

無人載具技術與應用

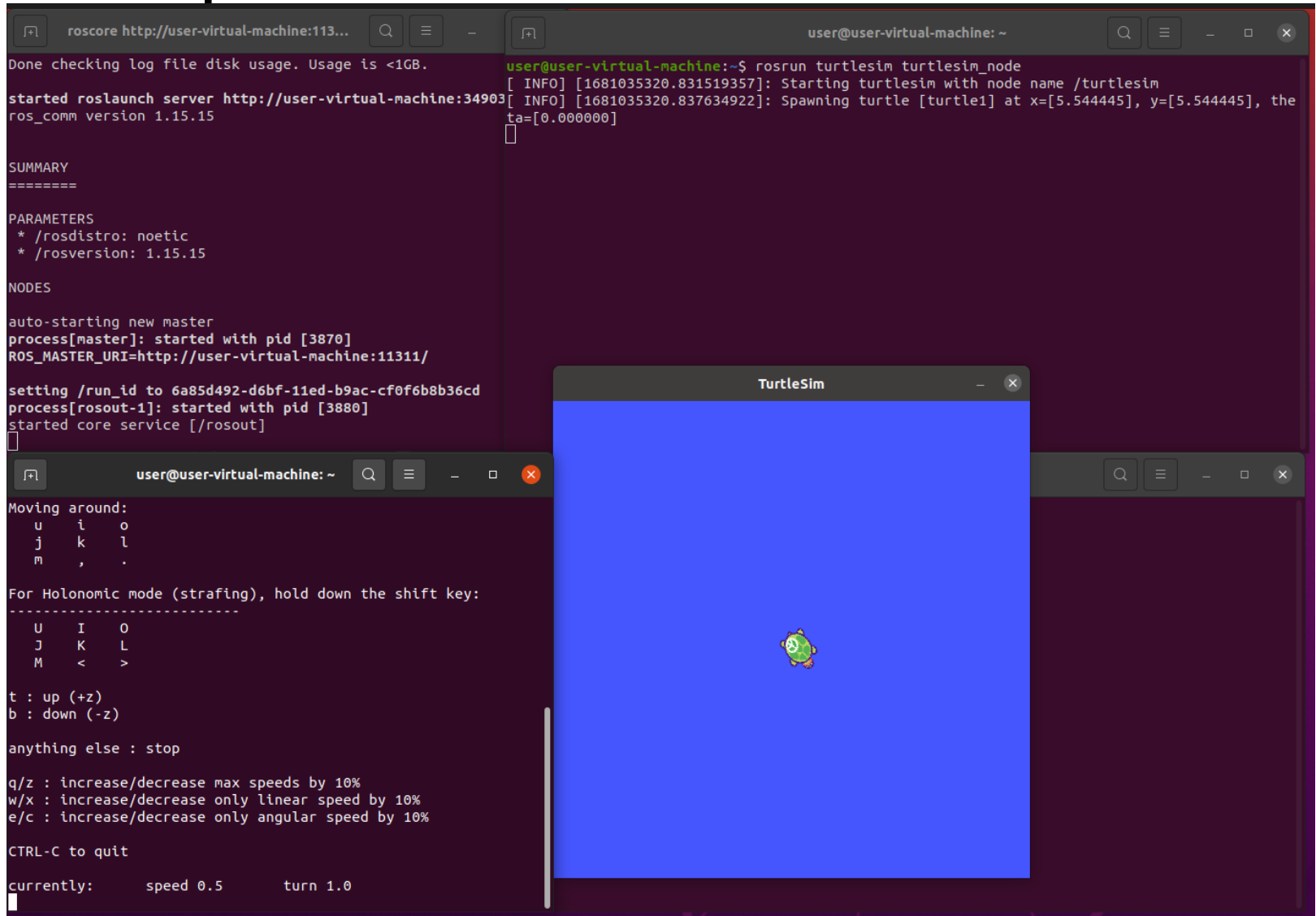
ROS

徐瑋隆

wlhsu304@gmail.com

ROS PARAMETER

ros parameter – turtlesim 01

The image shows a ROS environment with two terminal windows and a TurtleSim window. The top-left terminal window shows the output of 'roslaunch server http://user-virtual-machine:34903', including a summary of parameters (noetic, 1.15.15) and nodes (auto-starting new master, process[rosout-1]). The bottom-left terminal window shows the output of 'roslaunch turtlesim turtlesim_node', including a summary of parameters (noetic, 1.15.15) and nodes (auto-starting new master, process[turtlesim]). The right window is the TurtleSim GUI, which displays a blue square environment with a small green turtle icon in the center. The terminal windows are titled 'roscore http://user-virtual-machine:113...' and 'user@user-virtual-machine: ~'. The TurtleSim window is titled 'TurtleSim'.

roscore

roslaunch turtlesim turtlesim_node

roslaunch teleop_twist_keyboard teleop_twist_keyboard.py /cmd_vel:=/turtle1/cmd_vel

