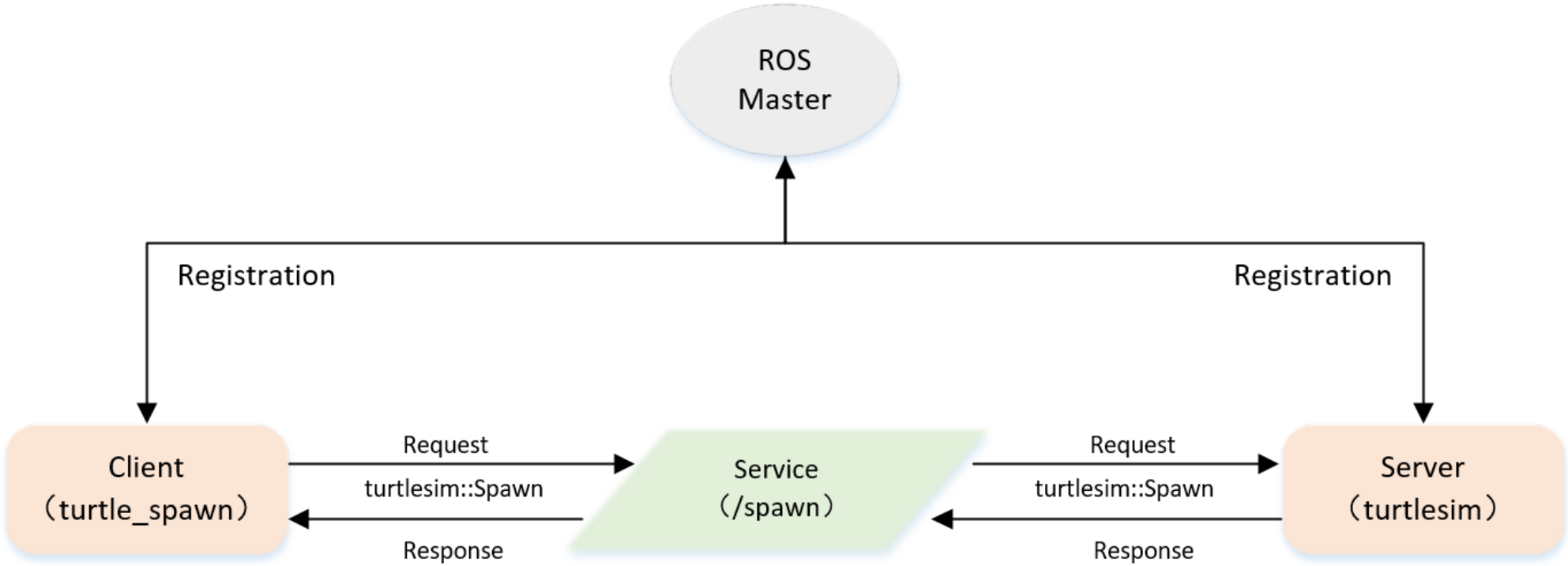
古月居 GYH.AI ROS

ROS入门
21讲

13.客户端Client的编程实现

主讲人：古月

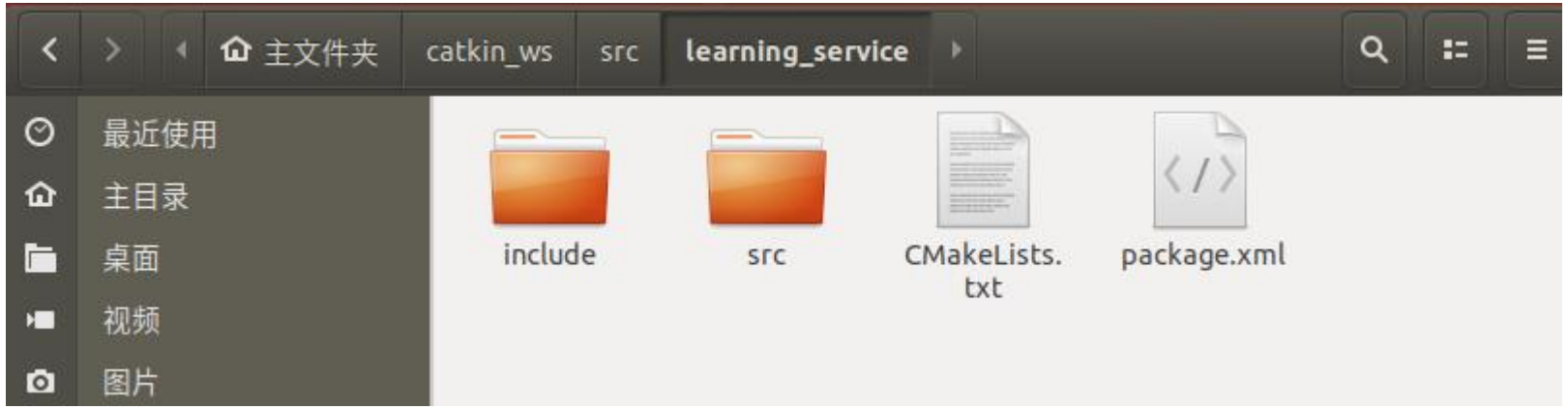
话题模型



服务模型（客户端/服务器）

创建功能包

```
$ cd ~/catkin_ws/src
$ catkin_create_pkg learning_service roscpp rospy std_msgs geometry_msgs
turtlesim
```



创建客户端代码（C++）

```cpp
/**
 * 该例程将请求/spawn服务，服务数据类型turtlesim::Spawn
 */

#include <ros/ros.h>
#include <turtlesim/Spawn.h>

int main(int argc, char** argv)
{
    // 初始化ROS节点
    ros::init(argc, argv, "turtle_spawn");

    // 创建节点句柄
    ros::NodeHandle node;

    // 发现/spawn服务后，创建一个服务客户端，连接名为/spawn的service
    ros::service::waitForService("/spawn");
    ros::ServiceClient add_turtle = node.serviceClient<turtlesim::Spawn>("/spawn");

    // 初始化turtlesim::Spawn的请求数据
    turtlesim::Spawn srv;
    srv.request.x = 2.0;
    srv.request.y = 2.0;
    srv.request.name = "turtle2";

    // 请求服务调用
    ROS_INFO("Call service to spwan turtle[x:%0.6f, y:%0.6f, name:%s]",
             srv.request.x, srv.request.y, srv.request.name.c_str());

    add_turtle.call(srv);

    // 显示服务调用结果
    ROS_INFO("Spwan turtle successfully [name:%s]", srv.response.name.c_str());

    return 0;
};
```

