

# Lab1: Rampup and ROS

CSCI 545

Fall 2019

## 1 Intro

In this lab, you will be setting up your system and getting acquainted with ROS. This lab is very important since it will setup the base on top of which you will work for subsequent labs. You will need to submit a lab report (pdf) and two python files (for the publisher/subscriber exercise).

Submission format: Your submission should be on your group's github repo. Create a folder called Lab1. Files should be placed in that folder like this:

```
Lab1/report.pdf
Lab1/code/<files>
```

## 2 Working with your project repository

1. First, clone your repo for the class

```
git clone https://github.com/usc-csci-545-fall-2019/<your-repo>.git
cd <your-repo>
```

2. Make your changes for the lab as necessary
3. (If you haven't already,) commit all your changes

```
git add <files to add>
git commit -m <Your commit message>
```

4. Push your changes to the repo (we will only look at the master branch)

```
git push origin master
```

## 3 Pre-requisites for this tutorial

- Your VM or native install should have ROS-kinetic installed setup with no catkin workspace (you create one yourself)

## 4 Verify ROS Install

1. Create a catkin workspace

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws
```

```
catkin build
source ~/catkin_ws/devel/setup.bash
Permanently add to path
echo 'source ~/catkin_ws/devel/setup.bash' >> ~/.bashrc
```

2. Check environment variables

```
printenv | grep ROS
```

Check if `ROS_ROOT`, `ROS_PACKAGE_PATH`, etc. are setup. If not, run

```
source /opt/ros/kinetic/setup.bash
```

and then

```
source ~/catkin_ws/devel/setup.bash
```

For a permanent solution, you should add these lines to your `~/.bashrc`

```
echo 'source /opt/ros/kinetic/setup.bash' >>~/.bashrc
```

```
echo 'source ~/catkin_ws/devel/setup.bash' >>~/.bashrc
```

3. You should be able to see which ROS packages are installed. `rospack list` You can see where a package is installed with `rospack find <package>`

## 5 Create your own package

The simplest possible package might have a structure which looks like this:

```
my_package/
  CMakeLists.txt
  package.xml
```

1. `cd ~/catkin_ws/src`
2. `catkin_create_pkg cs545_lab1 std_msgs rospy roscpp`  
This creates a package named `cs545_lab1` that lists `std_msgs`, `rospy`, and `roscpp` as dependencies
3. Build your catkin workspace `cd ~/catkin_ws/`  
`catkin build`
4. After build succeeds `source ~/catkin_ws/devel/setup.bash`
5. Verify that your package has installed  
`rospack list | grep cs545`  
Should return the name of your package

If you ever need to change dependencies for you package, you would have to edit the `package.xml` and `CMakeLists.txt`. See this tutorial page for more info.

## 6 ROS exercises

Note: You will need multiple terminals for this section. Consider using programs to handle this use-case, such as tmux or Terminator.

1. Run turtlesim
  - (a) Terminal1: `roscore`
  - (b) Terminal2: `roslaunch turtlesim turtlesim_node`
  - (c) Terminal3: `roslaunch turtlesim turtle_teleop_key`
  - (d) Select Terminal3 and then use the arrow keys to move the turtlebot
2. Find topics using rostopics
  - (a) Get list of topics with `rostopic -v`
  - (b) Run `rqt_graph` to see a graphical representation of the nodes. Explain what you see in 1-3 sentences.
  - (c) Get some info from a rostopic.
3. Record data using `rosbag record`
  - (a) Terminal4: `mkdir /tmp/bagfiles # gets cleared on reboot`
  - (b) `cd /tmp/bagfiles`
  - (c) `rosbag record -O baggy -a`
  - (d) Move turtlebot using arrow keys in Terminal3
  - (e) After recording for a few seconds, kill (Ctrl-C) the rosbag record in Terminal4
4. Play it back using `rosbag play`
  - (a) Kill (Ctrl-C) the turtle teleop key in Terminal3
  - (b) `cd /tmp/bagfiles`
  - (c) `rosbag info baggy.bag`
  - (d) `rosbag play baggy.bag`
5. Explain in 3-5 sentences the difference between `rosservice`, `rostopic`, `rosparams`, and `rosbag`.

## 7 Publisher and Subscriber Topics

This section requires you to write code to create a publisher and subscriber for a topic. Refer to the ROS tutorial slides presented in class or the official ROS tutorials for more information on this. You may need to use multiple terminals.

1. Write a publisher, using the skeleton code provided in `publisher_exercise.py`
2. Move the file to `~/catkin_ws/src/cs545_lab1/scripts/`
3. `catkin build cs545_lab1`

4. Make the publisher executable  
`chmod +x ~/catkin_ws/src/cs545_lab1/scripts/publisher_exercise.py`
5. Run `roscore` to start ROS
6. Run your publisher: `roslaunch cs545_lab1 publisher_exercise.py`
7. Verify that the topic is being published using `rostopic list` and `rostopic echo`
8. Write a subscriber, using the skeleton code provided in `subscriber_exercise.py`
9. Do the necessary steps to build and run the subscriber. Make sure you are able to receive the message and are printing it out.

## 8 Report

Write a report comprising of all the textual answers required in this lab. The report should be in .pdf format.