ros parameter – turtlesim 02

複習rosservice

rosservice list

rosservice call /spawn 1 2 0 kk

rosparam list

rosparam get /turtlesim/background_r

rosparam set /turtlesim/background_r 150

rosparam get /turtlesim/background_r

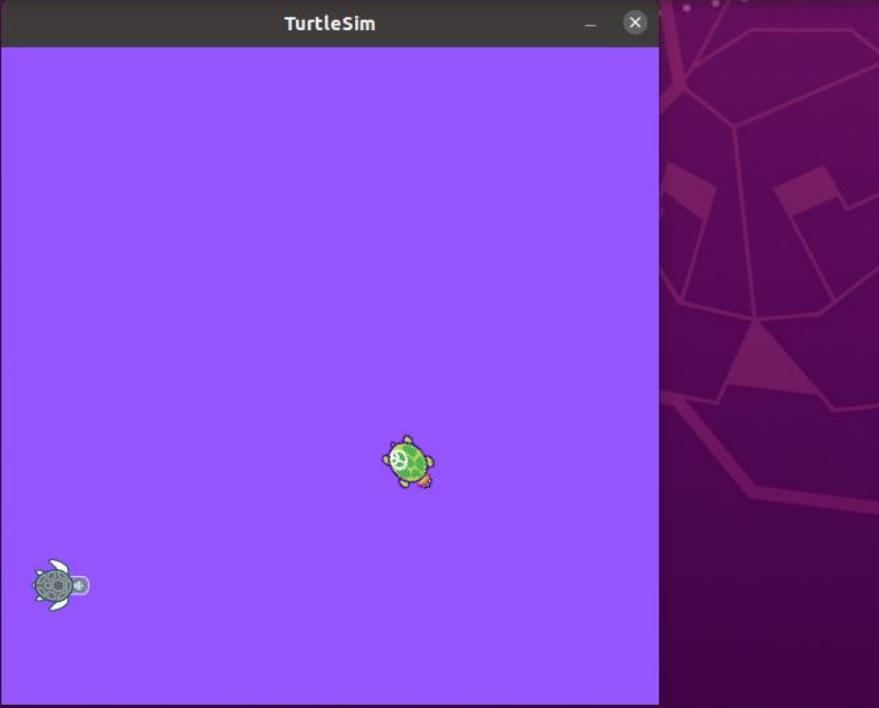
rosservice call /clear

清潔背景(強制程式讀取取ros param改變才會生效)

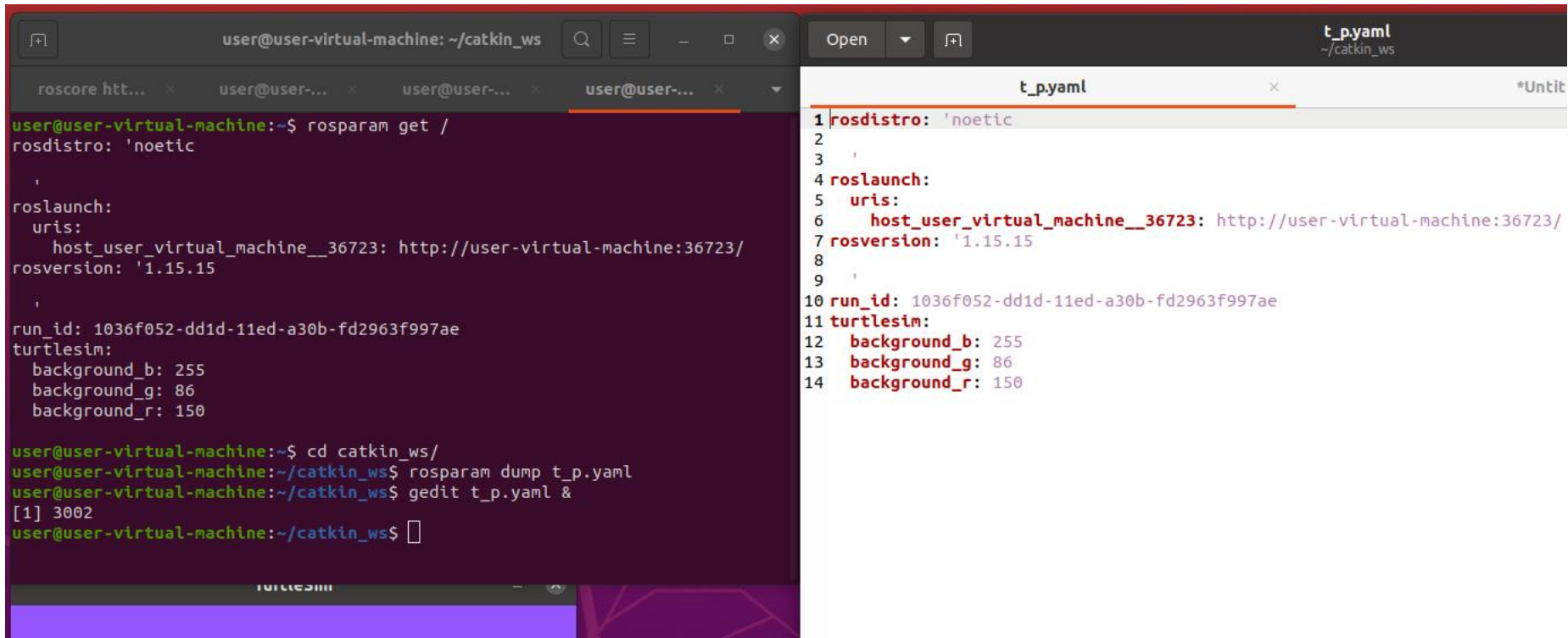
rosparam set /turtlesim/background_r 50

rosservice call /clear

```
roscore htt... x user@user-... x user@user-... x user@user-... x
user@user-virtual-machine:~$ rosparam list
/rosdistro
/roslaunch/uris/host_user_virtual_machine__36723
/rosversion
/run_id
/turtlesim/background_b
/turtlesim/background_g
/turtlesim/background_r
user@user-virtual-machine:~$ rosparam get /turtlesim/background_r
69
user@user-virtual-machine:~$ rosparam set /turtlesim/background_r 150
user@user-virtual-machine:~$ rosparam get /turtlesim/background_r
150
user@user-virtual-machine:~$ rosservice call /clear
user@user-virtual-machine:~$
```

The image shows a terminal window with ROS commands and their outputs, and a separate window titled 'TurtleSim'. The terminal window shows the following commands and outputs: 'rosparam list' lists system parameters; 'rosparam get /turtlesim/background_r' returns 69; 'rosparam set /turtlesim/background_r 150' sets the background red value to 150; 'rosparam get /turtlesim/background_r' returns 150; 'rosservice call /clear' is executed. The TurtleSim window shows a purple background, indicating the red value is set to 150 (which is outside the 0-255 range, so it wraps or is clamped). Two turtles are visible: a green one in the center and a grey one in the bottom left corner.

ros parameter – turtlesim 03



The image shows a terminal window on the left and a text editor on the right. The terminal window displays the output of the `rosparam get /` command, showing the current ROS configuration. The text editor shows the contents of the `t_p.yaml` file, which defines the ROS environment for the turtlesim simulation.

