ASSIGNMENT 4
ROS ACTIONS:

Asycnhronous call to another node functionality is called an action. An asynchrnous process here means that we don have to wait for the result and we can proceed to the other things that need to be done. The node providing the functionality will implement the Action Server.

Goal

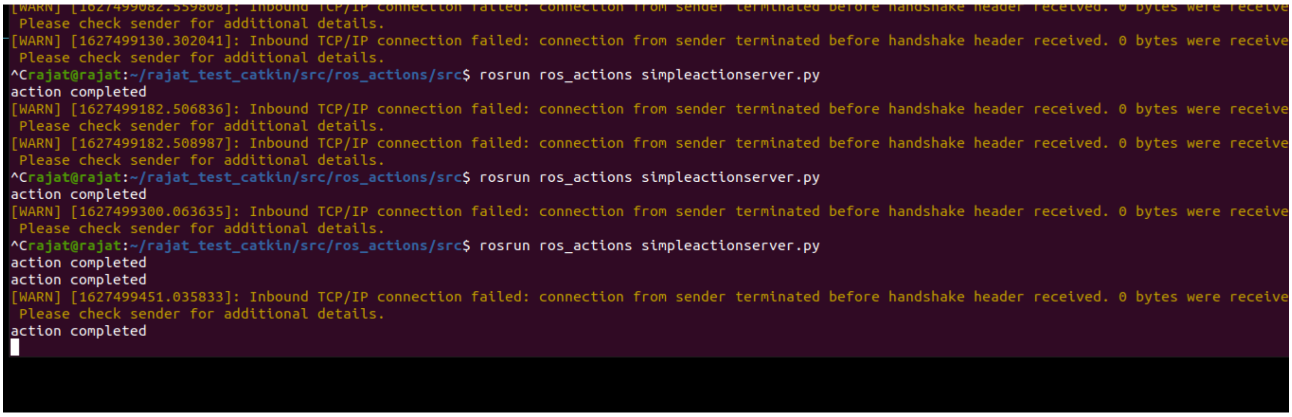
The goal is to complete the tasks using action. The notion of goal is introduced in such a way that it can be sent to an Action Server by an Action client. While moving the base the goal would be a PoseStamped message that contains information about where the rovot should move .For instance in a arm which moves objects will have a goal containing parameters such as min speed min distance min angle etc.

Feedback

Feedback provides server implementers a way to tell an ActionClient about the incremental progress of a goal. For moving the base, this might be the robot's current pose along the path. For controlling the tilting laser scanner, this might be the time left until the scan completes.

Result :

A result is sent from the ActionServer to the ActionClient upon completion of the goal. This is different from feedback, since it is sent exactly once. This is extremely useful when the purpose of the action is to provide some sort of information. For move base, the result isn't very important, but it might contain the final pose of the robot. For controlling the tilting laser scanner, the result might contain a point cloud generated from the requested scan.

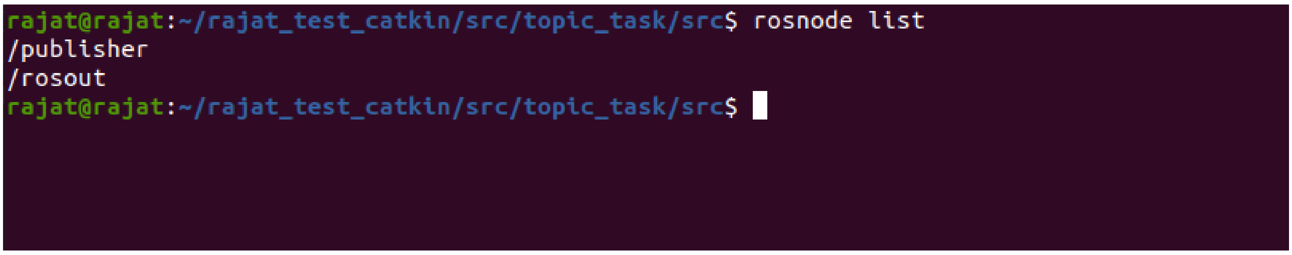


```
[WARN] [1627499082.559808]: Inbound TCP/IP connection failed: connection from sender terminated before handshake header received. 0 bytes were receive
 Please check sender for additional details.
[WARN] [1627499130.302041]: Inbound TCP/IP connection failed: connection from sender terminated before handshake header received. 0 bytes were receive
 Please check sender for additional details.
^Crajat@rajat:~/rajat_test_catkin/src/ros_actions/src$ rosrun ros_actions simpleactionserver.py
action completed
[WARN] [1627499182.506836]: Inbound TCP/IP connection failed: connection from sender terminated before handshake header received. 0 bytes were receive
 Please check sender for additional details.
[WARN] [1627499182.508987]: Inbound TCP/IP connection failed: connection from sender terminated before handshake header received. 0 bytes were receive
 Please check sender for additional details.
^Crajat@rajat:~/rajat_test_catkin/src/ros_actions/src$ rosrun ros_actions simpleactionserver.py
action completed
[WARN] [1627499300.063635]: Inbound TCP/IP connection failed: connection from sender terminated before handshake header received. 0 bytes were receive
 Please check sender for additional details.
^Crajat@rajat:~/rajat_test_catkin/src/ros_actions/src$ rosrun ros_actions simpleactionserver.py
action completed
action completed
[WARN] [1627499451.035833]: Inbound TCP/IP connection failed: connection from sender terminated before handshake header received. 0 bytes were receive
 Please check sender for additional details.
action completed
```

