

# ROS学习和理解分享



柯艺杰

E-mail: [jay@cloudkernel.cn](mailto:jay@cloudkernel.cn)





Started before 2007...

## ROS: an open-source Robot Operating System

Morgan Quigley\*, Brian Gerkey†, Ken Conley†, Josh Faust†, Tully Foote†,  
Jeremy Leibs‡, Eric Berger†, Rob Wheeler†, Andrew Ng\*

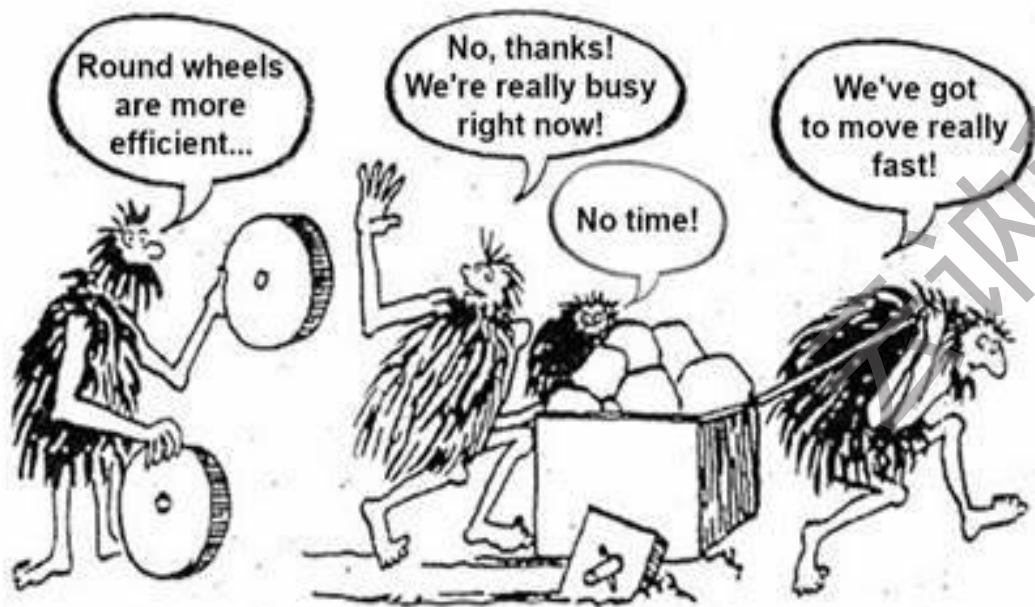
\*Computer Science Department, Stanford University, Stanford, CA

