جامعـــة خليفـــة
Khalifa University

Communicate with ROS Services

JUNE-2021

Communicate with ROS Services

1. What is a Service?

2. Create a Python Service Server

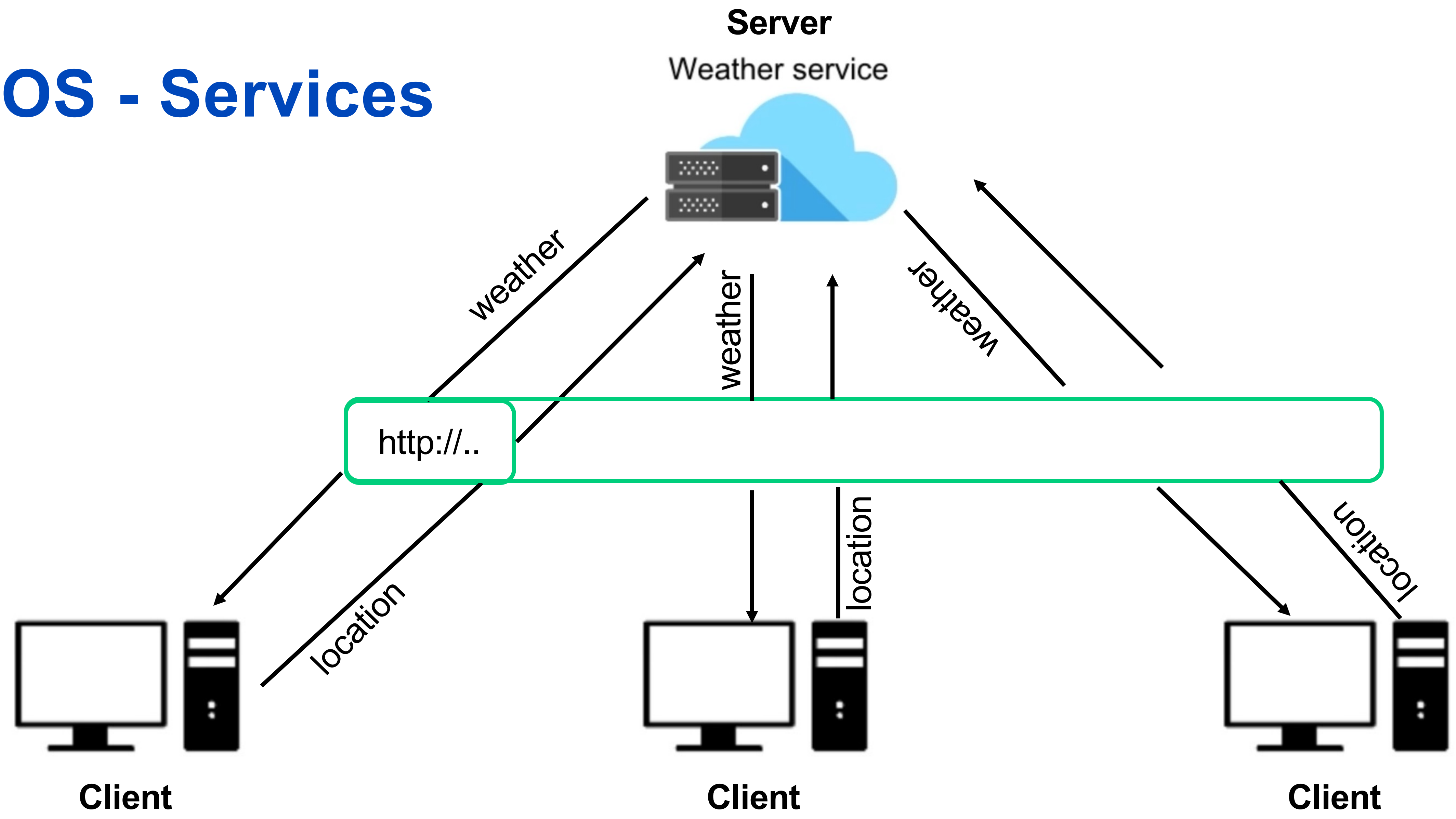
3. Create a Python Service Client

4. Debug Service with Command Line Tools

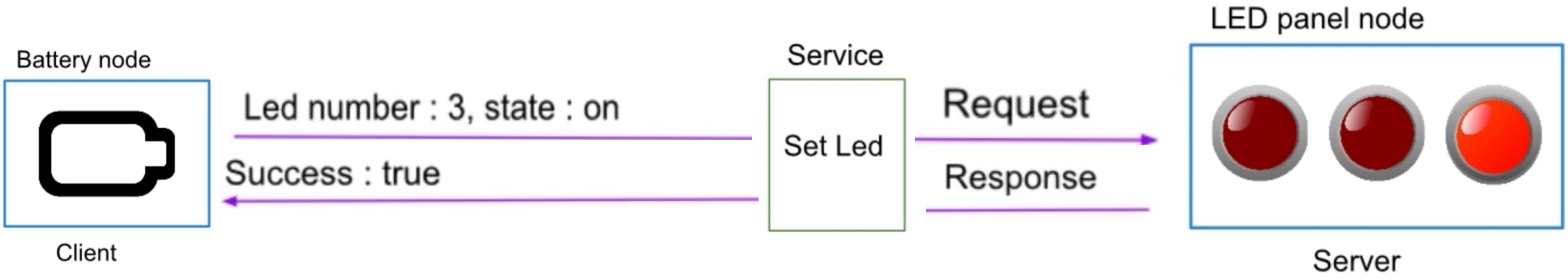
5. Experiments on Services with Turtlesim

What is ROS Services

