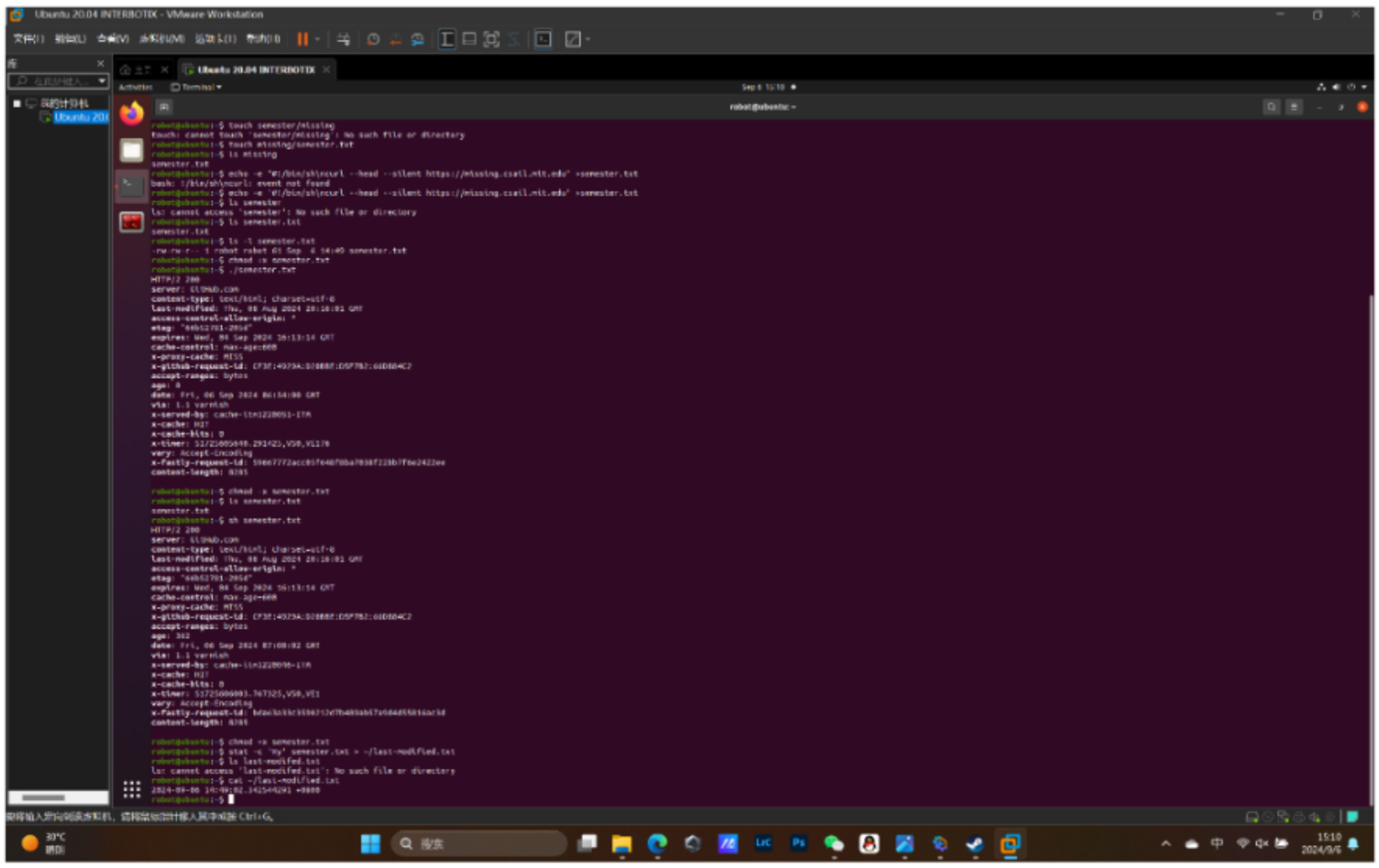
