古月居 GYH.AI ROS

ROS入门
21讲

8.ROS命令行工具的使用

主讲人：古月

ROS命令行工具

常用命令

- rostopic
- rosservice
- rosnode
- rosparam
- rosmsg
- rossrv

WORKSPACES

Create Workspace

```
mkdir catkin_ws && cd catkin_ws
wstool init src
catkin_make
source devel/setup.bash
```

Add Repo to Workspace

```
roscd; cd ../src
wstool set repo_name \
--git http://github.com/org/repo_name.git \
--version=kinetic-devel
wstool up
```

Resolve Dependencies in Workspace

```
sudo rosdep init  # only once
rosdep update
rosdep install --from-paths src --ignore-src \
--rosdistro=${ROS_DISTRO} -y
```

PACKAGES

Create a Package

```
catkin_create_pkg package_name [dependencies ...]
```

Package Folders

`include/package_name`	C++ header files
`src`	Source files. Python libraries in subdirectories
`scripts`	Python nodes and scripts
`msg, srv, action`	Message, Service, and Action definitions

Release Repo Packages

```
catkin_generate_changelog
# review & commit changelogs
catkin_prepare_release
bloom-release --track kinetic --ros-distro kinetic repo_name
```

Reminders

- Testable logic
- Publish diagnostics
- Desktop dependencies in a separate package

CMakeLists.txt

Skeleton

```
cmake_minimum_required(VERSION 2.8.3)
project(package_name)
find_package(catkin REQUIRED)
catkin_package()
```

Package Dependencies

To use headers or libraries in a package, or to use a package's exported CMake macros, express a build-time dependency:

```
find_package(catkin REQUIRED COMPONENTS roscpp)
```

Tell dependent packages what headers or libraries to pull in when your package is declared as a catkin component:

```
catkin_package(
  INCLUDE_DIRS include
  LIBRARIES ${PROJECT_NAME}
  CATKIN_DEPENDS roscpp)
```

Note that any packages listed as `CATKIN_DEPENDS` dependencies must also be declared as a `<run_depend>` in `package.xml`.

Messages, Services

These go after find_package(), but before catkin_package().
Example:

```
find_package(catkin REQUIRED COMPONENTS message_generation
std_msgs)
add_message_files(FILES MyMessage.msg)
add_service_files(FILES MyService.msg)
generate_messages(DEPENDENCIES std_msgs)
catkin_package(CATKIN_DEPENDS message_runtime std_msgs)ww
```

Build Libraries, Executables

Goes after the `catkin_package()` call.

```
add_library(${PROJECT_NAME} src/main)
add_executable(${PROJECT_NAME}_node src/main)
target_link_libraries(
  ${PROJECT_NAME}_node ${catkin_LIBRARIES})
```

Installation

```
install(TARGETS ${PROJECT_NAME}
  DESTINATION ${CATKIN_PACKAGE_LIB_DESTINATION})
install(TARGETS ${PROJECT_NAME}_node
  DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION})
install(PROGRAMS scripts/myscript
  DESTINATION ${CATKIN_PACKAGE_BIN_DESTINATION})
install(DIRECTORY launch
  DESTINATION ${CATKIN_PACKAGE_SHARE_DESTINATION})
```

RUNNING SYSTEM

Run ROS using plain:

```
roscore
```

Alternatively, roslaunch will run its own roscore automatically if it can't find one:

```
roslaunch my_package package_launchfile.launch
```

Suppress this behaviour with the `--wait` flag.

Nodes, Topics, Messages

```
rosnode list
rostopic list
rostopic echo cmd_vel
rostopic hz cmd_vel
rostopic info cmd_vel
rosmsg show geometry_msgs/Twist
```

Remote Connection

Master's ROS environment:

- `ROS_IP` or `ROS_HOSTNAME` set to this machine's network address.
- `ROS_MASTER_URI` set to URI containing that IP or hostname.

Your environment:

- `ROS_IP` or `ROS_HOSTNAME` set to your machine's network address.
- `ROS_MASTER_URI` set to the URI from the master.

To debug, check ping from each side to the other, run roswtf on each side.

ROS Console

Adjust using rqt_logger_level and monitor via rqt_console. To enable debug output across sessions, edit the `$HOME/.ros/config/rosconsole.config` and add a line for your package:

```
log4j.logger.ros.package_name=DEBUG
```

And then add the following to your session:

```
export ROSCONSOLE_CONFIG_FILE=$HOME/.ros/config/rosconsole.config
```

Use the roslaunch `--screen` flag to force all node output to the screen, as if each declared `<node>` had the `output="screen"` attribute.

