

ROS Crash Course

Class 3

The ROS logo consists of a 3x3 grid of nine blue dots on the left, followed by the letters "ROS" in a large, bold, blue sans-serif font.

Agenda

- Old HW

- ROS Concepts

- How to Make/Build ROS Packages

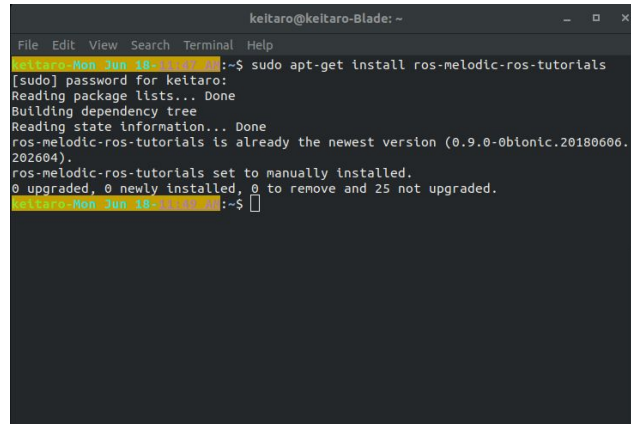
- New HW

HW

-Please start this download while everyone is showing their HW

-`$ sudo apt-get install ros-melodic-ros-tutorials ros-melodic-rqt
ros-melodic-rqt-common-plugins`

-Don't worry if you see 0 newly installed that just means you already have it

A terminal window titled 'keitaro@keitaro-Blade: ~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the command 'sudo apt-get install ros-melodic-ros-tutorials' being executed. The output indicates that the package is already installed and manually set. The terminal text is as follows:

```
keitaro@keitaro-Blade: ~  
File Edit View Search Terminal Help  
keitaro@keitaro-Blade: ~$ sudo apt-get install ros-melodic-ros-tutorials  
[sudo] password for keitaro:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
ros-melodic-ros-tutorials is already the newest version (0.9.0-0bionic.20180606.202604).  
ros-melodic-ros-tutorials set to manually installed.  
0 upgraded, 0 newly installed, 0 to remove and 25 not upgraded.  
keitaro@keitaro-Blade: ~$
```

ROS Concepts

-Master

-Node

-Publisher

-Subscriber

-Topic

-Message

ROS Concepts

-rosservice: a way to send a request and receive a response

-rosparameters: a way to edit the ROS Parameter Server

-ros commands:

- ros[command] convention is used for each shell command

- roscore: starts the rosmaster, rosout, and ros parameter server

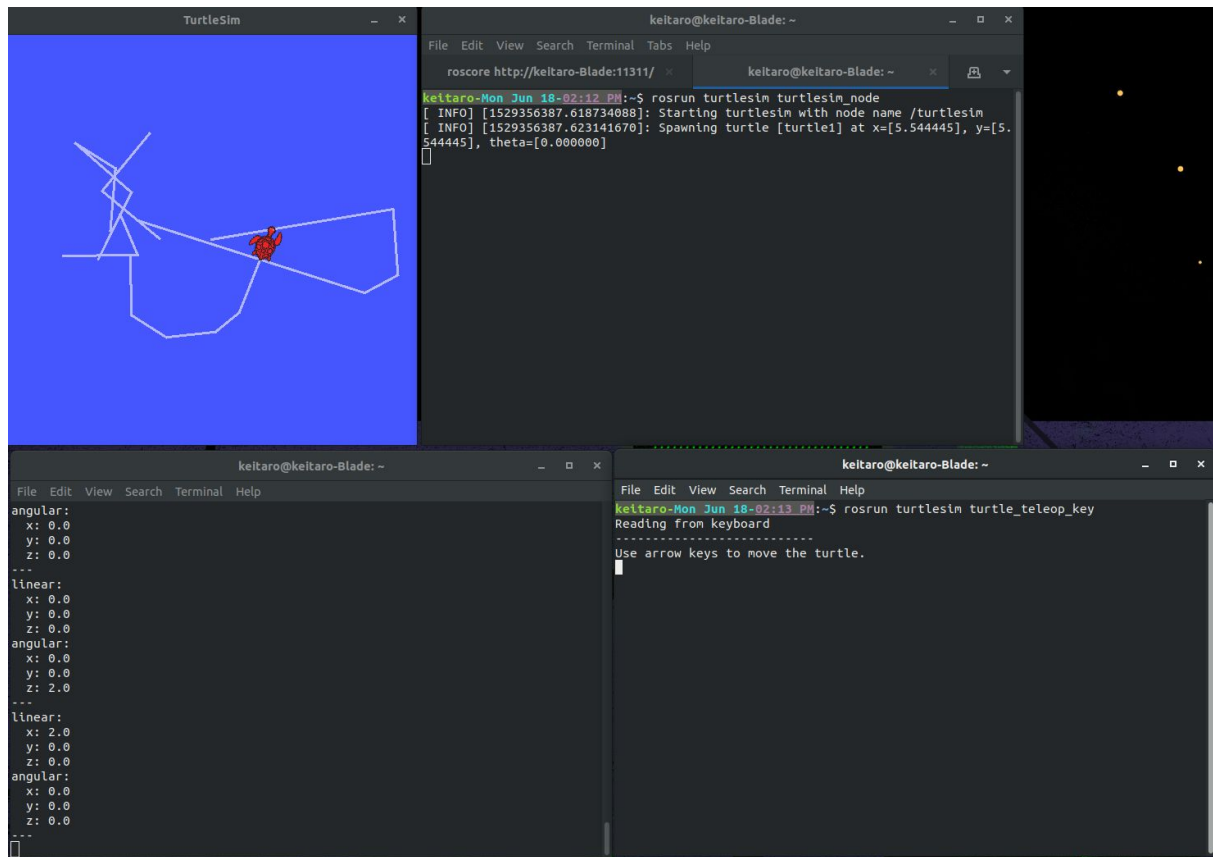
- roslaunch: runs a ros node

- rostopic: ros command tool used for rostopics

ROS Concepts

- Open a shell and run `$ roscore`
- In a new shell run `$ rosrun turtlesim turtlesim_node`
- In a new shell run `$ rosrun turtlesim turtle_teleop_key`
- In a new shell run `$ rostopic echo /turtle1/cmd_vel`
- Finally position all of them so that you can see the turtlesim screen and the last three terminals
- Using your arrows keys try to move the turtle around