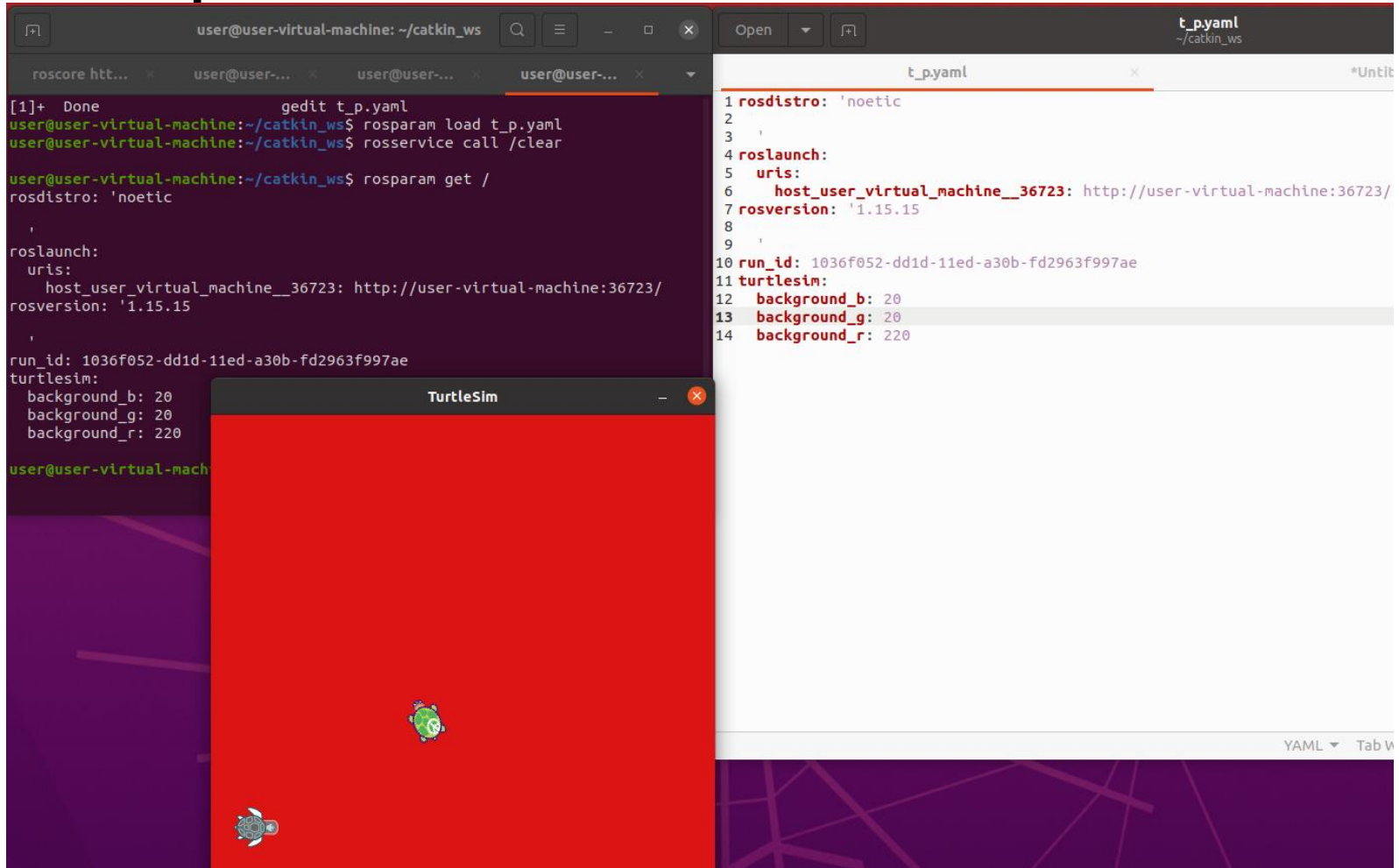
```
user@user-virtual-machine: ~/catkin_ws
roscore htt... x user@user-... x user@user-... x user@user-... x
user@user-virtual-machine:~$ rosparam get /
rostdistro: 'noetic'
'
'
roslaunch:
  uris:
    host_user_virtual_machine__36723: http://user-virtual-machine:36723/
  rosversion: '1.15.15'
'
run_id: 1036f052-dd1d-11ed-a30b-fd2963f997ae
turtlesim:
  background_b: 255
  background_g: 86
  background_r: 150

user@user-virtual-machine:~$ cd catkin_ws/
user@user-virtual-machine:~/catkin_ws$ rosparam dump t_p.yaml
user@user-virtual-machine:~/catkin_ws$ gedit t_p.yaml &
[1] 3002
user@user-virtual-machine:~/catkin_ws$
```

```
t_p.yaml
1 rostdistro: 'noetic'
2
3 '
4 roslaunch:
5   uris:
6     host_user_virtual_machine__36723: http://user-virtual-machine:36723/
7   rosversion: '1.15.15'
8
9 '
10 run_id: 1036f052-dd1d-11ed-a30b-fd2963f997ae
11 turtlesim:
12   background_b: 255
13   background_g: 86
14   background_r: 150
```

```
rosparam get /
cd catkin_ws/
rosparam dump t_p.yaml
gedit t_p.yaml &
```

ros parameter – turtlesim 04



The screenshot displays a ROS Noetic environment. On the left, a terminal window shows the following commands and output:

```
[1]+ Done gedit t_p.yaml
user@user-virtual-machine:~/catkin_ws$ rosparam load t_p.yaml
user@user-virtual-machine:~/catkin_ws$ rosservice call /clear

user@user-virtual-machine:~/catkin_ws$ rosparam get /
rostdistro: 'noetic'

'
roslaunch:
  uris:
    host_user_virtual_machine__36723: http://user-virtual-machine:36723/
rosversion: '1.15.15'

'
run_id: 1036f052-dd1d-11ed-a30b-fd2963f997ae
turtlesim:
  background_b: 20
  background_g: 20
  background_r: 220
user@user-virtual-machine:~/catkin_ws$
```

