배우기

ros

#ros

1: ros

(ROS)   .

2007   (Stanford Artificial Intelligence Laboratory) Stanford AI Robot STAIR   2008 2013
Willow Garage . 2013 ROS   .

ROS   . , , , ,   . ROS , BSD .

ROS   UAV, , ROS (   )   100   (   ). ROS   ,   . , / ,   .

ROS Unix . ROS   Mac OS X   ROS Fedora, Gentoo, Arch Linux   Linux .   ROS   C ++
Python   .

ROS *ROS Kinetic* 10 .

ROS ROS   http://www.ros.org/   .

	15.10, 16.04	2016-05-23
	14.04, 14.10, 15.04	2015-05-23
	13.10, 14.04	2014-07-22
	12.04, 12.10, 13.04	2013-09-04
	11.10, 12.04, 12.10	2012-12-31
	10.04, 11.10, 12.04	2012-04-23
Emys	10.04, 10.10, 11.04, 11.10	2011 8 30
	10.04, 10.10, 11.04	2011-03-02
C	9.04, 9.10, 10.04, 10.10	2010-08-02
	8.04	2010-03-02

Examples

ROS ( ). ROS ROS   ,   . ROS ,   .

ROS				
	16.04 (Xenial)	amd64 / i386 / armhf		-
	15.10 (Wily)	amd64 / i386		-

ROS				
	8 (Jessie)	amd64 / arm64		-
	OS X ()	-		-
		-		-
	OpenEmbedded / Yocto	-		-

...!

```
mkdir -p ~/catkin_ws/src
cd ~/catkin_ws/src
catkin_init_workspace
```

```
cd ~/catkin_ws/
catkin_make
```

.

```
source devel/setup.bash
```

hello_world .

```
catkin_create_pkg hello_world std_msgs rospy roscpp
```

src talker.cpp .

```
cd hello_world/src
touch talker.cpp
```

" " .

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

#include <sstream>

int main(int argc, char **argv)
{
  ros::init(argc, argv, "talker");

  ros::NodeHandle n;

  ros::Publisher chatter_pub = n.advertise<std_msgs::String>("chatter", 1000);

  ros::Rate loop_rate(10);

  int count = 0;
  while (ros::ok())
  {
```

```
    std_msgs::String msg;

    std::stringstream ss;
    ss << "hello world " << count;
    msg.data = ss.str();

    ROS_INFO("%s", msg.data.c_str());

    chatter_pub.publish(msg);

    ros::spinOnce();

    loop_rate.sleep();
    ++count;
  }

  return 0;
}
```

```
cd ..
```

CMakeLists.txt / .

```
catkin_package(
 INCLUDE_DIRS include
 LIBRARIES hello_world
#  CATKIN_DEPENDS roscpp rospy std_msgs
#  DEPENDS system_lib
)

include_directories(include ${catkin_INCLUDE_DIRS})

add_executable(talker src/talker.cpp)
target_link_libraries(talker ${catkin_LIBRARIES})
add_dependencies(talker hello_world_generate_messages_cpp)
```

```
cd ..
```

```
catkin_make
```

.

```
source devel/setup.bash
```

ROS

```
roscore
```

roscore   / .

```
rosrun hello_world talker
```

/ .

```
rostopic echo /chatter
```

ros : https://riptutorial.com/ko/ros/topic/7287/ros-

2:

' '''.

"" ""   .

Examples

ros   Yaml

roslaunch ROS   .   "* .launch" .

( ) .   ?

```
roscd stereo_camera
rosparam load marvin_cameras.yaml
rosrun stereo_camera stereo_camera __name:=bumblebeeLeft
rosrun stereo_camera stereo_camera __name:=bumblebeeCenter

roslaunch openni_launch_marvin kinect_left.launch
roslaunch openni_launch_marvin kinect_center.launch
```

, break up these commands in pieces .   *roscd* , *rosparam* , *rosrun roslaunch* 4 . !

roslaunch .   one and only roslaunch file in ROS one and only roslaunch file in ROS .
"solution.launch" .

XML ROS   "basic_example.launch" "roslaunch_example" ROS .

```
<launch>

</launch>
```

.

```
$ roslaunch roslaunch_example basic_example.launch
```

:

```
$ roslaunch package_name launch_file_name.launch
```

.

:
ROS ROS   .   . :

```
rosrun stereo_camera stereo_camera __name:=bumblebeeLeft
rosrun stereo_camera stereo_camera __name:=bumblebeeCenter
```

```
stereo_camera   stereo_camera   __name:=bumblebeeLeft __name:=bumblebeeCenter .
```

.

```
<launch>
    <node name="$(arg name)" pkg="stereo_camera" type="stereo_camera" output="screen">
            <param name="name" value="bumblebeeLeft" />
    </node>

    <node name="$(arg name)" pkg="stereo_camera" type="stereo_camera" output="screen">
            <param name="name" value="bumblebeeCenter" />
    </node>
</launch>
```

.

ROS   :

,   "" .

```
<param name="name" value="bumblebeeCenter" />
```

,   ( ) `"$(arg parameter_name)"`   `"$(arg parameter_name)"`   `"$(arg parameter_name)"`   .

    :

`(~/.ros)`   `on the terminal`   `on the terminal` , "log"   output "screen" .

ROS ROS :

XML   .   . include   .   .

```
roslaunch openni_launch_marvin kinect_left.launch
roslaunch openni_launch_marvin kinect_center.launch
```

.

```
<include file="$(find openni_launch_marvin)/launch/kinect_left.launch" />
<include file="$(find openni_launch_marvin)/launch/kinect_center.launch" />
```

    ROS   roscd

. , roslaunch `"$(find package_name)"` . `relative to the package racine`   .
"kinect_center.launch" "openni_launch_marvin / launch /" .

YAML   :

ROS YAML   ROS "rosparam" .   " ROS   rosparam YAML   .   ."

YAML   .

```
<rosparam command="load" file="$(find marvin_cameras)/config/marvin_cameras.yaml" />
```

YAML "marvin_cameras.yaml" "marvin_cameras / config /" .

"solution.launch" .

.

```
<launch>

    <rosparam command="load" file="$(find marvin_cameras)/config/marvin_cameras.yaml" />

    <node name="$(arg name)" pkg="stereo_camera" type="stereo_camera" output="screen">
         <param name="name" value="bumblebeeLeft" />
    </node>

    <node name="$(arg name)" pkg="stereo_camera" type="stereo_camera" output="screen">
         <param name="name" value="bumblebeeCenter" />
    </node>

    <include file="$(find openni_launch_marvin)/launch/kinect_left.launch" />
    <include file="$(find openni_launch_marvin)/launch/kinect_center.launch" />

</launch>
```

.

: https://riptutorial.com/ko/ros/topic/7361/-

3:

. ROS . ROS .

Examples

.

```
$ mkdir -p ~/workspace_name/src
$ cd ~/workspace_name/src
$ catkin_init_workspace
$ cd ~/workspace_name/
$ catkin_make
```

workspace_name `workspace_name` . .

```
$ source ~/workspace_name/devel/setup.bash
```

: https://riptutorial.com/ko/ros/topic/8313/--

4:

ROS . src . CMakeLists.txt package.xml .

Examples

rospy

workspace_name workspace_name   package_name   .

```
$ cd ~/workspace_name/src/
$ catkin_create_pkg package_name rospy
```

: https://riptutorial.com/ko/ros/topic/8314/-

S. No		Contributors
1	ros	ckirksey3, Community, Photon, Vtik
2		ensonic, Vtik
3		lmiguelvargasf
4		lmiguelvargasf, Michael