†Willow Garage, Menlo Park, CA

‡Computer Science Department, University of Southern California

**The Robot Operating System (ROS) is a set of software libraries and tools that help you build robot applications.**

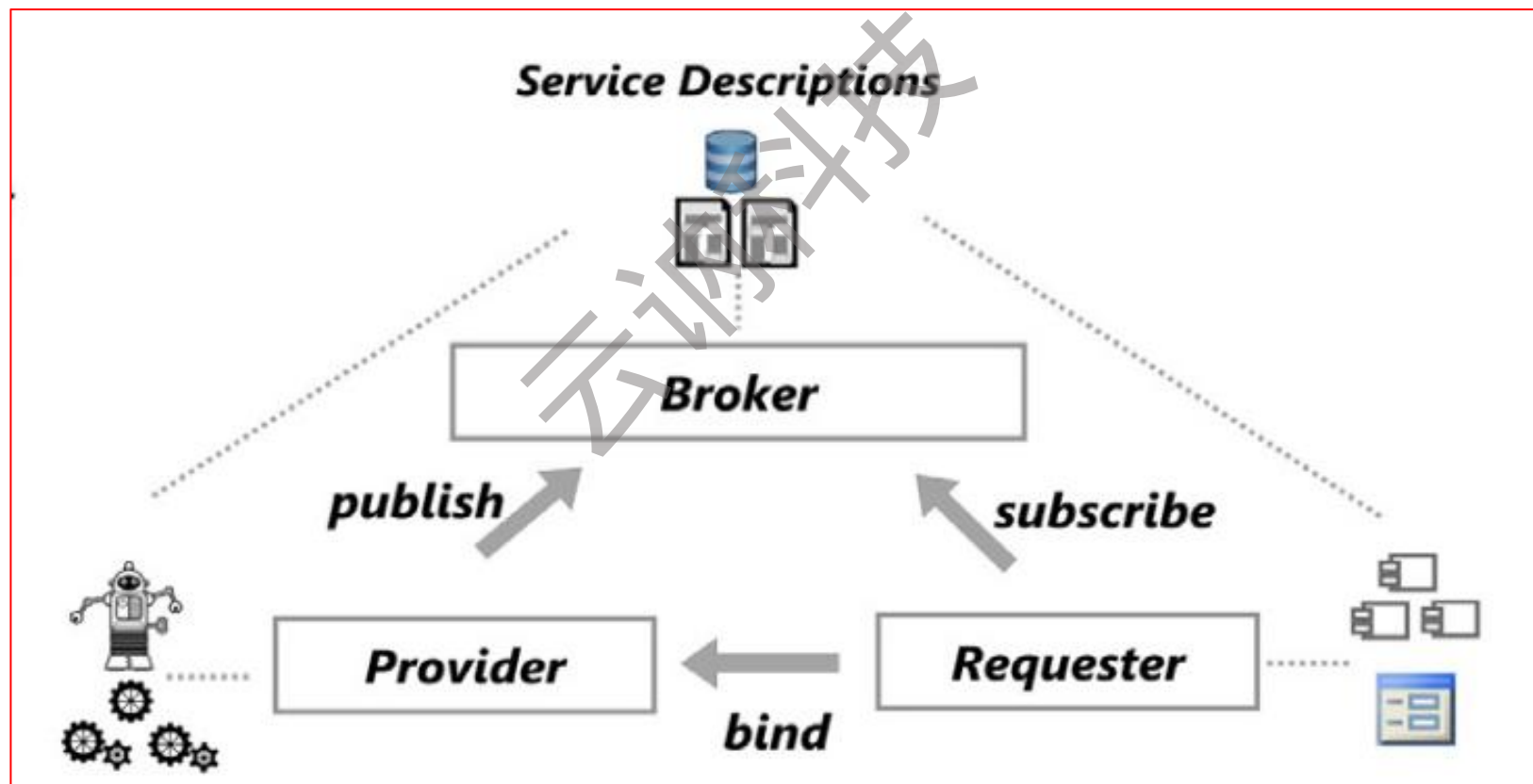
ROS解决了什么问题: **Don't reinvent the wheel**



