

编译 ROS 程序包

AI 航 团队

1. 图概念概述

Nodes: 节点,一个节点即为一个可执行文件,它可以通过 ROS 与其它节点进行通信。

Messages:消息, 消息是一种 ROS 数据类型,用于订阅或发布到一个话题。

Topics:话题, 节点可以发布消息到话题,也可以订阅话题以接收消息。

Master:节点管理器, ROS 名称服务 (比如帮助节点找到彼此)。

rosout: ROS 中相当于 stdout/stderr。

roscore: 主机+ rosout + 参数服务器 (参数服务器会在后面介绍)。

2. 节点

一个节点其实只不过是 ROS 程序包中的一个可执行文件。ROS 节点可以使用 ROS 客户库与其他节点通信。节点可以发布或接收一个话题。节点也可以提供或使用某种服务。

节点是 ros 中非常重要的一个概念,这里举一个通俗的例子:

例如,咱们有一个机器人,和一个遥控器,那么这个机器人和遥控器开始工作后,就是两个节点。遥控器起到了下达指令的作用;机器人负责监听遥控器下达的指令,完成相应动作。从这里我们可以看出,节点是一个能执行特定工作任务的工作单元,并且能够相互通信,从而实现一个机器人系统整体的功能。在这里我们把遥控器和机器人简单定义为两个节点,实际上在机器人中根据控制器、传感器、执行机构等不同组成模块,还可以将其进一步细分为更多的节点,这个是根据用户编写的程序来定义的。

3. 客户端

ROS 客户端库允许使用不同编程语言编写的节点之间互相通信:

`rospy` = python 客户端库

`roscpp` = c++ 客户端库

4. roscore

`roscore` 是你在运行所有 ROS 程序前首先要运行的命令。

```
$ roscore
```

然后你会看到类似下面的输出信息:

```
... logging to ~/.ros/log/9cf88ce4-b14d-11df-8a75-00251148e8cf/roslaunch-
machine_name-13039.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
started roslaunch server http://machine_name:33919/
ros_comm version 1.4.7
SUMMARY
=====
PARAMETERS
* /rosversion
* /rostdistro
NODES
auto-starting new master
process[master]: started with pid[13054]
ROS_MASTER_URI=http://machine_name:11311/
setting /run_id to 9cf88ce4-b14d-11df-8a75-00251148e8cf
process[rosout-1]: started with pid [13067]
started core service [/rosout]
```

如果 `roscore` 运行后无法正常初始化, 很有可能是存在网络配置问题。

5. 使用 `roscnode`

打开一个新的终端, 可以使用 `roscnode` 像运行 `roscore` 一样看看在

运行什么...

`roscnode` 显示当前运行的 ROS 节点信息。 `roscnode list` 指令列出活跃的节点:

```
$ roscnode list
```

6. 使用 `roscrun`

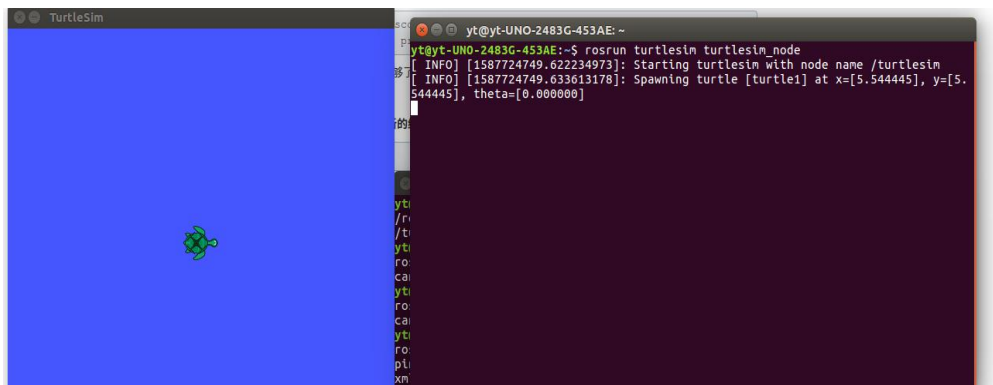
roslaunch 允许你使用包名直接运行一个包内的节点(而不需要知道这个包的路径)。

用法: 现在我们可以运行 turtlesim 包中的 turtlesim_node。

```
$ roslaunch turtlesim turtlesim_node
```

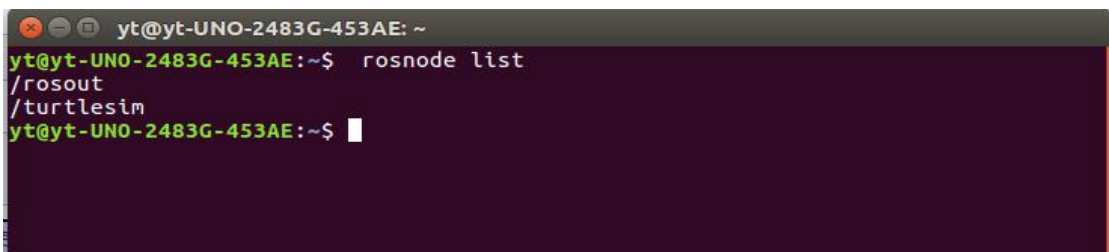
然后, 在一个新的终端:

你会看到 turtlesim 窗口:



再次执行 roslaunch

```
$ roslaunch
```



使用另外一个 roslaunch 指令, ping 来测试

关闭 turtlesim 窗口停止运行节点 (或者回到 roslaunch turtlesim 终端并使用`ctrl

-C`)。现在让我们重新运行它, 但是这一次使用 Remapping Argument 改变节点名称:

```
$ roslaunch turtlesim turtlesim_node __name:=my_turtle
```

```
yt@yt-UNO-2483G-453AE:~$ rosnode ping my_turtle
rosnode: node is [/my_turtle]
pinging /my_turtle with a timeout of 3.0s
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.525951ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.585079ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.424862ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.532866ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.446796ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.530005ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.450134ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.494003ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.454903ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.584126ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.540018ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.451088ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.509977ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.443935ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.482082ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.455141ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.483990ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.447035ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.529051ms
xmlrpc reply from http://yt-UNO-2483G-453AE:33797/      time=0.428915ms
```