16、ROS2常用命令工具

1、包管理工具ros2 pkg

1.1、ros2 pkg create

功能：创建功能包，创建时候需要指定包名、编译方式、依赖项等。

格式：

```
ros2 pkg create <package_name> --build-type <build-type> --dependencies
<dependencies>
```

ros2命令中：

- **pkg**：表示功能包相关的功能；
- **create**：表示创建功能包；
- **package_name**：必须项：新建功能包的名字；
- **build-type**：必须项：表示新创建的功能包是C++还是Python的，如果使用C++或者C，那这里就跟ament_cmake，如果使用Python，就跟ament_python；
- **dependencies**：可选项:表示功能包的依赖项，C++功能包需包含rclcpp，Python功能包需包含rclpy ,还有其它需要的依赖；

1.2、ros2 pkg list

功能：查看系统中功能包列表

格式：

```
ros2 pkg list
```

```
yahboom@yahboom-virtual-machine:~$ ros2 pkg list
action_msgs
action_tutorials_cpp
action_tutorials_interfaces
action_tutorials_py
actionlib_msgs
ament_cmake
ament_cmake_auto
ament_cmake_copyright
ament_cmake_core
ament_cmake_cppcheck
ament_cmake_cpplint
ament_cmake_export_definitions
ament_cmake_export_dependencies
ament_cmake_export_include_directories
ament_cmake_export_interfaces
ament_cmake_export_libraries
ament_cmake_export_link_flags
ament_cmake_export_targets
ament_cmake_flake8
ament_cmake_gmock
ament_cmake_gtest
ament_cmake_include_directories
ament_cmake_libraries
ament_cmake_lint_cmake
ament_cmake_pep257
ament_cmake_pytest
ament_cmake_python
ament_cmake_ros
ament_cmake_target_dependencies
```

1.3、ros2 pkg executeables

功能：查看某个包内所有可执行文件

格式：

```
ros2 pkg executables pkg_name
```

```
yahboom@yahboom-virtual-machine:~$ ros2 pkg executables turtlesim
turtlesim draw_square
turtlesim mimic
turtlesim turtle_teleop_key
turtlesim turtlesim_node
```

