

4、ROS2工作空间

1、工作空间简介

在ROS机器人开发中，我们针对机器人某些功能进行开发时，各种编写的代码、参数、脚本等文件，需要放置在某一个文件夹里进行管理，这个文件夹在ROS系统中就叫做工作空间。所以工作空间是一个存放项目开发相关文件的文件夹，也是开发过程中存放所有资料的大本营。

2、创建工作空间

- 创建一个文件夹，作为存放工程文件的工作空间，yahboomcar_ws是文件夹名称，可以任意命名

```
mkdir -p yahboomcar_ws/src
```

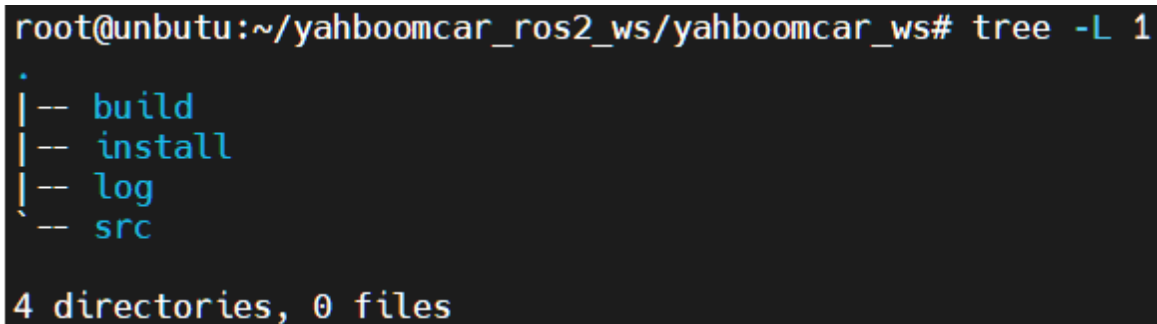
- 切换到工作空间文件夹

```
cd ~/yahboomcar_ws
```

3、编译工作空间

```
colcon build
```

构建完成后，我们应该看到 `build`、`install` 和 `log` 目录：



```
root@ubuntu:~/yahboomcar_ros2_ws/yahboomcar_ws# tree -L 1
.
|-- build
|-- install
|-- log
`-- src

4 directories, 0 files
```

ROS系统中一个典型的工作空间结构如上所示，这个yahboomcar_ws就是工作空间的根目录，里边会有四个子目录，或者叫做四个子空间。

- **src**，**代码空间**，未来编写的代码、脚本，都需要人为的放置到这里；
- **build**，**编译空间**，保存编译过程中产生的中间文件；
- **install**，**安装空间**，放置编译得到的可执行文件和脚本；
- **log**，**日志空间**，编译和运行过程中，保存各种警告、错误、信息等日志。

总体来讲，这四个空间的文件夹，我们绝大部分操作都是在src中进行的，编译成功后，就会执行install里边的结果，build和log两个文件夹用的很少。

这里也要强调一点，**工作空间的名称我们可以自己定义**，数量也并不是唯一的，比如：

工作空间1: `ros2_ws_a`, 用于A机器人的开发
工作空间1: `ros2_ws_b`, 用于B机器人的开发
工作空间1: `ros2_ws_c`, 用于C机器人的开发

以上情况是完全允许的, 就像是我们在集成开发环境中创建了多个新工程一样, 都是并列存在的关系。

4、设置环境变量

编译成功后, 为了让系统能够找到我们的功能包和可执行文件, 还需要设置环境变量:

```
# 仅在当前终端生效
source install/setup.bash
# 所有终端均生效
echo "source ~/yahboomcar_ws/install/setup.bash" >> ~/.bashrc
```

至此, 我们就完成了工作空间的创建、编译和配置。