On the right, a text editor shows the content of `t_p.yaml`:

```
1 rostdistro: 'noetic'
2
3 '
4 roslaunch:
5   uris:
6     host_user_virtual_machine__36723: http://user-virtual-machine:36723/
7 rosversion: '1.15.15'
8
9 '
10 run_id: 1036f052-dd1d-11ed-a30b-fd2963f997ae
11 turtlesim:
12   background_b: 20
13   background_g: 20
14   background_r: 220
```

In the foreground, the `TurtleSim` window is visible, showing a red background with two turtles.

`rosparam load`

`~/home/user/catkin_ws/t_p.yaml`

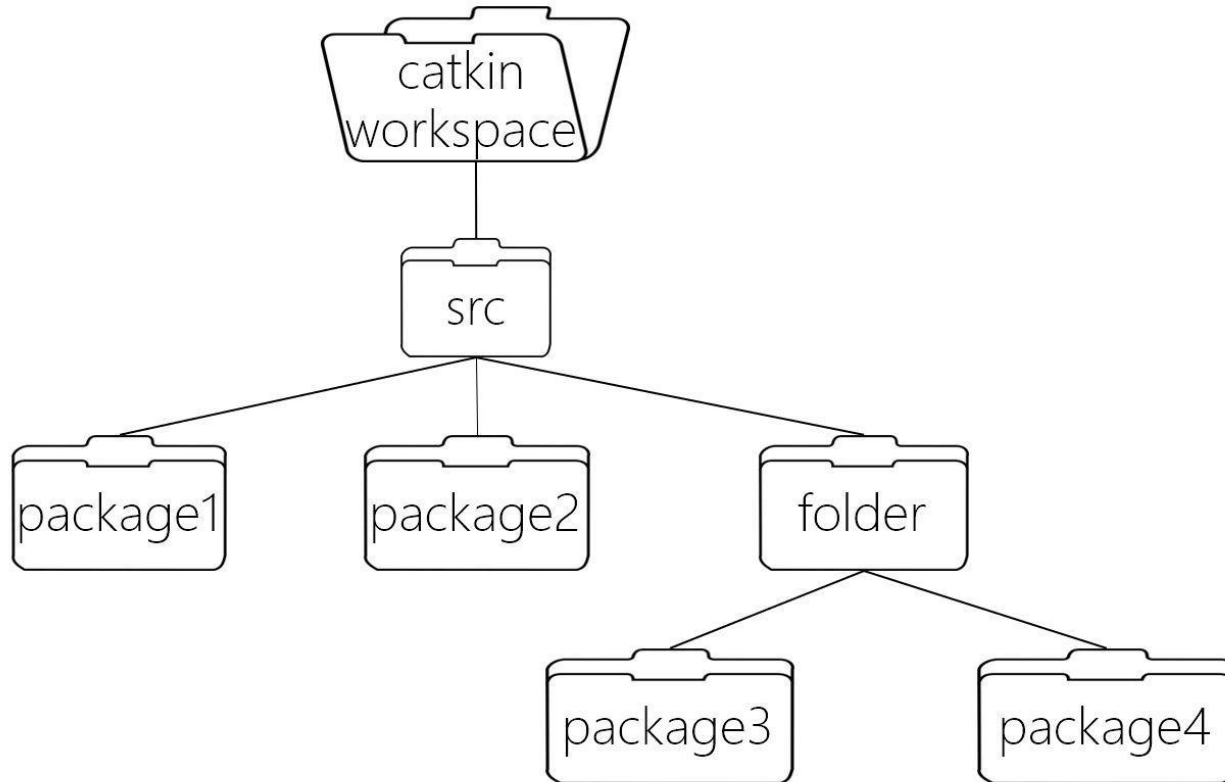
`rosservice call /clear`

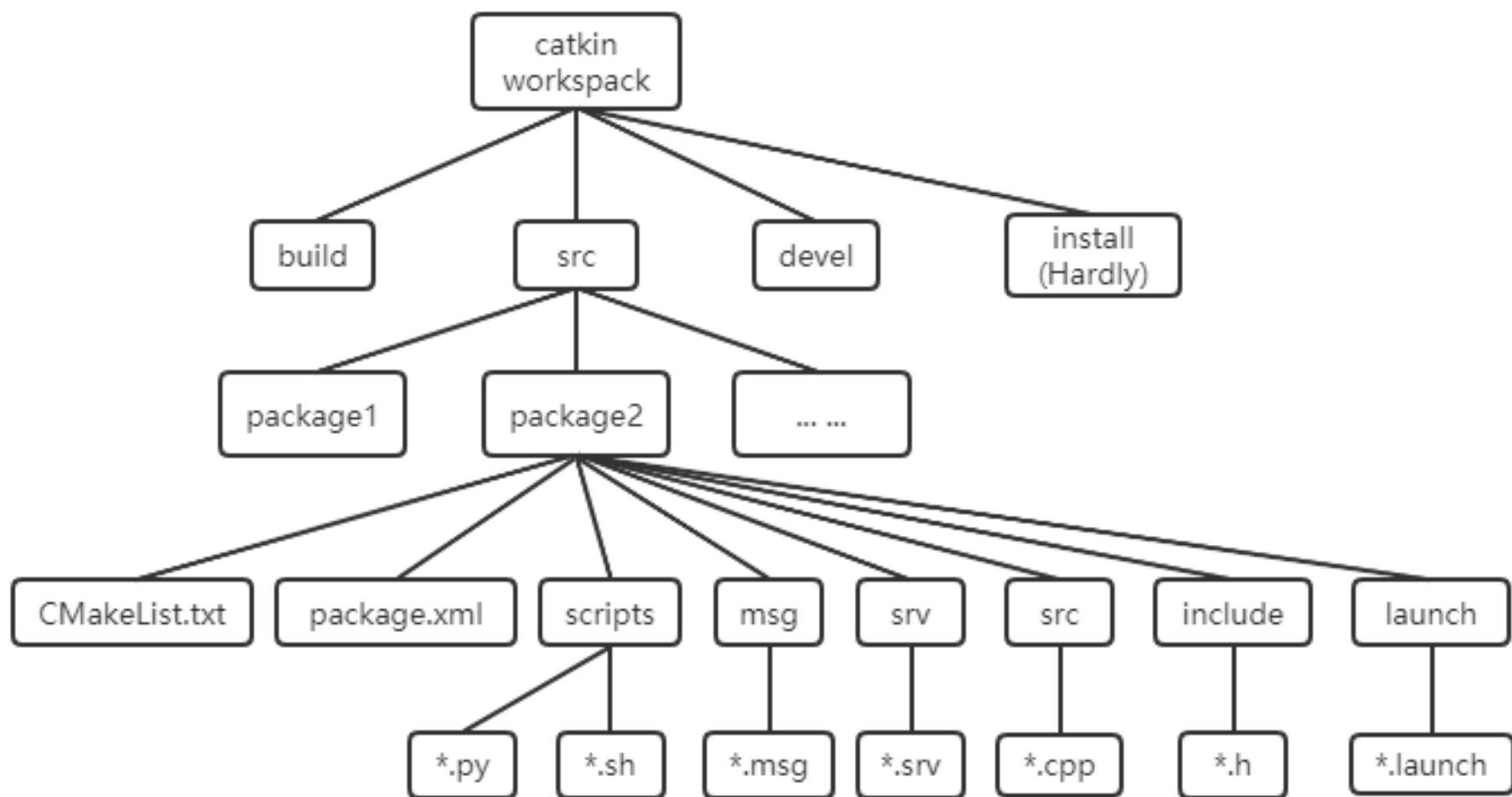
`rosparam get /`

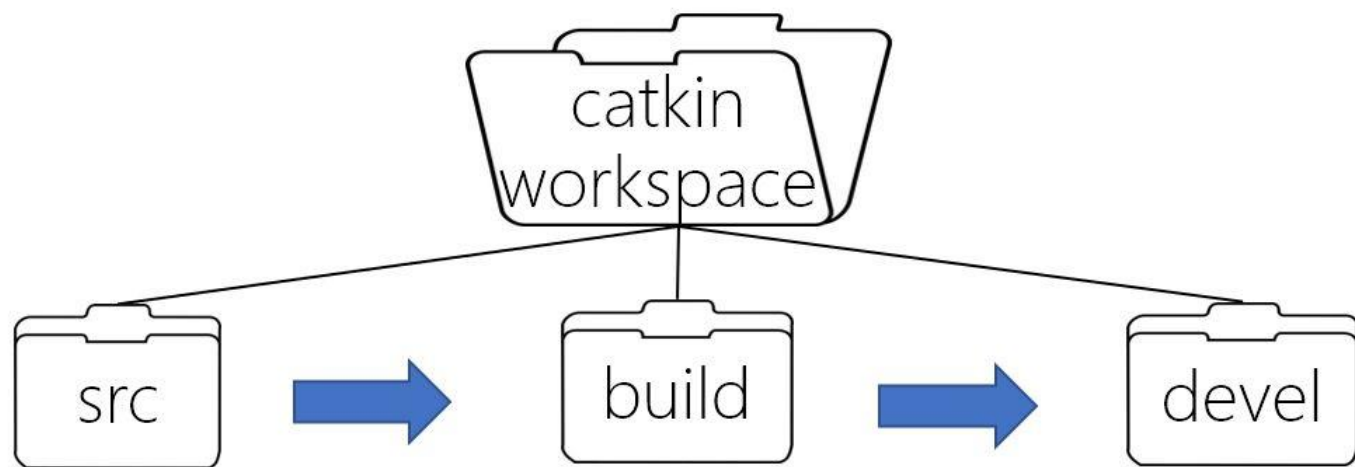
`rosparam get /turtlesim/background_r`

ros node coding

