

# 2023 秋《操作系统》课程实验报告

## 实验 0

21301039 黄钰淞

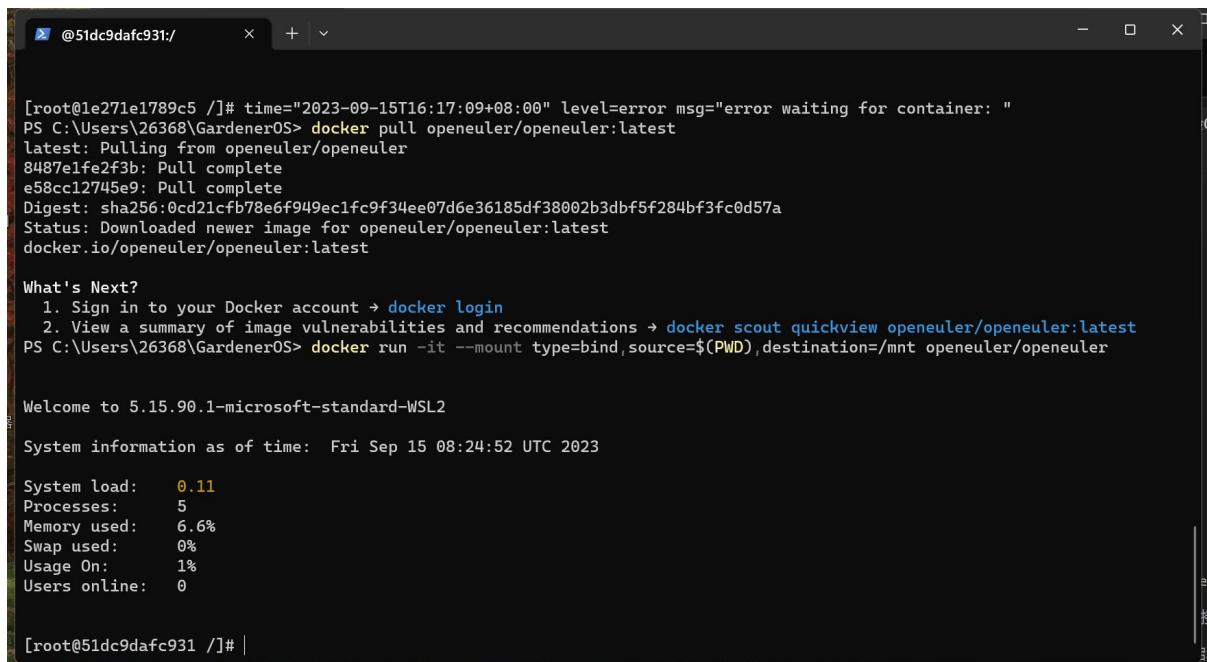
### 一、 实验步骤

#### 1. 准备实验基础平台

1.1 首先进行 docker desktop 等下载，之前下载过了不再赘述

1.2 拉取最新镜像

```
docker pull openeuler/openeuler:latest
```



The screenshot shows a terminal window with the following content:

```
[root@1e271e1789c5 ~]# time="2023-09-15T16:17:09+08:00" level=error msg="error waiting for container: "
PS C:\Users\26368\GardenerOS> docker pull openeuler/openeuler:latest
latest: Pulling from openeuler/openeuler
8487e1fe2f3b: Pull complete
e58cc12745e9: Pull complete
Digest: sha256:0cd21cfb78e6f949ec1fc9f34ee07d6e36185df38002b3dbf5f284bf3fc0d57a
Status: Downloaded newer image for openeuler/openeuler:latest
docker.io/openeuler/openeuler:latest

What's Next?
 1. Sign in to your Docker account → docker login
 2. View a summary of image vulnerabilities and recommendations → docker scout quickview openeuler/openeuler:latest
PS C:\Users\26368\GardenerOS> docker run -it --mount type=bind,source=$(PWD),destination=/mnt openeuler/openeuler

Welcome to 5.15.90.1-microsoft-standard-WSL2
System information as of time: Fri Sep 15 08:24:52 UTC 2023

System load: 0.11
Processes: 5
Memory used: 6.6%
Swap used: 0%
Usage On: 1%
Users online: 0

[root@51dc9dafc931 ~]# |
```