开源, 协作, 共享

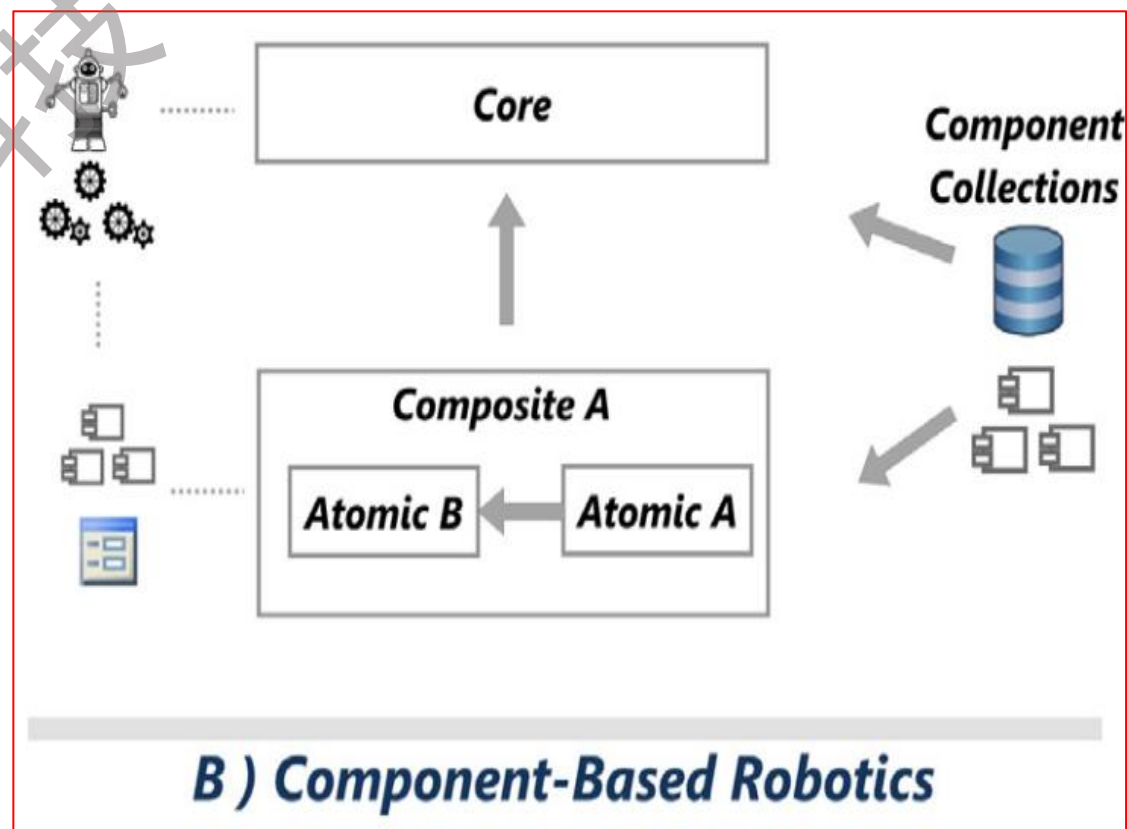
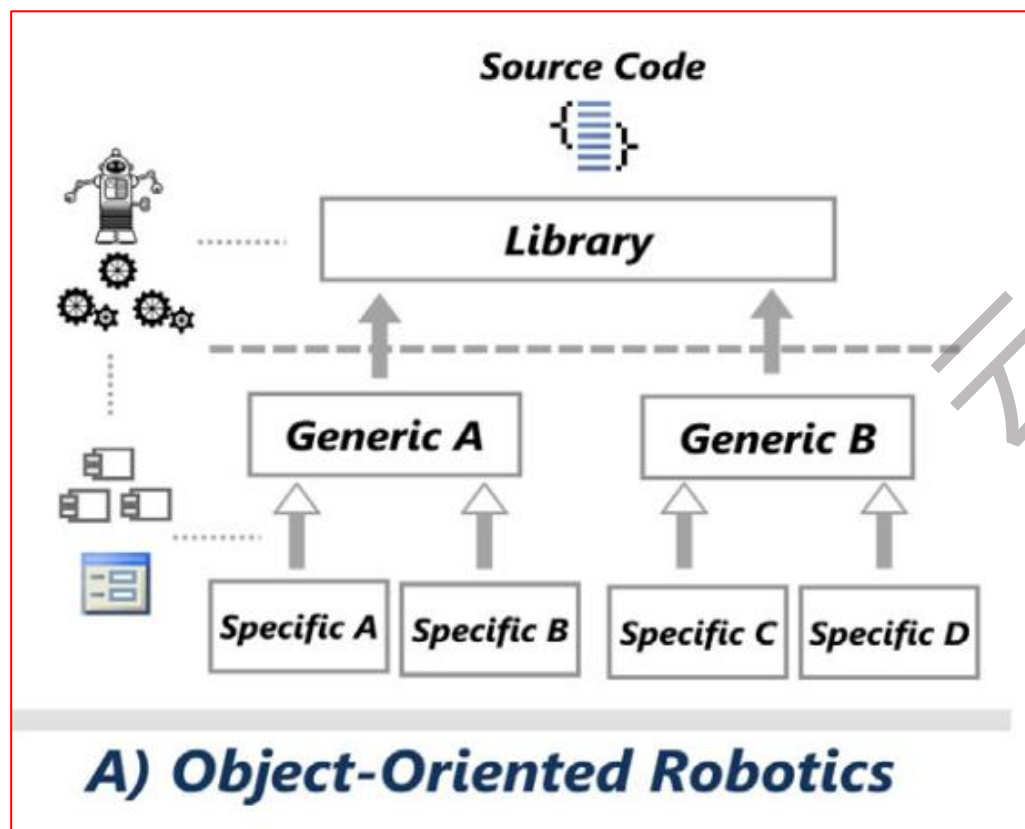
吸收和复用前人的成果