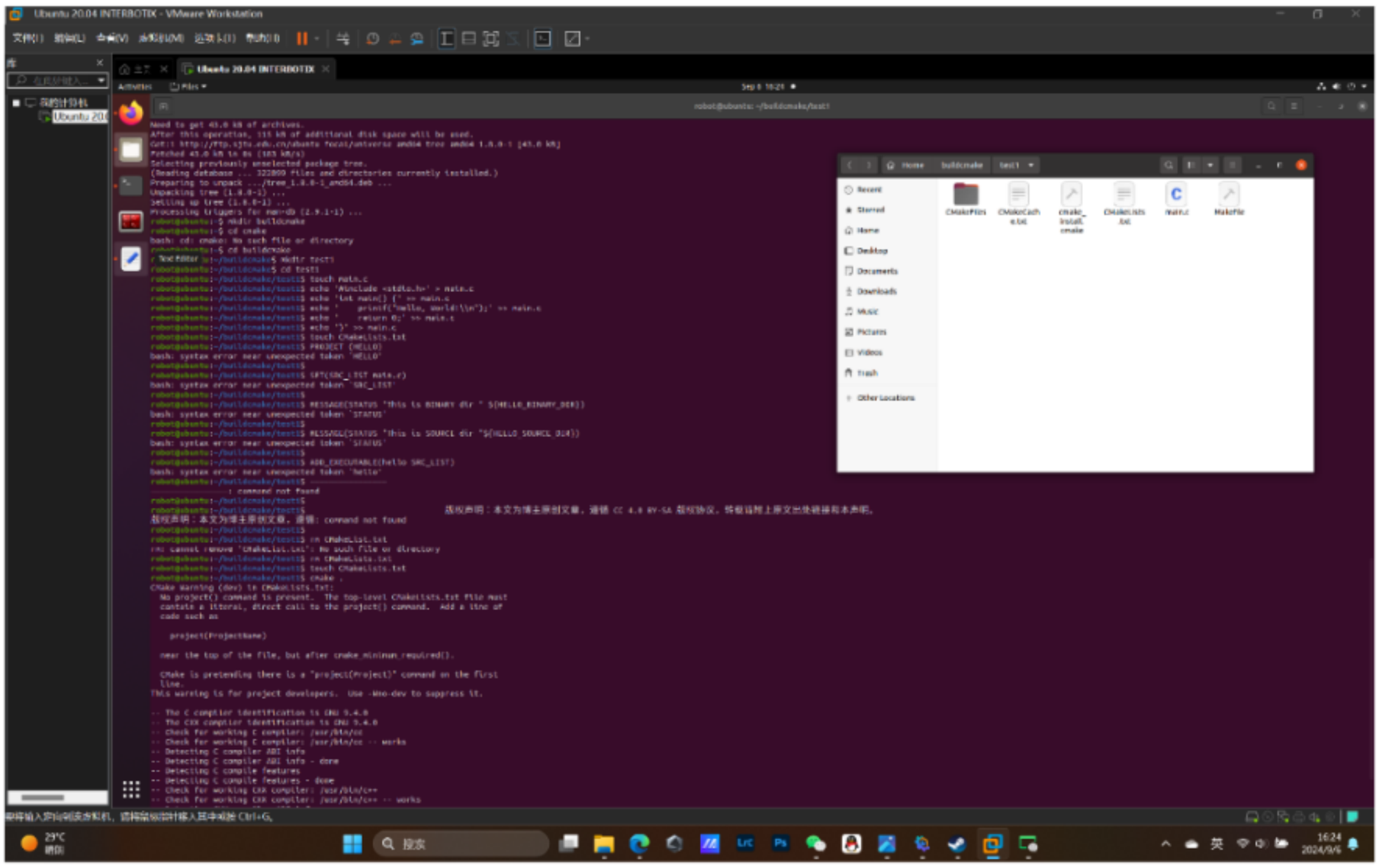
