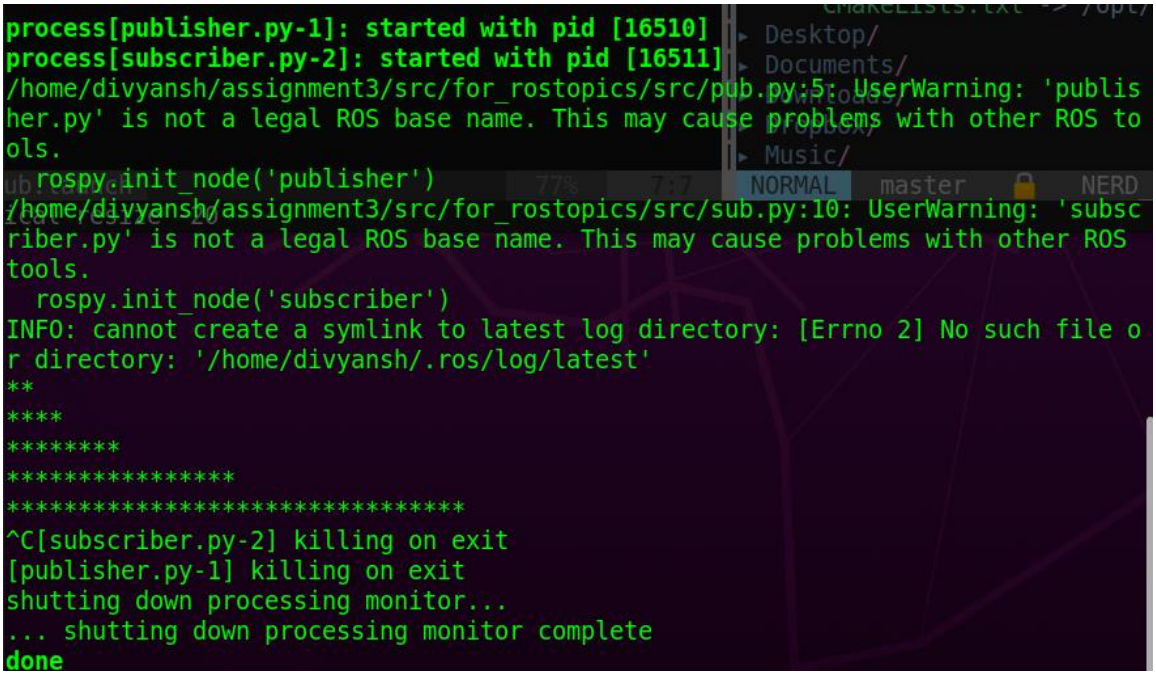
```
cnt+=1
rate.sleep()
```

Subscrbier

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

import rospy

from std_msgs.msg import String

def callback(msg):
print(msg.data)

rospy.init_node('subscriber')
sub=rospy.Subscriber('alpha',String,callback)

rospy.spin()
```



5. Create a lunch file to launch the Publisher node and subscriber node.



```
(base) divyansh@divyansh-HP:~$ rostopic info /alpha
Type: std_msgs/String

Publishers:
 * /publisher.py (http://divyansh-HP:38163/)

Subscribers:
 * /subscriber.py (http://divyansh-HP:43593/)
```

<launch>

<node pkg='for_rostopics' type='pub.py' name='publisher.py' output='screen'>

</node>

<node pkg='for_rostopics' type='sub.py' name='subscriber.py' output='screen'>

</node>

</launch>