ROS Concepts

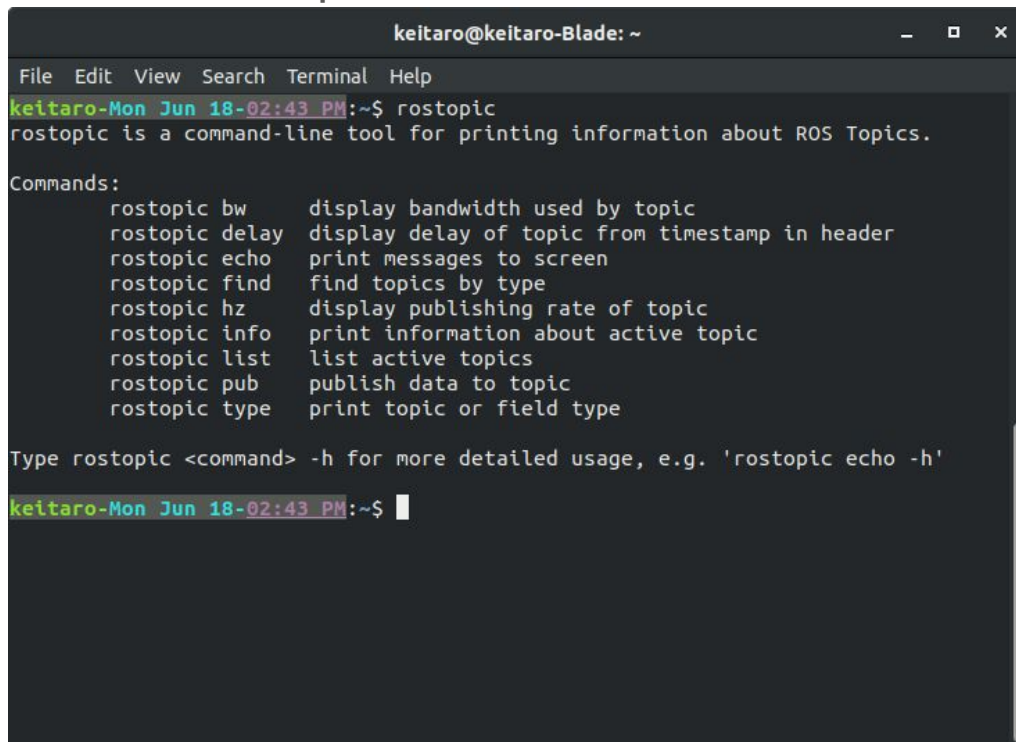


ROS Concepts

- Now open a new shell and run `$ rosservice call /clear`
- In the same terminal run `$ rosservice call /spawn 1 2 3 "test"`
- In the same terminal run `$ rosparam set /background_g 150`
- Then run `$ rosservice call /clear`

ROS Concepts

In the same terminal run `$ rostopic -h`



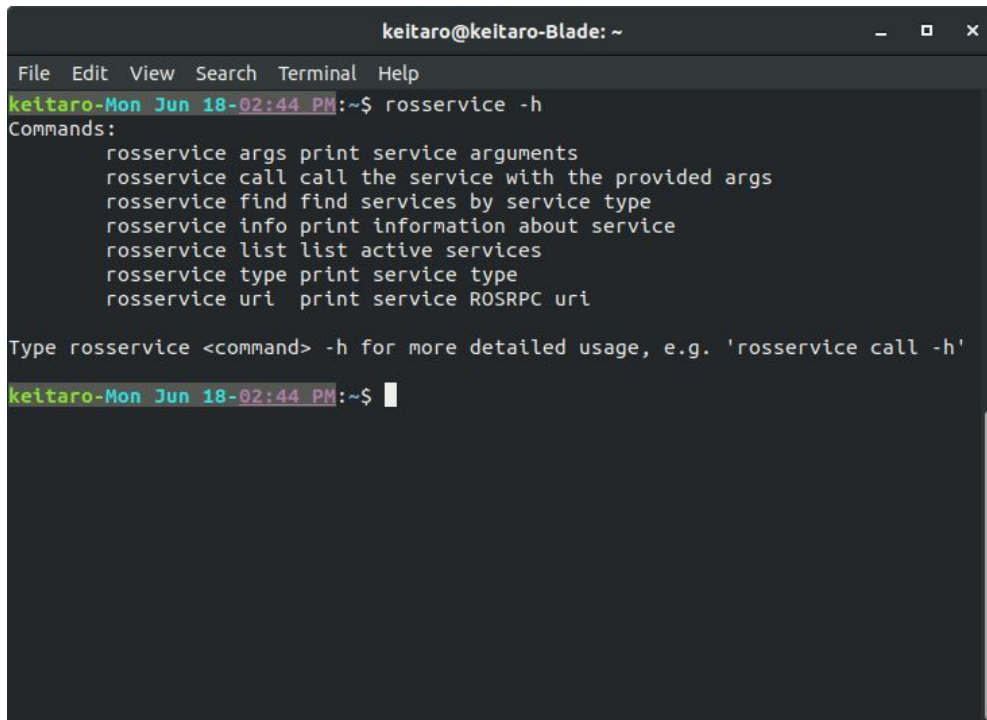
```
keitaro@keitaro-Blade: ~
File Edit View Search Terminal Help
keitaro-Mon Jun 18-02:43 PM:~$ rostopic
rostopic is a command-line tool for printing information about ROS Topics.

Commands:
  rostopic bw      display bandwidth used by topic
  rostopic delay   display delay of topic from timestamp in header
  rostopic echo    print messages to screen
  rostopic find    find topics by type
  rostopic hz     display publishing rate of topic
  rostopic info    print information about active topic
  rostopic list    list active topics
  rostopic pub     publish data to topic
  rostopic type    print topic or field type

Type rostopic <command> -h for more detailed usage, e.g. 'rostopic echo -h'
keitaro-Mon Jun 18-02:43 PM:~$
```