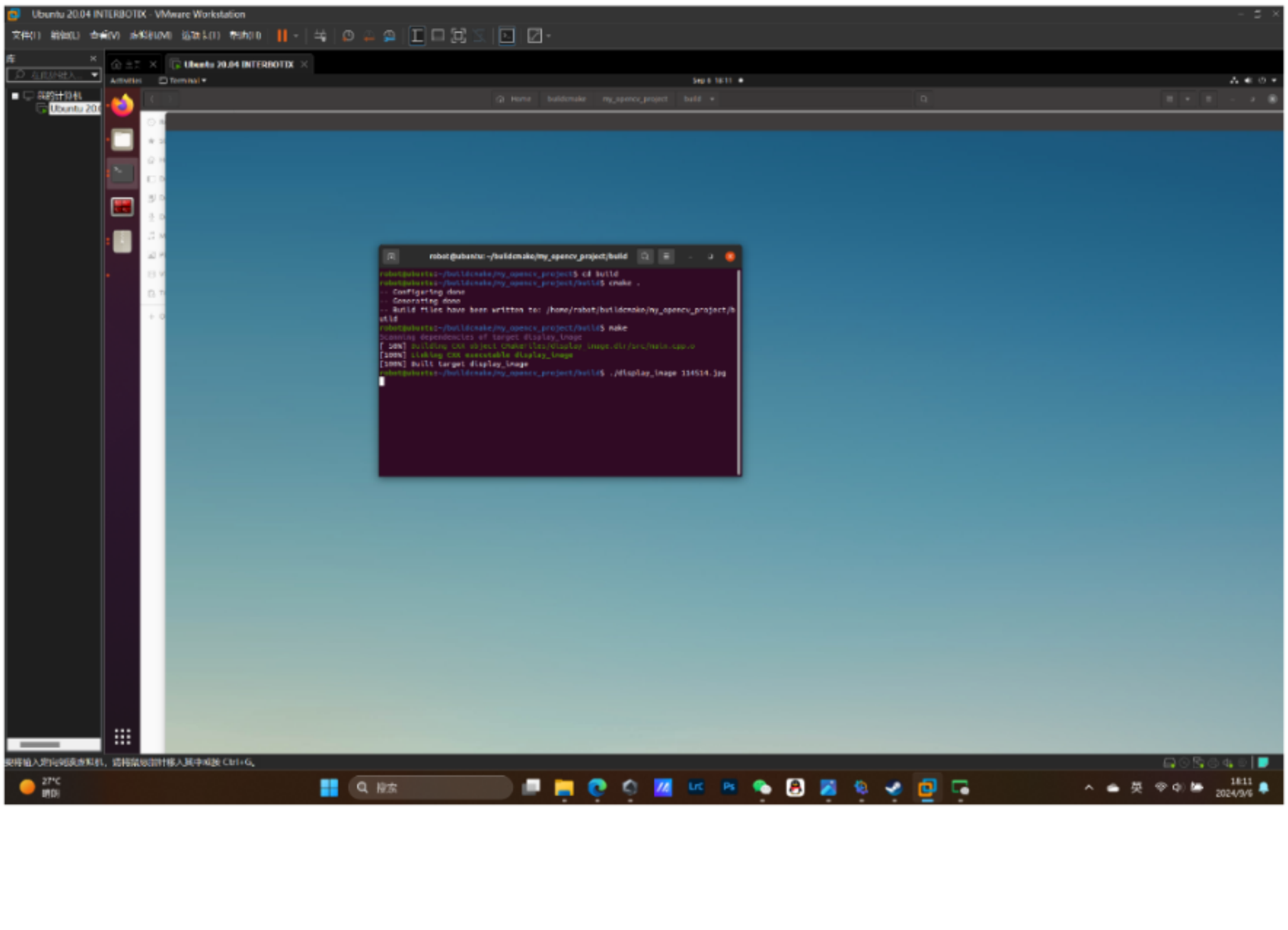
P2