ROS PARAMETER







package源代码包

cmake&catkin缓存和中间文件

目标文件

catkin_ws
(ROS package)

catkin



CMakeLists.txt

cmake



Makefile

make



hello.cpp

gcc/g++



hello.o

gcc/g++



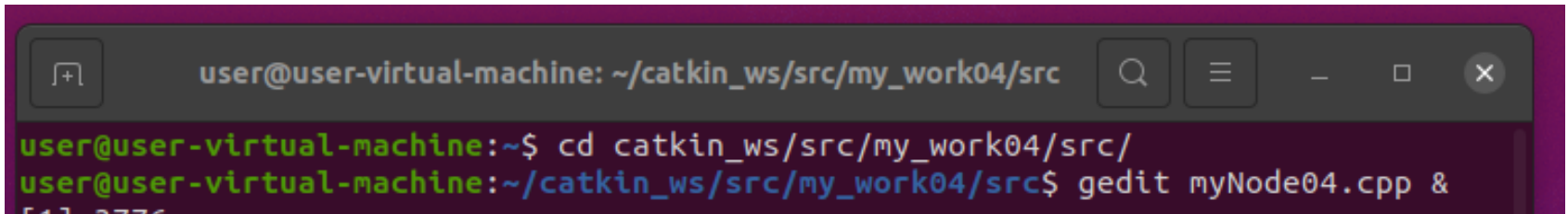
hello

ros node 01

```
user@user-virtual-machine:~$ cd catkin_ws/src/  
user@user-virtual-machine:~/catkin_ws/src$ catkin_create_pkg my_work03 roscpp ro  
spy std_msgs  
Created file my_work03/package.xml  
Created file my_work03/CMakeLists.txt  
Created folder my_work03/include/my_work03  
Created folder my_work03/src  
Successfully created files in /home/user/catkin_ws/src/my_work03. Please adjust  
the values in package.xml.
```

```
cd catkin_ws/src/  
catkin_create_pkg my_work04 roscpp rospy std_msgs  
ls
```

