



Universidad Politécnica de la Zona Metropolitana de Guadalajara

Actividad:

EV\_1\_4\_Ejercicio con Turtle

Descripción:

Desplazar la simulación de la tortuga ROS por medio de lineas de código.

Materia:

Dinámica y Control de Robots

Alumnos:

Beatris Núñez Bryan Alejandro

Carrillo Ochoa Ali Farid

Mejía Davalos Cesar

Rizo Martínez Alfredo

Rosales Ortiz Ian Alexis

Maestro:

Moran Garabito Carlos Enrique

Fecha:

Martes, 28 de mayo de 2019

Objetivo:

Realizar el desplazamiento del robot simulador Turtle por medio de líneas de código en terminal.

Materiales:

- Laptop
- Sistema operativo Linux
- ROS (Robot Operating System)

Procedimiento:

A continuación, se muestra el procedimiento a seguir para realizar nuestro objetivo. Cabe mencionar que al utilizar las líneas de código para desplazarlo, definiremos el valor de desplazamiento de X, Y, Z en los movimientos lineales y angulares.

1. Abrimos una terminal y ejecutamos ROS, con el siguiente comando.

```
$ roscore
```

Se muestra en la imagen.1 la forma correcta en que ROS está corriendo en nuestra máquina.

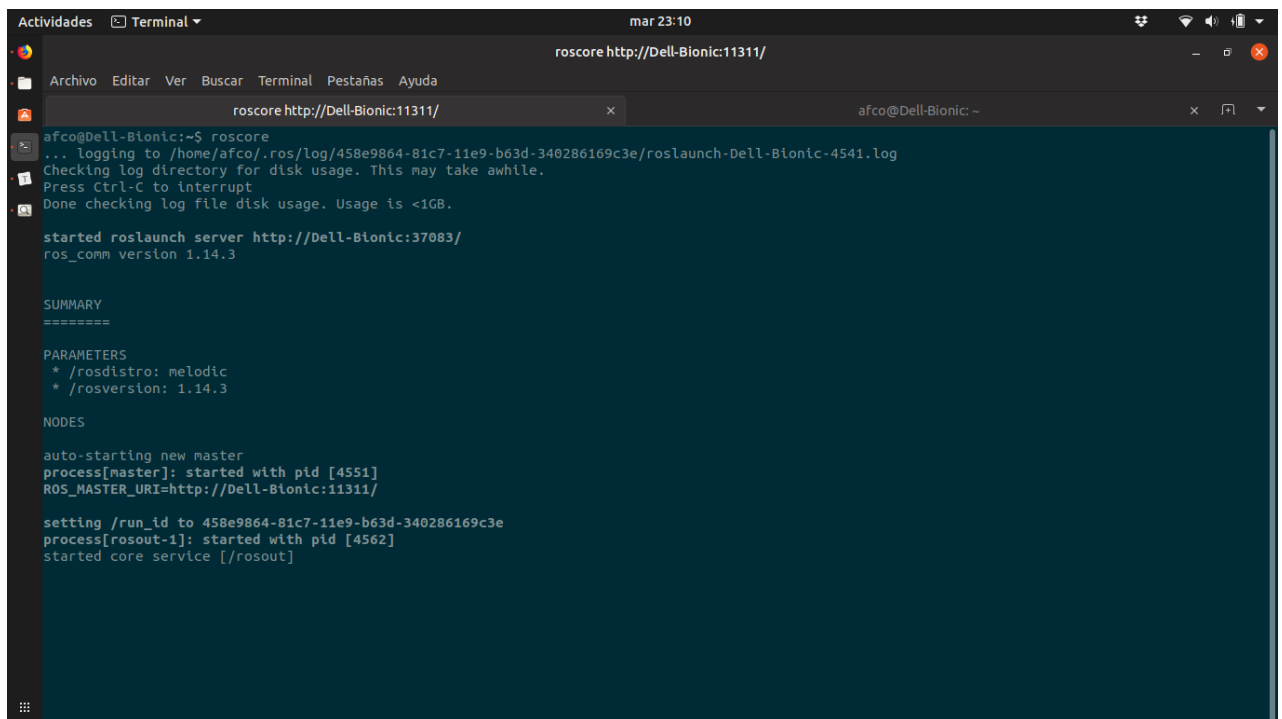
A screenshot of a Linux terminal window titled 'roscore http://Dell-Bionic:11311/'. The terminal shows the output of the 'roscore' command. It starts with logging information, then reports 'started roslaunch server http://Dell-Bionic:37083/' and 'ros\_comm version 1.14.3'. Below this, it shows a 'SUMMARY' section with 'PARAMETERS' (rosdistro: melodic, rosversion: 1.14.3) and 'NODES'. The 'NODES' section shows 'auto-starting new master', 'process[master]: started with pid [4551]', 'ROS\_MASTER\_URI=http://Dell-Bionic:11311/', 'setting /run\_id to 458e9864-81c7-11e9-b63d-340286169c3e', 'process[rosout-1]: started with pid [4562]', and 'started core service [/rosout]'. The terminal window has a dark background and standard Linux window controls at the top.