前面一部分个人认为是像 mkdir 创建目录，cd 到目录下后利用 touch 创建文件并利用 echo 进行编辑修改（应该还有 nano 工具和 vim 工具也可以达成类似的效果）而后是运行文件时出现了运行问题 sh 应该是强制运行，而 chmod 应该是修改其为可执行文件后有运行能力（？）





后半部分（opencv 部分）



P3

1.先是创建 ROS 包：

cd ~/catkin_ws/src

catkin_create_pkg my_msgs std_msgs message_generation

这个命令应该是在 ~/catkin_ws/src 目录下创建一个名为 my_msgs 的 ROS 包

2.创建消息文件夹和定义消息：

bashCopy Code

mkdir -p ~/catkin_ws/src/my_msgs/msg

nano ~/catkin_ws/src/my_msgs/msg/ThreeInts.msg

在 my_msgs 包内创建一个 msg 目录，并在其中创建 ThreeInts.msg 文件。

然后编辑入文件：

int64 a

int64 b

int64 c

3.修改 CMakeLists.txt 文件：

find_package(catkin REQUIRED COMPONENTS

  message_generation

  std_msgs

)

add_message_files(

  FILES

  ThreeInts.msg

)

generate_messages(

  DEPENDENCIES

  std_msgs

)

catkin_package(

  CATKIN_DEPENDS message_runtime std_msgs

)

分别作用如下：

  1．find_package(catkin REQUIRED COMPONENTS message_generation std_msgs):

查找 message_generation 和 std_msgs 包。

  2．add_message_files(FILES ThreeInts.msg): 添加自定义消息文件。

  3．generate_messages(DEPENDENCIES std_msgs): 生成自定义消息文件，并指定依赖的标准消息包。

  4.catkin_package(CATKIN_DEPENDS message_runtime std_msgs): 定义依赖关系。

4.修改 xml 文件：

<build_depend>message_generation</build_depend>

<exec_depend>message_runtime</exec_depend>

（message_generation：构建依赖，用于生成消息代码）

（message_runtime：运行时依赖，用于处理消息）

接着编译工作区
cd ~/catkin_ws
catkin_make

接着再创建新包名 publisherandsubscriber 存放发布者和订阅者的节点
~发布者节点：

```
#include <ros/ros.h>
#include <my_msgs/ThreeInts.h>

int main(int argc, char** argv) {
    ros::init(argc, argv, "publisher_node");
    ros::NodeHandle nh;
    ros::Publisher pub = nh.advertise<my_msgs::ThreeInts>("three_ints_topic", 10);

    ros::Rate rate(1); // 1 Hz
    while (ros::ok()) {
        my_msgs::ThreeInts msg;
        msg.a = 1;
        msg.b = 2;
        msg.c = 3;
        pub.publish(msg);
        ros::spinOnce();
        rate.sleep();
    }

    return 0;
}
```

~订阅者节点：

```
#include <ros/ros.h>
#include <my_msgs/ThreeInts.h>

void callback(const my_msgs::ThreeInts::ConstPtr& msg) {
    ROS_INFO("Received: a=%ld, b=%ld, c=%ld", msg->a, msg->b, msg->c);
}

int main(int argc, char** argv) {
    ros::init(argc, argv, "subscriber_node");
    ros::NodeHandle nh;
    ros::Subscriber sub = nh.subscribe("three_ints_topic", 10, callback);
```

```
    ros::spin();
    return 0;
}
```

接着在对应位置修改 CMakeLists.txt 文件（添加可执行文件，依赖和库）：

```
add_executable(publisher_node src/publisher_node.cpp)
add_executable(subscriber_node src/subscriber_node.cpp)

target_link_libraries(publisher_node ${catkin_LIBRARIES})
target_link_libraries(subscriber_node ${catkin_LIBRARIES})

add_dependencies(publisher_node ${${PROJECT_NAME}_EXPORTED_TARGETS}
${catkin_EXPORTED_TARGETS})
add_dependencies(subscriber_node ${${PROJECT_NAME}_EXPORTED_TARGETS}
${catkin_EXPORTED_TARGETS})
```

