1 新建工作空间

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

2 新建功能包

```
cd ~/catkin_ws/src
# catkin_create_pkg 功能包名 [依赖功能包1] [依赖功能包2]...
catkin_create_pkg msgtest std_msgs roscpp
```

3 新建msg文件

```
cd ~/catkin_ws/src/msgtest/
mkdir msg
cd msg/
vim msg1.msg
```

在msg1.msg中写入以下内容：

```
int32 num
string str
```

4 新建发布文件pub.cpp

```
cd ~/catkin_ws/src/msgtest/src
vim pub.cpp
```

写入以下内容：

```
#include "ros/ros.h"
// msgtest是功能包名，msg1是msg文件名，编译成功后会在devel/include自动生成一个头文件
#include "msgtest/msg1.h"

int main(int argc, char **argv)
{
 ros::init(argc, argv, "msgpub");  // msgpub为节点的名字

 // 创建一个句柄
   ros::NodeHandle n;

  // 新建一个pub对象，<>中是自定义的msg文件，pub_msg为topic的名字
   ros::Publisher msg_pub = n.advertise<msgtest::msg1>("mymsg", 1000);

 // 发送频率
   ros::Rate loop_rate(10);
   while (ros::ok())
   {
  // 自定义的msg类的一个新变量，并赋值
     msgtest::msg1 msg;
     msg.num = 1024;
  msg.str = "hello slam";

  // 发送msg
     msg_pub.publish(msg);
  //ROS的printf
     ROS_INFO("Publish : number = %d string = %s", msg.num, msg.str.c_str());
  //循环刷新
     ros::spinOnce();

     loop_rate.sleep();
  }
  return 0;
}
```

5 新建订阅文件sub.cpp

```
cd ~/catkin_ws/src/msgtest/src
vim sub.cpp
```

写入以下内容：

```
#include "ros/ros.h"
#include "msgtest/msg1.h"

// 回调函数
void chatterCallback(const msgtest::msg1::ConstPtr& msg)
{
  ROS_INFO("I heard: [num = %d, string = %s]", msg->num, msg->str.c_str());
}

int main(int argc, char **argv)
{
  ros::init(argc, argv, "msgsub");
  ros::NodeHandle n;

 //新建一个sub，chatterCallback是回调函数名
  ros::Subscriber sub = n.subscribe("mymsg", 1000, chatterCallback);

  ros::spin();
  return 0;
}
```

6 修改配置文件

CMakelists.txt

```
cd ~/catkin_ws/src/msgtest/
vim CMakeLists.txt
```

在文件中第13行处添加上message_generation：

```
find_package(catkin REQUIRED COMPONENTS
  std_msgs
 roscpp
  message_generation
)
```

将50行处注释取消掉， 并且按照下面代码修改：

```
 add_message_files(
   FILES
   msg1.msg
 )
```

将70行处注释取消掉，如下：

```
generate_messages(
   DEPENDENCIES
   std_msgs
 )
```

在文件最后添加如下代码：

```
# msgpub是自定义节点名，pub.cpp是文件名
add_executable(msgpub src/pub.cpp)
# msgpub是节点名
target_link_libraries(msgpub ${catkin_LIBRARIES})
# msgsub是自定义节点名，sub.cpp是文件名
add_executable(msgsub src/sub.cpp)
# msgsub是节点名
target_link_libraries(msgsub ${catkin_LIBRARIES})
```

package.xml

```
cd ~/catkin_ws/src/msgtest/
vim package.xml
```

在58行之后加入：

```
<build_depend>message_generation</build_depend>
<exec_depend>message_runtime</exec_depend>
```

