Workshop on ROS

6th January 2020 at 3:09pm

Understanding Robotics through ROS

LOCUS 2020, *IoE*, Pulchowk

ROS Architecutre

5th January 2020 at 10:32pm

- ROS Master
- ROS Node1
- ROS Node2
- Messages

ROS Packages

5th January 2020 at 10:35pm

> All the files that a specific ROS program contains; all its cpp files,python files, configuration files, compilation files, launch files, andparameters files.

General Package folder structure

- Launch folder
- src folder
- *CMakeLists*.txt
- package.xml

Creating a First ROS project/package

6th January 2020 at 2:22pm

Creating ROS workspace

```
$ mkdir -p ~/catkin_ws/src
$ cd ~/catkin_ws/src
$ catkin_init_workspace
$ cd ~/catkin_ws
$ catkin_make
```

Creating package

`$ catkin_create_pkg <package name> <dependencies>` * Use `src` directory to make package

```
$ cd ~/catkin_ws/src
$ catkin_create_pkg introduction roscpp rospy std_msgs
$ cd ~/catkin_ws
$ catkin_make
```

To build and add packages to the ROS path

```
$ souce ~/catkin_ws/devel/setup.bash
```

ROS Computation structure

6th January 2020 at 1:46pm

> ROS program with two nodes. One of them must publish a series of messages to a defined topic. The second one must listen to those messages and print it on the screen

- Nodes - main programs in ROS
- Topics - stream of messages with a defined type
- Service - Server - Client model, synchronous, blocking
- Action - Goal, Feedback, Result, Non-Blocking

Publisher, Subscriber and Topics

6th January 2020 at 3:12pm

- A publisher is a node that keeps publishing a message into a topic.
- A subscriber is node which subscribes to topic

- A topic is a channel that acts as an information high way, where other ROS nodes can either publish or read information.

Publisher code

6th January 2020 at 1:29pm

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

import rospy

from std_msgs.msg import Int32

# initializse node
rospy.init_node('topic_publisher')

#initialize what node is going to do publish/subscribe
pub = rospy.Publisher('counter', Int32)
rate = rospy.Rate(2)
count = 0


while not rospy.is_shutdown():
    pub.publish(count) #finally publish the topic
    count += 1
    rate.sleep()
```

Subscriber code

6th January 2020 at 1:47pm

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

import rospy

from std_msgs.msg import Int32

rospy.init_node('topic_publisher')
pub = rospy.Publisher('counter', Int32)
rate = rospy.Rate(2)
count = 0

while not rospy.is_shutdown():
    pub.publish(count)
    count += 1
    rate.sleep()
```

Running the individual node

6th January 2020 at 1:52pm

```
rosrun <package> <node executable>
```

Example:

```
rosrun introduction topic_publihser.py
```

Running codes using Launch files

6th January 2020 at 1:57pm

```
<launch>
<node pkg="intro_1" type="topic_publisher.py" name="topic_publisher"
output="screen">
</node>
<node pkg="intro_1" type="topic_subscriber.py" name="topic_subscriber"
output="screen">
</node>
</launch>
```

Controlling the turtle

6th January 2020 at 2:05pm

```
roscore
rosrun turtlesim turtlesim_node
rosrun turtlesim turtle_teleop_key
```

Use command learned to know the information about node

```
rosnode list
rostopic list
rostopic info /turtle1/pose
rosmsg show /turtlesim/pose
rostopic echo /turtle1/cmd_vel

rostopic pub /turtle1/cmd_vel geometry_msgs/Twist -- '[0.0, 0.0,0.0]' '[0.0,
0.0, 1.6]'
```

Exercises

5th January 2020 at 11:01pm

- Controlling the turtlebot3 in Gazebo
- Using basic commands to control the turtlebot