ROS COMMANDS

ROBOTICS

POLITECNICO
MILANO 1863

FILE SYSTEM TOOLS

Change directory in the ROS file system

```
roscd [package_name[/subdir]]
```

```
roscd roscpp && pwd      /opt/ros/indigo/share/roscpp
roscd roscpp/srv         /opt/ros/indigo/share/roscpp/srv
roscd robby_roboto       ~/catkin_ws/src/robby_roboto
```

FILE SYSTEM TOOLS

Getting information about installed packages

```
rospack <subcommand> [options] [package]
```

subcommands (among the others)

`depends [package]`	package dependencies
`find [package]`	find package directory
`list`	list available packages

```
rospack find roscpp      /opt/ros/indigo/share/roscpp
rospack list             <several packages>
```

PACKAGE CREATION

goo.gl/DBwhhC

Command to create a new package

```
catkin_create_pkg [package_name] [depend1] [depend2] [depend3]
```

```
catkin_create_pkg beginner_tutorials std_msgs rospy roscpp
```

Important Notes

- `roscpp` and `rospy` are client libraries to use C++ and Python
- Before being able to do that you should have creates a ros_workspace

STARTING THE MIDDLEWARE

goo.gl/DBwhhC

To start the ROS middleware just type in a terminal

```
roscore
```

Now it is possible to display information about the elements currently running

```
rosnode list
rostopic list
rostopic echo /rosout
rosservice list
rqt_graph
```

DEALING WITH NODES

goo.gl/DBwhhC

Getting information about running nodes

```
rosnode <command> [other_commands]
```

subcommands (among the others)

`ping`	test connectivity to node
`info`	print information about node
`kill`	kill a running node
`cleanup`	purge registration information of unreachable nodes

```
rosnode list
rosnode info /rosout
```

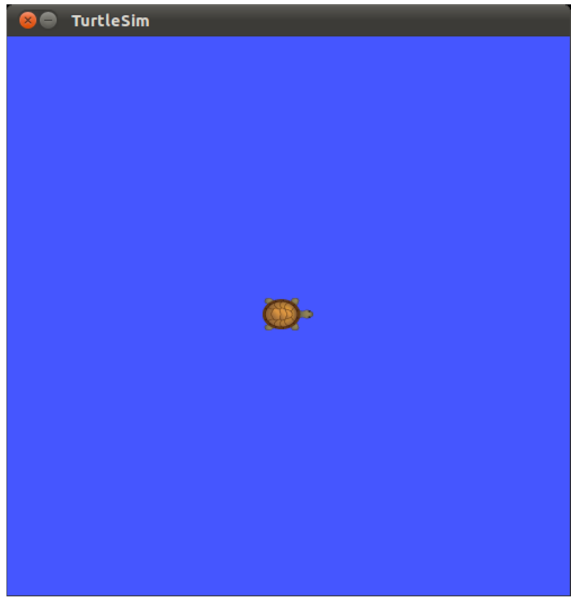
STARTING ROS NODES

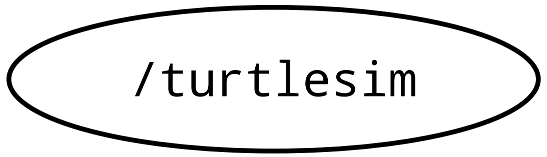
goo.gl/DBwhhC

To start a ROS node type in a terminal

```
rosrun [package_name] [node_name]
```

```
rosrun turtlesim turtlesim_node
rosnode ping turtlesim
rosnode info turtlesim
```