Imagen.1.ROS corriendo en terminal.

2. Abrimos otra terminal o puedes presionar la siguiente combinación de teclas `Ctrl+Shift+T` para abrir una nueva pestaña en la misma terminal donde ejecutamos ROS. A continuación ejecutamos la simulación del robot Turtle, escribiendo la siguiente línea de código en la terminal.

```
$ rosrunc turtlesim turtlesim_node
```

En la imagen.2 se muestra el nodo turtle corriendo correctamente.

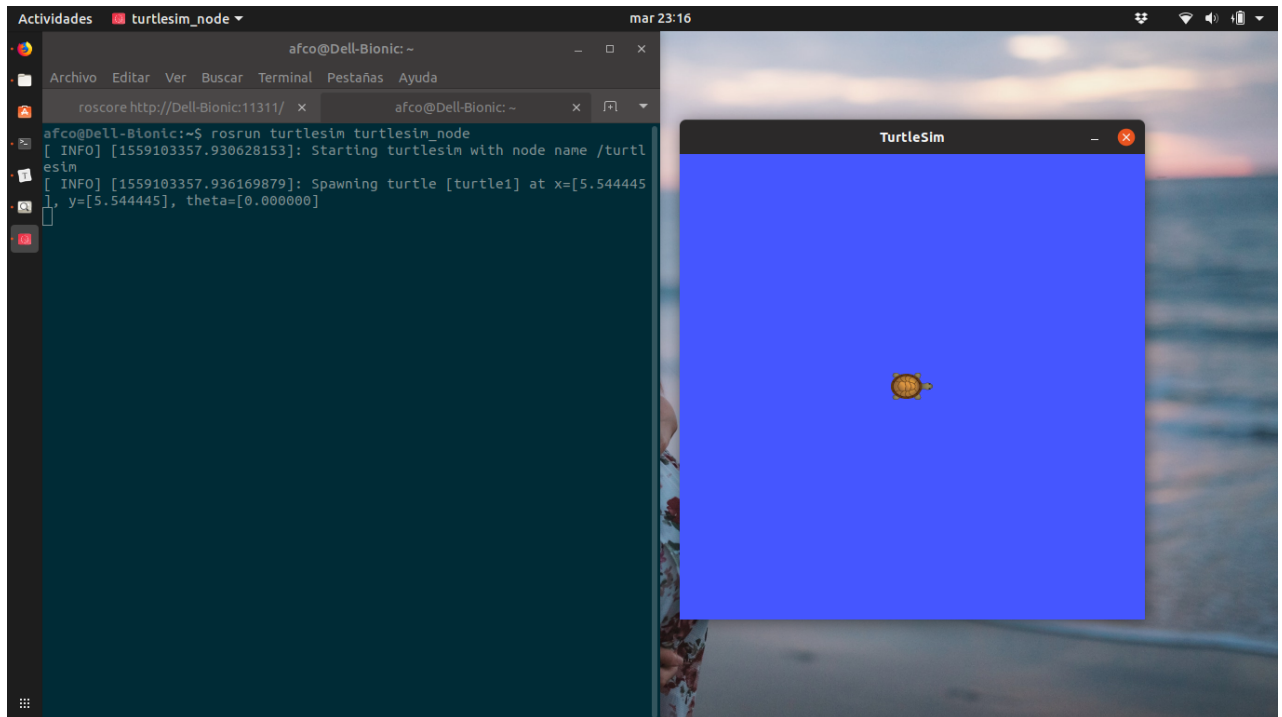


Imagen.2.Simulación del Robot Turtle corriendo en terminal.

3. Para finalizar, nuevamente abriremos una terminal más. En ella se introdujera el código de desplazamiento del robot, por medio de comando velocidad.