CLEARPATH ROBOTICS™

www.clearpathrobotics.com/ros-cheat-sheet

以小海龟为例

启动ROS Master

```
$ roscore
```

⬇

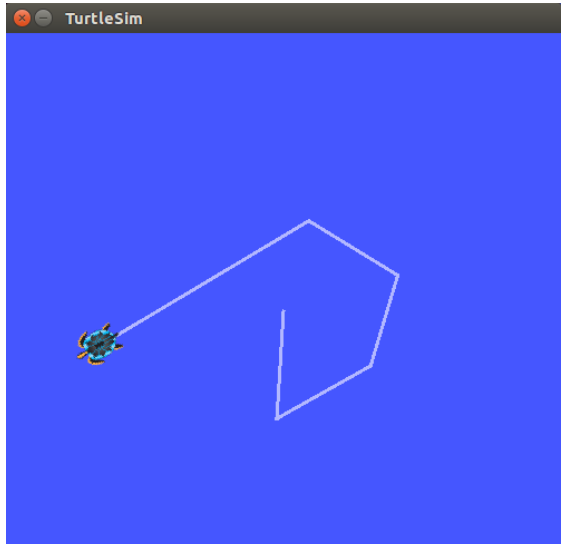
启动小海龟仿真器

```
$ rosrun turtlesim turtlesim_node
```

⬇

启动海龟控制节点

```
$ rosrun turtlesim turtle_teleop_key
```



小海龟仿真器界面

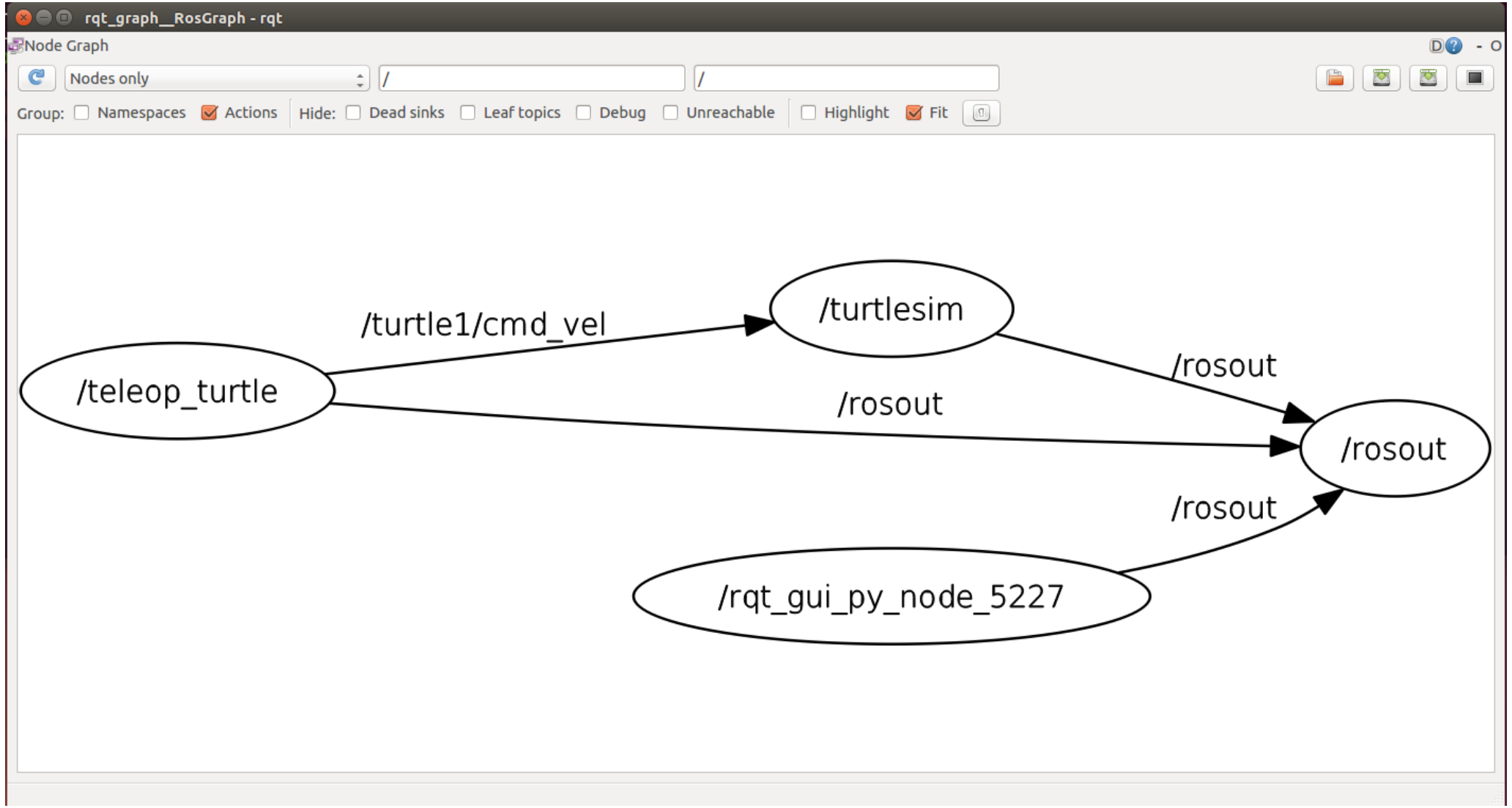
```
hcx@hcx-vpc:~$ rosrun turtlesim turtlesim_node
[ INFO] [1561200736.947992315]: Starting turtlesim with node name /turtlesim
[ INFO] [1561200736.954437402]: Spawning turtle [turtle1] at x=[5.544445], y=[5.
544445], theta=[0.000000]
```