STARTING ROS NODES

goo.gl/DBwhhC

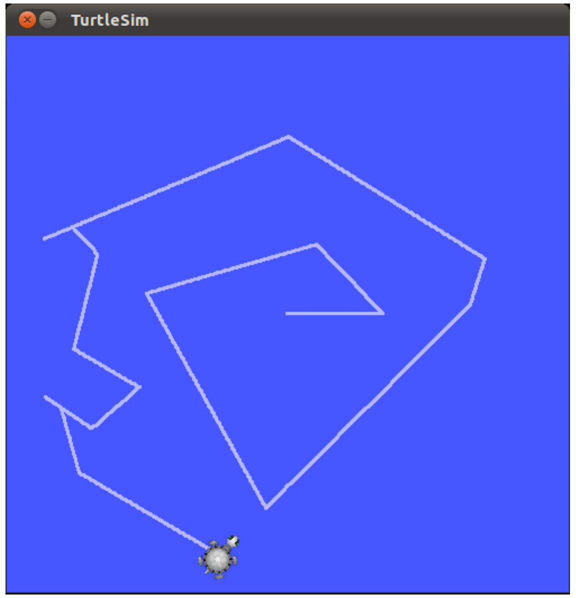
In a new terminal

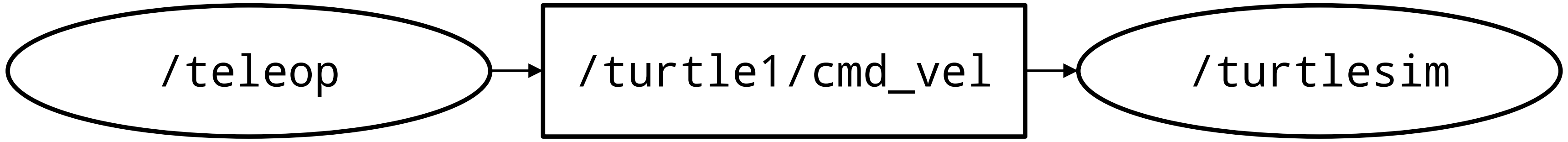
```
rosrun turtlesim turtle_teleop_key
```

Notes:

`turtle_teleop_key` is publishing the key strokes on a topic

`turtlesim` subscribes to the same topic to receive the key strokes





DEALING WITH TOPICS

goo.gl/DBwhhC

To show the running node type in a terminal

```
rqt_graph
```

To plot published data on a topic

```
rqt_plot /turtle1/pose/x /turtle1/pose/y
rqt_plot /turtle1/pose/x:y
```

To monitor a topic on a terminal type

```
rostopic echo /turtle1/cmd_vel
```

DEALING WITH TOPICS CONT.

goo.gl/DBwhhC

Getting information about ROS topics

```
rostopic <command> [topic_name]
```

subcommands (among the others)

`echo`	print messages to screen
`find`	find topics by type
`hz`	display publishing rate of topic
`info`	print information about active topic
`list`	list active topics
`pub`	publish data to topic
`type`	print topic type

DEALING WITH TOPICS CONT.

goo.gl/DBwhhC

Getting information about ROS topics

```
rostopic type [topic_name]


rostopic type /turtle1/cmd_vel
```

Publishing ROS topics

```
rostopic pub [topic] [msg type] [args]


rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist '{linear:  {x:
0.1, y: 0.0, z: 0.0}, angular: {x: 0.0,y: 0.0,z: 0.0
```

MESSAGES (ALSO SERVICES)

goo.gl/DBwhhC

Getting information about msg/srv files

```
rosmsg <command> [msg/srv_file]
```

subcommands (among the others)

`show`	Display the fields in the msg/srv.
`list`	Display names of all msg/srv.
`package`	List all the msg/srv in a package.
`packages`	List all packages containing the msg/srv.

```
rosmsg show Pose
rosmsg package nav_msgs
rosmsg packages sensor_msgs/CameraInfo
```

DEALING WITH SERVICES

goo.gl/DBwhhC

Calling services from command line and getting information:

```
rosservice <command> [other_commands]
```

subcommand (among the others)

`list`	Print information about active services.
`node`	Print name of node providing a service.
`call`	Call the service with the given args.
`args`	List the arguments of a service.
`type`	Print the service type.
`find`	Find services by service type

```
rosservice call /add_two_ints 1 2
rosservice type add_two_ints | rossrv show
```