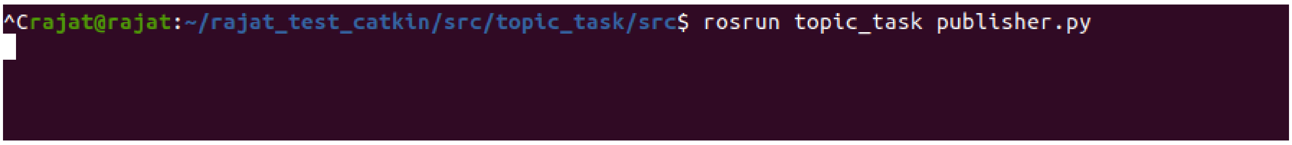
ROS Topics

Some nodes provide information for other nodes, as a camera feed would do, for example. Such a node is said to publish information that can be received by other nodes. The information in ROS is called a topic A topic defines the types of messages that will be sent concerning that topic.

The nodes that transmit data publish the topic name and the type of message to be sent. The actual data is published by the node. A node can subscribe to a topic and transmitted messages on that topic are received by the node subscribing.



```
rajat@rajat:~/rajat_test_catkin/src/topic_task/src$ rosnode list
/publisher
/rosout
rajat@rajat:~/rajat_test_catkin/src/topic_task/src$ 
```

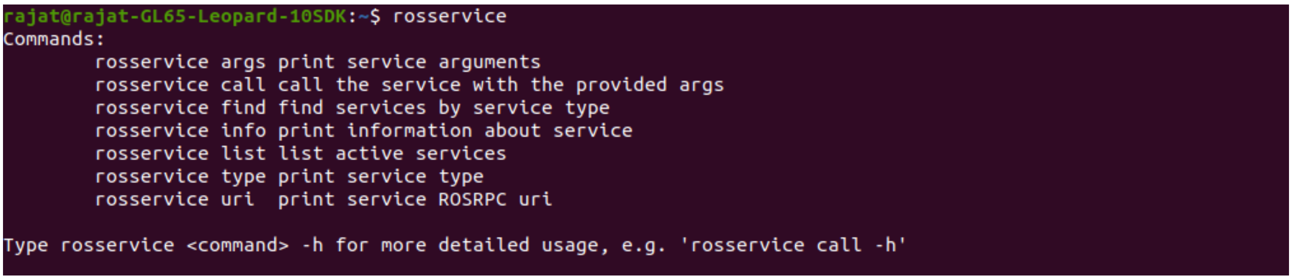


```
^Crajat@rajat:~/rajat_test_catkin/src/topic_task/src$ rosrun topic_task publisher.py
```

ROS SERVICES:

A Service, which is defined by a pair of messages: one for the request and one for the reply. A providing ROS node offers a service under a string name, and a client calls the service by sending the request message and awaiting the reply. Client libraries usually present this interaction to the programmer as if it were a remote procedure call.

Various related services:
1) clear: Clears the turtlesim background and sets the color to the value of the background parameters.
2) reset: Resets the turtlesim to the start configuration and sets the background color to the value of the background.
3) kill: Kills a turtle by name.



```
rajat@rajat-GL65-Leopard-10SDK:~$ rosservice
Commands:
	rosservice args	print service arguments
	rosservice call	call the service with the provided args
	rosservice find	find services by service type
	rosservice info	print information about service
	rosservice list	list active services
	rosservice type	print service type
	rosservice uri	print service ROSRPC uri

Type rosservice <command> -h for more detailed usage, e.g. 'rosservice call -h'
```