11、ROS2参数服务案例

1、参数简介

类似C++编程中的全局变量，可以便于在多个程序中共享某些数据，参数是ROS机器人系统中的全局字典，可以运行多个节点中共享数据。

在ROS系统中，参数是以全局字典的形态存在的，什么叫字典？就像真实的字典一样，由名称和数值组成，也叫做键和值，合成键值。或者我们也可以理解为，就像编程中的参数一样，有一个参数名 ，然后跟一个等号，后边就是参数值了，在使用的时候，访问这个参数名即可。

参数的特性非常丰富，比如某一个节点共享了一个参数，其他节点都可以访问，如果某一个节点对参数进行了修改，其他节点也有办法立刻知道，从而获取最新的数值。

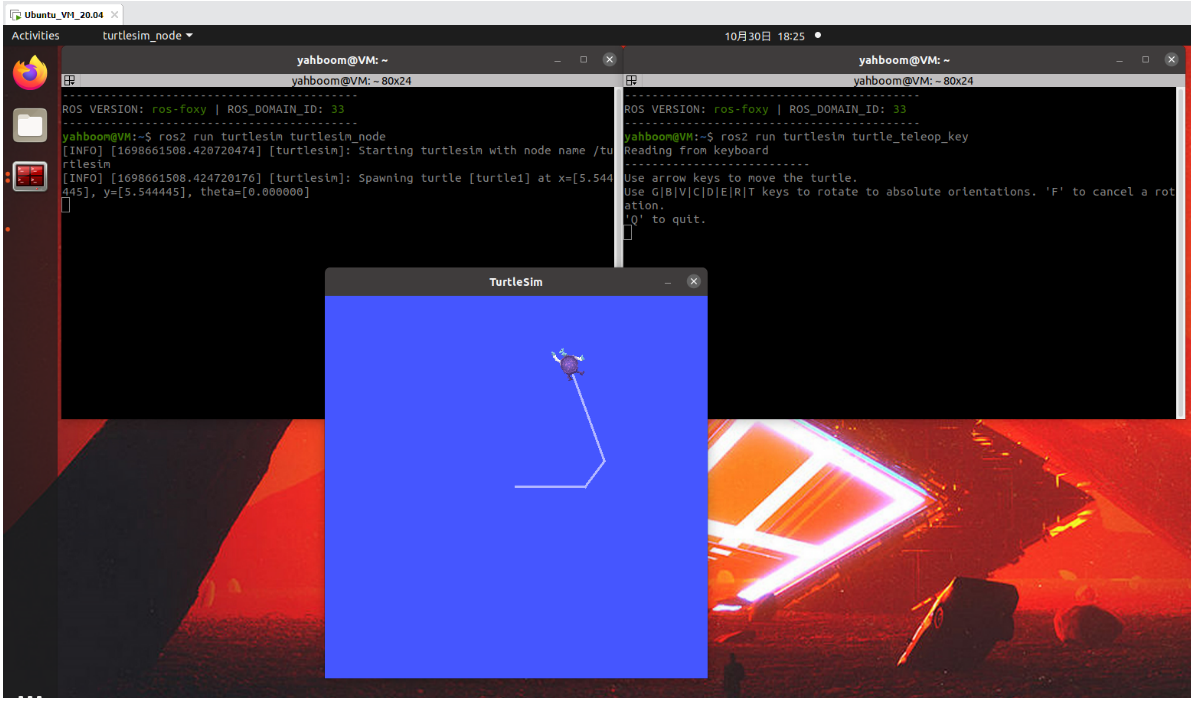
2、小海龟例程中的参数

在小海龟的例程中，仿真器也提供了不少参数，我们一起来通过这个例程，熟悉下参数的含义和命令行的使用方法。

由于这里要显示GUI，为方便操作，以下案例在教程配套的虚拟机中演示

1、虚拟机启动两个终端，分别运行小海龟仿真器和键盘控制节点：

```
ros2 run turtlesim turtlesim_node
ros2 run turtlesim turtle_teleop_key
```