ros node 02

A terminal window with a dark purple background. The title bar shows the user is at a virtual machine, in the directory ~/catkin_ws/src/my_work04/src. The terminal shows two commands: first, 'cd catkin_ws/src/my_work04/src/' which changes the directory, and second, 'gedit myNode04.cpp &' which opens a file editor. The prompt for the second command is partially cut off on the right.

```
user@user-virtual-machine: ~/catkin_ws/src/my_work04/src
user@user-virtual-machine:~$ cd catkin_ws/src/my_work04/src/
user@user-virtual-machine:~/catkin_ws/src/my_work04/src$ gedit myNode04.cpp &
```

```
#include "ros/ros.h"
```

```
int main(int argc, char **argv)
```

```
{
```

```
  ros::init(argc, argv, "myNode04");
```

```
  ros::NodeHandle n;
```

```
  double dPeriod = 3;
```

```
  int iCnt = 0;
```

```
  while (ros::ok())
```

```
{
```

```
  ROS_INFO("myNode04: hi :%d",iCnt++);
```

```
  ros::spinOnce();
```

```
  ros::Duration(dPeriod).sleep();
```

```
}
```

```
  return 0;
```

```
}
```

ros node 03

```
user@user-virtual-machine:~/catkin_ws/src/my_work03/src$ cd ..  
user@user-virtual-machine:~/catkin_ws/src/my_work03$ gedit CMakeLists.txt &  
[2] 24554  
user@user-virtual-machine:~/catkin_ws/src/my_work03$
```

```
113 ## Build ##  
114 #####  
115  
116 ## Specify additional locations of header files  
117 ## Your package locations should be listed before other locations  
118 include_directories(  
119 # include  
120 ${catkin_INCLUDE_DIRS}  
121 )  
122  
123 add_executable(myNode01 src/myNode01.cpp)  
124 target_link_libraries(myNode01 ${catkin_LIBRARIES})  
125
```

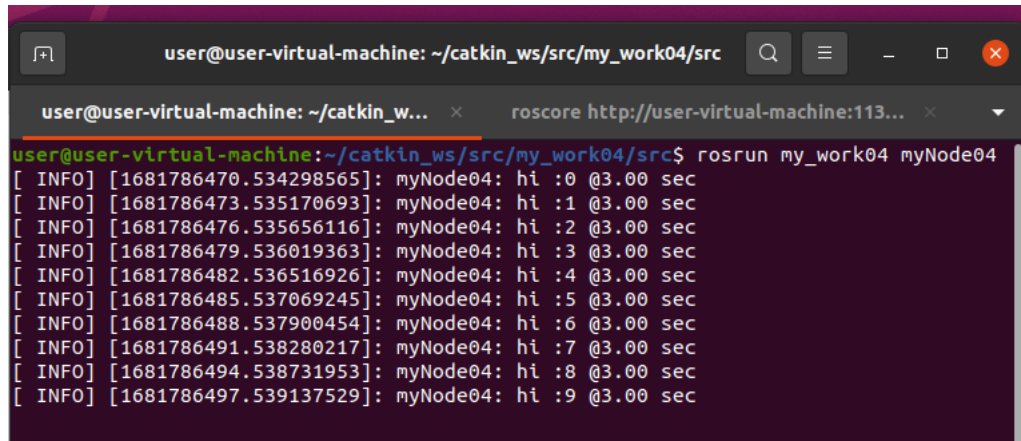
```
add_executable(myNode04 src/myNode04.cpp)  
target_link_libraries(myNode04 ${catkin_LIBRARIES})
```

```
user@user-virtual-machine:~/catkin_ws/src/my_work02/src$ cd ../../..  
user@user-virtual-machine:~/catkin_ws$ catkin_make
```

```
cd ~/catkin_ws  
catkin_make
```

編譯程式

ros node 04

A terminal window with a dark purple background. The title bar shows 'user@user-virtual-machine: ~/catkin_ws/src/my_work04/src'. The terminal content shows the command 'rosrun my_work04 myNode04' being executed, followed by ten lines of log output. Each line starts with '[INFO]' and contains a timestamp, a node name 'myNode04', and a message 'hi' followed by a counter from 0 to 9 and a 3-second interval. The window has standard Linux window controls and a search icon.

```
user@user-virtual-machine: ~/catkin_ws/src/my_work04/src
user@user-virtual-machine: ~/catkin_w... x  roscore http://user-virtual-machine:113... x
user@user-virtual-machine:~/catkin_ws/src/my_work04/src$ rosrun my_work04 myNode04
[ INFO] [1681786470.534298565]: myNode04: hi :0 @3.00 sec
[ INFO] [1681786473.535170693]: myNode04: hi :1 @3.00 sec
[ INFO] [1681786476.535656116]: myNode04: hi :2 @3.00 sec
[ INFO] [1681786479.536019363]: myNode04: hi :3 @3.00 sec
[ INFO] [1681786482.536516926]: myNode04: hi :4 @3.00 sec
[ INFO] [1681786485.537069245]: myNode04: hi :5 @3.00 sec
[ INFO] [1681786488.537900454]: myNode04: hi :6 @3.00 sec
[ INFO] [1681786491.538280217]: myNode04: hi :7 @3.00 sec
[ INFO] [1681786494.538731953]: myNode04: hi :8 @3.00 sec
[ INFO] [1681786497.539137529]: myNode04: hi :9 @3.00 sec
```

