



Mão na Massa com ROS

Ricardo B. Grando

Comandos

- **roscore** - Inicia o nodo Master.
- **roslaunch** [nome do pacote] [nome do nodo] - Executa um nodo.
- **rostopic** - Lida com nodos do ros.
- **rosservice** - Lida com serviços do ROS.
- **rostopic** - Lida com tópicos do ROS.
- **rosparam** - Lida com o serviço de parâmetros do ROS.

Mãos à obra

- **Instalando o ROS e definindo variáveis de ambiente.**
- **Trabalhando com Tópicos, Serviços e Parâmetros no TurtleSim.**
- **Criando um espaço de trabalho (catkin_workspace)**
- Criando um pacote
- Trabalhando com launch
- Escrevendo um nodo em roscpp que publica e um que se inscreve em um tópico
- Escrevendo um nodo em rospy para o TurtleSim
- Trabalhando com o simulado Stage
- Desenhando no RViz
- Utilizando o ROS em múltiplas máquinas.

Mãos à obra

- **Instalando o ROS e definindo variáveis de ambiente.**
- **Trabalhando com Tópicos, Serviços e Parâmetros no TurtleSim.**
- **Criando um espaço de trabalho (catkin_workspace)**
- **Criando um pacote**
- Trabalhando com launch
- Escrevendo um nodo em roscpp que publica e um que se inscreve em um tópico
- Escrevendo um nodo em rospy para o TurtleSim
- Trabalhando com o simulado Stage
- Desenhando no RViz
- Utilizando o ROS em múltiplas máquinas.

DICA: Use o TAB para completar os comandos

Criando um Pacote

- Acessando pasta src do workspace
 - `$ cd ~/catkin_ws/src`
- Executando o script de criação de pacotes
 - `# catkin_create_pkg <Nome Pacote> [Dependencia1] [Dependencia2]...`
 - `$ catkin_create_pkg c3 std_msgs rospy roscpp`
- Após a execução do script os seguintes arquivos são criados:
 - c3 - Pasta do pacote
 - package.xml - Arquivo com informações do pacote (versão, dependências, desenvolvedor)
 - CMakeLists.txt - Instruções de compilação do pacote
 - src - Pasta onde fica os códigos
 - include - Pasta onde fica os includes .h .hpp
- Para compilar o pacote basta usar o comando `catkin_make` na pasta principal `catkin_ws`.