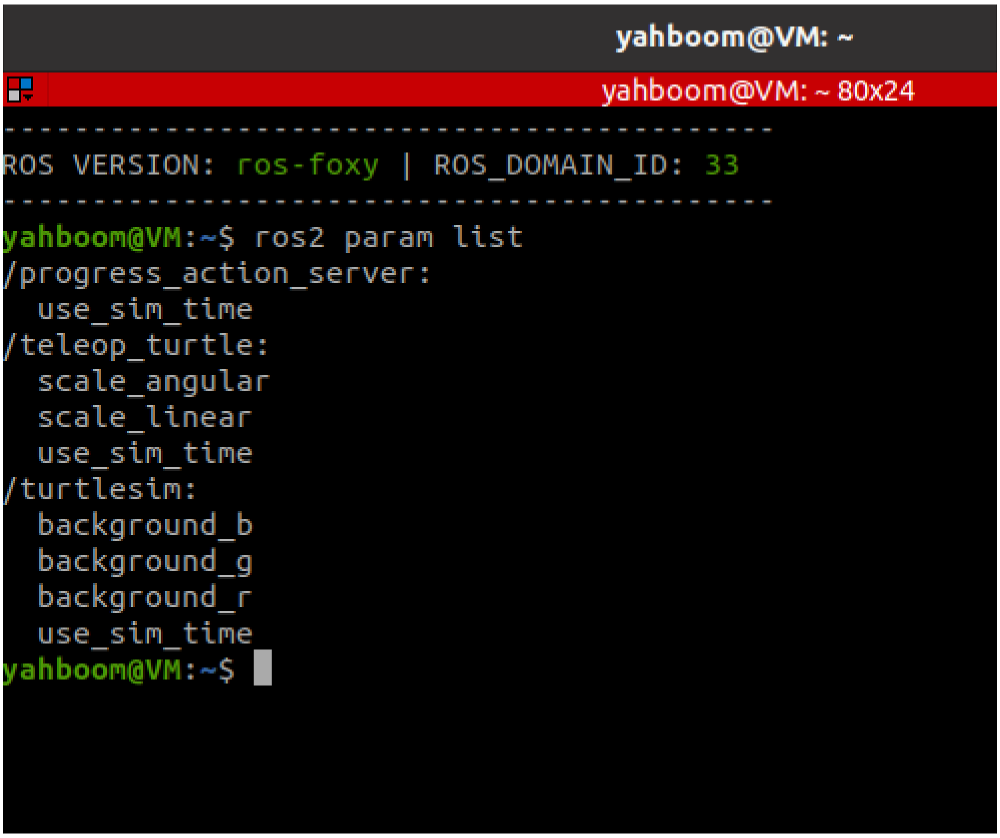


```
-----------------------------------------
ROS VERSION: ros-foxy | ROS_DOMAIN_ID: 33
-----------------------------------------
yahboom@VM:~$ ros2 run turtlesim turtlesim_node
[INFO] [1698661508.420720474] [turtlesim]: Starting turtlesim with node name /tu
rtlesim
[INFO] [1698661508.424720176] [turtlesim]: Spawning turtle [turtle1] at x=[5.544
445], y=[5.544445], theta=[0.000000]
```

```
-----------------------------------------
ROS VERSION: ros-foxy | ROS_DOMAIN_ID: 33
-----------------------------------------
yahboom@VM:~$ ros2 run turtlesim turtle_teleop_key
Reading from keyboard
---------------------------
Use arrow keys to move the turtle.
Use G|B|V|C|D|E|R|T keys to rotate to absolute orientations. 'F' to cancel a rot
ation.
'Q' to quit.
```