## 面向服务的解耦式软件设计思想 (publish-subscribe模式)



## 在这之前呢？

我们可以在很多设计中看到这两种模式的影子.....



# 学习ROS能收获什么

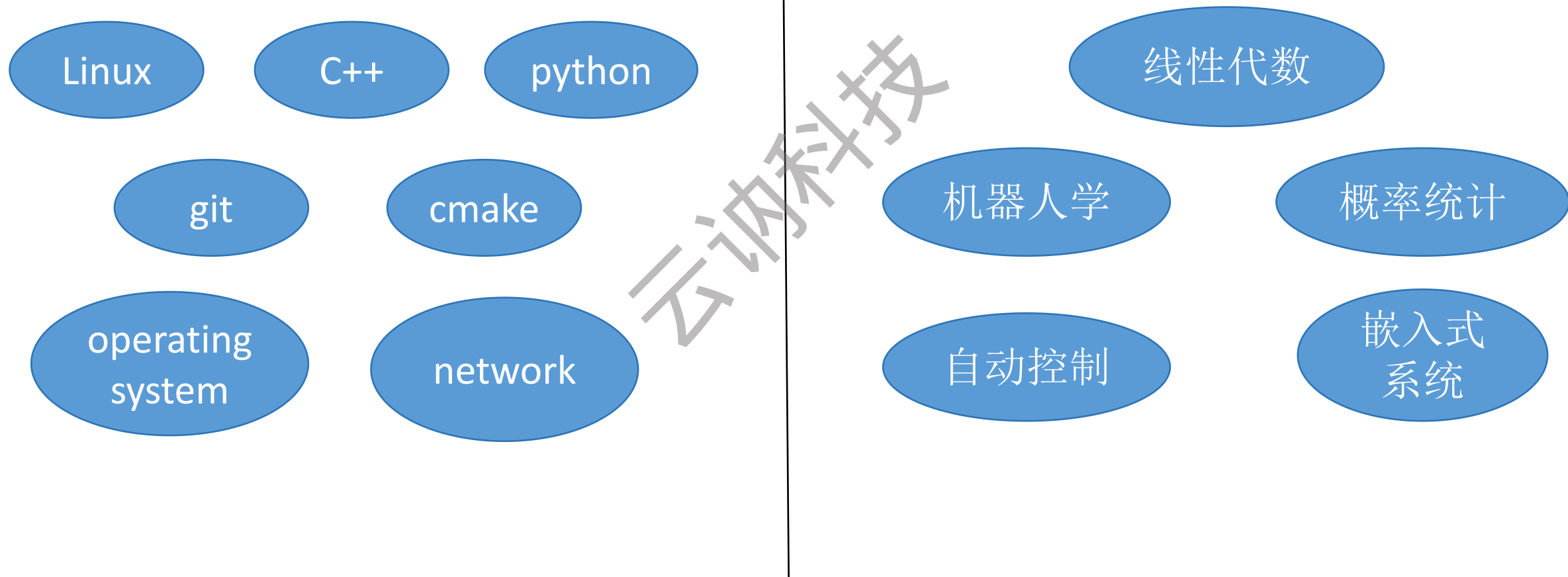
## Expected:

- 学习和提升编程能力
- 掌握机器人领域的软件开发技能
- 接触世界一流的ROS社区和优秀的开源代码
- 就业竞争力上升

## Tips:

- ROS只是一个工具，会用ROS不代表掌握了屠龙之术
- 避免成为调包侠，虽然市场上90%以上都沦为此类
- 专注一个行业技术领域即可，都能到达很高的技术地位，万金油的路是走不通的

# 工具和概念



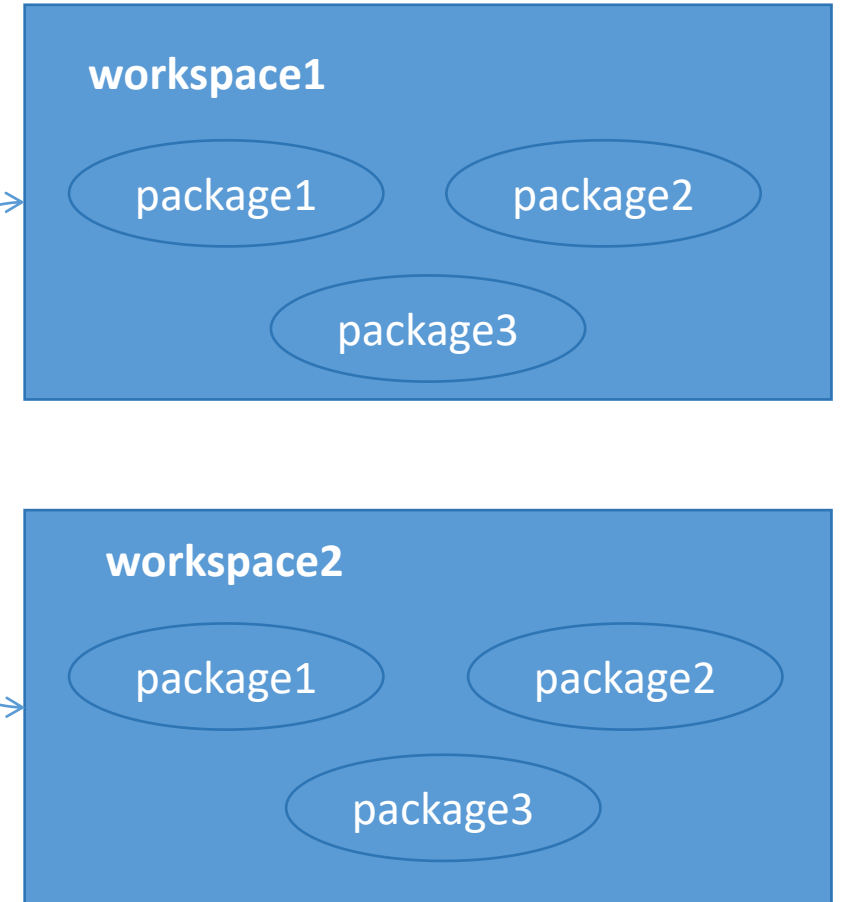
## Key concepts:

- publisher and subscriber
- services
- roslaunch
- node, package, workspace

## others

- rqt\_graph, rostopic, ROS\_INFO...
- tf, eigen

重要链接: <http://wiki.ros.org/ROS/Tutorials>



# 真正看懂一个package

Package example: loco\_driver

## Part 1. package.xml

定义该包编译和执行中所依赖的包，多数可以在开始创建包自动生成，命令：catkin\_create\_pkg

[http://wiki.ros.org/catkin/commands/catkin\\_create\\_pkg](http://wiki.ros.org/catkin/commands/catkin_create_pkg)

```
<buildtool_depend>catkin</buildtool_depend>  
<build_depend>roscpp</build_depend>  
<build_depend>rospy</build_depend>  
<build_depend>std_msgs</build_depend>  
<build_export_depend>roscpp</build_export_depend>  
<build_export_depend>rospy</build_export_depend>  
<build_export_depend>std_msgs</build_export_depend>  
<exec_depend>roscpp</exec_depend>  
<exec_depend>rospy</exec_depend>  
<exec_depend>std_msgs</exec_depend>
```

