

Docker hardware interaction and data processing

4.1. Hardware mounting (port binding)

1. Create udev rules in the host machine (/etc/udev/rules.d/), see [VI. Linux operating system --- - 6. Binding device ID] section
2. Then when starting the container, mount the devices with the rules set into the docker container through parameters such as --device=/dev/myserial --device=/dev/rplidar

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

3. The device can be found in the docker container

```
jetson@ubuntu:~$ docker images
REPOSITORY          TAG          IMAGE ID          CREATED          SIZE
ubuntu              1.0         78ca7be949b6     About an hour ago 69.2MB
pengan88/ubuntu     1.0         78ca7be949b6     About an hour ago 69.2MB
yahboomtechnology/ros-foxy 3.4.0       49581aa78b6b     6 hours ago     24.3GB
yahboomtechnology/ros-foxy 3.3.9       cefb5ac2ca02     4 days ago      20.5GB
yahboomtechnology/ros-foxy 3.3.8       49996806c64a     4 days ago      20.5GB
yahboomtechnology/ros-foxy 3.3.7       8989b8860d17     5 days ago      17.1GB
yahboomtechnology/ros-foxy 3.3.6       326531363d6e     5 days ago      16.1GB
mysql               latest      5371f8c3b63e     6 days ago      592MB
ubuntu              latest     bab8ce5c00ca     6 weeks ago     69.2MB
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 # myserial and rplidar are already in docker
console fd full mqueue myserial null ptmx pts random rplidar shm
stderr stdin stdout tty urandom zero
```

4.2. Display of GUI in docker

1. Install in the host machine:

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

2. Execute in the host machine: xhost +

After the following figure is displayed normally, perform 3 steps:

```
Jetson@jetson-desktop: ~
jetson@jetson-desktop: ~ 80x24
jetson@jetson-desktop:~$ xhost +
access control disabled, clients can connect from any host
jetson@jetson-desktop:~$
```

3. Execute the command in the host machine to enter the container, and pay attention to modify the corresponding version number:

```
docker run -it \
--privileged \
--net=host \
--ipc=bridge \
--ipc=host \
--pid=host \
--env="DISPLAY" \
--env="QT_X11_NO_MITSHM=1" \
-v /dev/bus/usb:/dev/bus/usb \
-v /tmp/.X11-unix:/tmp/.X11-unix \
-v ~/temp:/root/temp \
yahboomtechnology/jetcobot_noetic:3.1.3 /bin/bash
```

You can also edit the startup script, name it new_jetcobot_docker.sh, and run the following command to enter the container.

```
sh new_jetcobot_docker.sh
```

4. Testing

```
Execute in the container: rviz
```

4.3, Transfer files between docker container and host

4.3.1, Use cp naming

4.3.1.1, Copy files from container to host

```
# Command
docker cp container id: path in container destination host path

# Test
# Execute in container, create a file test
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
c54bf9efae47   ubuntu:latest  "/bin/bash"             2 hours ago   Up 9 minutes
              funny_hugle
3b9c01839579   hello-world    "/hello"                 3 hours ago   Exited (0) 3 hours ago
              jovial_brown
jetson@ubuntu:~$ docker attach c5
```

```

root@c54bf9efae47:/# ls
bin boot dev etc home lib media mnt opt proc root run sbin srv sys
tmp usr var
root@c54bf9efae47:/# cd
root@c54bf9efae47:~# ls
root@c54bf9efae47:~# touch test.txt
root@c54bf9efae47:~# ls
test.txt
root@c54bf9efae47:~# pwd
/root
root@c54bf9efae47:/# read escape sequence      #Press ctrl+P+Q to exit the
container without stopping
jetson@ubuntu:~$ docker cp c54bf9efae47:/root/test.txt ~/
jetson@ubuntu:~$ ls      # The test.txt file has been copied in
Desktop Documents Downloads fishros Music openvino Pictures Public
rootOnNVMe run_docker.sh sensors snap temp Templates test.txt Videos

```

4.3.1.2. Copy files from the host to the container

```

# Command
docker cp host file path container id: path in container

# Test
jetson@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
c54bf9efae47   ubuntu:latest  "/bin/bash"            2 hours ago   Up 5 minutes
              funny_hugle
3b9c01839579   hello-world    "/hello"                3 hours ago   Exited (0) 3 hours ago
              jovial_brown
jetson@ubuntu:~$ ls
Desktop Documents Downloads fishros Music openvino Pictures Public
rootOnNVMe run_docker.sh sensors snap temp Templates test.txt Videos
jetson@ubuntu:~$ touch 11.txt
jetson@ubuntu:~$ ls
11.txt Desktop Documents Downloads fishros Music openvino Pictures Public
rootOnNVMe run_docker.sh sensors snap temp Templates test.txt Videos
jetson@ubuntu:~$ docker cp 11.txt c54bf9efae47:/root/
jetson@ubuntu:~$ docker attach c5
root@c54bf9efae47:/# ls
bin boot dev etc home lib media mnt opt proc root run sbin srv sys
tmp usr var
root@c54bf9efae47:/# cd /root/
root@c54bf9efae47:~# ls      # 11.txt file has been copied in
11.txt test.txt

```

4.3.2. Using Data Volumes

4.3.2.1. Data Volume Overview

Package the application and the running environment into a container to run. The running can be accompanied by the container, but our requirement for data is that it can be persistent! For example, if you install a mysql, and then you delete the container, it is equivalent to deleting the library 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 is not generated through docker commit, so that the data is saved as part of the image, then when the container is deleted, the data will naturally disappear! This will not work!

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

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 the update of the image
4. The life cycle of the data volume lasts until no container uses it

4.3.2.2, Data volume usage

```
# Command
docker run -it -v host absolute path directory: container directory image name

# Test
docker run -it -v /home/jetson/temp:/root/temp yahboomtechnology/ros-foxy:3.4.0
/bin/bash
```

The /home/jetson/temp directory in the host and the /root/temp directory in the container can share data, or you can change it to other directories you want to share.