

5G智能物联网- 机器人案例课程

# 基于融合系统的机器人仿真实训

广和通大学计划项目组 2023Q2



# 目录

01

机器人方向参赛技术指引

02

Aidlux平台机器人开发环境配置

03

ROS机器人开发流程

04

Aidlux平台机器人仿真案例效果展示



01

# 机器人方向参赛技术指引

# 机器人方向参赛技术指引（需具备知识技能）

1 Linux基础

2 嵌入式基础

3 C/C++基础

4 ROS基础

5 电控基础



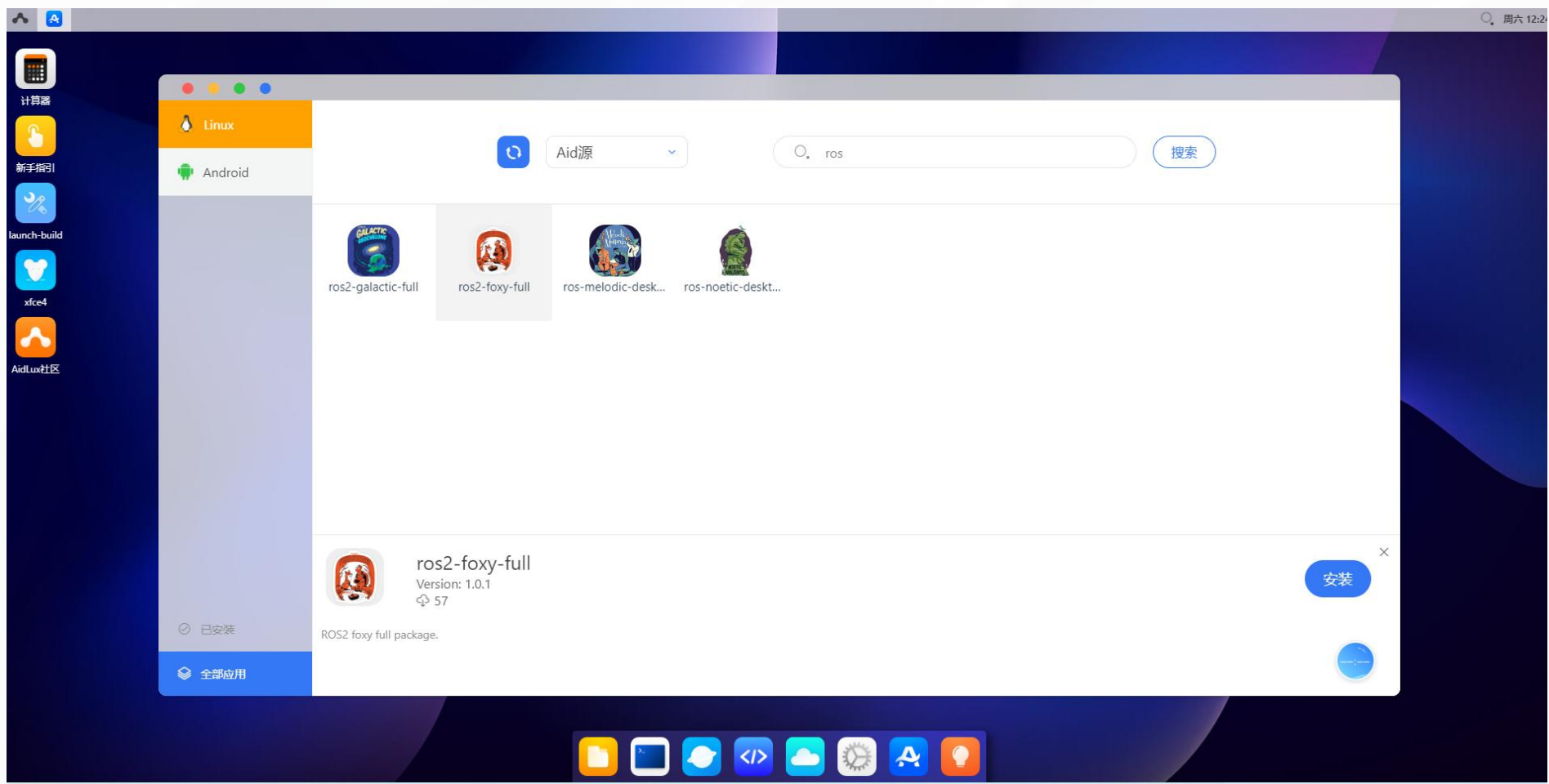
02

## Aidlux平台机器人开发环境配置

# Aidlux平台上机器人开发环境配置

- ROS/ROS2安装指南

推荐版本ros2 foxy，可以在AidLux应用中心下载；参赛者也可以自行安装其他版本ROS



# Aidlux平台上机器人开发环境配置

- gazebo仿真工具安装指南

```
apt-key adv --keyserver keyserver.ubuntu.com --recv-keys D2486D2DD83DB69272AFE98867170598AF249743
```

```
echo deb http://packages.osrfoundation.org/gazebo/debian\-stable buster main > /etc/apt/sources.list.d/gazebo-stable.list
```

```
curl -sSL https://ghproxy.com/https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -o /usr/share/keyrings/ros-archivekeyring.gpg
```

```
echo "deb [arch=arm64 signed-by=/usr/share/keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu buster main" | tee /etc/apt/sources.list.d/ros2.list > /dev/null
```

```
apt update
```

```
apt install -y python3-rosdep libgazebo11-dev libgraphicsmagick++1-dev libopencv-dev python3-netifaces python3-bson
```



03

## ROS机器人开发流程



# ROS机器人开发流程

- **第一步：建立工程**

在ROS开发中，这一步叫做创建工作空间，也就是保存后续开发工作涉及到所有文件的一个空间，在计算机中的体现其实就是文件夹，未来开发用到的文件都会保存在这个文件夹中。

- **第二步：在工作空间中创建写代码用的功能包**

类似我们在其他软件写代码时，会把同一功能的多个代码文件放置在一个文件夹中，这个文件夹在ROS中叫做功能包，每个功能包就是实现机器人某一功能的组织单元，多个功能包就组成了机器人应用的代码库。

- **第三步：最重要的一步，也就是写代码**

ROS开发常用的编码方式是C++和Python，无论我们使用哪种语言编写代码，都需要将代码放置在功能包中。

# ROS机器人开发流程

- **第四步：代码编译（C++）**

我们需要在功能包的CMakeLists.txt文件中设置如何编译代码的规则，python代码就不需要编译了，可以跳过这一步。

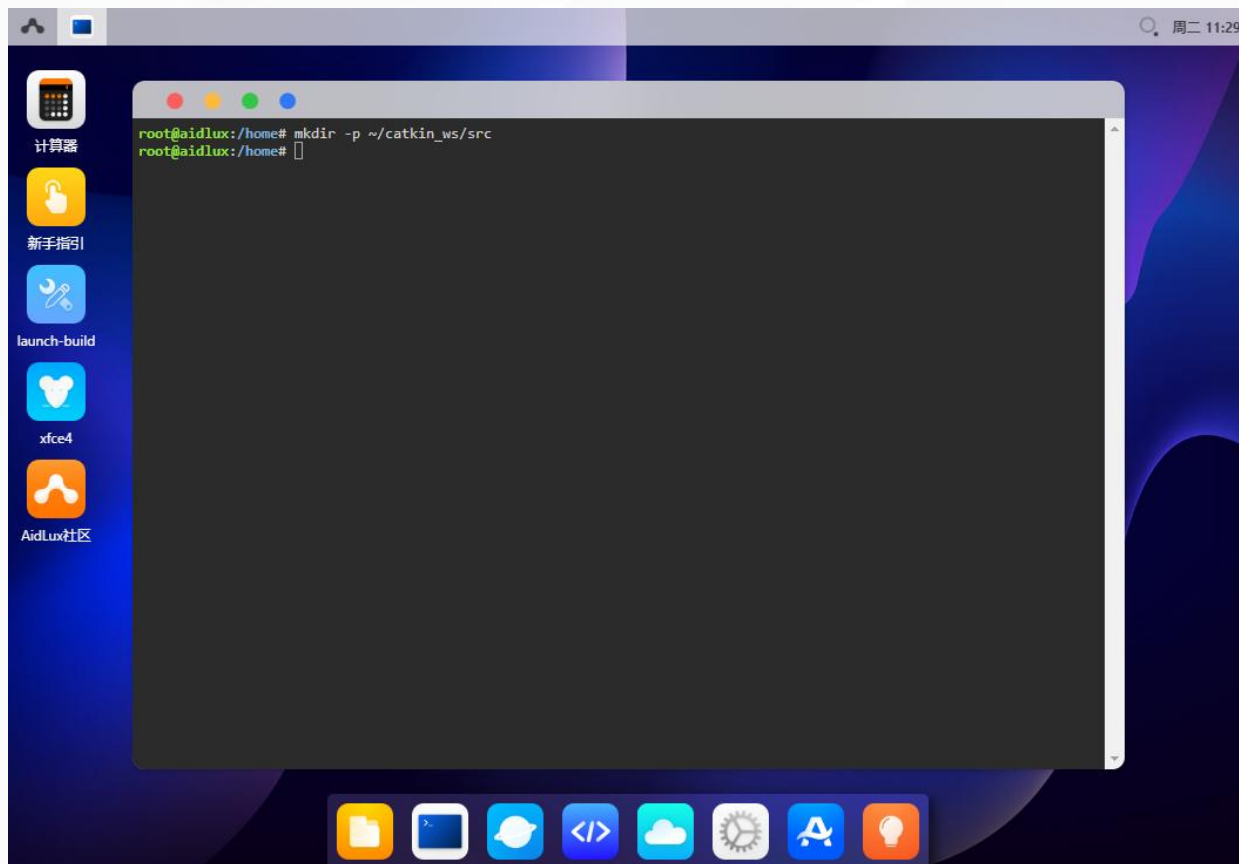
- **第五步：将工作空间打包为可执行文件**

我们把整个工程，也就是工作空间编译，将代码变成可执行文件，编译成功后就可以开心的运行了，假设运行出现问题，有可能就要回到代码编写的这一步，修改代码后再次编译运行，直到功能全部正常。

# 一、工作空间的创建和编译

- 创建工作空间:

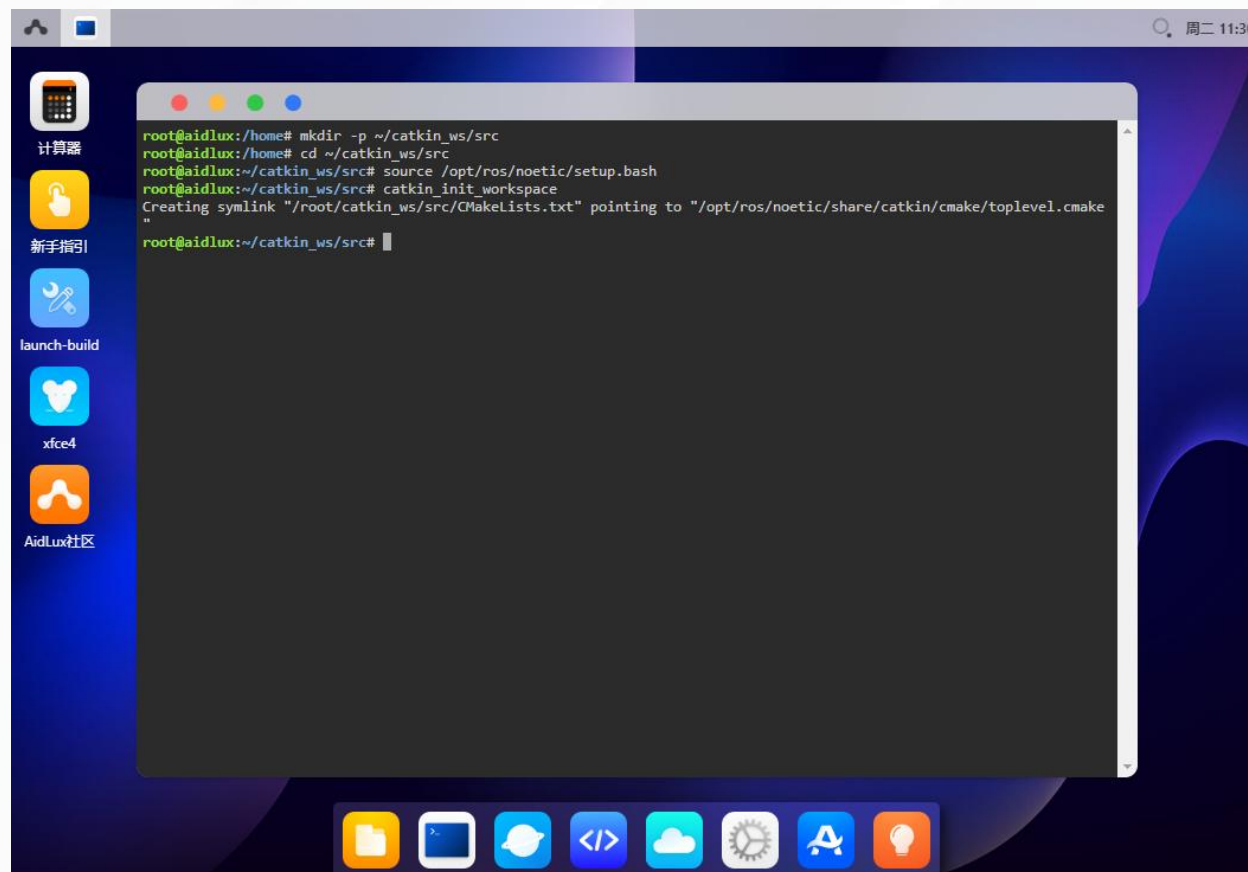
```
$ mkdir -p ~/catkin_ws/src
```



# 一、工作空间的创建和编译

- 将当前工作空间初始化为一个ROS里面的工作空间

```
$ catkin_init_workspace
```



A terminal window screenshot showing the process of initializing a ROS workspace. The terminal output is as follows:

```
root@aidlux:/home# mkdir -p ~/catkin_ws/src
root@aidlux:/home# cd ~/catkin_ws/src
root@aidlux:~/catkin_ws/src# source /opt/ros/noetic/setup.bash
root@aidlux:~/catkin_ws/src# catkin_init_workspace
Creating symlink "/root/catkin_ws/src/CMakeLists.txt" pointing to "/opt/ros/noetic/share/catkin/cmake/toplevel.cmake"
root@aidlux:~/catkin_ws/src#
```