ROS Concepts

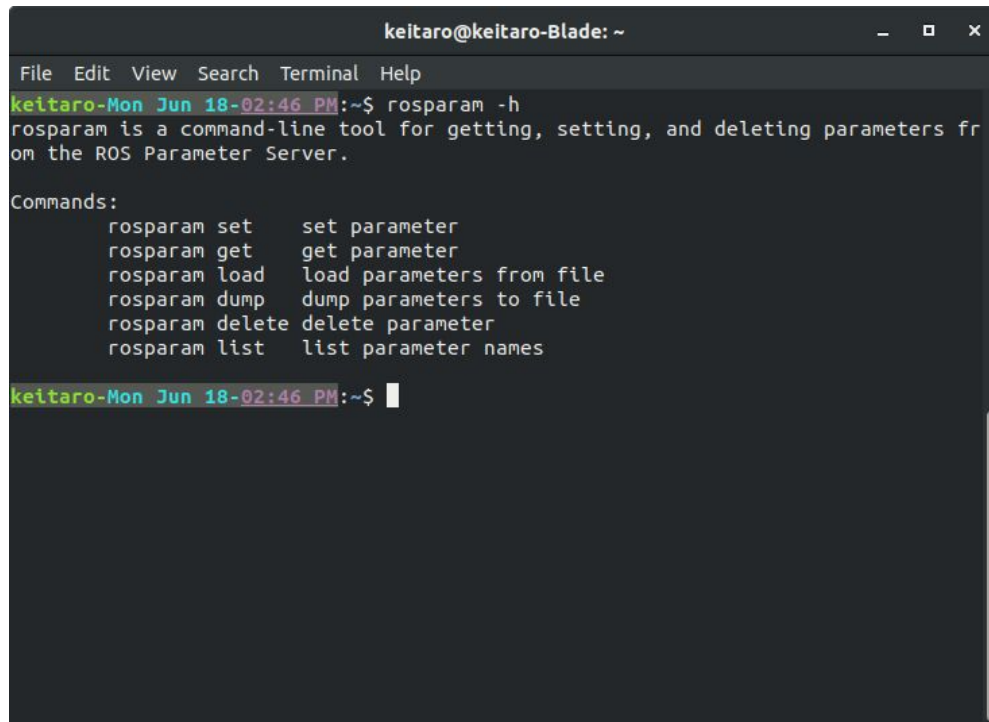
-Now try running `$ rosservice -h`



```
keitaro@keitaro-Blade: ~  
File Edit View Search Terminal Help  
keitaro-Mon Jun 18-02:44 PM:~$ rosservice -h  
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'  
keitaro-Mon Jun 18-02:44 PM:~$
```

ROS Concepts

-Finally try running `$ rosparam -h`



```
keitaro@keitaro-Blade: ~  
File Edit View Search Terminal Help  
keitaro-Mon Jun 18-02:46 PM:~$ rosparam -h  
rosparam is a command-line tool for getting, setting, and deleting parameters fr  
om the ROS Parameter Server.  
  
Commands:  
    rosparam set      set parameter  
    rosparam get      get parameter  
    rosparam load     load parameters from file  
    rosparam dump     dump parameters to file  
    rosparam delete   delete parameter  
    rosparam list     list parameter names  
  
keitaro-Mon Jun 18-02:46 PM:~$
```

ROS Concepts

-These ros commands are useful

- roscd

- rosls

- roscd

How to Make/Build ROS Package

-Catkin (http://wiki.ros.org/catkin/conceptual_overview)

- The compiler used for ROS code

-Catkin Workspace (<http://wiki.ros.org/catkin/workspaces>)

- Build: where the compiler builds the src code

- src: where the src code for any ros code live

- devel: a development environment when making install targets

How to Make/Build ROS Package

-Now in a new shell run the following commands:

```
-$ cd ~/catkin_ws/src
```

```
-$ catkin_create_pkg [your name]_test_pkg std_msgs geometry_msgs rospy  
roscpp
```

-The format for `catkin_create_pkg` is

```
-$ catkin_create_pkg [pkg name] [dependency 1] [dependency 2] ...
```

-You will now see the new package in the src folder

How to Make/Build ROS Package

-In the new package folder you will see the following

- include folder: for any header files you need

- src folder: for the main .cpp or .py files

- CMakeLists.txt: the instructions for catkin on how to build the package

- package.xml: for storing information about the package for ros

New HW

- Make 2 new nodes so that you can control 3 different turtles in turtlesim at the same time.
- Make them in the new package you just made this class
- Use the turtlesim and this tutorial for reference on source code (<http://wiki.ros.org/ROS/Tutorials/WritingPublisherSubscriber%28c%2B%2B%29>)
- Email me the finished package before class #4