重复编译工作区后就创见 launch 文件（启动节点并显示输出）：

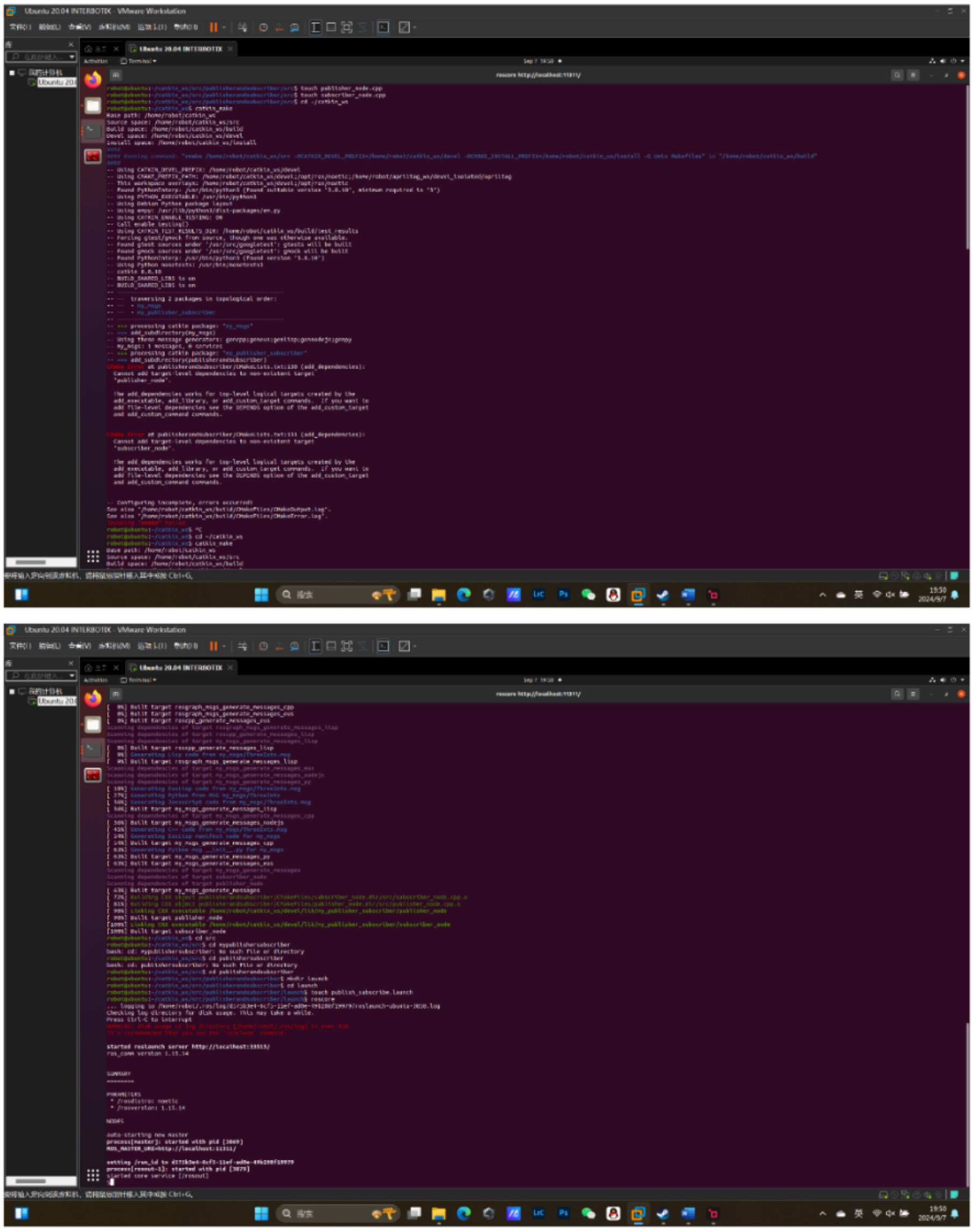
```
<launch>
  <node name="publisher_node" pkg="my_publisher_subscriber"
type="publisher_node" output="screen"/>
  <node name="subscriber_node" pkg="my_publisher_subscriber"
type="subscriber_node" output="screen"/>
</launch>
```