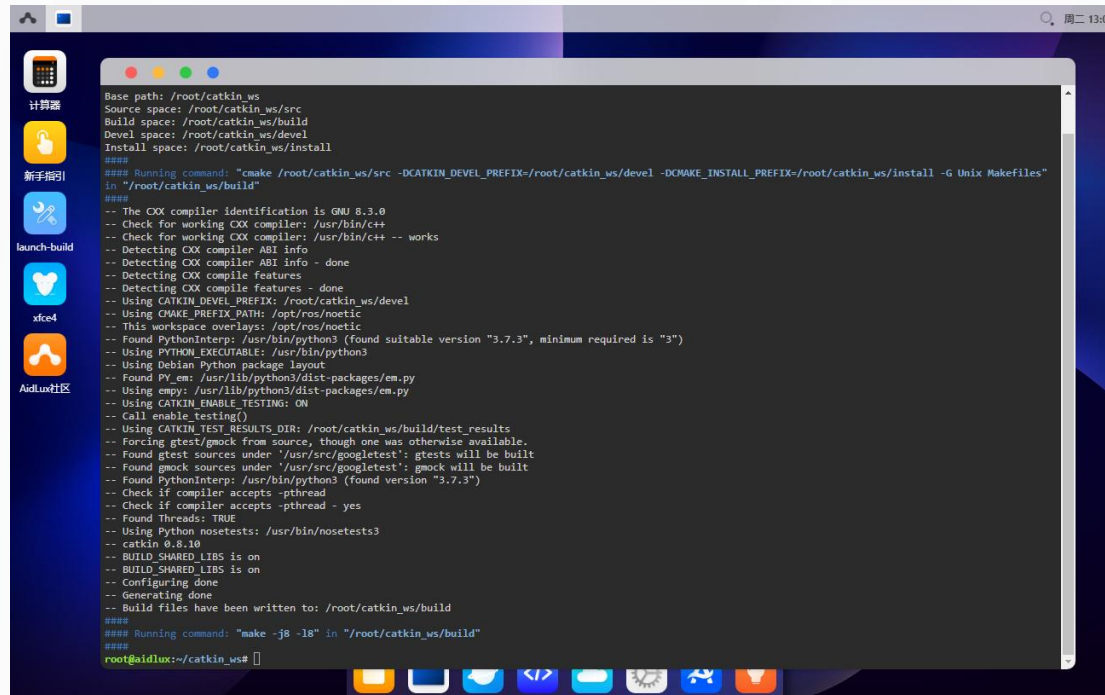
The terminal window is open in a desktop environment with a dark theme. The desktop background is dark blue with a purple gradient. On the left side, there is a dock with several application icons: a calculator, a hand icon, a wrench icon, a launch-build icon, an xfce4 icon, and an AidLux社区 icon. At the bottom, there is a taskbar with icons for a file manager, terminal, chat, code editor, cloud storage, settings, and a lightbulb icon. The top right corner of the terminal window shows the time as 周二 11:30.

# 一、工作空间的创建和编译

- 编译工作空间

我们的代码会放置在src里面，以后要编译代码就需要在工作空间的根目录下去进行编译，进入工作空间根目录后点击右键打开终端，输入如下命令进行编译空间。

```
$ catkin_make
```

A terminal window showing the output of the catkin\_make command. The terminal is titled '周二 13:06' and shows the following output:

```
Base path: /root/catkin_ws
Source space: /root/catkin_ws/src
Build space: /root/catkin_ws/build
Devel space: /root/catkin_ws/devel
Install space: /root/catkin_ws/install

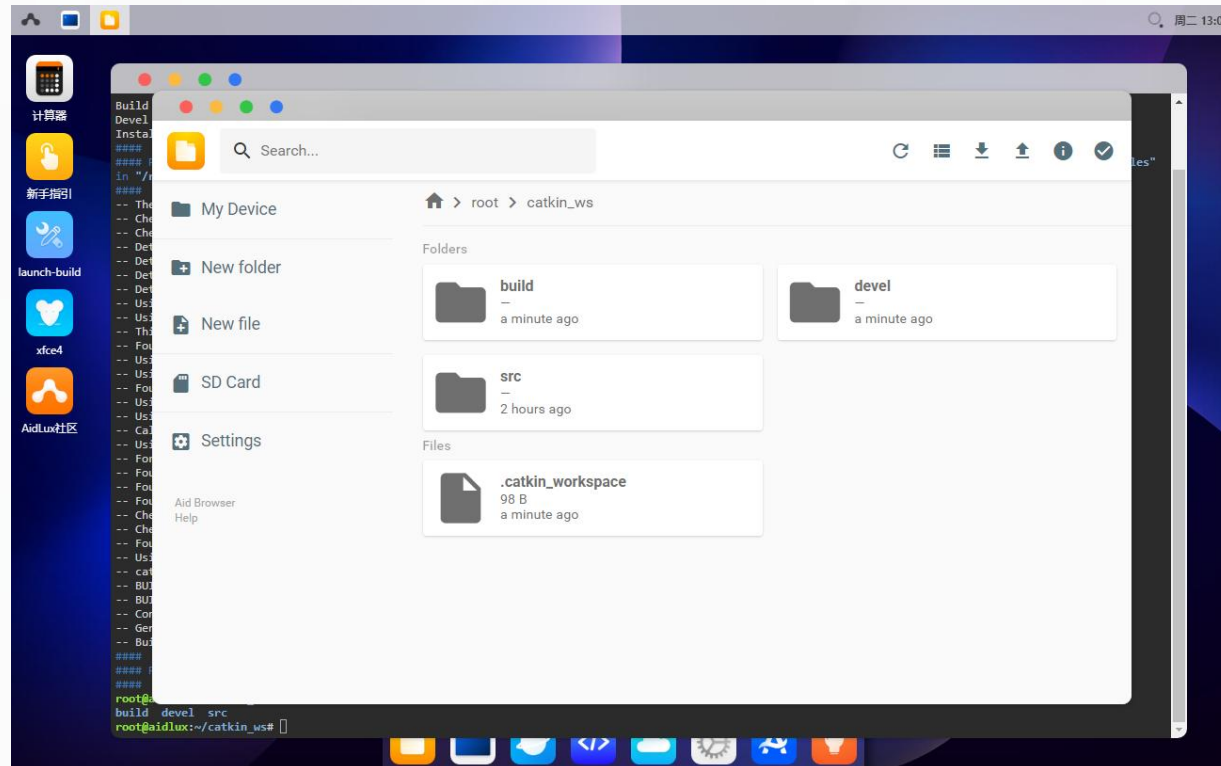
#### Running command: "cmake /root/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/root/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/root/catkin_ws/install -G Unix Makefiles"
in "/root/catkin_ws/build"
####
-- The CXX compiler identification is GNU 8.3.0
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /root/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.7.3", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Found PY_em: /usr/lib/python3/dist-packages/em.py
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /root/catkin_ws/build/test_results
-- Forcing gtest/gmock from source, though one was otherwise available.
-- Found gtest sources under '/usr/src/googletest': gtests will be built
-- Found gmock sources under '/usr/src/googletest': gmock will be built
-- Found PythonInterp: /usr/bin/python3 (found version "3.7.3")
-- Check if compiler accepts -pthread
-- Check if compiler accepts -pthread - yes
-- Found Threads: TRUE
-- Using Python nosetests: /usr/bin/nosetests3
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- Configuring done
-- Generating done
-- Build files have been written to: /root/catkin_ws/build

####
#### Running command: "make -j8 -l8" in "/root/catkin_ws/build"
####
root@idlux:~/catkin_ws#
```

# 一、工作空间的创建和编译

- 编译工作空间

编译完成后，会新生成两个文件夹，分别是build和devel文件夹，build是编译空间，主要用来放置编译的中间文件，devel是开发空间，主要来放置我们编译完成后的结果、程序的头文件、库等，另外src是代码空间，主要用来放置我们的源代码。

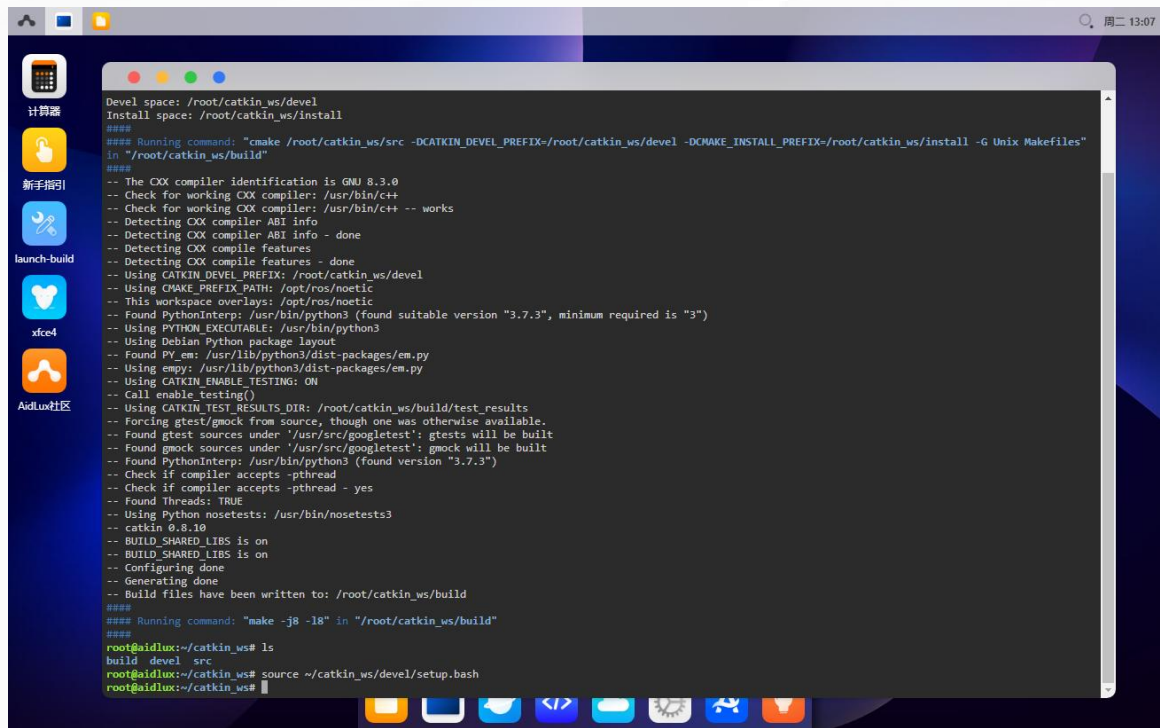


# 一、工作空间的创建和编译

- 设置环境变量

环境变量的设置也就是让Ubuntu知道当前功能包、开发内容的位置，打开终端，输入如下命令即可设置环境变量。

```
$ source ~/catkin_ws/devel/setup.bash
```



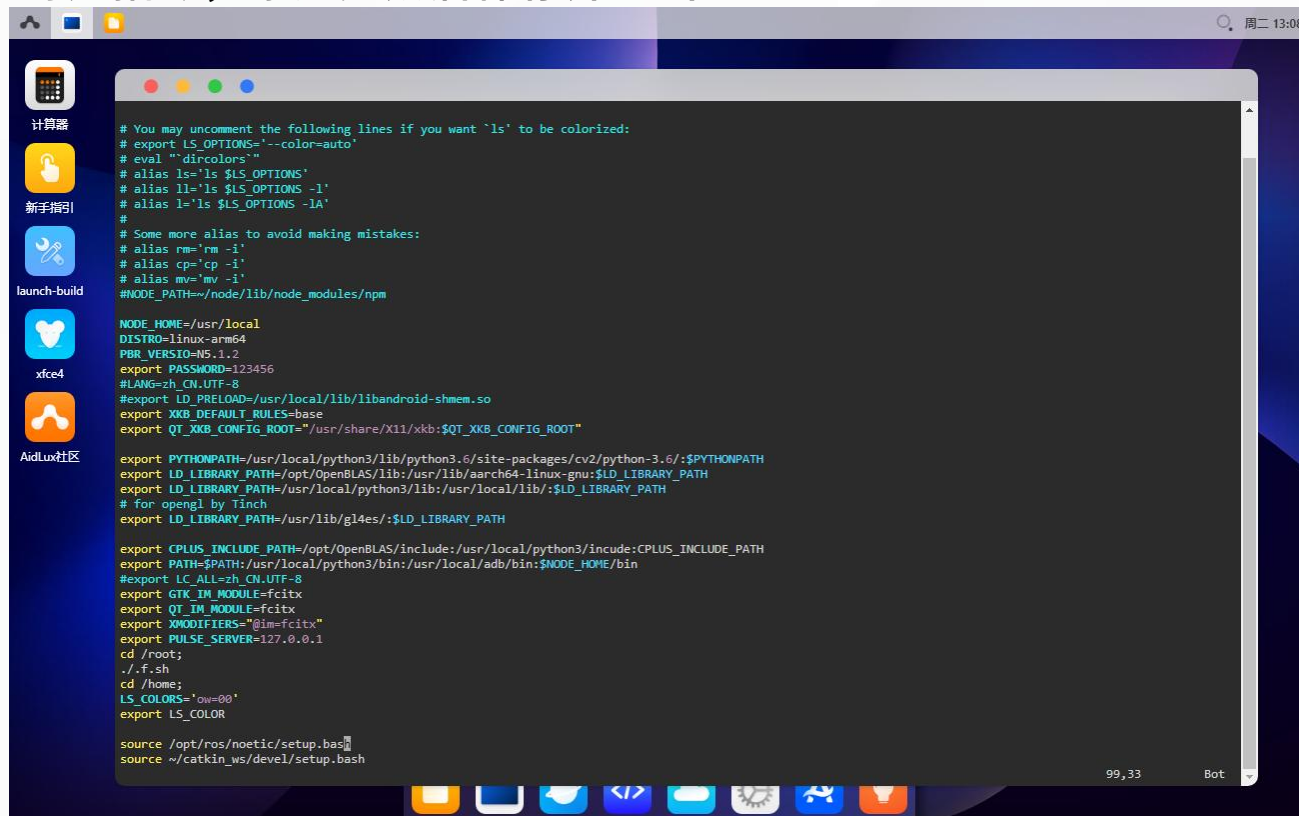
The image shows a terminal window in Ubuntu with a dark theme. The terminal output displays the process of setting up a ROS workspace. It starts with the command to source the setup.bash file, followed by the execution of 'catkin build' and 'catkin install'. The output shows various checks for compilers, Python versions, and other dependencies. The terminal ends with the prompt 'root@aidlux:~/catkin\_ws#', indicating that the workspace is ready for use.

```
Devel space: /root/catkin_ws/devel
Install space: /root/catkin_ws/install
####
#### Running command: "cmake /root/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/root/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/root/catkin_ws/install -G Unix Makefiles"
in "/root/catkin_ws/build"
####
-- The CXX compiler identification is GNU 8.3.0
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /root/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.7.3", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Found PY_em: /usr/lib/python3/dist-packages/em.py
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /root/catkin_ws/build/test_results
-- Forcing gtest/gmock from source, though one was otherwise available.
-- Found gtest sources under /usr/src/gtest: gtests will be built
-- Found gmock sources under /usr/src/gtest: gmock will be built
-- Found PythonInterp: /usr/bin/python3 (found version "3.7.3")
-- Check if compiler accepts -pthread
-- Check if compiler accepts -pthread - yes
-- Found Threads: TRUE
-- Using Python nosetests: /usr/bin/nosetests3
-- catkin 0.9.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- Configuring done
-- Generating done
-- Build files have been written to: /root/catkin_ws/build
####
#### Running command: "make -j8 -l8" in "/root/catkin_ws/build"
####
root@aidlux:~/catkin_ws# ls
build devel src
root@aidlux:~/catkin_ws# source ~/catkin_ws/devel/setup.bash
root@aidlux:~/catkin_ws#
```