# 真正看懂一个package

Package example: loco\_driver, 一个和硬件模块通信的驱动程序

## Part 2. CmakeLists

定义该包编译的详细过程，依赖包，消息和服务定义。Cmake自身也是功能强大的编译工具，在其他项目中也会经常使用。

<http://wiki.ros.org/catkin/CMakeLists.txt>

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

找到本包编译所依赖的其他包的位置，在编译中得到相应的环境变量

# 真正看懂一个package

Package example: loco\_driver

## Part 2. CmakeLists

```
## Generate messages in the 'msg' folder
add_message_files(
  FILES
  UwbTwaDistance.msg
)
```

```
## Generate added messages and services with any
dependencies listed here
generate_messages(
  DEPENDENCIES
  std_msgs
)
```

新增一个消息，服务类似

<http://wiki.ros.org/ROS/Tutorials/CreatingMsgAndSrv>

package.xml:

```
<build_depend>message_generation</build_depend>
```

```
<exec_depend>message_runtime</exec_depend>
```

# 真正看懂一个package

Package example: loco\_driver

## Part 2. CmakeLists

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

有其他包依赖loco\_driver，比如新增的message消息，我们需要把loco\_driver作为一个library传递给catkin build system

# 真正看懂一个package

Package example: loco\_driver

## Part 2. CmakeLists

```
add_executable(${PROJECT_NAME}_node
src/loco_driver_node.cpp)

add_dependencies(${PROJECT_NAME}_node
${${PROJECT_NAME}_EXPORTED_TARGETS}
${catkin_EXPORTED_TARGETS})

target_link_libraries(${PROJECT_NAME}_node
  ${catkin_LIBRARIES}
)
```

定义主程序执行文件（包含main入口），并把它需要的文件传递给它，如其他cpp文件，依赖的库等

# 真正看懂一个package

Package example: loco\_driver

## Part 3. loco\_driver\_node.cpp程序

```
#include "loco_driver/UwbTwaDistance.h" (自定义的消息文件)
```

(主程序入口, 一种标准的写法)

```
int main(int argc, char **argv)
{
    while(ros::ok()){
        ros::spinOnce();
        rate.sleep();
    }
}
```

**roscpp的深入理解:** <http://wiki.ros.org/roscpp/Overview>

有其他的模式存在, 比如timer, multi-threading, multiple node handlers

# 真正看懂一个package

Package example: loco\_driver

## Part 3. loco\_driver\_node.cpp程序

publisher, subscriber, callbacks.... (缓存, 优先级等)

```
int parseChar(uint8_t b);  
void decodeInit();  
void handle_message(uwb_message_t* msg);  
void handle_message_uwb_twa(uwb_message_t* msg);
```

和嵌入式写法一样的函数设计，但局部地方有注意

```
serial_obj.flushInput();
```

# 真正看懂一个package

Package example: loco\_driver

## Part 4. ROS是如何工作的？

读源码，roscpp的具体实现

以后ros是有可能闭源的

<https://github.com/ros/ros>

云纳科技

# 最后的建议

- 多写多做，胜于读死书，有时需要逼自己一下
- 读优秀的代码，先模仿再创造，  
<https://index.ros.org/packages/>，几百个优秀的软件包，看不完
- 做项目，打比赛，把程序写成肌肉记忆的行为
- 机器人技术元素：控制，传感，定位，建图，规划，决策，选择其中一到两个潜心数年都有所成，关键是能解决实际应用的问题，直接能用的开源程序很少

谢谢