启动海龟仿真器节点

```
hcx@hcx-vpc:~$ rosrun turtlesim turtle_teleop_key
Reading from keyboard
---------------------------
Use arrow keys to move the turtle.
```

启动海龟控制节点

以小海龟为例



使用rqt_graph可视化工具查看系统中运行的计算图

以小海龟为例

古月居 GYH.AI

查看话题列表

```
$ rosnode list
```

发布话题消息

```
$ rostopic pub -r 10 /turtle1/cmd_vel geometry_msgs/Twist "linear:
  x: 1.0
  y: 0.0
  z: 0.0
angular:
  x: 0.0
  y: 0.0
  z: 0.0"
```

发布服务请求

```
$ rosservice call /spawn "x: 5.0
y: 5.0
theta: 0.0
name: 'turtle2'"
```

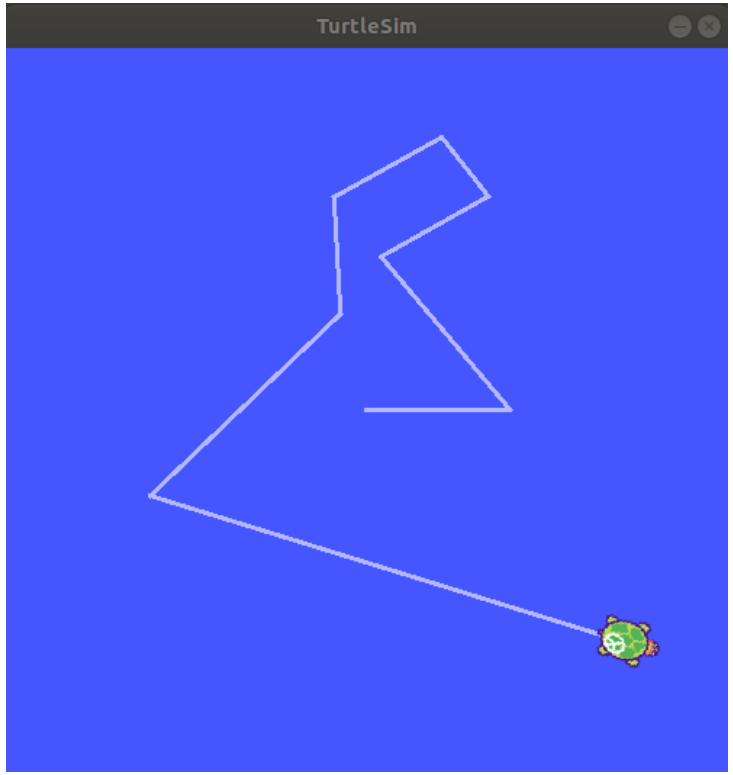
以小海龟为例

话题记录

```
$ rosbag record -a -O cmd_record
```

话题复现

```
$ rosbag play cmd_record.bag
```




```
hcx@hcx-vpc:~$ rosbag record -a -O cmd_record
[ INFO] [1562832989.601585958]: Subscribing to /turtle1/color_sensor
[ INFO] [1562832989.602647996]: Recording to cmd_record.bag.
[ INFO] [1562832989.605538325]: Subscribing to /turtle1/cmd_vel
[ INFO] [1562832989.608421198]: Subscribing to /rosout
[ INFO] [1562832989.611259134]: Subscribing to /rosout_agg
[ INFO] [1562832989.614049029]: Subscribing to /turtle1/pose
^Chcx@hcx-vpc:~$ ^C
hcx@hcx-vpc:~$ rosbag play cmd_record.bag
[ INFO] [1562833040.748840302]: Opening cmd_record.bag

Waiting 0.2 seconds after advertising topics... done.

Hit space to toggle paused, or 's' to step.
 [RUNNING]  Bag Time: 1562832989.617591   Duration: 0.000000 / 23.940571
 [RUNNING]  Bag Time: 1562832989.618096   Duration: 0.000505 / 23.940571
 [RUNNING]  Bag Time: 1562832989.718480   Duration: 0.100890 / 23.940571
 [RUNNING]  Bag Time: 1562832989.819050   Duration: 0.201459 / 23.940571
 [RUNNING]  Bag Time: 1562832989.859419   Duration: 0.241828 / 23.940571
 [RUNNING]  Bag Time: 1562832989.864236   Duration: 0.246645 / 23.940571
 [RUNNING]  Bag Time: 1562832989.879257   Duration: 0.261667 / 23.940571
 [RUNNING]  Bag Time: 1562832989.894307   Duration: 0.276716 / 23.940571
 [RUNNING]  Bag Time: 1562832989.911139   Duration: 0.293549 / 23.940571
 [RUNNING]  Bag Time: 1562832989.927119   Duration: 0.309528 / 23.940571
```

感谢观看

怕什么真理无穷，进一寸有一寸的欢喜

更多精彩，欢迎关注

古月居

古月春旭

bilibili 古月居GYH

ROS入门
21讲