

向服务器发送一个指令 在一个指令的位置 生存一个乌龟
这个乌龟是基于左下角的定义的

1、找到服务名称和服务的消息

显示所有的 service list

spawn 这个 service 的话题名称

```
qinghuan@qinghuan-System-Product-Name: ~/桌面$ rosservice list
/clear
/key/get_loggers
/key/set_logger_level
/kill
/reset
/rosout/get_loggers
/rosout/set_logger_level
/spawn
/sub_pose/get_loggers
/sub_pose/set_logger_level
```

service 的数据类型

Spawn 是 srv 的文件格式

---上面是客户提交的字段 下面是 server 返回的字段

```
qinghuan@qinghuan-System-Product-Name: ~/桌面$ rossrv info /turtlesim/Spawn
Unknown srv type [/turtlesim/Spawn]: invalid name [/turtlesim/Spawn]
qinghuan@qinghuan-System-Product-Name: ~/桌面$ rossrv info turtlesim/Spawn
float32 x
float32 y
float32 theta
string name
---
string name

qinghuan@qinghuan-System-Product-Name: ~/桌面$
```

rosservice list

发布的代码

```
> plumbing_test > src > test03_service_client.cpp > main(int, char * [])
1  ∨ #include "ros/ros.h"
2    #include "turtlesim/Spawn.h"
3
4    //这里都是头文件
5  ∨ int main(int argc, char* argv[])
6    {
7      setlocale(LC_ALL, "");
8      ros::init(argc, argv, "sercice_clientcall");
9      ros::NodeHandle nh;
10     ros::ServiceClient client= nh.serviceClient<turtlesim::Spawn>("/spawn");
11     turtlesim::Spawn srv;
12     srv.request.x=1.0;
13     srv.request.y=4.0;
14     srv.request.theta=1.57;//最大值是3.14 向右是-3.14 右转90
15     srv.request.name="turtle2";//不能重复调用的原因 是名字是一样的 如果想要重复i调用 需要把 名字该了 每个
16     client.waitForExistence();
17     //发送数据 返回的一个bool 代表是否成功响应
18     bool flag1 = client.call(srv);
19     if (flag1)
20     {
21       ROS_INFO("响应成功 %s ", srv.response.name.c_str());//这个是通过命令行看到spawn的命名方式了
22     }
23     ∨ else
24     {
25       ROS_INFO("响应失败");
26     }
27
28     return 0;
29 }
```

```
● qinghuan@qinghuan-System-Product-Name:~/env_cv/demo04_ws$ source ./devel/setup.bash
○ qinghuan@qinghuan-System-Product-Name:~/env_cv/demo04_ws$ roslaunch plumbing_test start_lauch.launch
... logging to /home/qinghuan/.ros/log/d36acc70-e9a5-11ee-94ab-b74ac8e46bd4/roslaunch-qinghuan-System-Product-Name-62527.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://qinghuan-System-Product-Name:46455/
```

问题 7 输出 调试控制台 终端 端口 注释

```
● qinghuan@qinghuan-System-Product-Name:~/env_cv/demo04_ws$ source ./devel/setup.bash
● qinghuan@qinghuan-System-Product-Name:~/env_cv/demo04_ws$ rosrn plumbing_test test03_service_client
[ INFO] [1711272496.902240661]: 响应成功 turtle2
○ qinghuan@qinghuan-System-Product-Name:~/env_cv/demo04_ws$ □
```