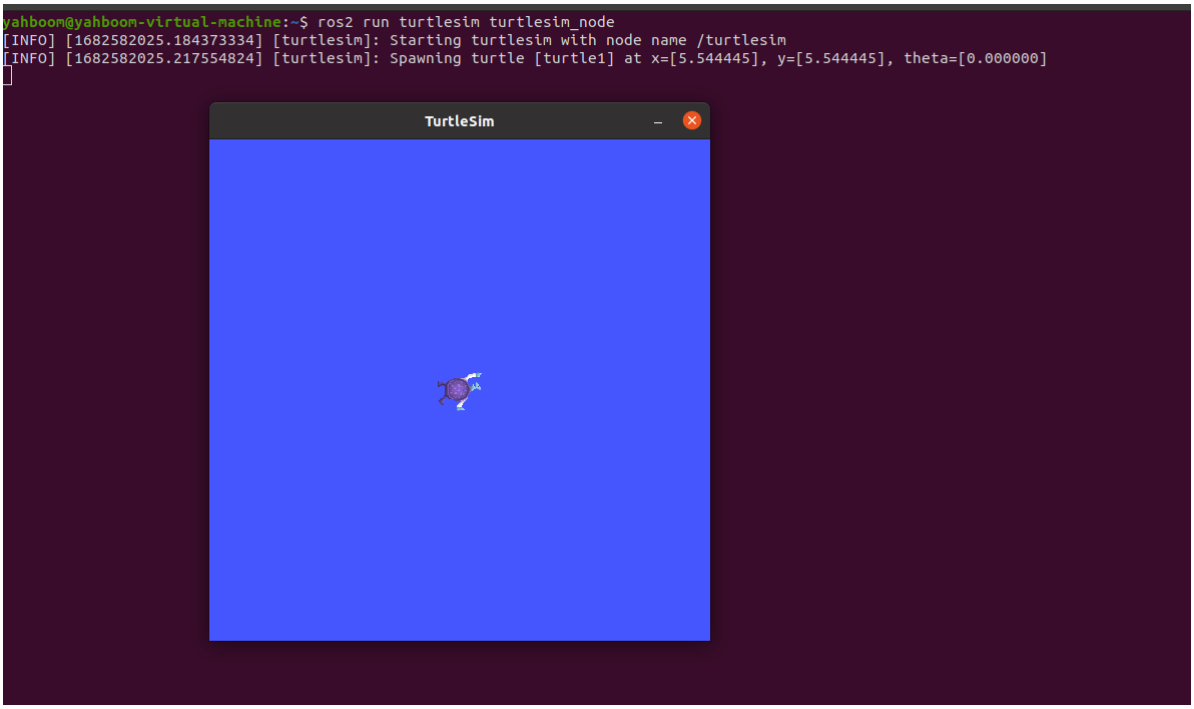
2、节点运行 ros2 run

功能：运行功能包节点程序

格式：

```
ros2 run pkg_name node_name
```

- pkg_name：功能包名字
- node_name：可执行程序的名字



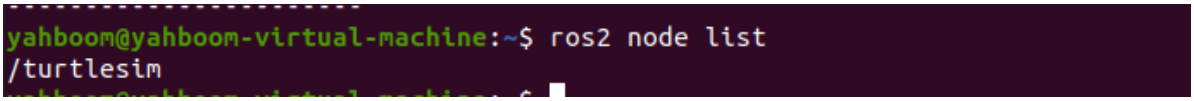
3、节点相关工具 ros2 node

3.1、ros2 node list

功能： 罗列出所有在当前域内节点名称

格式：

```
ros2 node list
```



3.2、ros2 node info

功能： 查看节点详细信息，包括订阅、发布的消息，开启的服务和动作等

格式：

```
ros2 node info node_name
```

- node_name：需要查看的节点名称

```
yahboom@yahboom-virtual-machine:~$ ros2 node info /turtlesim
/turtlesim
  Subscribers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /turtle1/cmd_vel: geometry_msgs/msg/Twist
  Publishers:
    /parameter_events: rcl_interfaces/msg/ParameterEvent
    /rosout: rcl_interfaces/msg/Log
    /turtle1/color_sensor: turtlesim/msg/Color
    /turtle1/pose: turtlesim/msg/Pose
  Service Servers:
    /clear: std_srvs/srv/Empty
    /kill: turtlesim/srv/Kill
    /reset: std_srvs/srv/Empty
    /spawn: turtlesim/srv/Spawn
    /turtle1/set_pen: turtlesim/srv/SetPen
    /turtle1/teleport_absolute: turtlesim/srv/TeleportAbsolute
    /turtle1/teleport_relative: turtlesim/srv/TeleportRelative
    /turtlesim/describe_parameters: rcl_interfaces/srv/DescribeParameters
    /turtlesim/get_parameter_types: rcl_interfaces/srv/GetParameterTypes
    /turtlesim/get_parameters: rcl_interfaces/srv/GetParameters
    /turtlesim/list_parameters: rcl_interfaces/srv/ListParameters
    /turtlesim/set_parameters: rcl_interfaces/srv/SetParameters
    /turtlesim/set_parameters_atomically: rcl_interfaces/srv/SetParametersAtomically
  Service Clients:

  Action Servers:
    /turtle1/rotate_absolute: turtlesim/action/RotateAbsolute
  Action Clients:
```

4、话题相关工具 ros2 topic

4.1、ros2 topic list

功能：罗列出当前域内的所有话题

格式：

```
ros2 topic list
```

```
yahboom@yahboom-virtual-machine:~$ ros2 topic list
/parameter_events
/rosout
/turtle1/cmd_vel
/turtle1/color_sensor
/turtle1/pose
```