# 一、工作空间的创建和编译

- 设置环境变量

注意source指令只在当前终端生效，如果终端关闭就没有效了，所以我们需要放置到bashrc里面，输入指令，打开bashrc，在该文件里把对应的工作空间、变量放置进来，在文件最后的位置加上对应指令，添加完成后保存并退出。



```
# You may uncomment the following lines if you want `ls` to be colorized:
# export LS_OPTIONS='--color=auto'
# eval "$(dircolors)"
# alias ls='ls $LS_OPTIONS'
# alias ll='ls $LS_OPTIONS -l'
# alias l='ls $LS_OPTIONS -lA'
#
# Some more alias to avoid making mistakes:
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'
#NODE_PATH=~/.node/lib/node_modules/npm

NODE_HOME=/usr/local
DISTRO=linux-arm64
PBR_VERSION=N5.1.2
export PASSWORD=123456
#LANG=zh_CN.UTF-8
#export LD_PRELOAD=/usr/local/lib/libandroid-shmem.so
export XKB_DEFAULT_RULES=base
export QT_XKB_CONFIG_ROOT="/usr/share/X11/xkb:$QT_XKB_CONFIG_ROOT"

export PYTHONPATH=/usr/local/python3/lib/python3.6/site-packages/cv2/python-3.6/:$PYTHONPATH
export LD_LIBRARY_PATH=/opt/OpenBLAS/lib:/usr/lib/aarch64-linux-gnu:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=/usr/local/python3/lib:/usr/local/lib:$LD_LIBRARY_PATH
# for opengl by Tinech
export LD_LIBRARY_PATH=/usr/lib/g14es:$LD_LIBRARY_PATH

export CPLUS_INCLUDE_PATH=/opt/OpenBLAS/include:/usr/local/python3/include:CPLUS_INCLUDE_PATH
export PATH=$PATH:/usr/local/python3/bin:/usr/local/adb/bin:$NODE_HOME/bin
#export LC_ALL=zh_CN.UTF-8
export GTK_IM_MODULE=fcitx
export QT_IM_MODULE=fcitx
export XMODIFIERS="@im=fcitx"
export PULSE_SERVER=127.0.0.1
cd /root;
./f.sh
cd /home;
LS_COLORS='ow=00'
export LS_COLOR

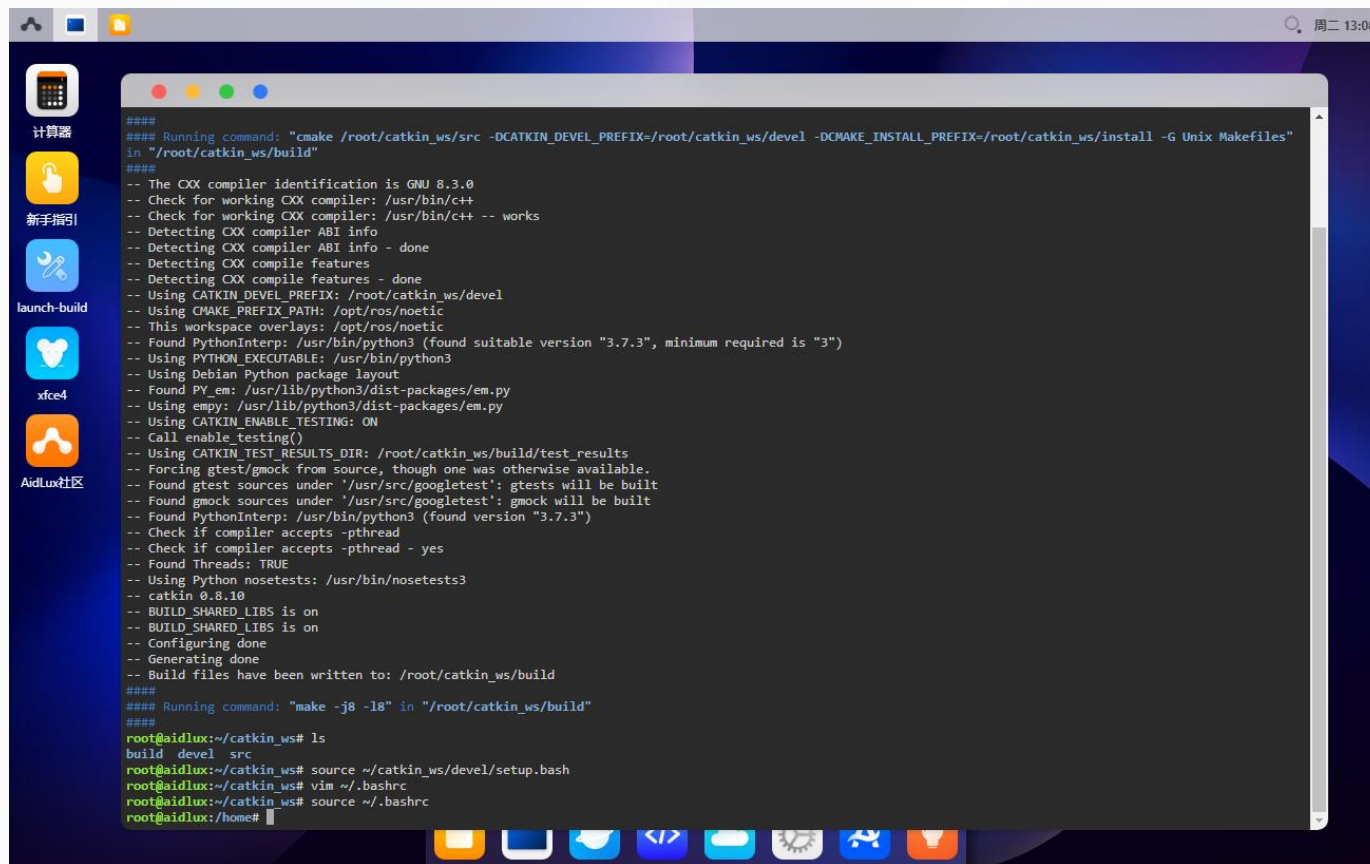
source /opt/ros/noetic/setup.bash
source ~/catkin_ws/devel/setup.bash
```



# 一、工作空间的创建和编译

- 设置环境变量

然后在终端里面进行source脚本，让脚本在当前终端里面立刻生效。或者关闭当前终端，重新打开终端也可以重新运行所有的脚本。



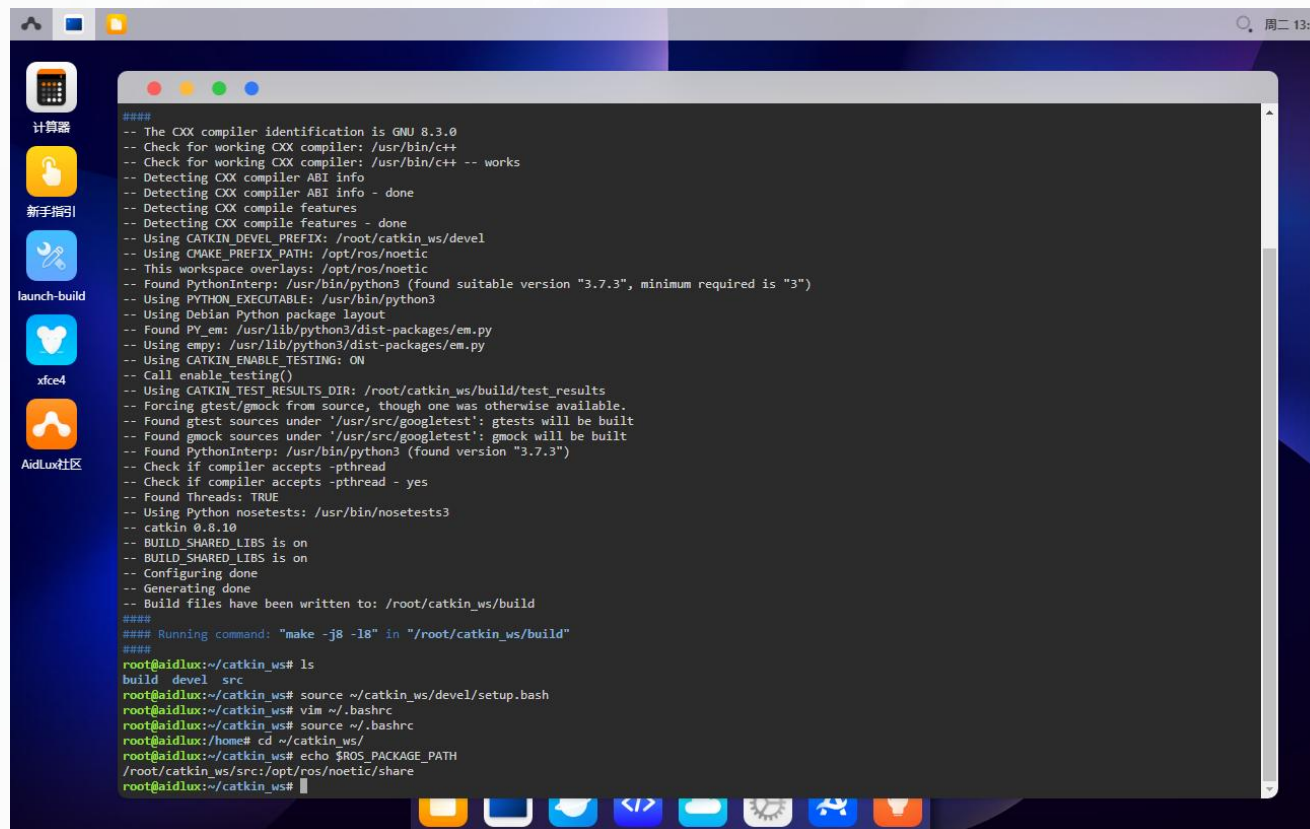
```
#####
##### Running command: "cmake /root/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/root/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/root/catkin_ws/install -G Unix Makefiles"
in "/root/catkin_ws/build"
#####
-- The CXX compiler identification is GNU 8.3.0
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /root/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.7.3", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Found PY_em: /usr/lib/python3/dist-packages/em.py
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /root/catkin_ws/build/test_results
-- Forcing gtest/gmock from source, though one was otherwise available.
-- Found gtest sources under /usr/src/googletest: gtests will be built
-- Found gmock sources under /usr/src/googletest: gmock will be built
-- Found PythonInterp: /usr/bin/python3 (found version "3.7.3")
-- Check if compiler accepts -pthread
-- Check if compiler accepts -pthread - yes
-- Found Threads: TRUE
-- Using Python nosetests: /usr/bin/nosetests3
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- Configuring done
-- Generating done
-- Build files have been written to: /root/catkin_ws/build
#####
##### Running command: "make -j8 -l8" in "/root/catkin_ws/build"
#####
root@aidlux:~/catkin_ws# ls
build  devel  src
root@aidlux:~/catkin_ws# source ~/catkin_ws/devel/setup.bash
root@aidlux:~/catkin_ws# vim ~/.bashrc
root@aidlux:~/catkin_ws# source ~/.bashrc
root@aidlux:~/catkin_ws# source ~/.bashrc
root@aidlux:~/home#
```

# 一、工作空间的创建和编译

- 验证环境变量

如果想要验证之前设置的环境变量有没有生效，就可以输入如下命令进行验证

```
$ echo $ROS_PACKAGE_PATH
```



The image shows a terminal window with a dark background and a light-colored text. The terminal output displays the results of a ROS workspace configuration process. It starts with a series of diagnostic messages for the CXX compiler (GNU 8.3.0) and Python (3.7.3). The workspace is configured with the following settings:

- CATKIN\_DEVEL\_PREFIX: /root/catkin\_ws/devel
- CMAKE\_PREFIX\_PATH: /opt/ros/noetic
- Workspace overlays: /opt/ros/noetic
- PythonInterp: /usr/bin/python3 (found suitable version "3.7.3", minimum required is "3")
- PYTHON\_EXECUTABLE: /usr/bin/python3
- Debian Python package layout
- PY\_em: /usr/lib/python3/dist-packages/em.py
- empy: /usr/lib/python3/dist-packages/em.py
- CATKIN\_ENABLE\_TESTING: ON
- enable\_testing() is called
- CATKIN\_TEST\_RESULTS\_DIR: /root/catkin\_ws/build/test\_results
- gtest/gmock sources are found and built
- PythonInterp: /usr/bin/python3 (found version "3.7.3")
- Compiler options: -pthread, -yes
- Threads: TRUE
- Python nosetests: /usr/bin/nosetests3
- catkin 0.8.10
- BUILD\_SHARED\_LIBS is on
- Configuring and generating done
- Build files are written to: /root/catkin\_ws/build

The terminal then shows the execution of the following commands:

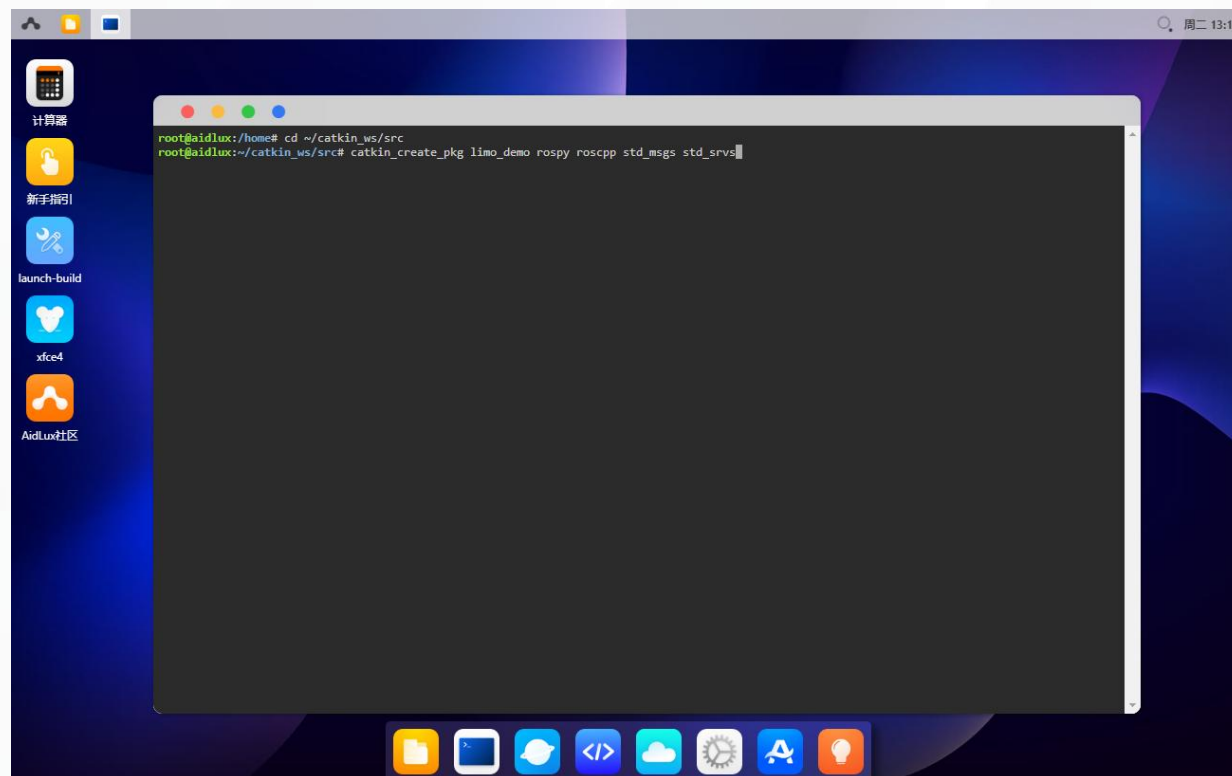
```
root@aIDLux:~/catkin_ws# ls
root@aIDLux:~/catkin_ws# build devel src
root@aIDLux:~/catkin_ws# source ~/catkin_ws/devel/setup.bash
root@aIDLux:~/catkin_ws# vim ~/.bashrc
root@aIDLux:~/catkin_ws# source ~/.bashrc
root@aIDLux:~/catkin_ws# cd ~/catkin_ws/
root@aIDLux:~/catkin_ws# echo $ROS_PACKAGE_PATH
/root/catkin_ws/src:/opt/ros/noetic/share
root@aIDLux:~/catkin_ws#
```

## 二、创建功能包

- 创建功能包

先进入到src里面，右键打开终端，输入如下命令来创建功能包

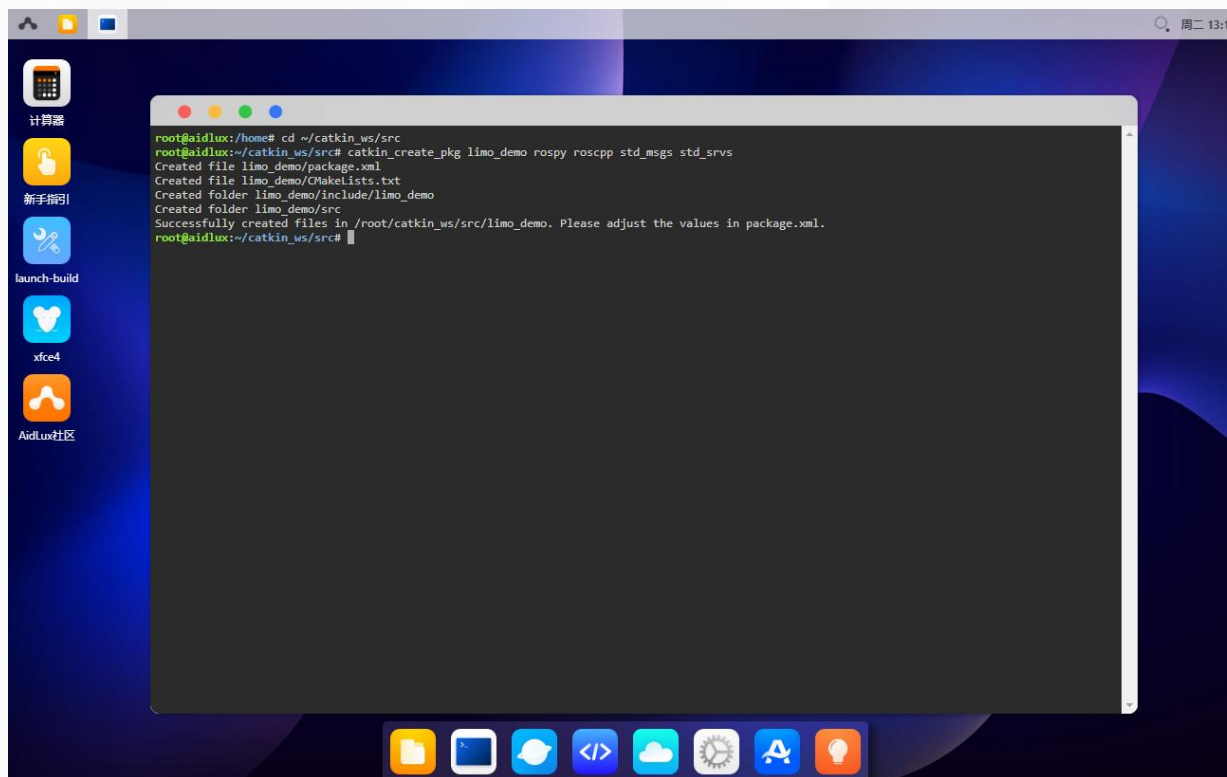
```
$ catkin_create_pkg limo_demo rospy roscpp std_msgs std_srvs
```



## 二、创建功能包

- 创建功能包

创建完成后，会显示很多日志信息，表明我们已经创建了对应的功能包，可以看到新生成了几个文件夹，其中CMakeList.txt文件是用来后续设定一些编译规则的文件，package.xml是用来编写功能包相关的描述内容。后续我们的代码开发也可以放置到src文件夹下面



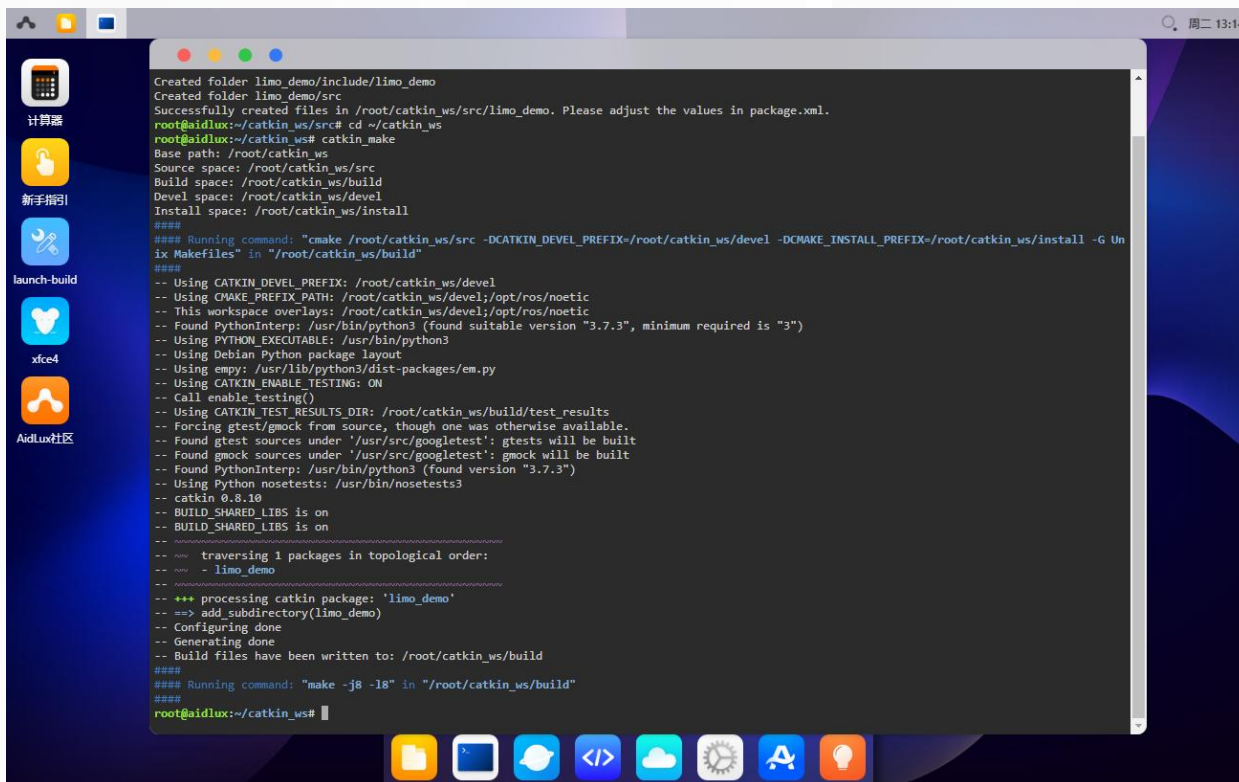
```
root@aidlux:~/home# cd ~/catkin_ws/src
root@aidlux:~/catkin_ws/src# catkin_create_pkg limo_demo roscpp std_msgs std_srvs
Created file limo_demo/package.xml
Created file limo_demo/CMakeLists.txt
Created folder limo_demo/include/limo_demo
Created folder limo_demo/src
Successfully created files in /root/catkin_ws/src/limo_demo. Please adjust the values in package.xml.
root@aidlux:~/catkin_ws/src#
```

## 二、创建功能包

- 编译功能包

功能包创建完成后，我们就可以回到根目录进行编译功能包，打开终端，输入如下命令进行编译功能包。

```
$ catkin_make
```



```
Created folder limo_demo/include/limo_demo
Created folder limo_demo/src
Successfully created files in /root/catkin_ws/src/limo_demo. Please adjust the values in package.xml.
root@aidlux:~/catkin_ws/src# cd ~/catkin_ws
root@aidlux:~/catkin_ws# catkin_make
Base path: /root/catkin_ws
Source space: /root/catkin_ws/src
Build space: /root/catkin_ws/build
Devel space: /root/catkin_ws/devel
Install space: /root/catkin_ws/install
####
#### Running command: "cmake /root/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/root/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/root/catkin_ws/install -G Unix Makefiles" in "/root/catkin_ws/build"
####
-- Using CATKIN_DEVEL_PREFIX: /root/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /root/catkin_ws/devel;/opt/ros/noetic
-- This workspace overlays: /root/catkin_ws/devel;/opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.7.3", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /root/catkin_ws/build/test_results
-- Forcing gtest/gmock from source, though one was otherwise available.
-- Found gtest sources under '/usr/src/googletest': gtests will be built
-- Found gmock sources under '/usr/src/googletest': gmock will be built
-- Found PythonInterp: /usr/bin/python3 (found version "3.7.3")
-- Using Python nosetests: /usr/bin/nosetests3
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-----
-- ~~~ traversing 1 packages in topological order:
-- ~~~ - limo_demo
-- ~~~
-- +++ processing catkin package: 'limo_demo'
-- ==> add_subdirectory(limo_demo)
-- Configuring done
-- Generating done
-- Build files have been written to: /root/catkin_ws/build
####
#### Running command: "make -j8 -l8" in "/root/catkin_ws/build"
####
root@aidlux:~/catkin_ws#
```

## 二、创建功能包

- 编译功能包

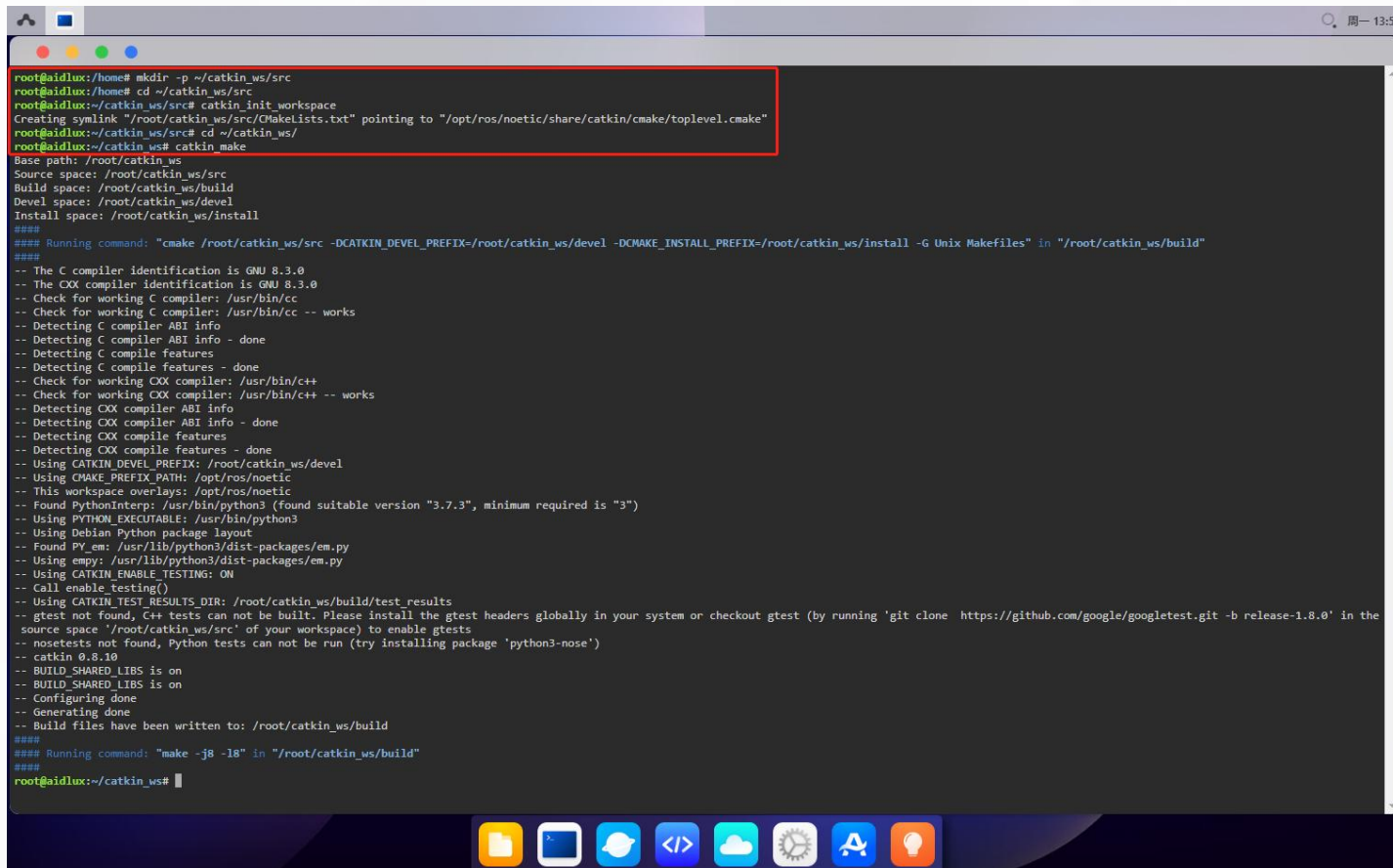
ROS的工作空间中，功能包不仅不能重名，还不能嵌套，比如把一个功能包放在另外一个功能包中，编译就会报错，但是文件夹是可以嵌套的，比如把一些同样类别的功能包放在一个文件夹中，是允许的。