ROS - Services



ROS - Services

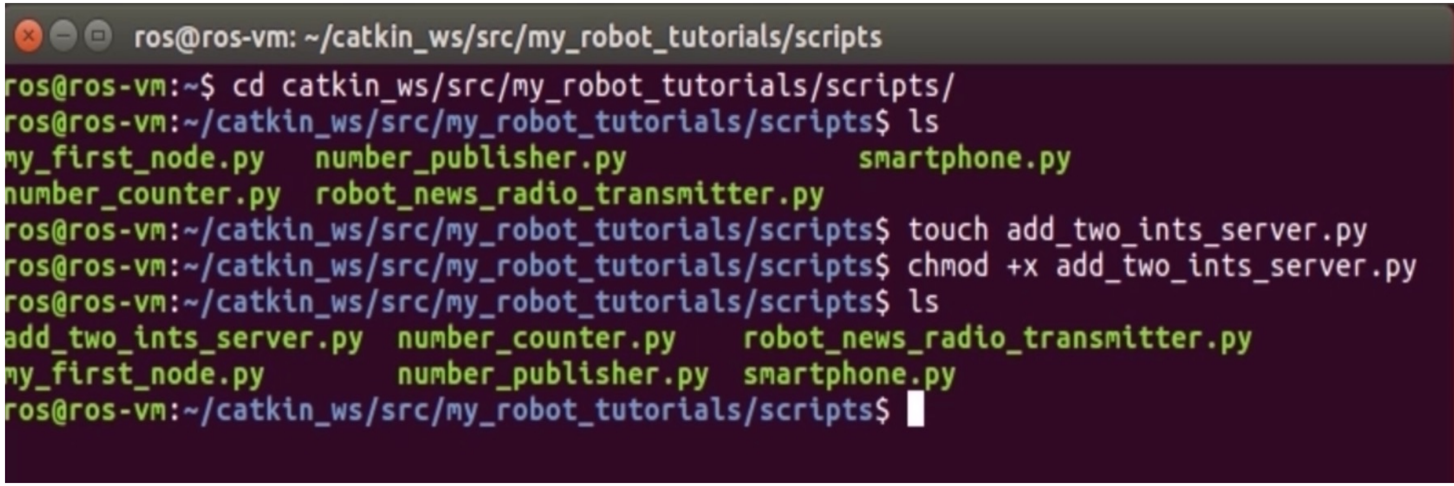


ROS - Services

- A ROS service is a client/server system
- Synchronous
- For computations and quick actions
- One message type for Request, one message type for Response
- Can be written in Python, C++,.. Directly inside ROS nodes
- A service server can only exist once, but can have many clients