键入 roscore 回车

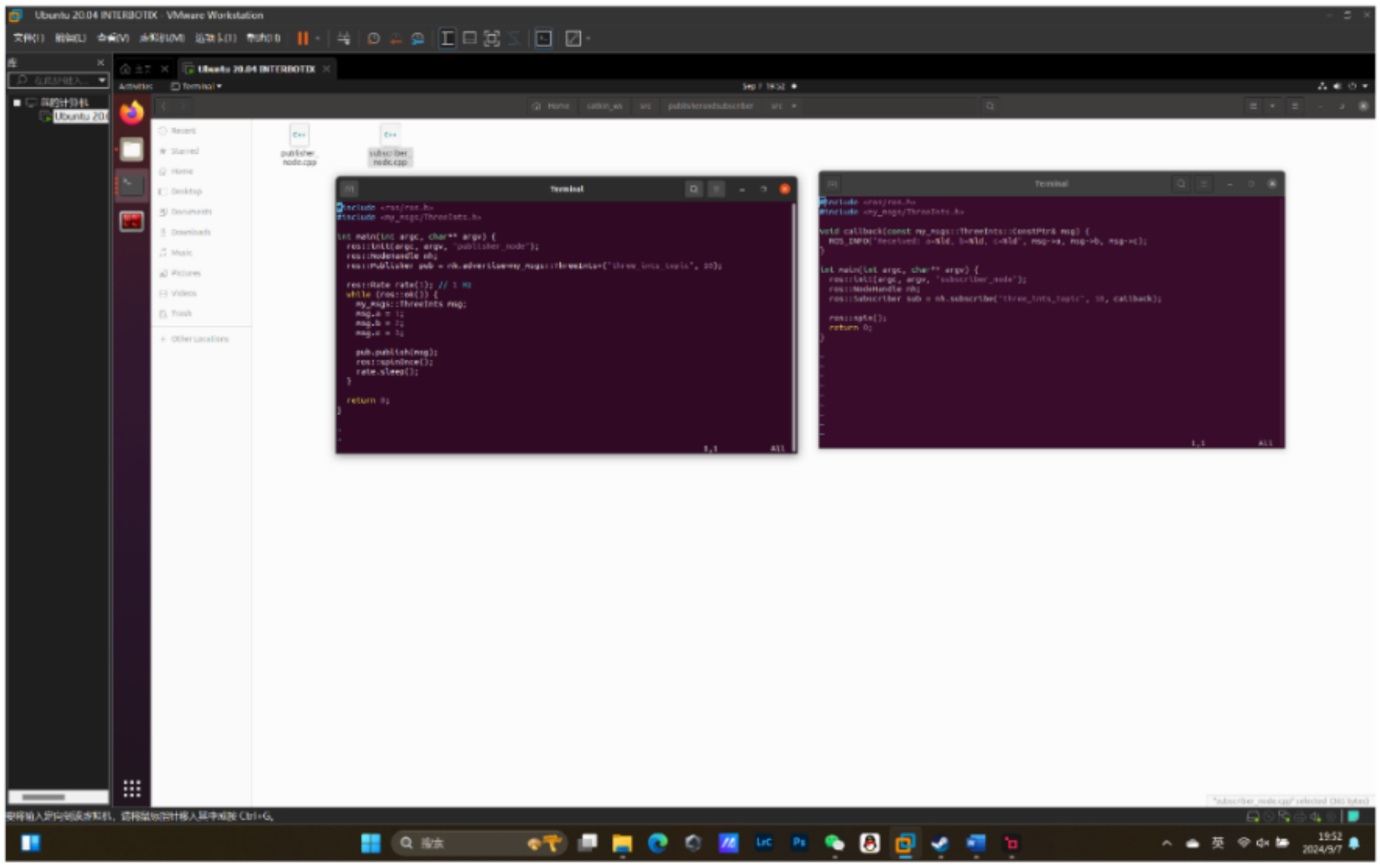
再在一个新的 terminal 中输入：

roslaunch my_publisher_subscriber publish_subscribe.launch 开始运行

4c7672d2527f6cd96641d45d7d02a38a.mp4


Ubuntu 20.04 INTERBOTIX - VMware Workstation
Ubuntu 20.04 INTERBOTIX
Activities
Terminal
Sep 7 19:50
roscore http://localhost:11311/
robot@ubuntu:~/catkin_ws/src/publisherandsubscriber/src$ touch publisher_node.cpp
robot@ubuntu:~/catkin_ws/src/publisherandsubscriber/src$ touch subscriber_node.cpp
robot@ubuntu:~/catkin_ws/src/publisherandsubscriber/src$ cd ~/catkin_ws
robot@ubuntu:~/catkin_ws$ catkin_make
Base path: /home/robot/catkin_ws
Source space: /home/robot/catkin_ws/src
Build space: /home/robot/catkin_ws/build
Devel space: /home/robot/catkin_ws/devel
Install space: /home/robot/catkin_ws/install
-- Using CATKIN_DEVEL_PREFIX: /home/robot/catkin_ws/devel
-- Using CMAKE_PREFIX_PATH: /home/robot/catkin_ws/devel;/opt/ros/noetic;/home/robot/aprilTag_ws/devel_isolated/apriltag
-- This workspace overlays: /home/robot/catkin_ws/devel;/opt/ros/noetic
-- Found PythonInterp: /usr/bin/python3 (found suitable version "3.8.10", 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: /home/robot/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 "3.8.10")
-- Using Python nosetests: /usr/bin/nosetests3
-- catkin 0.8.10
-- BUILD_SHARED_LIBS is on
-- BUILD_SHARED_LIBS is on
-- traversing 2 packages in topological order:
-- - my_msgs
-- - my_publisher_subscriber
-- +++ processing catkin package: 'my_msgs'
-- ==> add_subdirectory(my_msgs)
