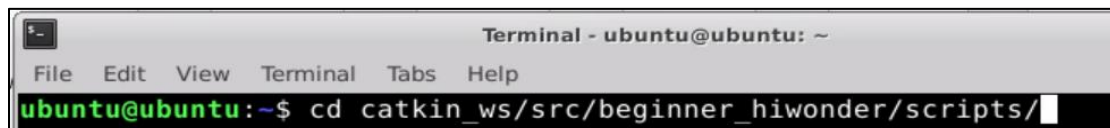


# Lesson 9 Write A Simple Server

## 1. Create Service Code

Note: Before creating service code, you need to create workspace and package first. The specific operation steps can be viewed in file "ROS Basic Lessons->Lesson 3 Create Workspace and Package".

- 1) Enter "**cd catkin\_ws/src/beginner\_hiwonder/scripts/**" command and press "Enter".



```
Terminal - ubuntu@ubuntu: ~
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~$ cd catkin_ws/src/beginner_hiwonder/scripts/
```

- 2) Enter "**vi turtle\_command\_server.py**" command to edit program and copy the following program. If want to modify, you can press "i". After modifying, press "Esc" and enter ":wq" to save and exit.



```
Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder/scripts$ vi turtle_command_server.py
```

```
#!/usr/bin/env python

# -*- coding: utf-8 -*-

import rospy

import thread,time

from geometry_msgs.msg import Twist

from std_srvs.srv import Trigger, TriggerResponse

pubCommand = False;

turtle_vel_pub = rospy.Publisher('/turtle1/cmd_vel', Twist, queue_size=10)

def command_thread():
```

```
while True:

    if pubCommand:

        vel_msg = Twist()

        vel_msg.linear.x = 0.5

        vel_msg.angular.z = 0.2

        turtle_vel_pub.publish(vel_msg)

    time.sleep(0.1)

def commandCallback(req):

    global pubCommand

    pubCommand = bool(1-pubCommand)

    rospy.loginfo("Publish turtle velocity command![%d]", pubCommand)

    return TriggerResponse(1, "Change turtle command state!")

def turtle_command_server():

    rospy.init_node('turtle_command_server')

    s = rospy.Service('/turtle_command', Trigger, commandCallback)

    print "Ready to receive turtle command."

    thread.start_new_thread(command_thread, ())

    rospy.spin()

if __name__ == "__main__":

    turtle_command_server()
```

```

Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts
File Edit View Terminal Tabs Help
8 turtle_vel_pub = rospy.Publisher('/turtle1/cmd_vel', Twist, queue_size=10)
9 def command_thread():
10     while True:
11         if pubCommand:
12             vel_msg = Twist()
13             vel_msg.linear.x = 0.5
14             vel_msg.angular.z = 0.2
15             turtle_vel_pub.publish(vel_msg)
16
17             time.sleep(0.1)
18 def commandCallback(req):
19     global pubCommand
20     pubCommand = bool(1-pubCommand)
21     rospy.loginfo("Publish turtle velocity command![%d]", pubCommand)
22     return TriggerResponse(1, "Change turtle command state!")
23 def turtle_command_server():
24     rospy.init_node('turtle_command_server')
25     s = rospy.Service('/turtle_command', Trigger, commandCallback)
26     print "Ready to receive turtle command."
27     thread.start_new_thread(command_thread, ())
28     rospy.spin()
29 if __name__ == "__main__":
30     turtle_command_server()
:wq

```

- 3) Enter “**chmod +x turtle\_command\_server.py**” command and press “Enter” to give the executable permission to the saved turtle\_command\_server.py.

```

Terminal - ubuntu@ubuntu: ~/catkin_ws/src/beginner_hiwonder/scripts
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws/src/beginner_hiwonder/scripts$ chmod +x turtle_command
_server.py

```

## 2. Run Server Node

- 1) Enter “roscore” command to start node manager.

```

ubuntu@ubuntu:~/catkin_ws$ roscore

```

- 2) Enter “roslaunch turtlesim turtlesim\_node” command and press “Enter” to start turtlesim simulator window.

```

Terminal - ubuntu@ubuntu: ~/catkin_ws
File Edit View Terminal Tabs Help
ubuntu@ubuntu:~/catkin_ws$ roslaunch turtlesim turtlesim_node

```

- 3) Open a new terminal. Then enter “roslaunch beginner\_hiwonder turtle\_command\_server.py” and press “Enter” to run service node. If need

to stop running the node, you can press “Ctrl+C”.

```
Terminal - ubuntu@ubuntu: ~  
File Edit View Terminal Tabs Help  
ubuntu@ubuntu:~$ rosrn beginner_hiwonder turtle_command_server.py
```

- 4) Open a new terminal again. Enter “rosservice call /turtle\_command “{}”” command and press “Enter” to move the turtle along the circular path.

```
Terminal - ubuntu@ubuntu: ~  
File Edit View Terminal Tabs Help  
ubuntu@ubuntu:~$ rosservice call /turtle_command "{}"  
success: True  
message: "Change turtle command state!"
```