Create a Python Service Server

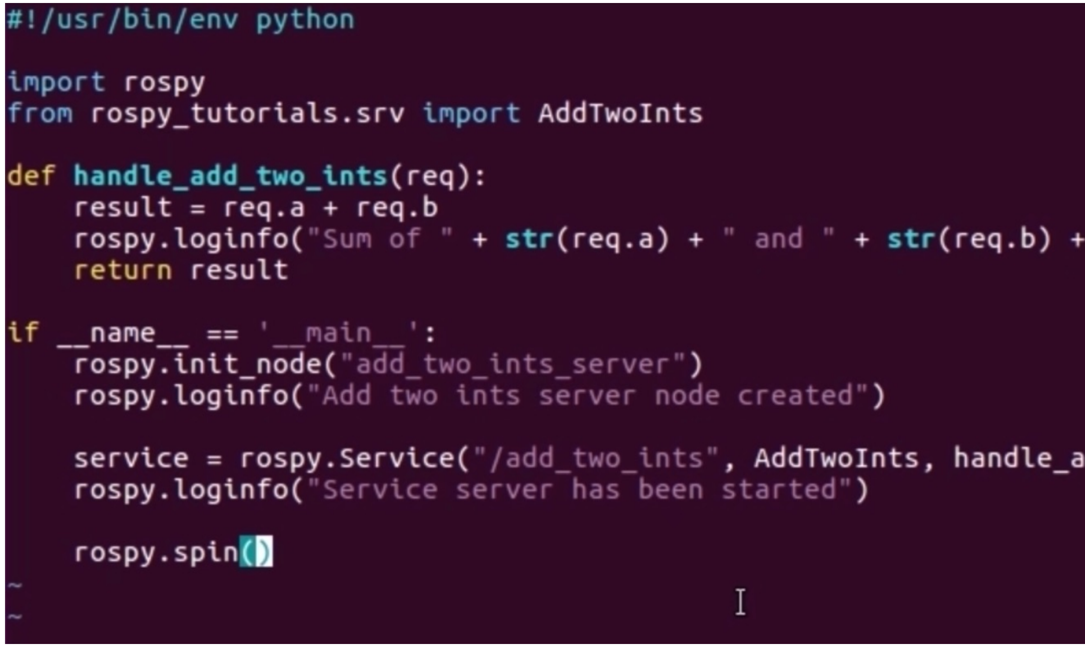
```
$ cd catkin_ws/src/my_robot_tutorials/scripts/
$ ls
$ touch add_two_ints_server.py
$ chmod +x add_two_ints_server.py
$ gedit add_two_ints_server.py
```



```
ros@ros-vm:~$ cd catkin_ws/src/my_robot_tutorials/scripts/
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ ls
my_first_node.py   number_publisher.py              smartphone.py
number_counter.py  robot_news_radio_transmitter.py
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ touch add_two_ints_server.py
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ chmod +x add_two_ints_server.py
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ ls
add_two_ints_server.py  number_counter.py    robot_news_radio_transmitter.py
my_first_node.py        number_publisher.py  smartphone.py
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$
```

Now edit the file

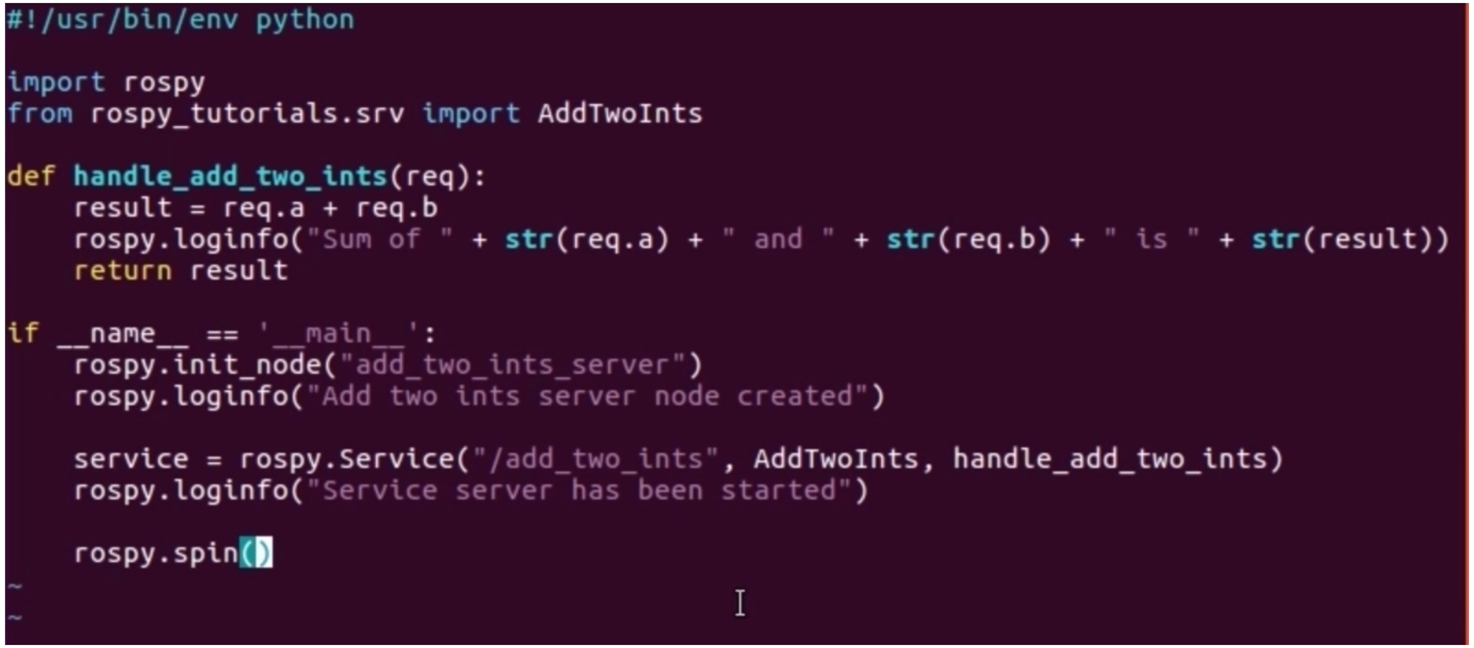
```
#!/usr/bin/env python3
Import rospy
From rospy_tutorials.srv import AddTwoInts
If __name__ ==‘__main__’:
        rospy.init_node(“add_two_ints_server”)
        rospy.loginfo(“Add two ints server node created”)
        service = rospy.Service(“/add_two_ints”, AddTwoInts)
```