4.2、ros2 topic info

功能：显示话题消息类型，订阅者/发布者数量

格式：

```
ros2 topic info topic_name
```

- topic_name：需要查询的话题的名字

```
yahboom@yahboom-virtual-machine:~$ ros2 topic info /turtle1/cmd_vel
Type: geometry_msgs/msg/Twist
Publisher count: 0
Subscription count: 1
```

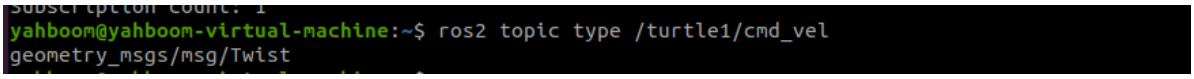
4.3、ros2 topic type

功能：查看话题的消息类型

格式：

```
ros2 topic type topic_name
```

- topic_name：需要查询话题类型的名字



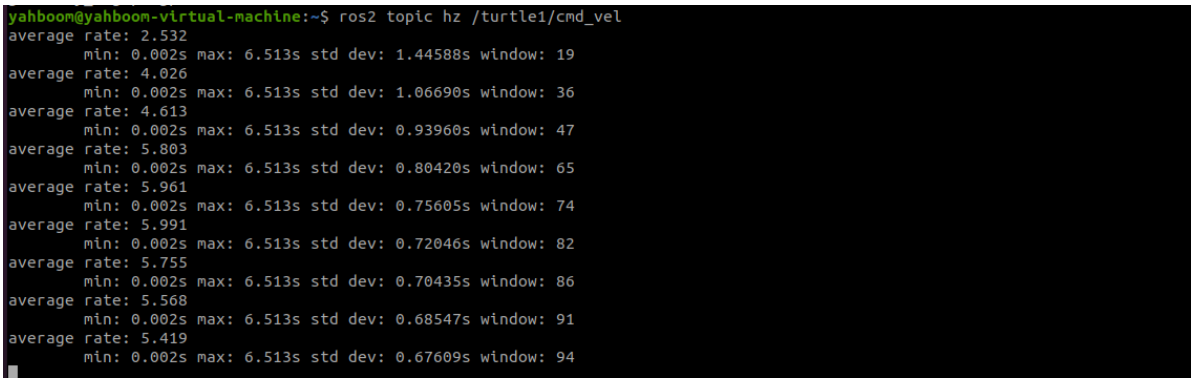
4.4、ros2 topic hz

功能：显示话题平均发布频率

格式：

```
ros2 topic hz topic_name
```

- topic_name：需要查询话题频率的名字



4.5、ros2 topic echo

功能：在终端打印话题消息，类似于一个订阅者

格式：ros2 topic echo topic_name

- topic_name：需要打印消息的话题的名字

```
yahboom@yahboom-virtual-machine:~$ ros2 topic echo /turtle1/cmd_vel
linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
---
linear:
  x: 2.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0
```