那如何判断一个文件夹到底是普通文件夹，还是功能包呢，就是看它有没有上面我们介绍到的CMakeList.txt和package.xml两个文件，如果有的话，就是功能包，没有的话，就是文件夹，大家需要记住这一点。

## 五、工作空间编译与运行程序

- 编译工作空间

首先按照上面的步骤，创建对应的工作空间，并编译。如图中红框命令所示：

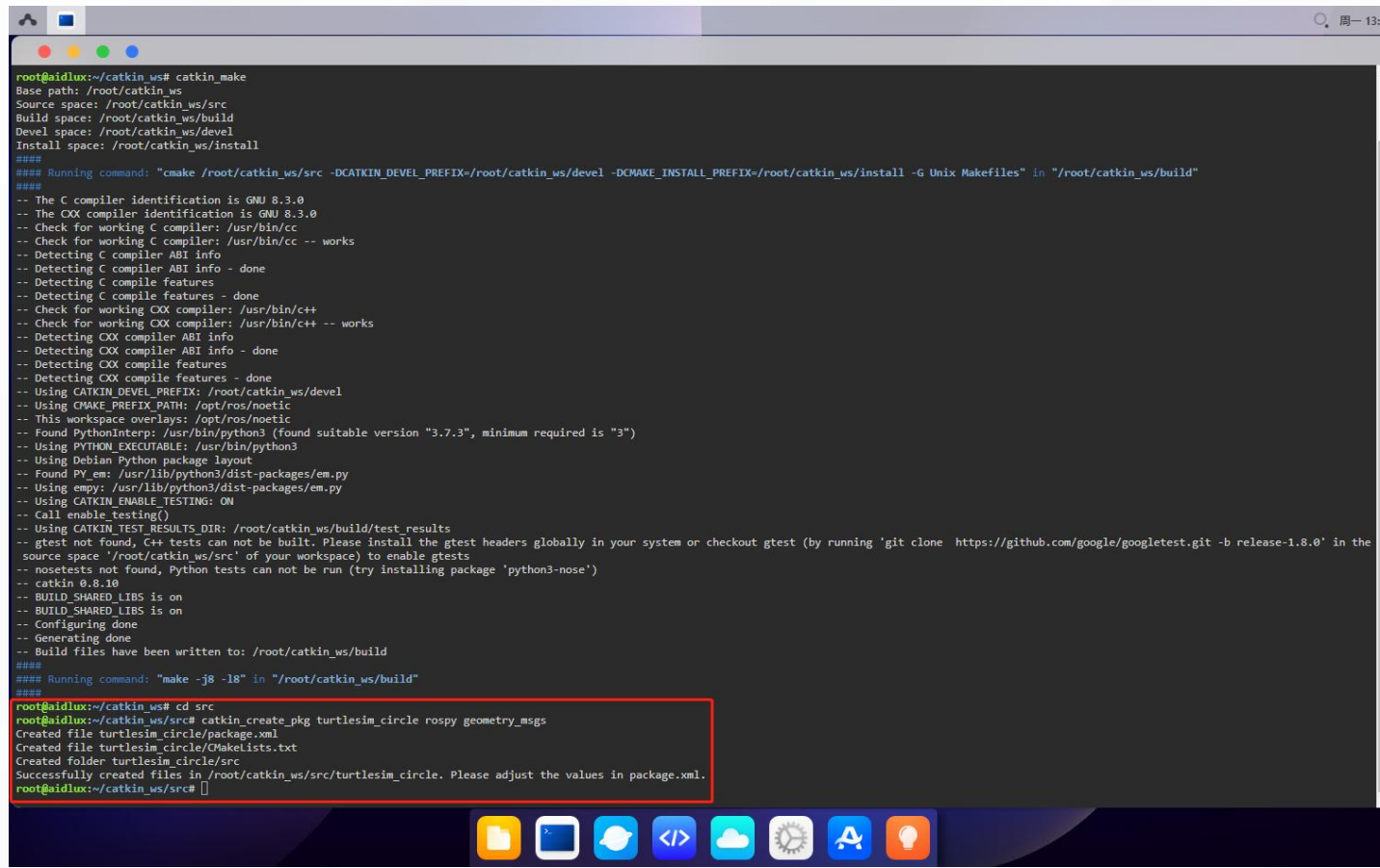


```
root@aidlux:/home# mkdir -p ~/catkin_ws/src
root@aidlux:/home# cd ~/catkin_ws/src
root@aidlux:~/catkin_ws/src# catkin_init_workspace
Creating symlink "/root/catkin_ws/src/MakeLists.txt" pointing to "/opt/ros/noetic/share/catkin/cmake/toplevel.cmake"
root@aidlux:~/catkin_ws/src# cd ~/catkin_ws/
root@aidlux:~/catkin_ws# catkin_make
Base path: /root/catkin_ws
Source space: /root/catkin_ws/src
Build space: /root/catkin_ws/build
Devel space: /root/catkin_ws/devel
Install space: /root/catkin_ws/install
####
#### Running command: "cmake /root/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/root/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/root/catkin_ws/install -G Unix Makefiles" in "/root/catkin_ws/build"
####
-- The C compiler identification is GNU 8.3.0
-- The CXX compiler identification is GNU 8.3.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /root/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.7.3", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Found PY_em: /usr/lib/python3/dist-packages/em.py
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /root/catkin_ws/build/test_results
-- gtest not found, C++ tests can not be built. Please install the gtest headers globally in your system or checkout gtest (by running 'git clone https://github.com/google/googletest.git -b release-1.8.0' in the source space '/root/catkin_ws/src' of your workspace) to enable gtests
-- nosetests not found, Python tests can not be run (try installing package 'python3-nose')
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- Configuring done
-- Generating done
-- Build files have been written to: /root/catkin_ws/build
####
#### Running command: "make -j8 -l8" in "/root/catkin_ws/build"
####
root@aidlux:~/catkin_ws#
```

# 五、工作空间编译与运行程序

- 运行程序

本次实例中，我们编写一个控制小乌龟走圆圈的控制模块，因此在工作空间中创建功能包：



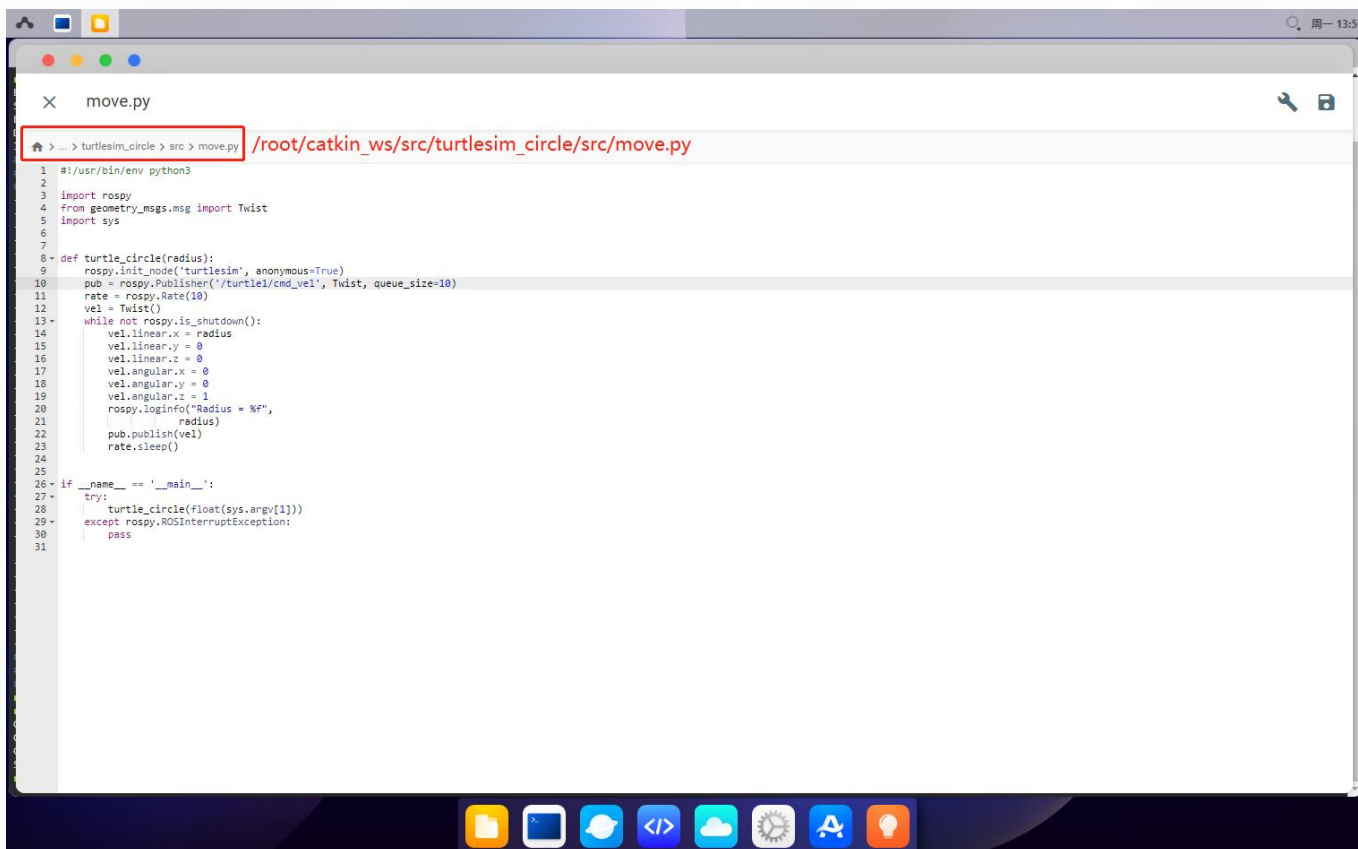
```
root@aidlux:~/catkin_ws# catkin_make
Base path: /root/catkin_ws
Source space: /root/catkin_ws/src
Build space: /root/catkin_ws/build
Devel space: /root/catkin_ws/devel
Install space: /root/catkin_ws/install
####
#### Running command: "cmake /root/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/root/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/root/catkin_ws/install -G Unix Makefiles" in "/root/catkin_ws/build"
####
-- The C compiler identification is GNU 8.3.0
-- The CXX compiler identification is GNU 8.3.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /root/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.7.3", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Found PY_em: /usr/lib/python3/dist-packages/em.py
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /root/catkin_ws/build/test_results
-- gtest not found, C++ tests can not be built. Please install the gtest headers globally in your system or checkout gtest (by running 'git clone https://github.com/google/googletest.git -b release-1.8.0' in the source space '/root/catkin_ws/src' of your workspace) to enable gtests
-- noretsts not found, Python tests can not be run (try installing package 'python3-nose')
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- Configuring done
-- Generating done
-- Build files have been written to: /root/catkin_ws/build
####
#### Running command: "make -j8 -l8" in "/root/catkin_ws/build"
####
root@aidlux:~/catkin_ws# cd src
root@aidlux:~/catkin_ws/src# catkin_create_pkg turtlesim_circle rospy geometry_msgs
Created file turtlesim_circle/package.xml
Created file turtlesim_circle/CMakeLists.txt
Created folder turtlesim_circle/src
Successfully created files in /root/catkin_ws/src/turtlesim_circle. Please adjust the values in package.xml.
root@aidlux:~/catkin_ws/src#
```



## 五、工作空间编译与运行程序

- 运行程序

当功能包创建完成后，进入功能包的源码目录中，编写源代码，具体目录路径参考图中红字所示：

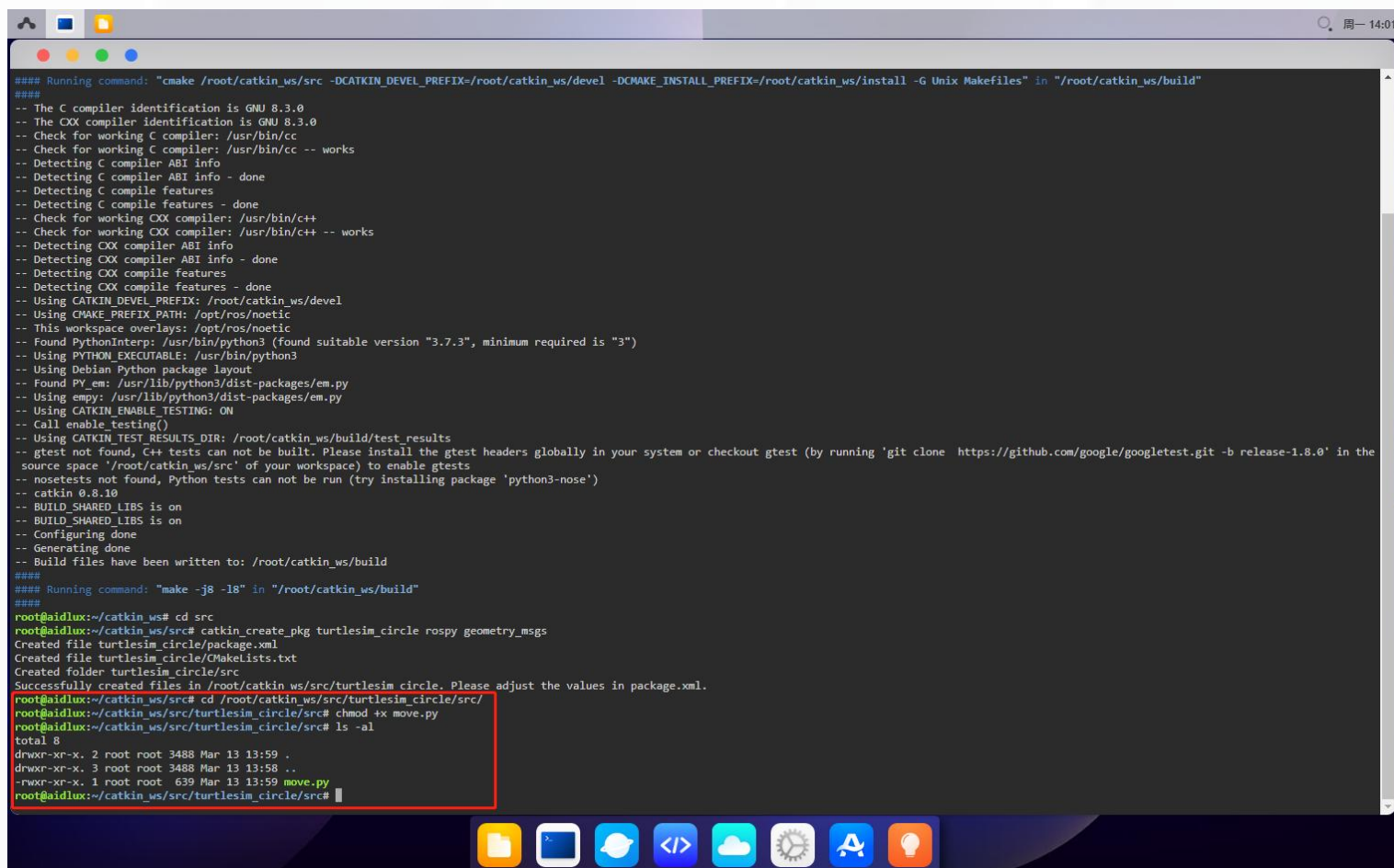


```
1 #!/usr/bin/env python3
2
3 import rospy
4 from geometry_msgs.msg import Twist
5 import sys
6
7
8 def turtle_circle(radius):
9     rospy.init_node('turtlesim', anonymous=True)
10    pub = rospy.Publisher('/turtle1/cmd_vel', Twist, queue_size=10)
11    rate = rospy.Rate(10)
12    vel = Twist()
13    while not rospy.is_shutdown():
14        vel.linear.x = radius
15        vel.linear.y = 0
16        vel.linear.z = 0
17        vel.angular.x = 0
18        vel.angular.y = 0
19        vel.angular.z = 1
20        rospy.loginfo("Radius = %f",
21                    radius)
22        pub.publish(vel)
23        rate.sleep()
24
25
26 if __name__ == '__main__':
27     try:
28         turtle_circle(float(sys.argv[1]))
29     except rospy.ROSInterruptException:
30         pass
31
```

