Ros学习笔记

目录

[1 Ros环境搭建 1](#_Toc134993134)

[2 Ros通信机制 1](#_Toc134993135)

[3 Ros通信机制进阶 1](#_Toc134993136)

[4 Ros运行管理 1](#_Toc134993137)

[5 Ros常用组件 3](#_Toc134993138)

[6 Ros机器人仿真 3](#_Toc134993139)

1 Ros环境搭建

* 1. Ros下载
  2. 终端快捷命令

新建功能包

mkdir -p 自定义空间名称/src

cd 自定义空间名称

catkin\_make

source ./devel/setup.bash 刷新环境变量

source .bashrc 刷新环境变量

rviz 打开图形化工具

2 Ros通信机制

3 Ros通信机制进阶

时间api：

1.api时间时刻能相加减

2.api时刻不能相加能相减

2.api持续时间可以相互加减

节点退出方式：

1. Ctrl+c
2. 同名节点启动
3. ros::shutdown()

4 Ros运行管理

4.4ros节点重名

1.rosrun

a.增加前缀

/abc 🡪 /ergouzi/abc

rosrun abc abc \_\_ns:=ergouzi

b.取别名

rosrun abc abc \_\_name:=xiaoqiang(别名)

/abc 🡪 /xiaoqiang

2.launch文件设置

launch文件

<launch>

<node pkg="turtlesim" type="turtlesim\_node" name="t1" />

<node pkg="turtlesim" type="turtlesim\_node" name="t2" />

<node pkg="turtlesim" type="turtlesim\_node" name="t1" ns="hello"/>

</launch>

在 node 标签中，name 属性是必须的，ns 可选。

编码设置命名空间

1.名称别名设置

核心代码:ros::init(argc,argv,"zhangsan",ros::init\_options::AnonymousName);

会在名称后面添加时间戳。

2.C++ 实现:命名空间

命名空间设置

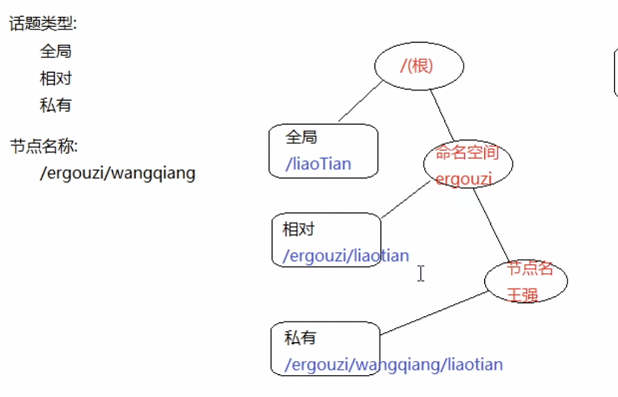
核心代码

std::map<std::string, std::string> map;

map["\_\_ns"] = "xxxx";

ros::init(map,"wangqiang");

4.5ros话题重名



Rosrun

1.方案1

将 teleop\_twist\_keyboard 节点的话题设置为/turtle1/cmd\_vel

启动键盘控制节点:rosrun teleop\_twist\_keyboard teleop\_twist\_keyboard.py /cmd\_vel:=/turtle1/cmd\_vel

启动乌龟显示节点: rosrun turtlesim turtlesim\_node

二者可以实现正常通信

2.方案2

将乌龟显示节点的话题设置为 /cmd\_vel

启动键盘控制节点:rosrun teleop\_twist\_keyboard teleop\_twist\_keyboard.py

启动乌龟显示节点: rosrun turtlesim turtlesim\_node /turtle1/cmd\_vel:=/cmd\_vel

二者可以实现正常通信

Launch

**launch 文件设置话题重映射语法:**

<node pkg="xxx" type="xxx" name="xxx">

<remap from="原话题" to="新话题" />

</node>

将 teleop\_twist\_keyboard 节点的话题设置为/turtle1/cmd\_vel

<launch>

<node pkg="turtlesim" type="turtlesim\_node" name="t1" />

<node pkg="teleop\_twist\_keyboard" type="teleop\_twist\_keyboard.py" name="key">

<remap from="/cmd\_vel" to="/turtle1/cmd\_vel" />

</node>

</launch>

编码实现

1.1全局名称

**格式:**以/开头的名称，和节点名称无关

**比如:**/xxx/yyy/zzz

ros::Publisher pub = nh.advertise<std\_msgs::String>("/chatter",1000);

**结果:**/chatter

1.2相对名称

**格式:**非/开头的名称,参考命名空间(与节点名称平级)来确定话题名称

ros::Publisher pub = nh.advertise<std\_msgs::String>("chatter",1000);

**结果:**xxx/chatter

1.3私有名称

**格式:**以~开头的名称

ros::NodeHandle nh("~");

ros::Publisher pub = nh.advertise<std\_msgs::String>("chatter",1000);

**结果：**/xxx/hello/chatter

4.6参数重名

Rosrun设置

启动乌龟显示节点，并设置参数 A = 100

rosrun turtlesim turtlesim\_node \_A:=100

5 Ros常用组件

5.1tf坐标变换

6 Ros机器人仿真

1. urdf

连杆link

关节joint

语法：开标签+闭标签

<robot name=””>（机器人名）

<link/joint name=””>（杆/关节）

<visual>（描述外观）

<geometry>（形状)

1. xacro

必须

<robot name="mycar" xmlns:xacro="http://wiki.ros.org/xacro">

属性封装

<xacro:property name="wheel\_radius" value="0.0325" />

定义宏

<xacro:macro name="wheel\_func" params="wheel\_name flag" >

内容（省略）

调用宏

<xacro:wheel\_func wheel\_name="left" flag="1" />

其他文件如launch调用

<xacro:include filename="my\_base.xacro" />

1. gazebo
2. rviz