4.5、ros2 topic pub

功能：在终端发布指定话题消息

格式：

```
ros2 topic pub topic_name message_type message_content
```

- topic_name：需要发布话题消息的话题的名字
- message_type：话题的数据类型
- message_content：消息内容

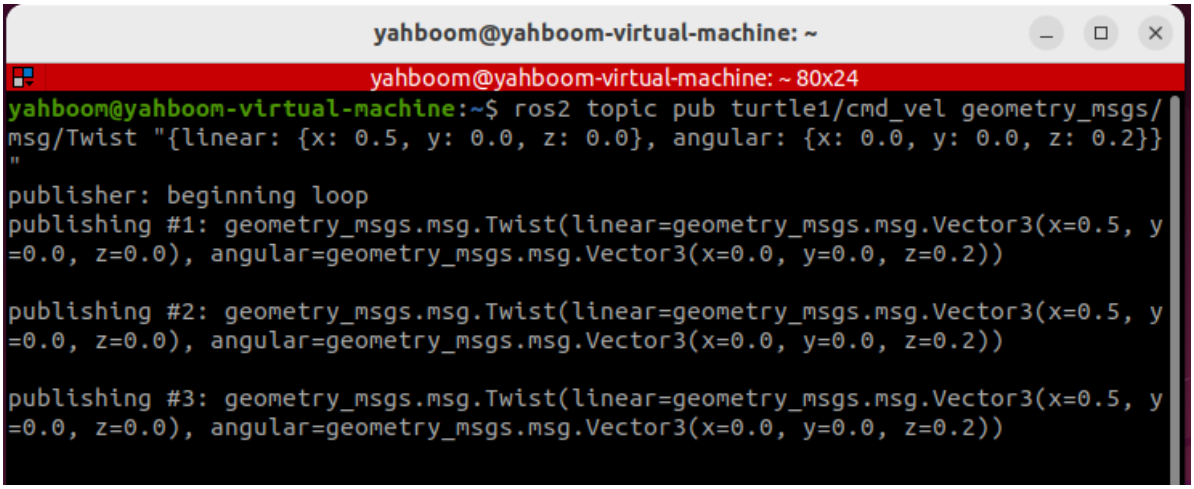
默认是以1Hz的频率循环发布，可以设置以下参数，

- 参数-1只发布一次，ros2 topic pub -1 topic_name message_type message_content
- 参数-t count循环发布count次结束，ros2 topic pub -t count topic_name message_type message_content
- 参数-r count以count Hz的频率循环发布，ros2 topic pub -r count topic_name message_type message_content

示例：

- 通过命令行发布速度指令
- 这里需要注意的是每个冒号后是有个空格，否则的话会提示格式错误

```
ros2 topic pub turtle1/cmd_vel geometry_msgs/msg/Twist "{linear: {x: 0.5, y: 0.0, z: 0.0}, angular: {x: 0.0, y: 0.0, z: 0.2}}"
```