# 五、工作空间编译与运行程序

- 运行程序

编写完成后，为这个功能包的源码授予可执行权限：

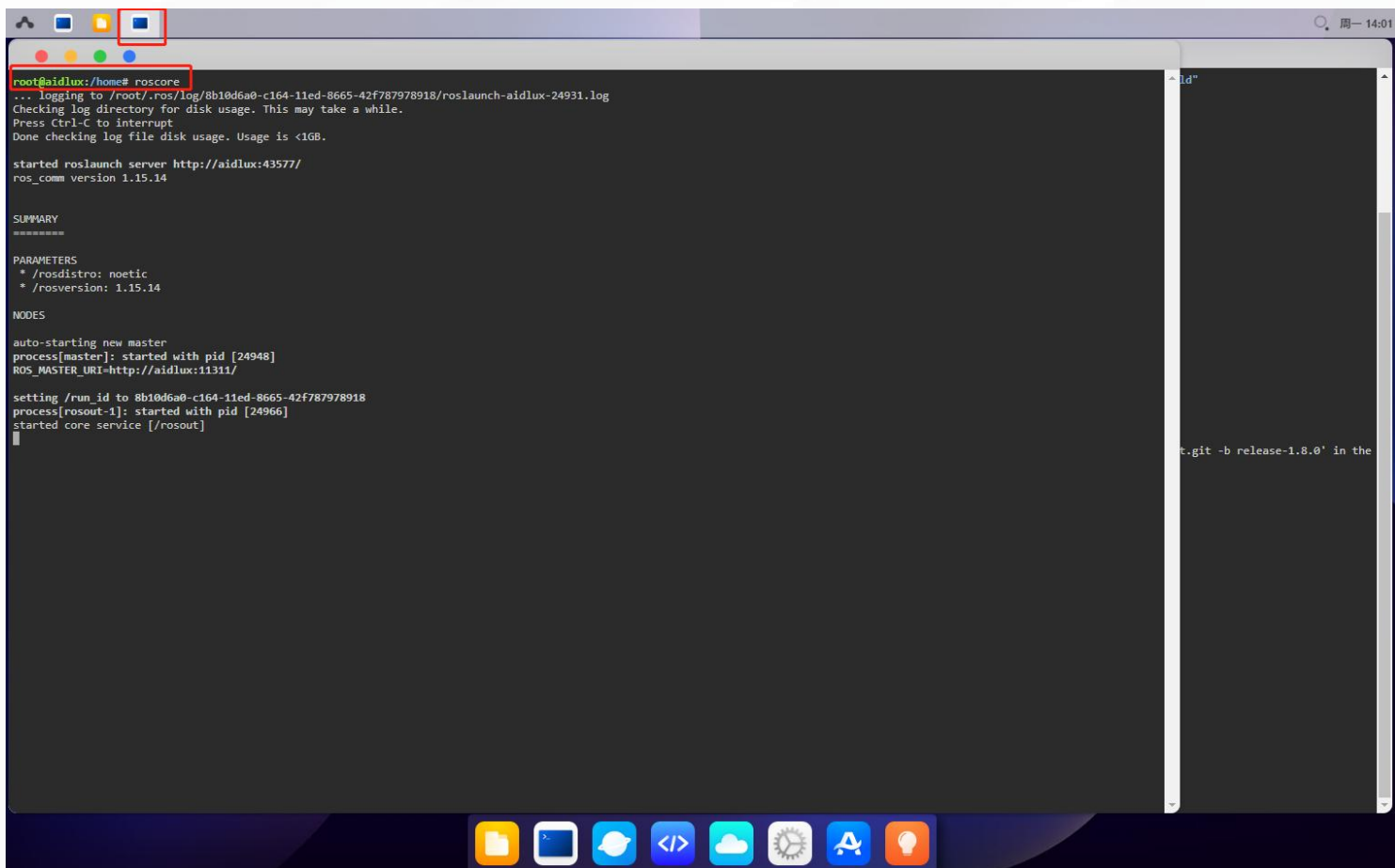


```
### Running command: "cmake /root/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/root/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/root/catkin_ws/install -G Unix Makefiles" in "/root/catkin_ws/build"
###
-- The C compiler identification is GNU 8.3.0
-- The CXX compiler identification is GNU 8.3.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- Using CATKIN_DEVEL_PREFIX: /root/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.7.3", minimum required is "3")
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Found PY_em: /usr/lib/python3/dist-packages/em.py
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /root/catkin_ws/build/test_results
-- gtest not found, C++ tests can not be built. Please install the gtest headers globally in your system or checkout gtest (by running 'git clone https://github.com/google/googletest.git -b release-1.8.0' in the
source space "/root/catkin_ws/src" of your workspace) to enable gtests
-- nosetests not found, Python tests can not be run (try installing package 'python3-nose')
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- Configuring done
-- Generating done
-- Build files have been written to: /root/catkin_ws/build
###
### Running command: "make -j8 -l8" in "/root/catkin_ws/build"
###
root@aidlux:~/catkin_ws# cd src
root@aidlux:~/catkin_ws/src# catkin_create_pkg turtlesim_circle rospy geometry_msgs
Created file turtlesim_circle/package.xml
Created file turtlesim_circle/CMakeLists.txt
Created folder turtlesim_circle/src
Successfully created files in /root/catkin_ws/src/turtlesim_circle. Please adjust the values in package.xml.
root@aidlux:~/catkin_ws/src# cd /root/catkin_ws/src/turtlesim_circle/src/
root@aidlux:~/catkin_ws/src/turtlesim_circle/src# chmod +x move.py
root@aidlux:~/catkin_ws/src/turtlesim_circle/src# ls -al
total 8
drwxr-xr-x. 2 root root 3488 Mar 13 13:59 .
drwxr-xr-x. 3 root root 3488 Mar 13 13:58 ..
-rwxr-xr-x. 1 root root 639 Mar 13 13:59 move.py
root@aidlux:~/catkin_ws/src/turtlesim_circle/src#
```

## 五、工作空间编译与运行程序

- 运行程序

然后可以启动ros进行测试，先启动roscore：



```
root@aidlux:/home# roscore
... logging to /root/.ros/log/8b10d6a0-c164-11ed-8665-42f787978918/roslaunch-aidlux-24931.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://aidlux:43577/
ros_comm version 1.15.14

SUMMARY
=====
PARAMETERS
 * /roscore: noetic
 * /rosversion: 1.15.14

NODES

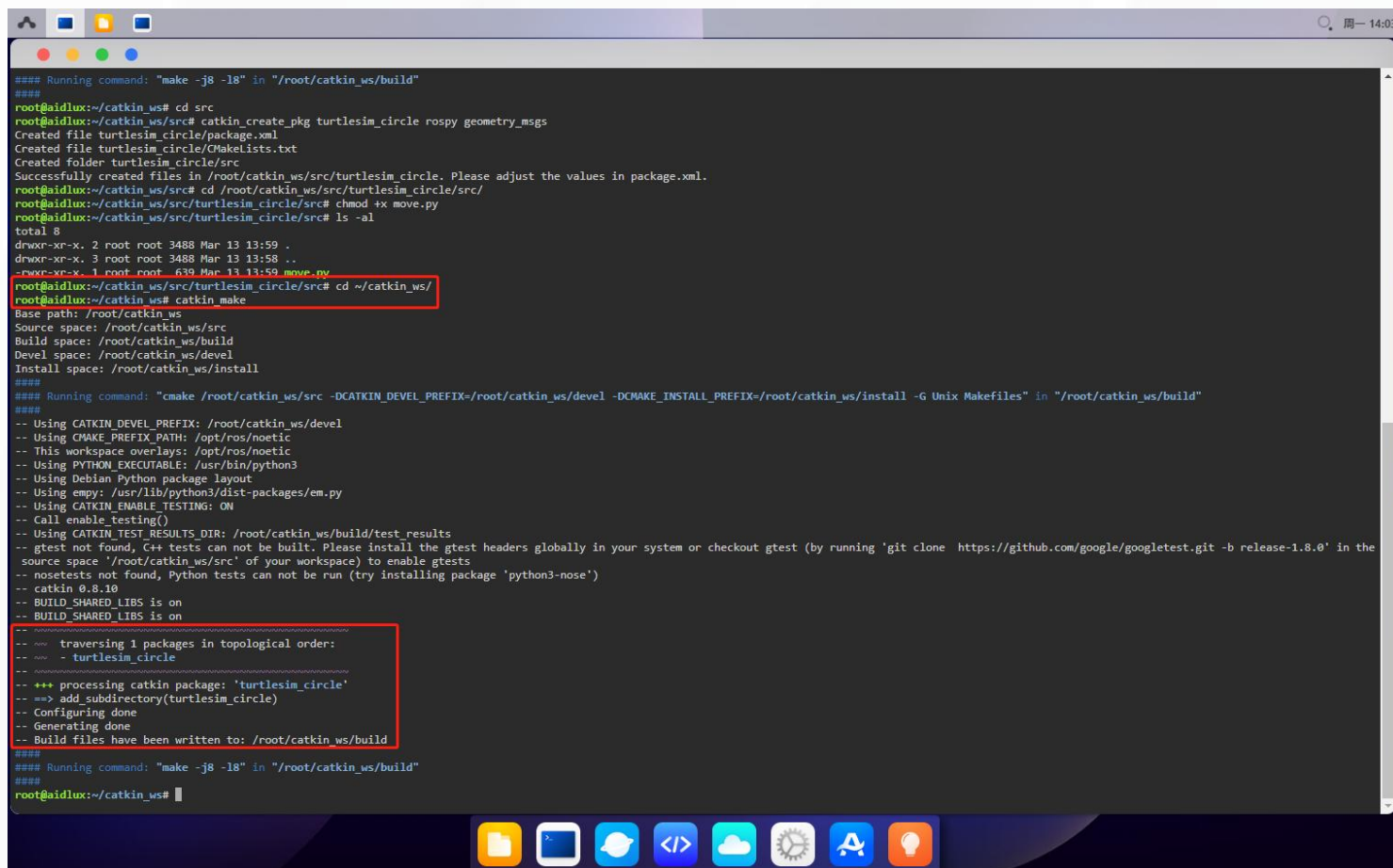
auto-starting new master
process[master]: started with pid [24948]
ROS_MASTER_URI=http://aidlux:11311/

setting /run_id to 8b10d6a0-c164-11ed-8665-42f787978918
process[rosout-1]: started with pid [24966]
started core service [/rosout]
```