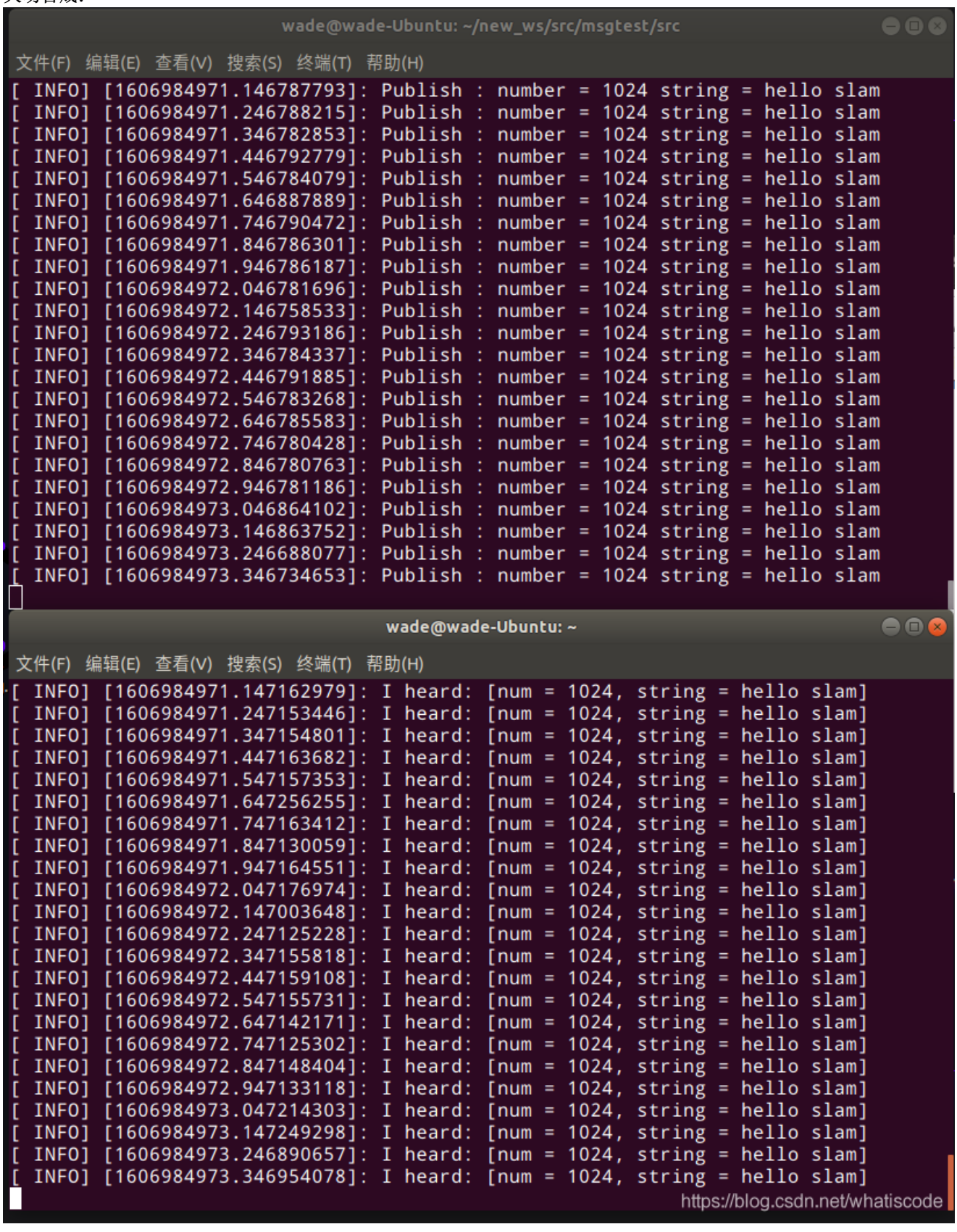
7 编译并运行

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

新建一个终端：

```
source ~/catkin_ws/devel/setup.bash
rosrun msgtest msgpub
```

再新建一个终端：

```
source ~/catkin_ws/devel/setup.bash
rosrun msgtest msgsub
```

大功告成：