

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.

Trabalhando com launch

- Launch é uma ferramenta utilizada para **configurar e executar vários os nodos** de uma só vez.
- O **launch executa o nodo master** automaticamente eliminando a necessidade de abrir um terminal para o comando \$roscore
- Um launch é definido por um **arquivo xml** com extensão .launch
- Normalmente os launches ficam salvos na pasta “launch” do pacote (boas práticas), entretanto eles podem ser salvos em qualquer lugar dentro da pasta do pacote.
- Comando de execução de um launch
 - \$ **roslaunch** [NomePacote] [NomeArquivo.launch]

Trabalhando com Launcher

- Usando launch para reproduzir as tarefas anteriores:

Arquivo: **rodaTurtle.launch**

```
<launch>
  <!-- Executa o primeiro simulador -->
  <node pkg="turtlesim" name="sim" type="turtlesim_node"/>
  <!-- Executa o publisher -->
  <node pkg="rostopic" name="publisher" type="rostopic"
    args="pub /turtle1/cmd_vel geometry_msgs/Twist -r 1 -- '[2.0, 0.0, 0.0]' '[0.0, 0.0, -1.8]'" />
  <!-- Cria outra tartaruga -->
  <node pkg="rosservice" name="navaTurtle" type="rosservice"
    args="call /spawn 8 5 3.0 'Michelangelo'" />
  <!-- Define um parâmetro -->
  <param name="teste" value="Michelangelo" />
</launch>
```

shorturl.at/agzJP

Trabalhando com Launcher

- Tags do launch:
 - `<launch>` - Principal TAG do launcher.
 - `<node>` - Especifica um nodo que será executado.
 - `<machine>` - Utilizado para execuções remotas.
 - `<include>` - Importar outro launcher dentro do launcher atual.
 - `<remap>` - Realiza re-mapeamento de nomes.
 - `<param>` ou `<rosparam>` - Define um parâmetro no serviço de parâmetros distribuído.
 - `<group>` - Cria um grupo de nodos, útil para aplicar uma configuração apenas a um grupo de nodos.
 - `<test>` - Executa um nodo em modo de teste.
 - `<arg>` - Passagem de parâmetros como argumento para o nodo.

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.