# ros笔记2

1. 指令

rosrun turtlesim turtlesim\_node turtle节点

rosrun turtlesim turtle\_teleop\_key turtle 键盘控制节点

rosrun rqt\_graph rqt\_graph 图形化查看节点关系

rosrun rqt\_plot rqt\_plot 图形化显示指定topic中某一个量的变化图线

rostopic

rostopic echo *topic* 显示topic收到的消息

rostopic list [-v] 列出现有topic

rostopic pub -1 *topic* *msg\_type* *data* 向topic发布1次类型为msg\_type数据为data的消息

rostopic pub -r 1 *topic* *msg\_type* *data*向topic以1Hz速度发布类型为msg\_type数据为data的消息

rostopic type *topic* 显示topic对应的消息类型

rostopic hz *topic* 显示topic收到消息的频率

rosmsg show *msg\_type* 显示指定类型的消息的结构（该指令可前接管道）

rossrv show srv\_type 显示指定类型的服务的request/response结构（该指令可前接管道）

rosservice

rosservice list 列出服务

rosservice call *srv [request data]* 向srv发出请求，得到response返回

rosservice type *srv* 查看srv的类型

rosparam

rosparam list 列出参数

rosparam get/set *param* 存取参数

rosparam dump/load *file* 从文件存取参数

rosed package filename 编辑个包中得某个文件（不必提供完整路径）

roscp *package* *from* *to* 将某个东西复制到包内某个路径处

1. 概念

节点间通信通过对一个topic订阅/发布来实现，但订阅/发布的消息类型必须是同一类型。从此意义上说，一个topic对应的消息类型确定了这个topic的类型。

Service：节点间同行的一种方式。采用服务器/客户端模式，向一个service发出request并得到response。

Parameter Server：动态参数服务器。可以存取某些参数，把参数保存在文件中，或者从文件中加载参数

1. rqt工具

rqt 主程序

rqt\_console 控制台，显示消息

rqt\_graph 显示节点间得关系

rqt\_logger\_level 设置日志显示得级别

rqt\_plot 画曲线图

1. roslaunch

用于一次启动多个节点

roslaunch *package* *filename.launch*

需要一个.launch后缀得XML文件。

1. msg创建与使用

自定义消息类型存放在msg文件夹中，.msg后缀，每行一个字段类型+一个字段名。类型可以是基本类型、数组或是别的msg文件。

创建自定义msg后，要修改CMakeList.txt并重新catkin\_make install确保消息类型能够使用

1. srv的创建和使用

自定义服务类型放在srv文件夹里，.srv后缀，格式同msg，但要分request和response两部分，用---隔开。其余与msg一致。

1. 创建节点

* 在包的src文件夹里新建源文件，包含ros/ros.h，调用ros::init(argc, argv, nodename)，创建ros::NodeHandler对象。第一个NodeHandle创建时节点完全初始化好，所有NodeHandle销毁时节点关闭。NodeHandle是Node网络的主要访问点。
* ros::ok()在节点被踢出网络或是没有NodeHandle实例或是被ros::shutdown()或是被Ctrl-c时返回false
* ROS\_INFO用以代替printf/cout，用rqt\_console可以查看。

1. Publisher

ros::Publisher pub = nh.advertise<*msg\_type*>(*topic\_name*, *queue\_max\_length*)

*msg\_type* *msg*

pub.pulish(*msg*)

ros::Rate 用于sleep节点以控制publish频率

1. Subscriber

ros::Subscriber sub = nh.subscribe(topic, queue\_max\_length, callback)

callback要求接受一个msg\_type::ConstPtr（boost的智能指针），用以访问接收到的msg

ros::spin() 开始等待消息的循环

在Subscriber析构时会自动停止订阅