# 五、工作空间编译与运行程序

- 运行程序

然后编译功能包：

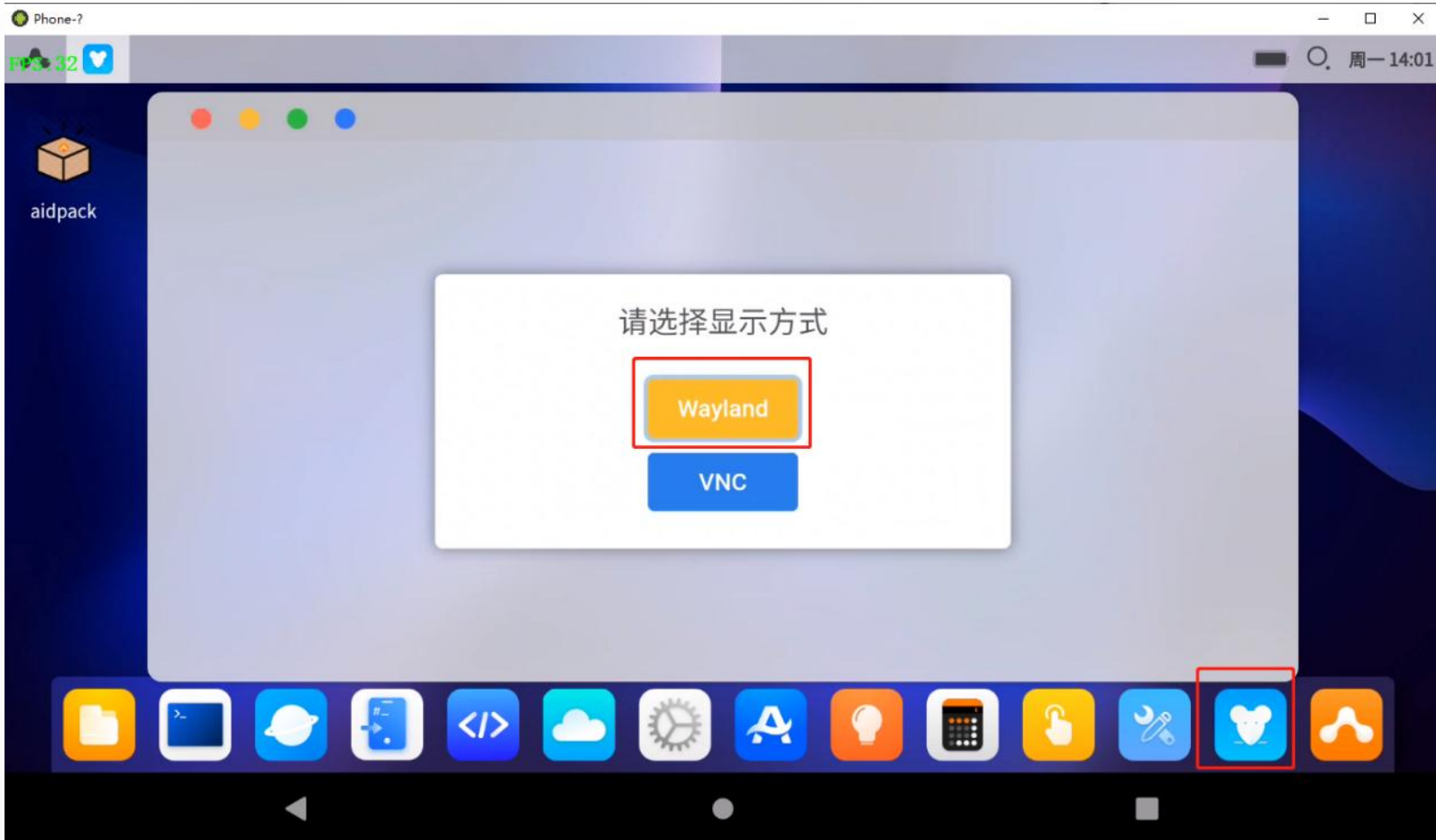


```
### Running command: "make -j8 -l8" in "/root/catkin_ws/build"
###
root@aidlux:~/catkin_ws# cd src
root@aidlux:~/catkin_ws/src# catkin_create_pkg turtlesim_circle rospy geometry_msgs
Created file turtlesim_circle/package.xml
Created file turtlesim_circle/CMakeLists.txt
Created folder turtlesim_circle/src
Successfully created files in /root/catkin_ws/src/turtlesim_circle. Please adjust the values in package.xml.
root@aidlux:~/catkin_ws/src# cd /root/catkin_ws/src/turtlesim_circle/src/
root@aidlux:~/catkin_ws/src/turtlesim_circle/src# chmod +x move.py
root@aidlux:~/catkin_ws/src/turtlesim_circle/src# ls -al
total 8
drwxr-xr-x. 2 root root 3488 Mar 13 13:59 .
drwxr-xr-x. 3 root root 3488 Mar 13 13:58 ..
-rwxr-xr-x. 1 root root 639 Mar 13 13:59 move.py
root@aidlux:~/catkin_ws/src/turtlesim_circle/src# cd ~/catkin_ws/
root@aidlux:~/catkin_ws# catkin_make
Base path: /root/catkin_ws
Source space: /root/catkin_ws/src
Build space: /root/catkin_ws/build
Devel space: /root/catkin_ws/devel
Install space: /root/catkin_ws/install
###
### Running command: "cmake /root/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/root/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/root/catkin_ws/install -G Unix Makefiles" in "/root/catkin_ws/build"
###
-- Using CATKIN_DEVEL_PREFIX: /root/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /root/catkin_ws/build/test_results
-- gtest not found, C++ tests can not be built. Please install the gtest headers globally in your system or checkout gtest (by running 'git clone https://github.com/google/googletest.git -b release-1.8.0' in the source space '/root/catkin_ws/src' of your workspace) to enable gtests
-- nosetests not found, Python tests can not be run (try installing package 'python3-nose')
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-----
-- traversing 1 packages in topological order:
-- - turtlesim_circle
-----
-- +++ processing catkin package: 'turtlesim_circle'
-- ==> add_subdirectory(turtlesim_circle)
-- Configuring done
-- Generating done
-- Build files have been written to: /root/catkin_ws/build
###
### Running command: "make -j8 -l8" in "/root/catkin_ws/build"
###
root@aidlux:~/catkin_ws#
```

## 五、工作空间编译与运行程序

- 运行程序

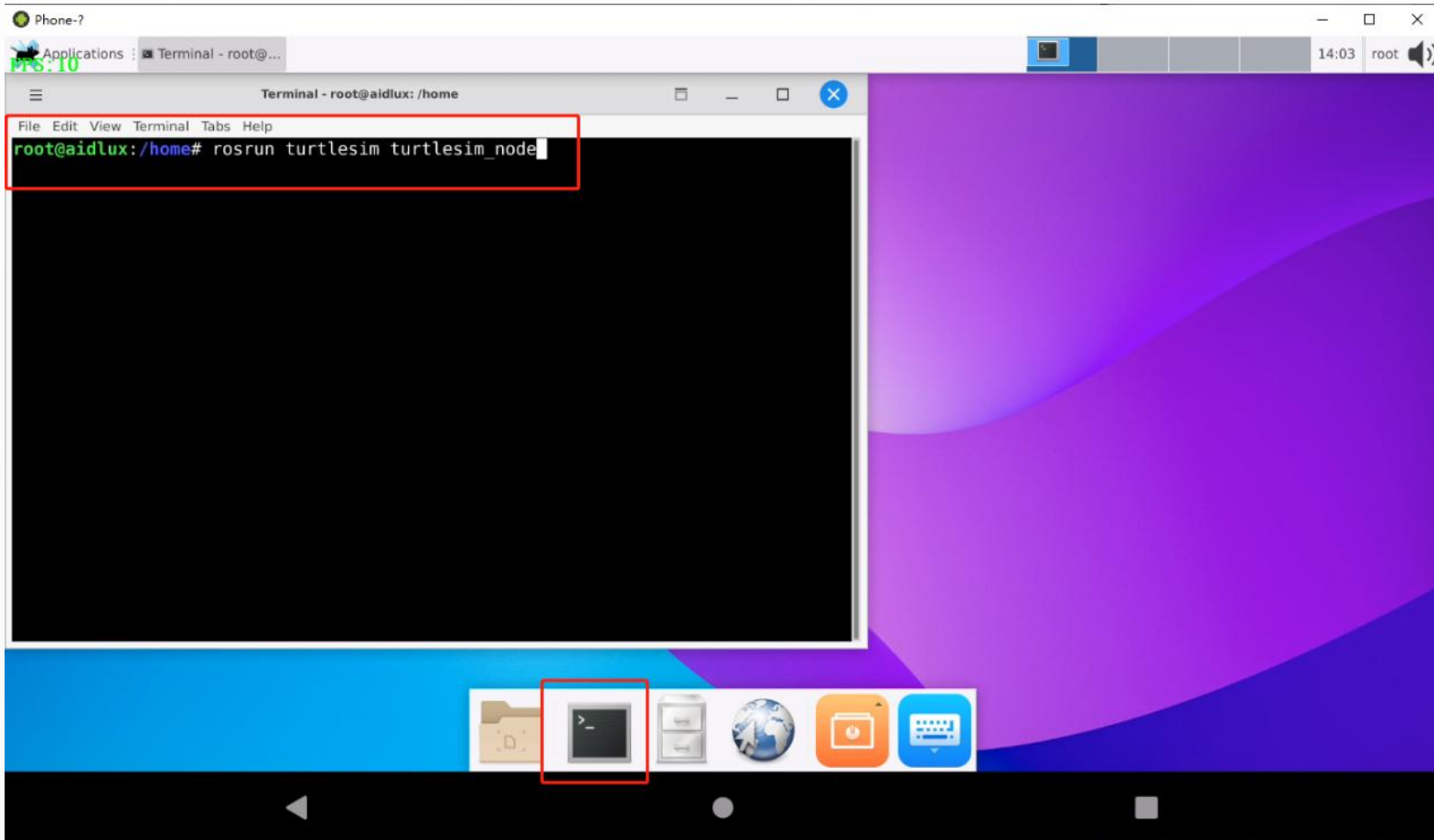
接下在盒子上，打开 xfce 桌面环境，因为小乌龟模拟器必须在桌面环境才能正常打开窗口：