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

import rospy
from rospy_tutorials.srv import AddTwoInts

def handle_add_two_ints(req):
    result = req.a + req.b
    rospy.loginfo("Sum of " + str(req.a) + " and " + str(req.b) +
    return result

if __name__ == '__main__':
    rospy.init_node("add_two_ints_server")
    rospy.loginfo("Add two ints server node created")

    service = rospy.Service("/add_two_ints", AddTwoInts, handle_a
    rospy.loginfo("Service server has been started")

    rospy.spin()
~
~
```

Now edit the file

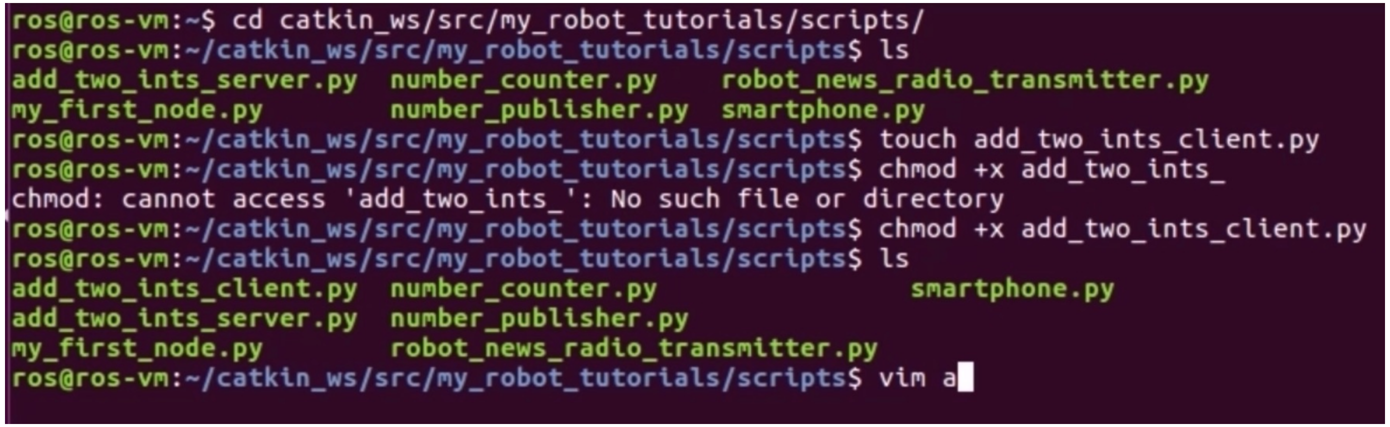
```
#!/usr/bin/env python3
Import rospy
from rospy_tutorials.srv import AddTwoInts
If __name__ ==‘__main__’:
        rospy.init_node(“add_two_ints_server”)
        rospy.loginfo(“Add two ints server node created”)
        service = rospy.Service(“/add_two_ints”, AddTwoInts)
```



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

import rospy
from rospy_tutorials.srv import AddTwoInts

def handle_add_two_ints(req):
    result = req.a + req.b
    rospy.loginfo("Sum of " + str(req.a) + " and " + str(req.b) + " is " + str(result))
    return result

if __name__ == '__main__':
    rospy.init_node("add_two_ints_server")
    rospy.loginfo("Add two ints server node created")

    service = rospy.Service("/add_two_ints", AddTwoInts, handle_add_two_ints)
    rospy.loginfo("Service server has been started")

    rospy.spin()
~
~
```