roscore

rosrun my_work04 myNode04

ros node coding

ROS PARAMETER

ros parameter 01

https://github.com/Waywrong/ros_course/

```
user@user-virtual-machine: ~/catkin_ws/src/my_work04/src
user@user-virtual-machine:~$ cd catkin_ws/src/my_work04/src/
user@user-virtual-machine:~/catkin_ws/src/my_work04/src$ gedit myNode04.cpp &
```

```
myNode04.cpp
~/catkin_ws/src/my_work04/src

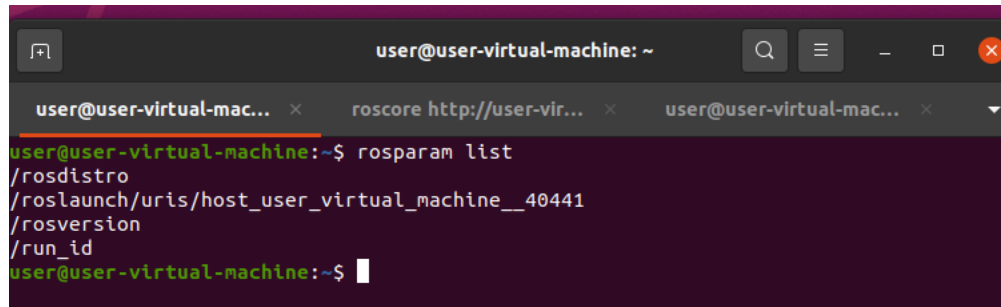
1 #include "ros/ros.h"
2
3 int main(int argc, char **argv)
4 {
5     ros::init(argc, argv, "myNode04");
6     ros::NodeHandle n;
7
8     double dPeriod = 3;
9     n.getParam("/para_Period", dPeriod);
10    int iCnt = 0;
11
12    while (ros::ok())
13    {
14        ROS_INFO("myNode04: hi :%d @%4.2f sec", iCnt++, dPeriod);
15        ros::spinOnce();
16        ros::Duration(dPeriod).sleep();
17    }
18    return 0;
19 }
```

```
^Cuser@user-virtual-machine:~/catkin_ws$ catkin_make
Base path: /home/user/catkin_ws
Source space: /home/user/catkin_ws/src
Build space: /home/user/catkin_ws/build
Devel space: /home/user/catkin_ws/devel
Install space: /home/user/catkin_ws/install
####
#### Running command: "make cmake_check_build_system" in "/home/user/catkin_ws/build"
```

ros parameter 02

roscore

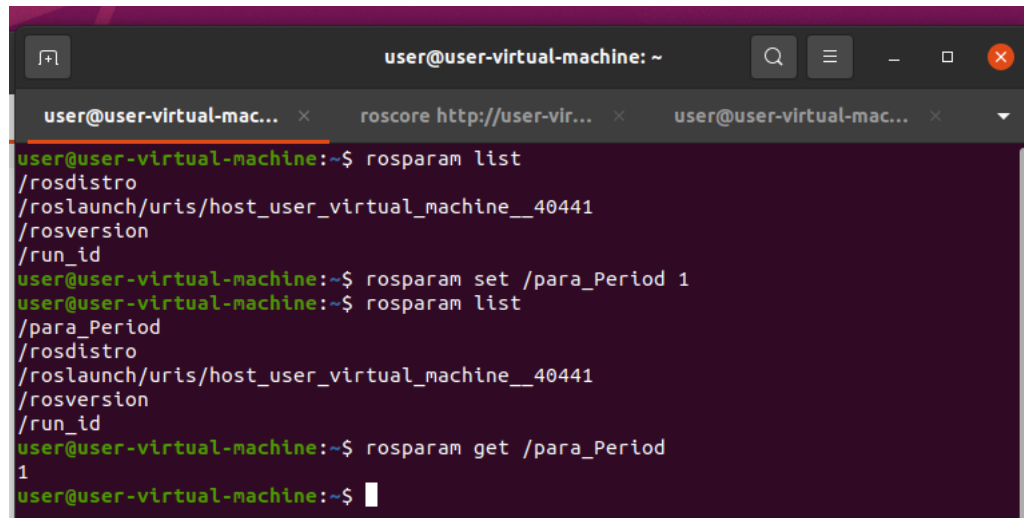
rosparam list

A terminal window titled 'user@user-virtual-machine: ~' with three tabs. The first tab is active and shows the command 'rosparam list' and its output: '/rostdistro', '/roslaunch/uris/host_user_virtual_machine__40441', '/rosversion', and '/run_id'.

```
user@user-virtual-machine: ~  
user@user-virtual-machine:~$ rosparam list  
/rostdistro  
/roslaunch/uris/host_user_virtual_machine__40441  
/rosversion  
/run_id  
user@user-virtual-machine:~$
```

rosparam set /para_Period 1

rosparam list

A terminal window titled 'user@user-virtual-machine: ~' with three tabs. The first tab is active and shows the command 'rosparam list' and its output, followed by 'rosparam set /para_Period 1', another 'rosparam list' command, and its output which now includes '/para_Period', and finally 'rosparam get /para_Period' with output '1'.

```
user@user-virtual-machine: ~  
user@user-virtual-machine:~$ rosparam list  
/rostdistro  
/roslaunch/uris/host_user_virtual_machine__40441  
/rosversion  
/run_id  
user@user-virtual-machine:~$ rosparam set /para_Period 1  
user@user-virtual-machine:~$ rosparam list  
/para_Period  
/rostdistro  
/roslaunch/uris/host_user_virtual_machine__40441  
/rosversion  
/run_id  
user@user-virtual-machine:~$ rosparam get /para_Period  
1  
user@user-virtual-machine:~$
```

