

Práctica 1

Instalación de Herramientas de Software

Laboratorio de Bio-Robótica
Construcción de Robots Móviles

Objetivos

- Familiarizar al alumno con el uso del sistema operativo Linux, distribución Ubuntu versión 16.04.
- Aprender los conceptos básicos de la plataforma ROS (Robot Operating System).

1. Introducción

ROS es un *middleware* de código abierto (*open source*) que provee la funcionalidad comúnmente necesaria en el desarrollo de software para robots móviles autónomos, como paso de mensajes y manejo de paquetes. El robot móvil que se construirá durante el curso utiliza ROS instalado en la minicomputadora RaspberryPi, por lo que es importante conocerlo.

ROS puede describirse en dos niveles conceptuales: el sistema de archivos y el grafo de procesos.

El sistema de archivos. Se refiere al modo en que están organizados los recursos en disco:

- **Workspace:** Se refiere a las carpetas que contienen paquetes de ROS.
- **Paquete:** Es la principal unidad de organización de software en ROS. Pueden contener nodos, bibliotecas, datasets, archivos de configuración y otros.
- **Manifiesto:** Definido por el archivo `package.xml` en cada paquete. Provee metadatos acerca de cada paquete.
- **Mensaje:** Archivos con extensión `.msg`. Definen estructuras de datos para el paso de mensajes en ROS.
- **Servicio:** Archivos con extensión `.srv`. Definen estructuras de tipo *request-response*. Utilizan mensajes para dicha definición.

Grafo de procesos. Es una red *peer-to-peer* de procesos. Los componentes básicos son:

- **Roscore:** Inicializa el sistema ROS: un máster + rosout + un servidor de parámetros.
- **Nodos:** Es simplemente un ejecutable que usa ROS para comunicarse con otros nodos.
- **Tópicos:** Algo similar a una variable cuyo contenido puede ser compartido entre todos los nodos mediante un patrón de publicación y suscripción.
- **Servicios:** Otra forma de comunicar nodos pero con un patrón de petición y respuesta.

- **Servidor de parámetros:** Es un diccionario compartido. Todos los nodos pueden leer y escribir parámetros en tiempo de ejecución.

2. Desarrollo

2.1. Instalación de Ubuntu 16.04

Instale Ubuntu 16.04, para ello, descargue la imagen de la siguiente dirección:

<https://www.ubuntu.com/download/desktop>

Grabe la imagen en una memoria USB siguiendo las instrucciones de la siguiente página:

<https://tutorials.ubuntu.com/tutorial/tutorial-create-a-usb-stick-on-windows>

Instale Ubuntu 16.04 desde la memoria USB siguiendo las instrucciones de la siguiente página:

<https://tutorials.ubuntu.com/tutorial/tutorial-install-ubuntu-desktop>

Asegúrese de instalar la versión correcta, de 32 o 64 bits.

2.2. Instalación de ROS

Una vez instalado Ubuntu, instale ROS Kinetic. Las instrucciones se encuentran en la siguiente página:

<http://wiki.ros.org/kinetic/Installation/Ubuntu>

En el paso 1.4, seleccione la opción **Desktop-Full**. Es importante haber instalado Ubuntu 16.04, dado que algunas de las instrucciones contenidas en esta página sólo funcionan correctamente en Ubuntu 15.10 y 16.04. También es muy recomendable instalar todo con las opciones por default, lo que facilitará los desarrollos del resto del curso.

2.3. Tutoriales

Ejecute las instrucciones contenidas en los tutoriales 1, 3, 4, 11 y 14 del nivel principiante de la siguiente dirección:

<http://wiki.ros.org/ROS/Tutorials>

Estos tutoriales sirven para crear dos *paquetes*: el primero contiene dos *nodos*, uno de los cuales publica un *tópico* y el otro se suscribe al mismo; el segundo contiene otros dos nodos, uno que atiende un *servicio* y el otro lo solicita y espera una respuesta.

3. Evaluación

La práctica se considerará entregada si los cuatro nodos del punto 2.3 funcionan correctamente, esto es, que la salida sea la descrita en los tutoriales.