设置参数

获取参数信息

通过背景 background

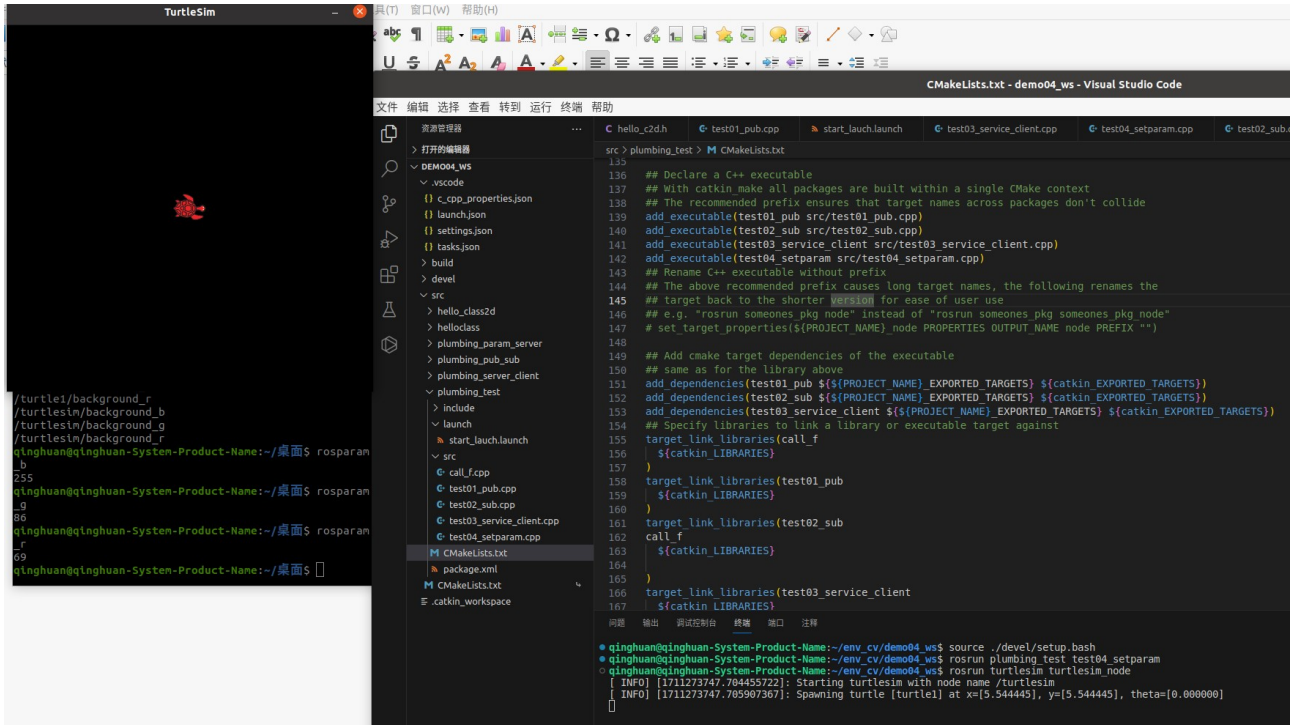
```
qinghuan@qinghuan-System-Product-Name: ~/桌面 80x23
qinghuan@qinghuan-System-Product-Name: ~/桌面$ rosparam list
/roscdistro
/roslaunch/uris/host_qinghuan_system_product_name__36269
/roslaunch/uris/host_qinghuan_system_product_name__37469
/roslaunch/uris/host_qinghuan_system_product_name__37545
/roslaunch/uris/host_qinghuan_system_product_name__37633
/roslaunch/uris/host_qinghuan_system_product_name__42915
/roslaunch/uris/host_qinghuan_system_product_name__46455
/roslaunch/uris/host_qinghuan_system_product_name__46519
/rosversion
/run_id
/turtle1/background_b
/turtle1/background_g
/turtle1/background_r
/turtlesim/background_b
/turtlesim/background_g
/turtlesim/background_r
qinghuan@qinghuan-System-Product-Name: ~/桌面$
```

通过设置颜色颜色

```
qinghuan@qinghuan-System-Product-Name: ~/桌面$ rosparam get /turtlesim/background_b
255
qinghuan@qinghuan-System-Product-Name: ~/桌面$ rosparam get /turtlesim/background_g
86
qinghuan@qinghuan-System-Product-Name: ~/桌面$ rosparam get /turtlesim/background_r
69
qinghuan@qinghuan-System-Product-Name: ~/桌面$
```

不用 roslaunch 启动 launch 的问题是 因为用这个把 roscore 和参数服务器全部启动 关闭的时候全部关闭没有保存

先运行自己的文件 对 param 做一个设置 结束后 在启动在会有效果
直接启动 是默认值



nh 的时候说明了 命名空间 调用了重载函数的方法 这个需要查阅 NodeHandle 的重载版本
这个是最后的效果 作的对比

