4.ROS2 workspace

1. Workspace introduction

In ROS robot development, when we develop certain functions of the robot, various written codes, parameters, scripts and other files need to be placed in a certain folder for management. This folder is called a workspace in the ROS system. Therefore, the workspace is a folder that stores files related to project development, and it is also the base camp where all materials are stored during the development process.
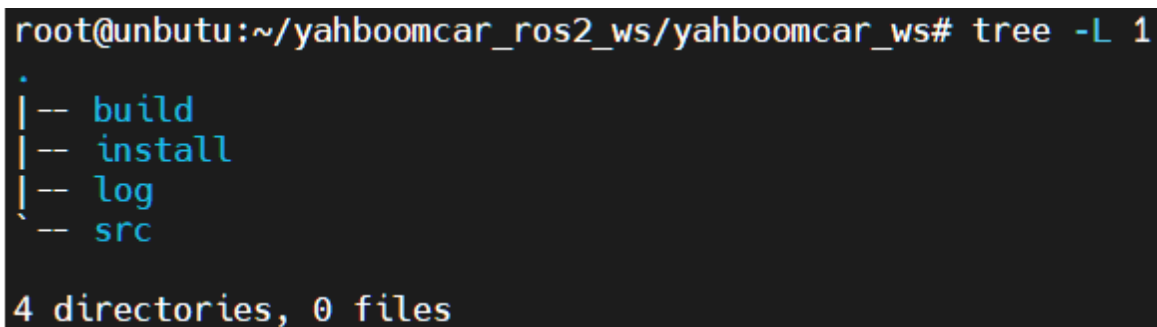
2. Create a workspace

```
# Select a storage directory, or place it in the root directory
cd ~/yahboomcar_ros2_ws
# Create a workspace
mkdir -p yahboomcar_ws/src
cd yahboomcar_ws
```

3. Compile workspace

```
colcon build
```

After the build is complete, we should see the `build`, `install` and `log` directories:

A typical workspace structure in the ROS system is as shown above. This yahboomcar_ws is the root directory of the workspace, and there will be four subdirectories, or four subspaces.

- **src, code space**, codes and scripts written in the future need to be placed here manually;
- **build, compilation space**, saves the intermediate files generated during the compilation process;
- **install, installation space**, where compiled executable files and scripts are placed;
- **log, log space**, during the compilation and running process, various warnings, errors, information and other logs are saved.

Generally speaking, most of the operations on the folders in these four spaces are performed in src. After the compilation is successful, the results in the install will be executed. The build and log folders are rarely used.

It should also be emphasized here that we can define the name of the workspace ourselves, and the number is not unique, for example:

```
Workspace 1: ros2_ws_a, used for the development of robot A
Workspace 1: ros2_ws_b, used for the development of B robot
Workspace 1: ros2_ws_c, used for the development of C robots
```

The above situation is completely allowed, just like we create multiple new projects in the integrated development environment, they all exist side by side.

4. Set environment variables

After successful compilation, in order for the system to find our function package and executable file, environment variables need to be set:

```
#Only takes effect in the current terminal
source install/setup.bash
# Valid on all terminals
echo "source ~/yahboomcar_ros2_ws/yahboomcar_ws/install/setup.bash" >> ~/.bashrc
```

At this point, we have completed the creation, compilation and configuration of the workspace.