```
$ rostopic pub -1 /turtle1/cmd_vel geometry_msgs/Twist -- '[0.0,0.0,0.0]' '[0.0,0.0,0.0]'
```

Lo importante de aquí es destacar el código de cordenadas que se emplea en el linea, determinanado la primera sección de cordenadas como lineales y la segunda como angulares, acomodas por los vectores X, Y, Z.

```
linear '[x,y,z]' angular '[x,y,z]'
```

En el ejemplo realizado desplazamos el robot a 3 unidades negativas de su posición actual, definiendolo en parte lineal en el vector `x = -3`. Se muestra en la imagen.3 el código funcionando.

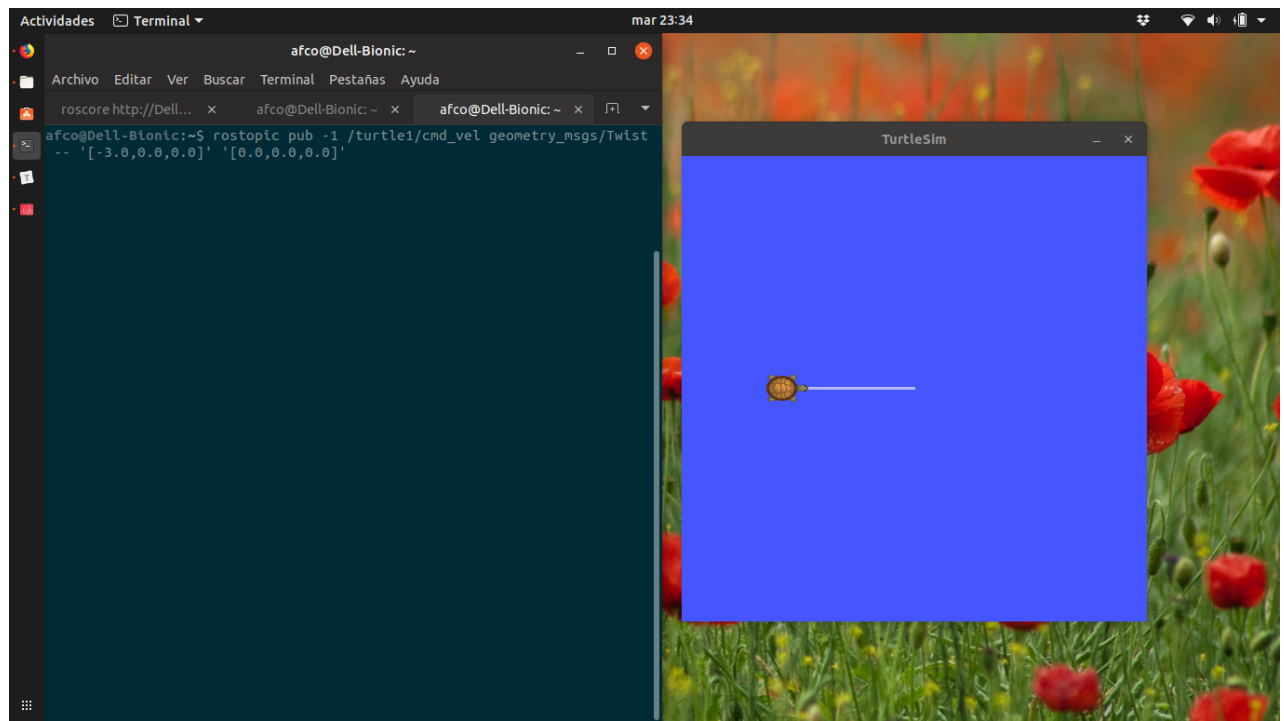


Imagen.3.Desplazamiento del Robot Turtle por linea de código velocidad.