ros parameter 03

```
roslaunch my_work04 myNode04
rosparam get /para_Period
```

注意目前的寫法，rosparam set /para_Period 5
會沒辦法套用上，解法看後續PPT

```
user@user-virtual-machine:~$ roslaunch my_work04 myNode04
[ INFO] [1681786979.481918699]: myNode04: hi :0 @3.00 sec
[ INFO] [1681786982.483815762]: myNode04: hi :1 @3.00 sec
[ INFO] [1681786985.484211082]: myNode04: hi :2 @3.00 sec
[ INFO] [1681786988.484920812]: myNode04: hi :3 @3.00 sec
[ INFO] [1681786991.486081720]: myNode04: hi :4 @3.00 sec
[ INFO] [1681786994.486521119]: myNode04: hi :5 @3.00 sec
^C^Cuser@user-virtual-machine:~$ ^C
user@user-virtual-machine:~$ roslaunch my_work04 myNode04
[ INFO] [1681787234.719517045]: myNode04: hi :0 @1.00 sec
[ INFO] [1681787235.720898206]: myNode04: hi :1 @1.00 sec
[ INFO] [1681787236.721925833]: myNode04: hi :2 @1.00 sec
^Cuser@user-virtual-machine:~$
```

ros parameter 04

https://github.com/Waywrong/ros_course/

```
user@user-virtual-machine: ~/catkin_ws/src/my_work04/src
user@user-virtual-machine:~$ cd catkin_ws/src/my_work04/src/
user@user-virtual-machine:~/catkin_ws/src/my_work04/src$ gedit myNode04.cpp &

myNode04.cpp
~/catkin_ws/src/my_work04/src

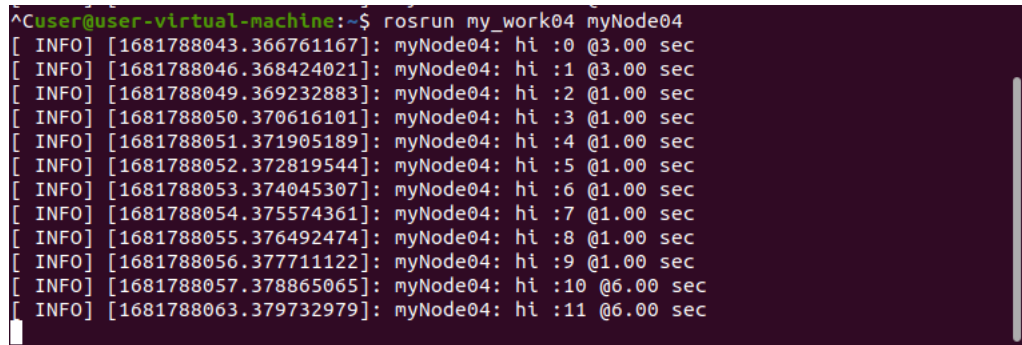
1#include "ros/ros.h"
2
3int main(int argc, char **argv)
4{
5    ros::init(argc, argv, "myNode04");
6    ros::NodeHandle n;
7
8    double dPeriod = 3;
9    n.getParam("/para_Period", dPeriod);
10    int iCnt = 0;
11
12    while (ros::ok())
13    {
14        n.getParam("/para_Period", dPeriod);
15        ROS_INFO("myNode04: hi :%d @%4.2f sec", iCnt++, dPeriod);
16        ros::spinOnce();
17        ros::Duration(dPeriod).sleep();
18    }
19    return 0;
20}

^Cuser@user-virtual-machine:~/catkin_ws$ catkin_make
Base path: /home/user/catkin_ws
Source space: /home/user/catkin_ws/src
Build space: /home/user/catkin_ws/build
Devel space: /home/user/catkin_ws/devel
Install space: /home/user/catkin_ws/install
####
#### Running command: "make cmake_check_build_system" in "/home/user/catkin_ws/build"
```

ros parameter 03

```
roslaunch my_work04 myNode04  
rosparam get /para_Period
```

```
rosparam set /para_Period 1  
rosparam set /para_Period 6
```

A terminal window with a dark purple background and green text. The prompt is 'Cuser@user-virtual-machine:~\$'. The command 'roslaunch my_work04 myNode04' has been executed. The output consists of 12 lines of log messages, each starting with '[INFO] [timestamp]: myNode04: hi :X @Y.YY sec' where X is the iteration number from 0 to 11 and Y is the period in seconds. The first two iterations (0 and 1) have a period of 3.00 seconds, and the remaining ten iterations (2 through 11) have a period of 1.00 second. The final two iterations (10 and 11) show a period of 6.00 seconds, indicating a parameter change.

```
Cuser@user-virtual-machine:~$ roslaunch my_work04 myNode04  
[ INFO] [1681788043.366761167]: myNode04: hi :0 @3.00 sec  
[ INFO] [1681788046.368424021]: myNode04: hi :1 @3.00 sec  
[ INFO] [1681788049.369232883]: myNode04: hi :2 @1.00 sec  
[ INFO] [1681788050.370616101]: myNode04: hi :3 @1.00 sec  
[ INFO] [1681788051.371905189]: myNode04: hi :4 @1.00 sec  
[ INFO] [1681788052.372819544]: myNode04: hi :5 @1.00 sec  
[ INFO] [1681788053.374045307]: myNode04: hi :6 @1.00 sec  
[ INFO] [1681788054.375574361]: myNode04: hi :7 @1.00 sec  
[ INFO] [1681788055.376492474]: myNode04: hi :8 @1.00 sec  
[ INFO] [1681788056.377711122]: myNode04: hi :9 @1.00 sec  
[ INFO] [1681788057.378865065]: myNode04: hi :10 @6.00 sec  
[ INFO] [1681788063.379732979]: myNode04: hi :11 @6.00 sec
```

ros launch

ROS PARAMETER

roslaunch

```
1 <?xml version="1.0"?>
2
3 <launch>
4 <!-- 參數定義在這邊 -->
5     <arg name="arg_para_Period" value="1" />
6     <arg name="arg1" value="true" />
7     <arg name="arg2" value="false" />
8
9     <param name="/para_Period" value="$(arg arg_para_Period)" />
10
11 <!-- 執行node檔 -->
12     <node pkg="my_work04" type="myNode04" name="myNode04" output="screen" if="$(arg arg1)"/>
13
14 <!-- 用launch執行launch檔-->
15     <include file="$(find my_work01)/launch/tt.launch" if="$(arg arg2)"/>
16
17 </launch>
```

roslaunch my_work04 my_work04.launch