2、虚拟机启动一个终端，并使用如下命令查看参数列表

```
ros2 param list
```



yahboom@VM: ~

yahboom@VM: ~ 80x24

```
----------------------------------------------
ROS VERSION: ros-foxy | ROS_DOMAIN_ID: 33
----------------------------------------------
yahboom@VM:~$ ros2 param list
/progress_action_server:
  use_sim_time
/teleop_turtle:
  scale_angular
  scale_linear
  use_sim_time
/turtlesim:
  background_b
  background_g
  background_r
  use_sim_time
yahboom@VM:~$ 
```

3、参数查询与修改

如果想要查询或者修改某个参数的值，可以在param命令后边跟get或者set子命令：

```
ros2 param describe turtlesim background_b   # 查看某个参数的描述信息
ros2 param get turtlesim background_b        # 查询某个参数的值
ros2 param set turtlesim background_b 10     # 修改某个参数的值
```

4、参数文件保存与加载

一个一个查询/修改参数太麻烦了，不如试一试参数文件，ROS中的参数文件使用yaml格式，可以在param命令后边跟dump子命令，将某个节点的参数都保存到文件中，或者通过load命令一次性加载某个参数文件中的所有内容：

```
ros2 param dump turtlesim >> turtlesim.yaml  # 将某个节点的参数保存到参数文件中
ros2 param load turtlesim turtlesim.yaml     # 一次性加载某一个文件中的所有参数
```

