4、Docker hardware interaction and data processing

4.1. Hardware mounting (port binding)

1、Create udev rules in the host (/etc/udev/rules.d/)

2、Then when opening the container, mount the devices with the rules set into the docker container through parameters such as --device=/dev/myserial --device=/dev/rplidar and so on.

```
docker run -it --device=/dev/myserial --device=/dev/rplidar ubuntu:latest /bin/bash
```

3、The device can be discovered in the docker container

```
jetson@ubuntu:~$ docker images
REPOSITORY                    TAG       IMAGE ID       CREATED             SIZE
hello-world                   latest    46331d942d63   13 months ago       9.14kB
jetson@ubuntu:~$ ll /dev | grep ttyUSB*
lrwxrwxrwx   1 root   root           7 Apr 23 18:07 myserial -> ttyUSB0
lrwxrwxrwx   1 root   root           7 Apr 23 18:07 rplidar -> ttyUSB1
crwxrwxrwx   1 root   dialout 188,   0 Apr 23 18:07 ttyUSB0
crwxrwxrwx   1 root   dialout 188,   1 Apr 23 18:07 ttyUSB1
jetson@ubuntu:~$ docker run -it   --device=/dev/myserial --device=/dev/rplidar
ubuntu:latest  /bin/bash
root@03522257ba30:/# ls /dev   # docker中已经有myserial和rplidar
console  fd  full  mqueue  myserial  null  ptmx  pts  random  rplidar  shm  stderr
stdin  stdout  tty  urandom  zero
```

4.2、Display of GUI in docker

note：The system provided by yahboom is already installed, no need to install it.

1、Install on the host.

```
sudo apt-get install tigervnc-standalone-server tigervnc-viewer
sudo apt-get install x11-xserver-utils
```

2、Execute in host machine：xhost +

After the following picture is displayed normally, perform 3 steps

```
yahboom@yahboom-VM:~$ xhost +
access control disabled, clients can connect from any host
yahboom@yahboom-VM:~$
```

3、Execute the command on the host to enter the container:

```
docker run -it \                                    # Run docker image interactively
--env="DISPLAY" \                                   # Turn on the display GUI
interface
--env="QT_X11_NO_MITSHM=1" \                        # Use port 1 of X11 for display
-v /tmp/.X11-unix:/tmp/.X11-unix \                  # Mapping shows service node
directory
microros/micro-ros-agent:humble                     # Image name to start
/bin/bash                                           # Execute the /bin/bash command
within the container
```

4、test

```
Execute in container: rviz2
```

4.3、Transfer files between docker container and host

4.3.1、Use cp naming

- Copy files from the container to the host

The command used is, `docker cp container id:path in container destination host path`

- Copy files from the host to the container

The command used is, `docker cp host file path container id: path within the container`

4.3.2、Use data volumes

Data volume overview

The application and running environment are packaged to form a container for operation. The operation can be accompanied by the container, but our requirements for data are that we hope to be persistent! It's like if you install a mysql and delete the container, it's equivalent to deleting the database and running away. This is definitely not okay! So we hope that it is possible to share data between containers. If the data generated by the docker container does not generate a new image through docker commit, so that the data is saved as part of the image, then when the container is deleted, the data will naturally be gone! This won't work!

In order to save data in docker we can use volumes! Let the data be mounted locally! This way the data will not be lost due to deletion of the container!

Features:

1、Data volumes can share or reuse data between containers

2、Changes in the volume can take effect directly

3、Changes in the data volume will not be included in updates to the mirror

4、The life cycle of a data volume lasts until no containers use it.

Data volume usage

```
# 命令
docker run -it -v Host absolute path directory: directory in the container image name
# Test: The /home/jetson/temp directory in the host and the /root/temp directory in the container can share data, or they can be changed to other directories you want to share.
docker run -it -v /home/jetson/temp:/root/temp ubuntu:latest /bin/bash
```