Create a Python Service Client

```
$ cd catkin_ws/src/my_robot_tutorials/scripts/
$ ls
$ touch add_two_ints_client.py
$ chmod +x add_two_ints_client.py
$ gedit add_two_ints_client.py
```



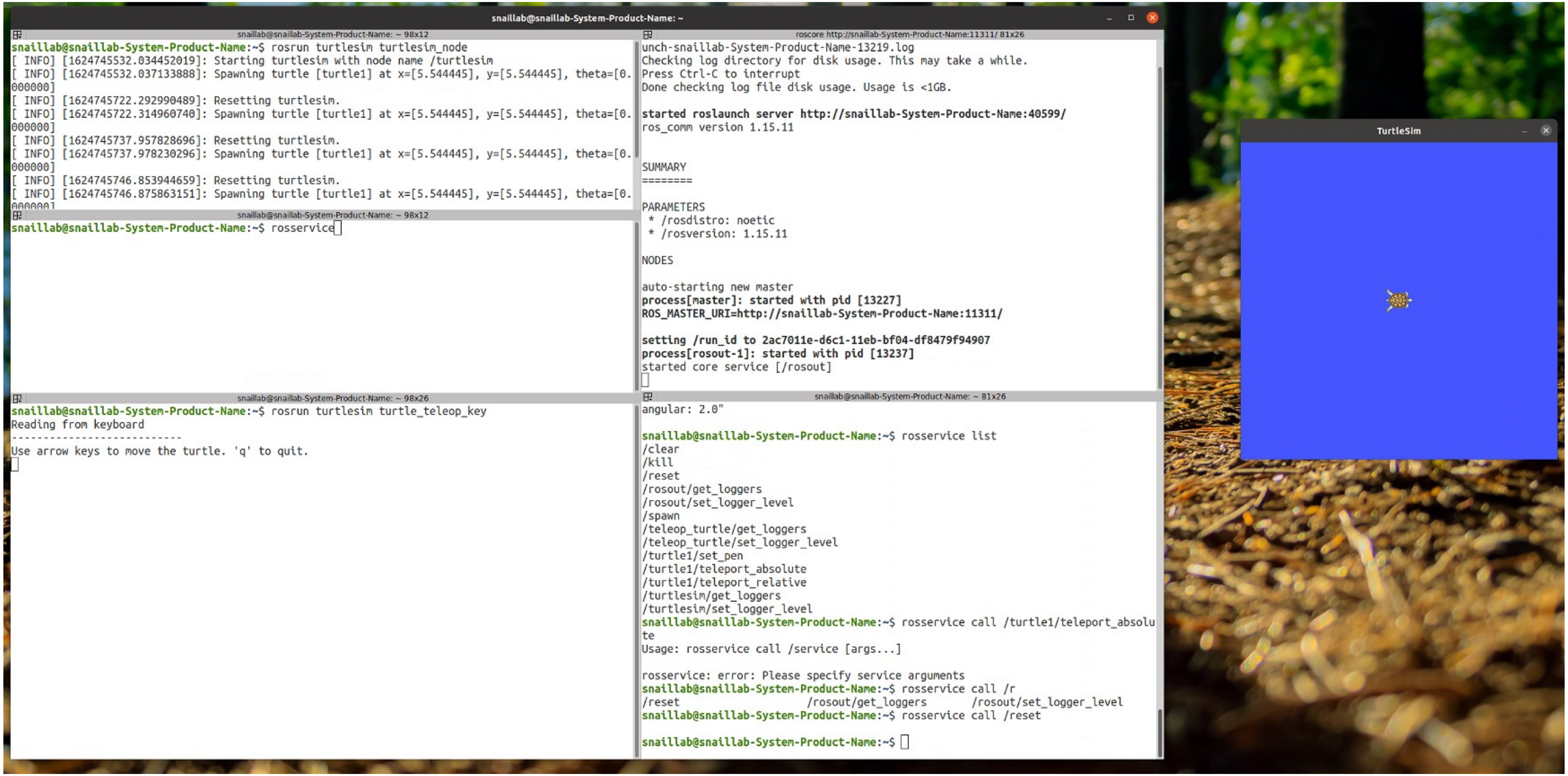
```
ros@ros-vm:~$ cd catkin_ws/src/my_robot_tutorials/scripts/
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ ls
add_two_ints_server.py  number_counter.py    robot_news_radio_transmitter.py
my_first_node.py        number_publisher.py  smartphone.py
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ touch add_two_ints_client.py
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ chmod +x add_two_ints_
chmod: cannot access 'add_two_ints_': No such file or directory
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ chmod +x add_two_ints_client.py
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ ls
add_two_ints_client.py  number_counter.py                smartphone.py
add_two_ints_server.py  number_publisher.py
my_first_node.py        robot_news_radio_transmitter.py
ros@ros-vm:~/catkin_ws/src/my_robot_tutorials/scripts$ vim a
```