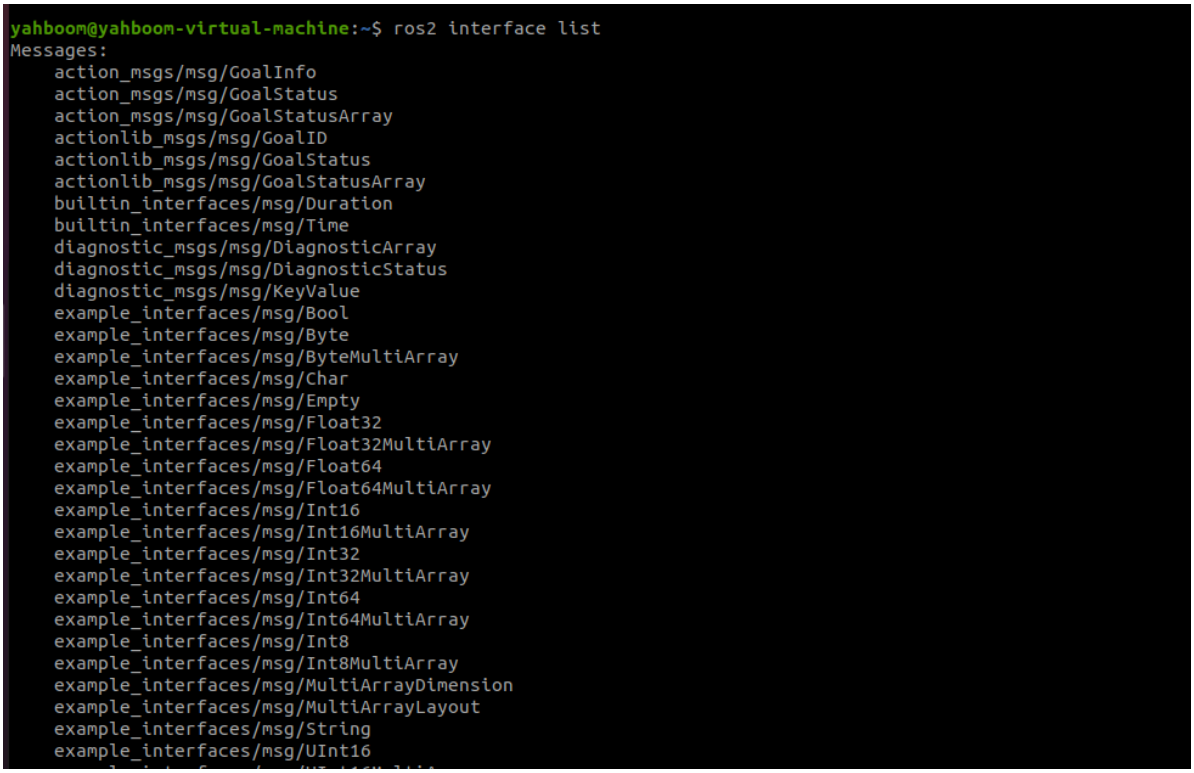
5、接口相关工具ros2 interface

5.1、ros2 interface list

功能：罗列当前系统的所有接口，包括话题、服务、动作。

格式：

```
ros2 interface list
```



5.2、ros2 interface show

功能：显示指定接口的详细内容

格式：

```
ros2 interface show interface_name
```

- interface_name：需要显示的接口内容的名字

```
yahboom@yahboom-virtual-machine:~$ ros2 interface show sensor_msgs/msg/LaserScan
# Single scan from a planar laser range-finder
#
# If you have another ranging device with different behavior (e.g. a sonar
# array), please find or create a different message, since applications
# will make fairly laser-specific assumptions about this data

std_msgs/Header header # timestamp in the header is the acquisition time of
                      # the first ray in the scan.
                      #
                      # in frame frame_id, angles are measured around
                      # the positive Z axis (counterclockwise, if Z is up)
                      # with zero angle being forward along the x axis

float32 angle_min     # start angle of the scan [rad]
float32 angle_max     # end angle of the scan [rad]
float32 angle_increment  # angular distance between measurements [rad]

float32 time_increment   # time between measurements [seconds] - if your scanner
                         # is moving, this will be used in interpolating position
                         # of 3d points
float32 scan_time        # time between scans [seconds]

float32 range_min        # minimum range value [m]
float32 range_max        # maximum range value [m]

float32[] ranges         # range data [m]
                         # (Note: values < range_min or > range_max should be discarded)
float32[] intensities    # intensity data [device-specific units].  If your
                         # device does not provide intensities, please leave
                         # the array empty
```

6、服务相关工具 ros2 service

6.1、ros2 service list

功能：罗列出当前域内所有的服务

格式：

```
ros2 interface show interface_name
```

```
yahboom@yahboom-virtual-machine:~$ ros2 service list
/clear
/kill
/reset
/spawn
/teleop_turtle/describe_parameters
/teleop_turtle/get_parameter_types
/teleop_turtle/get_parameters
/teleop_turtle/list_parameters
/teleop_turtle/set_parameters
/teleop_turtle/set_parameters_atomically
/turtle1/set_pen
/turtle1/teleport_absolute
/turtle1/teleport_relative
/turtlesim/describe_parameters
/turtlesim/get_parameter_types
/turtlesim/get_parameters
/turtlesim/list_parameters
/turtlesim/set_parameters
/turtlesim/set_parameters_atomically
yahboom@yahboom-virtual-machine:~$
```

6.2、ros2 service call

功能：调用指定服务

格式：

```
ros2 interface call service_name service_Type arguments
```

- service_name：需要调用的服务
- service_Type：服务数据类型
- arguments：提供服务需要的参数

例如，调用生成海龟服务

```
ros2 service call /spawn turtlesim/srv/Spawn "{x: 2, y: 2, theta: 0.2, name: 'turtle10'}"
```