## 五、工作空间编译与运行程序

- 运行程序

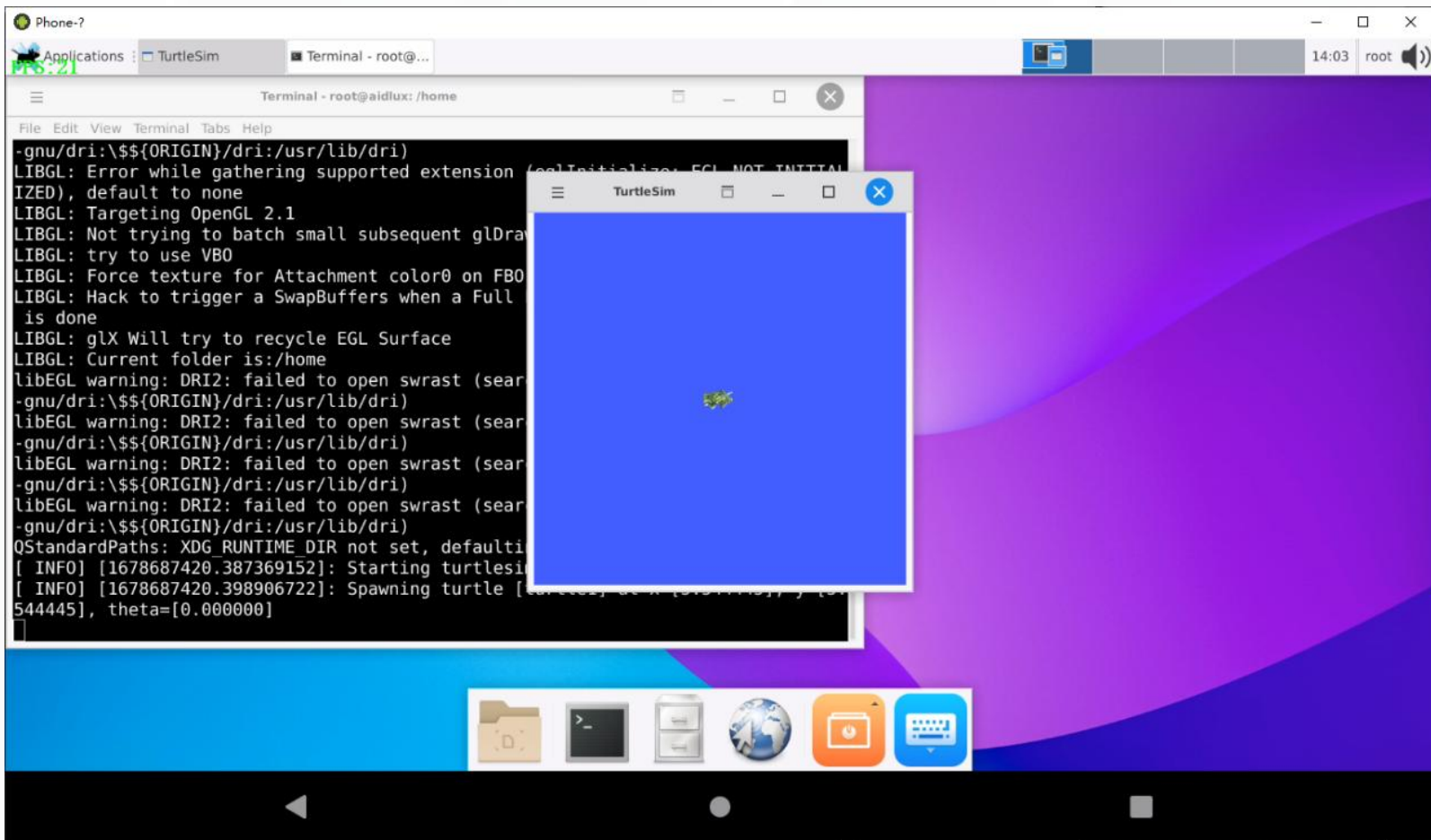
进入桌面环境后，打开终端工具，运行图中所示命令，等待小乌龟模拟器节点启动完成：



## 五、工作空间编译与运行程序

- 运行程序

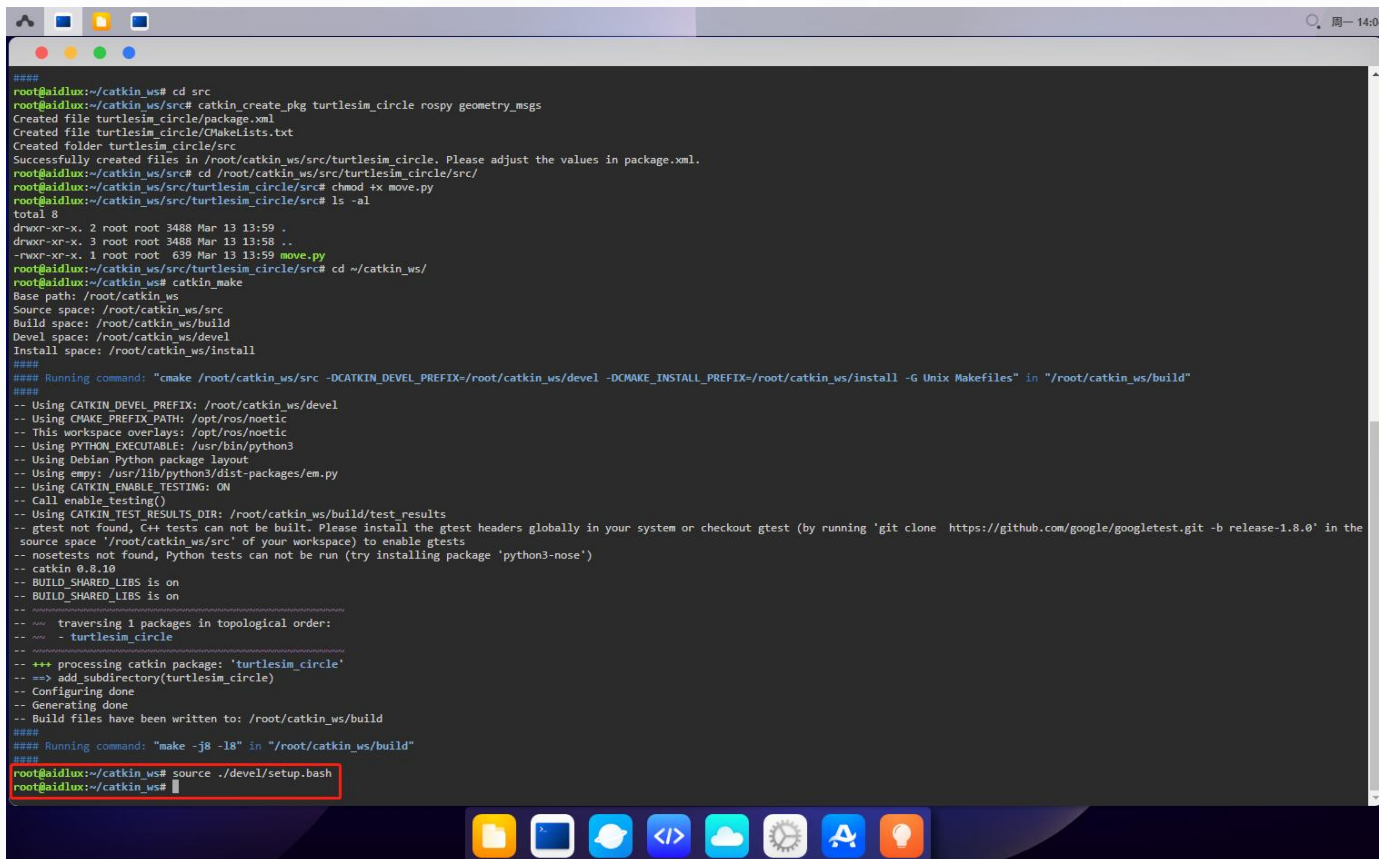
进入桌面环境后，打开终端工具，运行图中所示命令，等待小乌龟模拟器节点启动完成：



## 五、工作空间编译与运行程序

- 运行程序

当小乌龟模拟器窗口成功打开后，回到浏览器中的web桌面，将工作空间加入到环境变量中，这一步必不可少，否则ros无法知晓我们自己开发的功能包位置：



```
#####
root@aidlux:~/catkin_ws# cd src
root@aidlux:~/catkin_ws/src# catkin_create_pkg turtlesim_circle rospy geometry_msgs
Created file turtlesim_circle/package.xml
Created file turtlesim_circle/CMakeLists.txt
Created folder turtlesim_circle/src
Successfully created files in /root/catkin_ws/src/turtlesim_circle. Please adjust the values in package.xml.
root@aidlux:~/catkin_ws/src# cd /root/catkin_ws/src/turtlesim_circle/src/
root@aidlux:~/catkin_ws/src/turtlesim_circle/src# chmod +x move.py
root@aidlux:~/catkin_ws/src/turtlesim_circle/src# ls -al
total 8
drwxr-xr-x. 2 root root 3488 Mar 13 13:59 .
drwxr-xr-x. 3 root root 3488 Mar 13 13:58 ..
-rwxr-xr-x. 1 root root 639 Mar 13 13:59 move.py
root@aidlux:~/catkin_ws/src/turtlesim_circle/src# cd ~/catkin_ws/
root@aidlux:~/catkin_ws# catkin_make
Base path: /root/catkin_ws
Source space: /root/catkin_ws/src
Build space: /root/catkin_ws/build
Devel space: /root/catkin_ws/devel
Install space: /root/catkin_ws/install

#####
Running command: "cmake /root/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/root/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/root/catkin_ws/install -G Unix Makefiles" in "/root/catkin_ws/build"
#####
-- Using CATKIN_DEVEL_PREFIX: /root/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /root/catkin_ws/build/test_results
-- gtest not found, C++ tests can not be built. Please install the gtest headers globally in your system or checkout gtest (by running 'git clone https://github.com/google/googletest.git -b release-1.8.0' in the source space '/root/catkin_ws/src' of your workspace) to enable gtests
-- nosetests not found, Python tests can not be run (try installing package 'python3-nose')
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- ~~~~ traversing 1 packages in topological order:
-- ~~~~ - turtlesim_circle
-- ~~~~ ~~~~
-- ~~~~ +++ processing catkin package: 'turtlesim_circle'
-- ~~~~ ==> add_subdirectory(turtlesim_circle)
-- ~~~~ Configuring done
-- ~~~~ Generating done
-- ~~~~ Build files have been written to: /root/catkin_ws/build

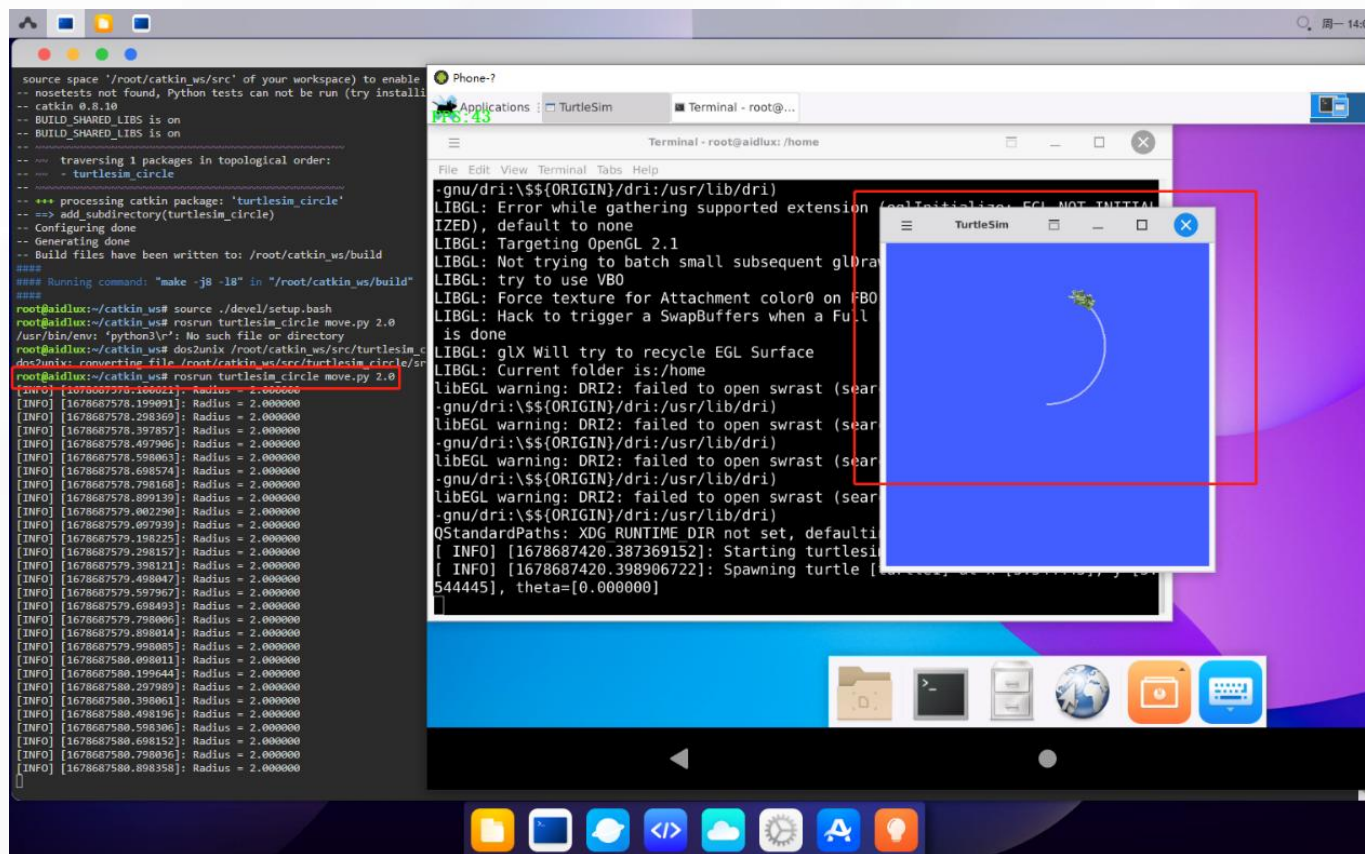
#####
Running command: "make -j8 -l8" in "/root/catkin_ws/build"
#####
root@aidlux:~/catkin_ws# source ./devel/setup.bash
root@aidlux:~/catkin_ws#
```



## 五、工作空间编译与运行程序

- 运行程序

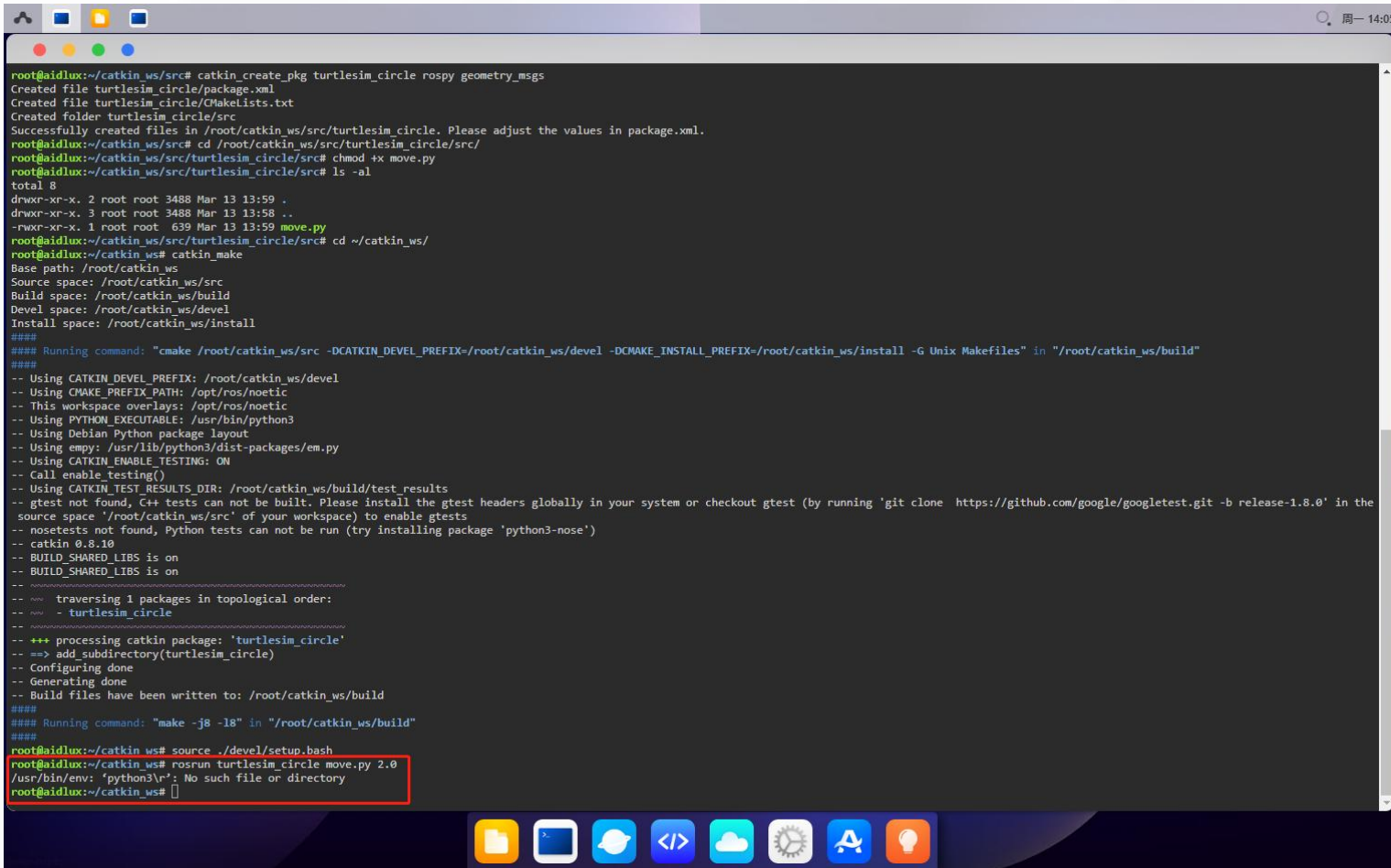
最后运行命令，启动功能模块，并查看盒子上的 xfce 桌面环境中，小乌龟是否按照指令走圆圈



# 五、工作空间编译与运行程序

- 运行程序

如果这一步出现如下报错：



```
root@aidlux:~/catkin_ws/src# catkin_create_pkg turtlesim_circle rospy geometry_msgs
Created file turtlesim_circle/package.xml
Created file turtlesim_circle/CMakeLists.txt
Created folder turtlesim_circle/src
Successfully created files in /root/catkin_ws/src/turtlesim_circle. Please adjust the values in package.xml.
root@aidlux:~/catkin_ws/src# cd /root/catkin_ws/src/turtlesim_circle/src/
root@aidlux:~/catkin_ws/src/turtlesim_circle/src# chmod +x move.py
root@aidlux:~/catkin_ws/src/turtlesim_circle/src# ls -al
total 8
drwxr-xr-x. 2 root root 3488 Mar 13 13:59 .
drwxr-xr-x. 3 root root 3488 Mar 13 13:58 ..
-rwxr-xr-x. 1 root root 639 Mar 13 13:59 move.py
root@aidlux:~/catkin_ws/src/turtlesim_circle/src# cd ~/catkin_ws/
root@aidlux:~/catkin_ws# catkin_make
Base path: /root/catkin_ws
Source space: /root/catkin_ws/src
Build space: /root/catkin_ws/build
Devel space: /root/catkin_ws/devel
Install space: /root/catkin_ws/install
####
### Running command: "cmake /root/catkin_ws/src -DCATKIN_DEVEL_PREFIX=/root/catkin_ws/devel -DCMAKE_INSTALL_PREFIX=/root/catkin_ws/install -G Unix Makefiles" in "/root/catkin_ws/build"
###
-- Using CATKIN_DEVEL_PREFIX: /root/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /opt/ros/noetic
-- This workspace overlays: /opt/ros/noetic
-- Using PYTHON_EXECUTABLE: /usr/bin/python3
-- Using Debian Python package layout
-- Using empy: /usr/lib/python3/dist-packages/em.py
-- Using CATKIN_ENABLE_TESTING: ON
-- Call enable_testing()
-- Using CATKIN_TEST_RESULTS_DIR: /root/catkin_ws/build/test_results
-- gtest not found, C++ tests can not be built. Please install the gtest headers globally in your system or checkout gtest (by running 'git clone https://github.com/google/googletest.git -b release-1.8.0' in the source space '/root/catkin_ws/src' or your workspace) to enable gtests
-- nosetests not found, Python tests can not be run (try installing package 'python3-nose')
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- ~~~~ traversing 1 packages in topological order:
-- ~~~~ - turtlesim_circle
-- ~~~~
-- +++ processing catkin package: 'turtlesim_circle'
==> add_subdirectory(turtlesim_circle)
-- Configuring done
-- Generating done
-- Build files have been written to: /root/catkin_ws/build
####
### Running command: "make -j8 -l8" in "/root/catkin_ws/build"
###
root@aidlux:~/catkin_ws# source ./devel/setup.bash
root@aidlux:~/catkin_ws# rosrun turtlesim_circle move.py 2.0
/usr/bin/env: 'python3/r': No such file or directory
root@aidlux:~/catkin_ws#
```

## 五、工作空间编译与运行程序

- 运行程序

说明编码存在问题，可以通过 `apt install -y dos2unix` 安装编码转换工具，然后使用命令 `dos2unix /root/catkin_ws/src/turtlesim_circle/src/*` 对源码文件进行编码转换，转换完成后，再次启动，应该就可以正常使用了。



04

## Aidlux平台机器人仿真案例效果展示

**AidLux**

**融合系统机器人解决方案**

# THANK YOU

AidLux社区: [community.aidlux.com](http://community.aidlux.com)