-- Using these message generators: gencpp;geneus;genlisp;gennodejs;genpy
-- my_msgs: 1 messages, 0 services
-- +++ processing catkin package: 'my_publisher_subscriber'
-- ==> add_subdirectory(publisherandsubscriber)
CMake Error at publisherandsubscriber/CMakeLists.txt:130 (add_dependencies):
Cannot add target-level dependencies to non-existent target "publisher_node".
The add_dependencies works for top-level logical targets created by the add_executable, add_library, or add_custom_target commands. If you want to add file-level dependencies see the DEPENDS option of the add_custom_target and add_custom_command commands.
CMake Error at publisherandsubscriber/CMakeLists.txt:131 (add_dependencies):
Cannot add target-level dependencies to non-existent target "subscriber_node".
The add_dependencies works for top-level logical targets created by the add_executable, add_library, or add_custom_target commands. If you want to add file-level dependencies see the DEPENDS option of the add_custom_target and add_custom_command commands.
-- Configuring incomplete, errors occurred!
See also "/home/robot/catkin_ws/build/CMakeFiles/CMakeOutput.log".
See also "/home/robot/catkin_ws/build/CMakeFiles/CMakeError.log".
Invoking "cmake" failed
robot@ubuntu:~/catkin_ws$ ^C
robot@ubuntu:~/catkin_ws$ cd ~/catkin_ws
robot@ubuntu:~/catkin_ws$ catkin_make
Base path: /home/robot/catkin_ws
Source space: /home/robot/catkin_ws/src
Build space: /home/robot/catkin_ws/build
要将输入定向到该虚拟机，请将鼠标指针移入其中或按 Ctrl+G。
19:50
2024/9/7