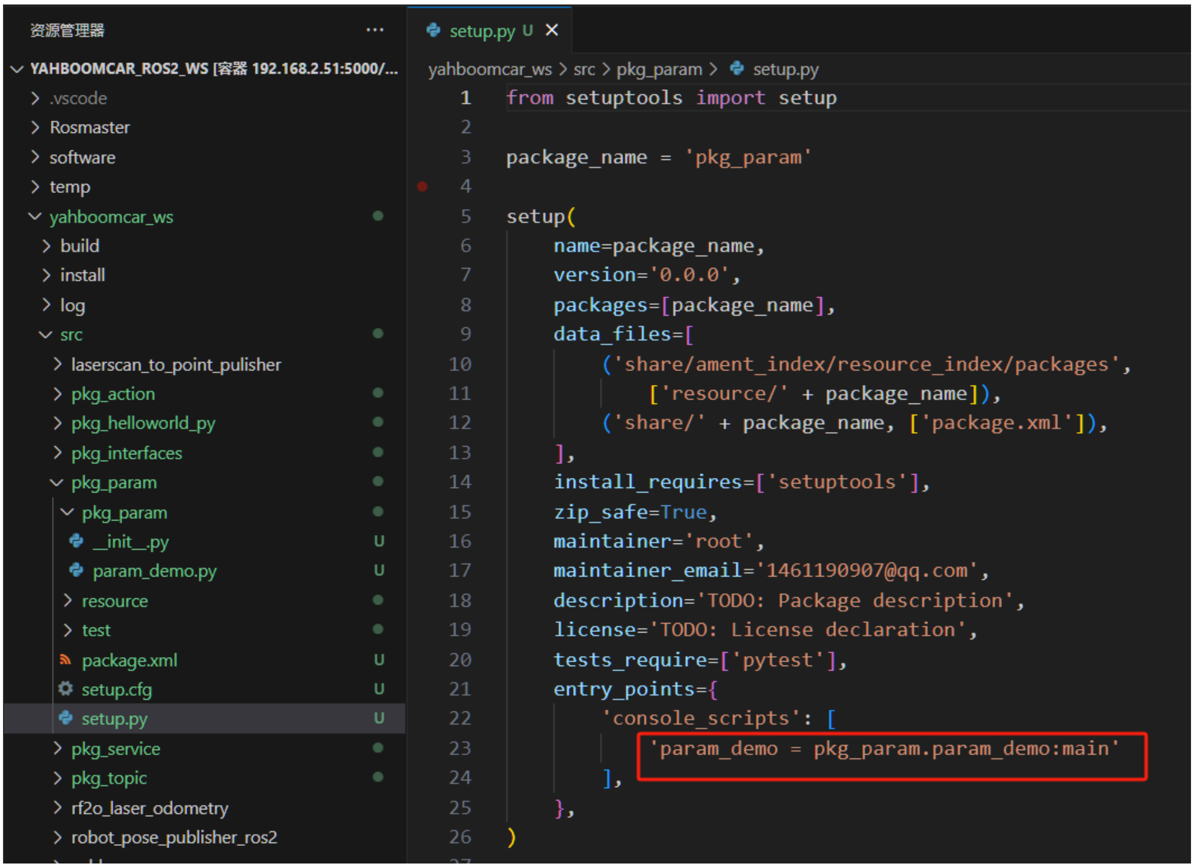
3、参数案例

3.1、新建功能包

- 在工作空间的src目录下新建功能包

```
ros2 pkg create pkg_param --build-type ament_python --dependencies rclpy --node-name param_demo
```

执行完上述命令，会创建pkg_param功能包，同时会创建一个param_demo的节点，并且已经配置好相关的配置文件



```python
from setuptools import setup

package_name = 'pkg_param'

setup(
    name=package_name,
    version='0.0.0',
    packages=[package_name],
    data_files=[
        ('share/ament_index/resource_index/packages',
            ['resource/' + package_name]),
        ('share/' + package_name, ['package.xml']),
    ],
    install_requires=['setuptools'],
    zip_safe=True,
    maintainer='root',
    maintainer_email='1461190907@qq.com',
    description='TODO: Package description',
    license='TODO: License declaration',
    tests_require=['pytest'],
    entry_points={
        'console_scripts': [
            'param_demo = pkg_param.param_demo:main'
        ],
    },
)
```

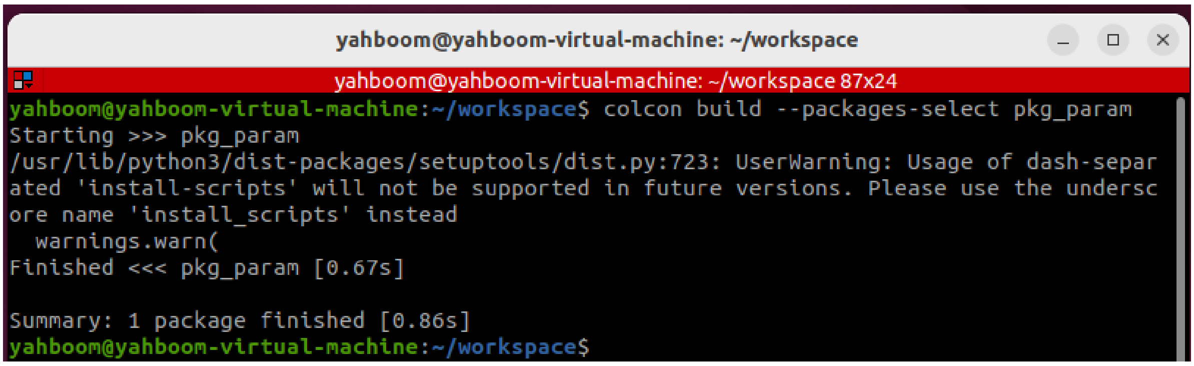
3.2、代码实现

接下来编辑【param_demo.py】实现发布方的功能，添加如下代码：

```python
import rclpy                                     # ROS2 Python接口库
from rclpy.node import Node                     # ROS2 节点类

class ParameterNode(Node):
    def __init__(self, name):
        super().__init__(name)                                    # ROS2节点父类初始化
        self.timer = self.create_timer(2, self.timer_callback)    # 创建一个定时器（单位为秒的周期，定时执行的回调函数）
        self.declare_parameter('robot_name', 'muto')              # 创建一个参数，并设置参数的默认值

    def timer_callback(self):                                      # 创建定时器周期执行的回调函数
        robot_name_param = self.get_parameter('robot_name').get_parameter_value().string_value   # 从ROS2系统中读取参数的值
        self.get_logger().info('Hello %s!' % robot_name_param)     # 输出日志信息，打印读取到的参数值

def main(args=None):                                 # ROS2节点主入口main函数
    rclpy.init(args=args)                            # ROS2 Python接口初始化
    node = ParameterNode("param_declare")            # 创建ROS2节点对象并进行初始化
    rclpy.spin(node)                                 # 循环等待ROS2退出
    node.destroy_node()                                      # 销毁节点对象
    rclpy.shutdown()                                         # 关闭ROS2 Python接口
```