Now edit the file

```
#!/usr/bin/env python3
Import rospy
From rospy_tutorials.srv import AddTwoInts
If __name__ =='__main__':
            rospy.init_node("add_two_ints_server")
            rospy.loginfo("Add two ints server node created")
            service = rospy.Service("/add_two_ints", AddTwoInts)
```

Debug Services with Command Line Tools

Experiment on Services with Turtlesim


snaillab@snaillab-System-Product-Name: ~
snaillab@snaillab-System-Product-Name: ~ 98x12
roscore http://snaillab-System-Product-Name:11311/ 81x26
snaillab@snaillab-System-Product-Name: ~ 98x12
snaillab@snaillab-System-Product-Name: ~ 98x26
snaillab@snaillab-System-Product-Name: ~ 81x26
TurtleSim

```
snaillab@snaillab-System-Product-Name:~$ rosrun turtlesim turtlesim_node
[ INFO] [1624745532.034452019]: Starting turtlesim with node name /turtlesim
[ INFO] [1624745532.037133888]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.
000000]
[ INFO] [1624745722.292990489]: Resetting turtlesim.
[ INFO] [1624745722.314960740]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.
000000]
[ INFO] [1624745737.957828696]: Resetting turtlesim.
[ INFO] [1624745737.978230296]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.
000000]
[ INFO] [1624745746.853944659]: Resetting turtlesim.
[ INFO] [1624745746.875863151]: Spawning turtle [turtle1] at x=[5.544445], y=[5.544445], theta=[0.
000000]
```

```
snaillab@snaillab-System-Product-Name:~$ rosservice
```

```
snaillab@snaillab-System-Product-Name:~$ rosrun turtlesim turtle_teleop_key
Reading from keyboard
---------------------------
Use arrow keys to move the turtle. 'q' to quit.
```

```
unch-snaillab-System-Product-Name-13219.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://snaillab-System-Product-Name:40599/
ros_comm version 1.15.11


SUMMARY
========

PARAMETERS
 * /rosdistro: noetic
 * /rosversion: 1.15.11

NODES

auto-starting new master
process[master]: started with pid [13227]
ROS_MASTER_URI=http://snaillab-System-Product-Name:11311/

setting /run_id to 2ac7011e-d6c1-11eb-bf04-df8479f94907
process[rosout-1]: started with pid [13237]
started core service [/rosout]
```

```
angular: 2.0"

snaillab@snaillab-System-Product-Name:~$ rosservice list
/clear
/kill
/reset
/rosout/get_loggers
/rosout/set_logger_level
/spawn
/teleop_turtle/get_loggers
/teleop_turtle/set_logger_level
/turtle1/set_pen
/turtle1/teleport_absolute
/turtle1/teleport_relative
/turtlesim/get_loggers
/turtlesim/set_logger_level
snaillab@snaillab-System-Product-Name:~$ rosservice call /turtle1/teleport_absolu
te
Usage: rosservice call /service [args...]

rosservice: error: Please specify service arguments
snaillab@snaillab-System-Product-Name:~$ rosservice call /r
/reset                    /rosout/get_loggers       /rosout/set_logger_level
snaillab@snaillab-System-Product-Name:~$ rosservice call /reset

snaillab@snaillab-System-Product-Name:~$
```

Experiment on Services with Turtlesim