Ubuntu 20.04 INTERBOTIX - VMware Workstation
Ubuntu 20.04 INTERBOTIX
Activities
Terminal
roscore http://localhost:11311/
[0%] Built target roscpp_generate_messages_lisp
[0%] Generating Lisp code from my_msgs/ThreeInts.msg
[0%] Built target rosgraph_msgs_generate_messages_lisp
[10%] Generating EusLisp code from my_msgs/ThreeInts.msg
[27%] Generating Python from MSG my_msgs/ThreeInts
[36%] Generating Javascript code from my_msgs/ThreeInts.msg
[36%] Built target my_msgs_generate_messages_lisp
[36%] Built target my_msgs_generate_messages_nodejs
[45%] Generating C++ code from my_msgs/ThreeInts.msg
[54%] Generating EusLisp manifest code for my_msgs
[54%] Built target my_msgs_generate_messages_cpp
[63%] Generating Python msg __init__.py for my_msgs
[63%] Built target my_msgs_generate_messages_py
[63%] Built target my_msgs_generate_messages_eus
[63%] Built target my_msgs_generate_messages
[72%] Building CXX object publisherandsubscriber/CMakeFiles/subscriber_node.dir/src/subscriber_node.cpp.o
[81%] Building CXX object publisherandsubscriber/CMakeFiles/publisher_node.dir/src/publisher_node.cpp.o
[90%] Linking CXX executable /home/robot/catkin_ws/devel/lib/my_publisher_subscriber/publisher_node
[90%] Built target publisher_node
[100%] Linking CXX executable /home/robot/catkin_ws/devel/lib/my_publisher_subscriber/subscriber_node
[100%] Built target subscriber_node
robot@ubuntu:~/catkin_ws$ cd src
robot@ubuntu:~/catkin_ws/src$ cd mypublishersubscriber
bash: cd: mypublishersubscriber: No such file or directory
robot@ubuntu:~/catkin_ws/src$ cd publishersubscriber
bash: cd: publishersubscriber: No such file or directory
robot@ubuntu:~/catkin_ws/src$ cd publisherandsubscriber
robot@ubuntu:~/catkin_ws/src/publisherandsubscriber$ mkdir launch
robot@ubuntu:~/catkin_ws/src/publisherandsubscriber$ cd launch
robot@ubuntu:~/catkin_ws/src/publisherandsubscriber/launch$ touch publish_subscribe.launch
robot@ubuntu:~/catkin_ws/src/publisherandsubscriber/launch$ roscore
... logging to /home/robot/.ros/log/d3f3b3e4-8cf3-11ef-ad0e-49b280f19979/roslaunch-ubuntu-3038.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
WARNING: disk usage in log directory [/home/robot/.ros/log] is over 1GB.
It's recommended that you use the 'rosclean' command.
started roslaunch server http://localhost:33515/
ros_comm version 1.15.14
SUMMARY
PARAMETERS
* /rosdistro: noetic
* /rosversion: 1.15.14
NODES
auto-starting new master
process[master]: started with pid [3069]
ROS_MASTER_URI=http://localhost:11311/
setting /run_id to d3f3b3e4-8cf3-11ef-ad0e-49b280f19979
process[rosout-1]: started with pid [3079]
started core service [/rosout]
要将输入定向到该虚拟机，请将鼠标指针移入其中或按 Ctrl+G。
19:50
2024/9/7



P4

先是配置环境，python, pytorch, pycocotools, pycharm 以及像 cuda，cudnn 等驱动，在一个选定文件夹里面创建虚拟环境并完成对应的环境配置，而后在 pycharm 中对应载入虚拟环境，下载一些预训练模型和源码并结合 python 版本和项目需要进行更改。接下来写一个爬虫程序爬取需要的图片，并分为测试集，训练集和验证集，最后再下载 labelimg 进行数据标注就可以交给 pycharm 进行训练

其余代码和 pt 文件在目录可见