3.3、编译功能包

```
colcon build --packages-select pkg_param
```



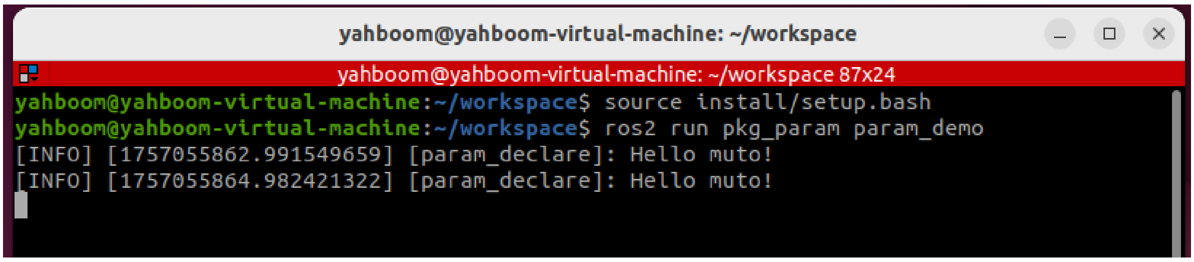
yahboom@yahboom-virtual-machine: ~/workspace
yahboom@yahboom-virtual-machine: ~/workspace 87x24

```
yahboom@yahboom-virtual-machine:~/workspace$ colcon build --packages-select pkg_param
Starting >>> pkg_param
/usr/lib/python3/dist-packages/setuptools/dist.py:723: UserWarning: Usage of dash-separ
ated 'install-scripts' will not be supported in future versions. Please use the undersc
ore name 'install_scripts' instead
  warnings.warn(
Finished <<< pkg_param [0.67s]

Summary: 1 package finished [0.86s]
yahboom@yahboom-virtual-machine:~/workspace$
```