turtle_spawn.cpp

如何实现一个客户端

- 初始化ROS节点；
- 创建一个Client实例；
- 发布服务请求数据；
- 等待Server处理之后的应答结果。

配置客户端代码编译规则

```
## Declare a C++ executable
## With catkin_make all packages are built within a single CMake context
## The recommended prefix ensures that target names across packages don't collide
# add_executable(${PROJECT_NAME}_node src/learning_service_node.cpp)

## Specify libraries to link a library or executable target against
# target_link_libraries(${PROJECT_NAME}_node
#   ${catkin_LIBRARIES}
# )

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

如何配置CMakeLists.txt中的编译规则

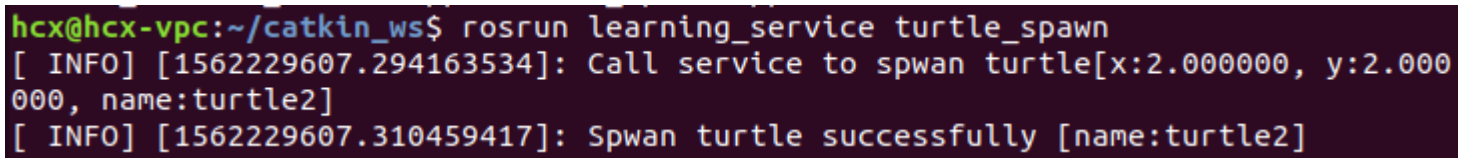
- 设置需要编译的代码和生成的可执行文件；
- 设置链接库；

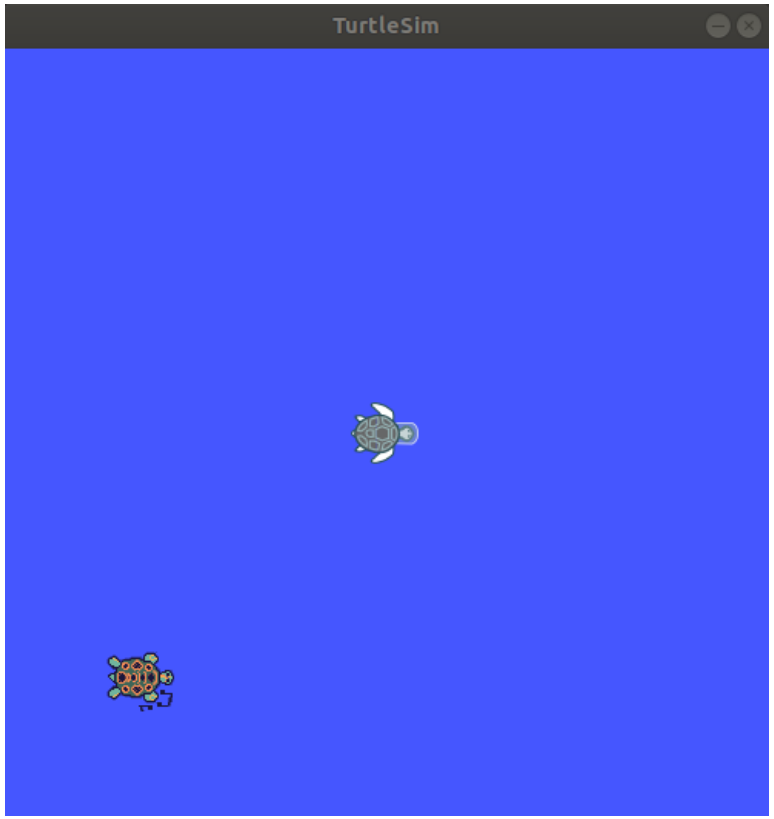
```
add_executable(turtle_spawn src/turtle_spawn.cpp)
target_link_libraries(turtle_spawn ${catkin_LIBRARIES})
```

CMakeLists.txt

编译并运行客户端

```
$ cd ~/catkin_ws
$ catkin_make
$ source devel/setup.bash
$ roscore
$ rosrun turtlesim turtlesim_node
$ rosrun learning_service turtle_spawn
```





创建客户端代码（Python）

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# 该例程将请求/spawn服务，服务数据类型turtlesim::Spawn

import sys
import rospy
from turtlesim.srv import Spawn

def turtle_spawn():
    # ROS节点初始化
    rospy.init_node('turtle_spawn')

    # 发现/spawn服务后，创建一个服务客户端，连接名为/spawn的service
    rospy.wait_for_service('/spawn')
    try:
        add_turtle = rospy.ServiceProxy('/spawn', Spawn)

        # 请求服务调用，输入请求数据
        response = add_turtle(2.0, 2.0, 0.0, "turtle2")
        return response.name
    except rospy.ServiceException, e:
        print "Service call failed: %s"%e

if __name__ == "__main__":
    #服务调用并显示调用结果
    print "Spwan turtle successfully [name:%s]" %(turtle_spawn())
```

turtle_spawn.py

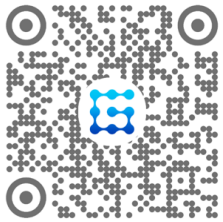
如何实现一个客户端

- 初始化ROS节点；
- 创建一个Client实例；
- 发布服务请求数据；
- 等待Server处理之后的应答结果。

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

古月学院

古月居GYH