

在安装好 ubuntu 系统之后，我们就可以安装 ros 了。首先配置安装 ros 需要的 key。用 `ctrl+alt+t` 打开一个终端，在终端中键入

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main"
> /etc/apt/sources.list.d/ros-latest.list'
```

再在终端中键入

```
sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --rec
v-key 421C365BD9FF1F717815A3895523BAEEB01FA116
```

此时可能会遇到无法连接到服务器的问题，可以把代码中的 `hkp://ha.pool.sks-keyservers.net:80` 网址换成 `hkp://pgp.mit.edu:80` 或 `hkp://keyserver.ubuntu.com:80`

接下来进行 ros 的安装。在终端中键入

```
sudo apt update
```

此时可能会报错如下

```
Could not get lock/var/lib/dpkg/lock - open
```

这是因为 apt 还在运行，我们可以通过键入

```
sudo rm /var/cache/apt/archives/lock
sudo rm /var/lib/dpkg/lock
```

来解决。在完成这一步后，我们接着键入

```
sudo apt install ros-melodic-desktop-full
```

来安装 ros。这一步的执行时间会比较长。执行完毕后，ros 就算安装完成了。在开始使用之前我们还需要对 ros 进行一些初始化及环境设定。

在终端中键入

```
sudo rosdep init
rosdep update
```

来对 ros 进行初始化。然后在终端中键入

```
echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

来进行初始环境设定。接下来我们还要安装一个使用 ros 时常用的包。在终端中键入

```
sudo apt install python-rosinstall python-rosinstall-generator python-
wstool build-essential
```

至此，我们就可以开始使用 ros 了。

使用 ros 时，我们首先要创建一个工作空间。在终端中键入如下代码

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

此时我们创建了一个 catkin_ws 工作空间，该空间下有一个 src 文件夹，一个 build 文件夹和一个 devel 文件夹。在这个工作空间中，catkin 包可以被编译。接下来我们创建一个 catkin 包。一个 catkin 的包必须包括一个 package.xml 文件，一个 CMakeLists.txt 文件，且每个文件夹下只有一包，包之间不允许嵌套。在终端中键入

```
cd ~/catkin_ws/src  
  
catkin_create_pkg beginner_tutorials std_msgs roscpp
```

这是一个名为 beginner_tutorials 的包。这样我们的第一个包就创建好了。

ROS 有以下几个概念需要理解

- Nodes:节点,一个节点即为一个可执行文件，它可以通过 ROS 与其它节点进行通信。
- Messages:消息，消息是一种 ROS 数据类型，用于订阅或发布到一个话题。
- Topics:话题,节点可以发布消息到话题，也可以订阅话题以接收消息。
- Master:节点管理器，ROS 名称服务（比如帮助节点找到彼此）。
- roscout: ROS 中相当于 stdout/stderr。
- roscore: 主机+ roscout + 参数服务器

我们在运行所有 ros 程序前需要最先运行 roscore。在终端中键入

```
roscout
```

室友 roscore 可以查看现在活跃的节点。打开一个新的终端，键入

```
roscout list
```

它会列出所以当前活跃的节点。此时应该只显示一个 roscout，因为这个节点用于收集和记录节点调试输出信息，总是在运行。

ROS 可以允许试用包名直接运行一个包内的节点。其格式如下

```
roscout [package_name] [node_name]
```

当节点之间有了通讯时，我们可以查看在某个话题上发布的数据，其格式如下

```
roscout echo [topic]
```

我们也可以查看所有发布和订阅的话题及其类型的详细信息。命令如下

```
roscout list -v
```

使用 roscout pub 可以把数据发送到某个正在运行的 topic 上，其格式如下

```
rostopic pub [topic] [msg_type] [args]
```