3.4、运行程序

- 先刷新环境变量，然后运行节点

```
ros2 run pkg_param param_demo
```



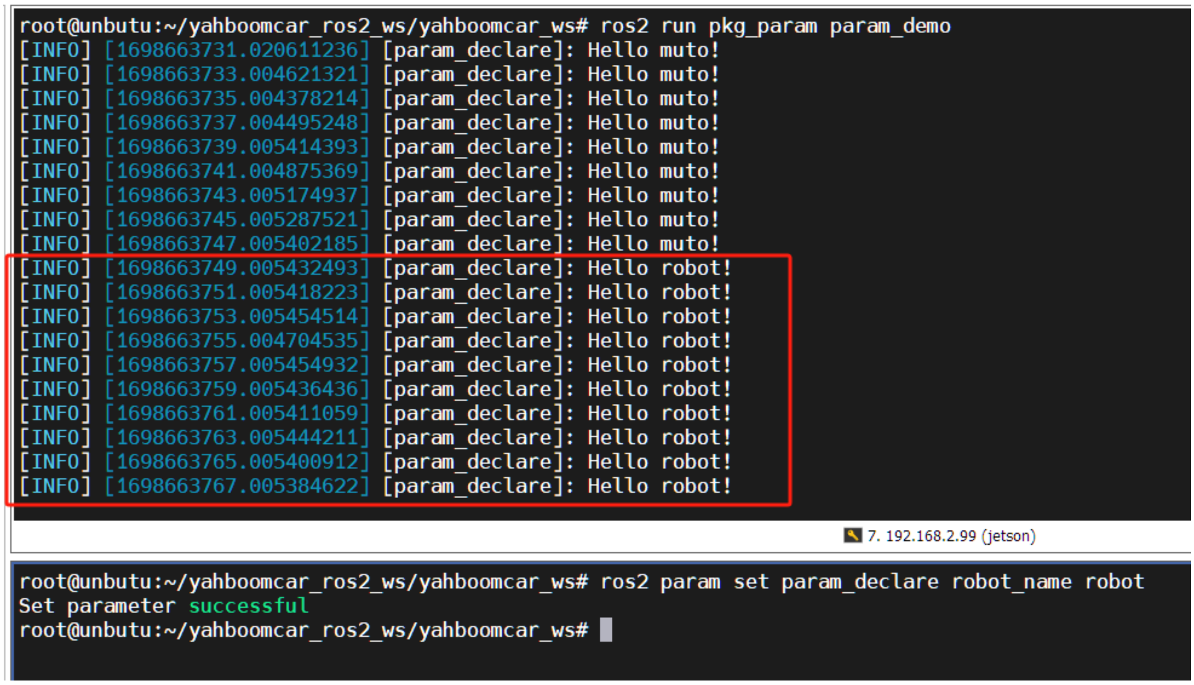
yahboom@yahboom-virtual-machine: ~/workspace
yahboom@yahboom-virtual-machine: ~/workspace 87x24

```
yahboom@yahboom-virtual-machine:~/workspace$ source install/setup.bash
yahboom@yahboom-virtual-machine:~/workspace$ ros2 run pkg_param param_demo
[INFO] [1757055862.991549659] [param_declare]: Hello muto!
[INFO] [1757055864.982421322] [param_declare]: Hello muto!
```

开启另一个终端，将robot_name设置为robot：

```
ros2 param set param_declare robot_name robot
```

终端中可以看到循环打印的日志信息，其中的“muto”是我们默认设置的一个参数值，参数名称是“robot_name”，通过命令行修改这个参数后，看到终端中也跟着变化了。



```
root@unbutu:~/yahboomcar_ros2_ws/yahboomcar_ws# ros2 run pkg_param param_demo
[INFO] [1698663731.020611236] [param_declare]: Hello muto!
[INFO] [1698663733.004621321] [param_declare]: Hello muto!
[INFO] [1698663735.004378214] [param_declare]: Hello muto!
[INFO] [1698663737.004495248] [param_declare]: Hello muto!
[INFO] [1698663739.005414393] [param_declare]: Hello muto!
[INFO] [1698663741.004875369] [param_declare]: Hello muto!
[INFO] [1698663743.005174937] [param_declare]: Hello muto!
[INFO] [1698663745.005287521] [param_declare]: Hello muto!
[INFO] [1698663747.005402185] [param_declare]: Hello muto!
[INFO] [1698663749.005432493] [param_declare]: Hello robot!
[INFO] [1698663751.005418223] [param_declare]: Hello robot!
[INFO] [1698663753.005454514] [param_declare]: Hello robot!
[INFO] [1698663755.004704535] [param_declare]: Hello robot!
[INFO] [1698663757.005454932] [param_declare]: Hello robot!
[INFO] [1698663759.005436436] [param_declare]: Hello robot!
[INFO] [1698663761.005411059] [param_declare]: Hello robot!
[INFO] [1698663763.005444211] [param_declare]: Hello robot!
[INFO] [1698663765.005400912] [param_declare]: Hello robot!
[INFO] [1698663767.005384622] [param_declare]: Hello robot!
```

```
root@unbutu:~/yahboomcar_ros2_ws/yahboomcar_ws# ros2 param set param_declare robot_name robot
Set parameter successful
root@unbutu:~/yahboomcar_ros2_ws/yahboomcar_ws# 
```