Resultados:

A continuación se muestra en la imagen.4 la practica de procedimiento desarrollador por el compañero Ian Alexis, desplazando la tortuga tanto de forma lineal y angular.

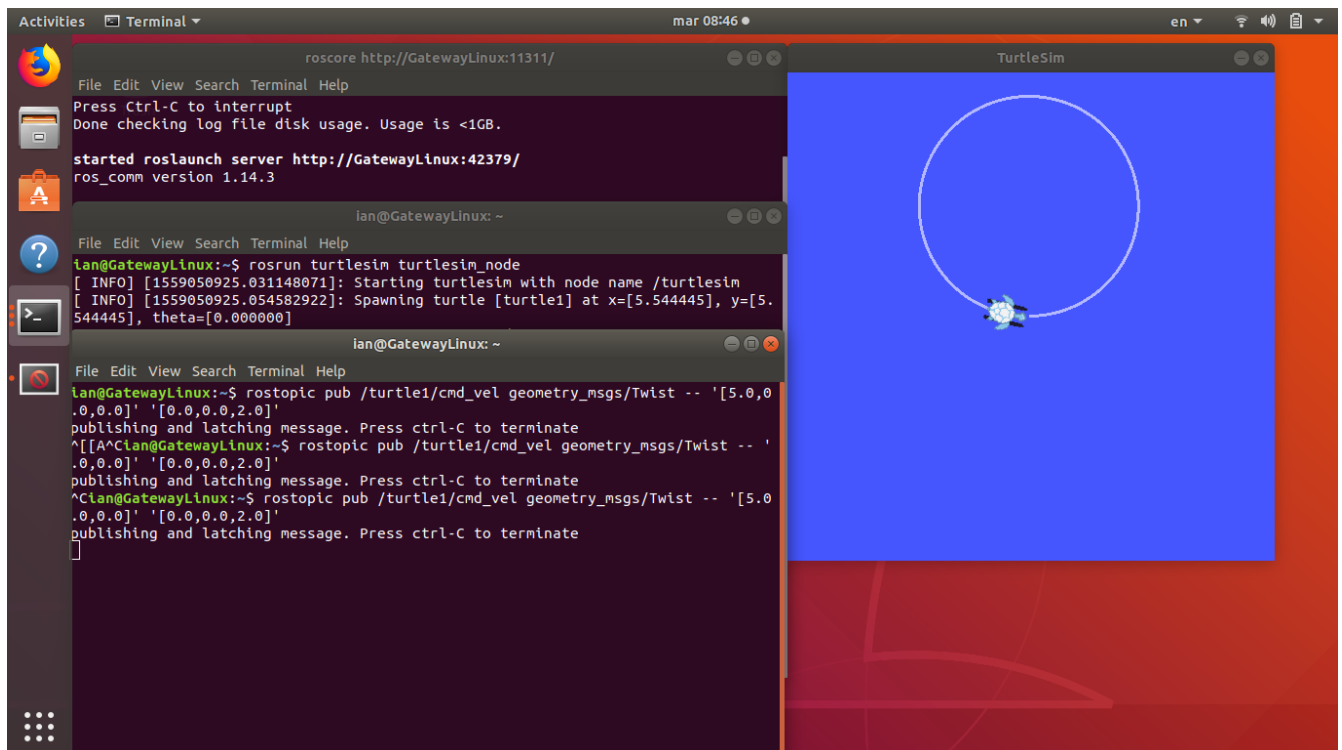


Imagen.4.Simualación del Robot Turtle ejecuto desde el computador del compañero Ian.