可以看到拉取成功

#### 2. 创建开发目录

Git 相关操作参考 <https://blog.csdn.net/Thinkingcao/article/details/106059446>

当前步骤最终截图如下

```
MINGW64:/c/Users/26368/GardenerOS
26368@HYSLAPTOP MINGW64 ~/GardenerOS (master)
$ git branch -a
* master
  remotes/origin/exp0
  remotes/origin/master

26368@HYSLAPTOP MINGW64 ~/GardenerOS (master)
$ git checkout -b exp0
Switched to a new branch 'exp0'

26368@HYSLAPTOP MINGW64 ~/GardenerOS (exp0)
$ git branch -a
* exp0
  master
  remotes/origin/exp0
  remotes/origin/master

26368@HYSLAPTOP MINGW64 ~/GardenerOS (exp0)
$ git log --oneline --graph --decorate --all
* e9bca4f (HEAD -> exp0, origin/master, origin/exp0, master) expRec0 initial

26368@HYSLAPTOP MINGW64 ~/GardenerOS (exp0)
$
```

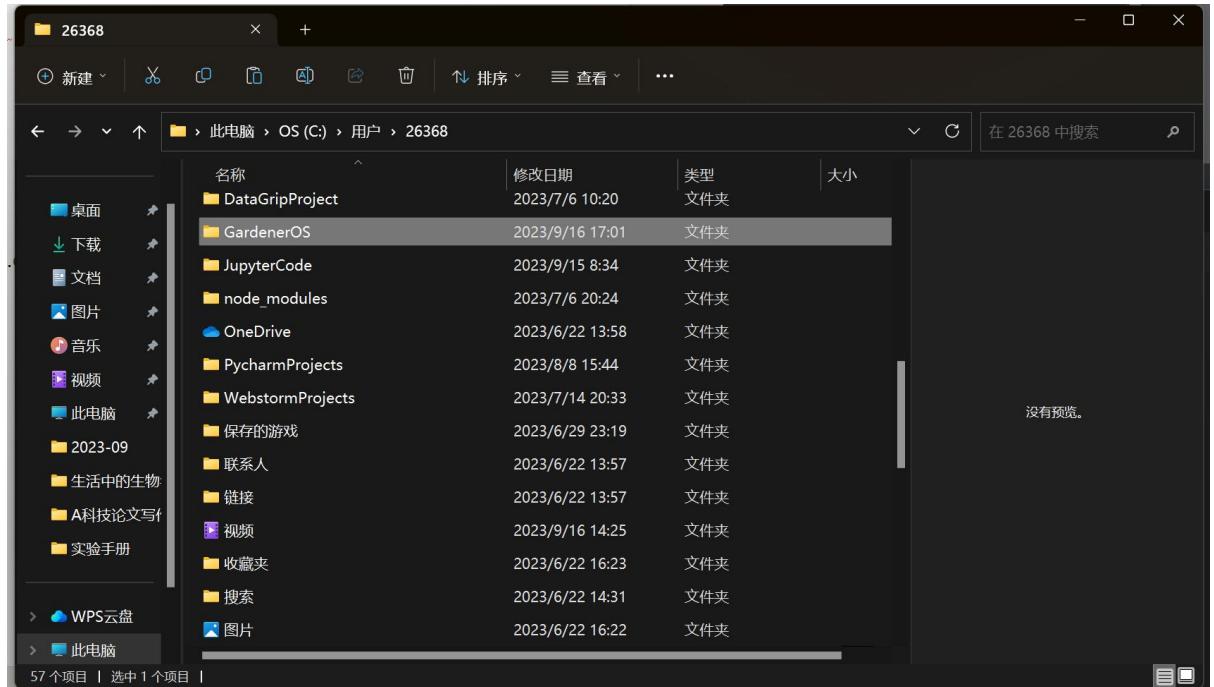
详细步骤如下

## 2.1 创建本地开发目录

首先在用户目录下创建 GardenerOS 目录作为本地开发目录

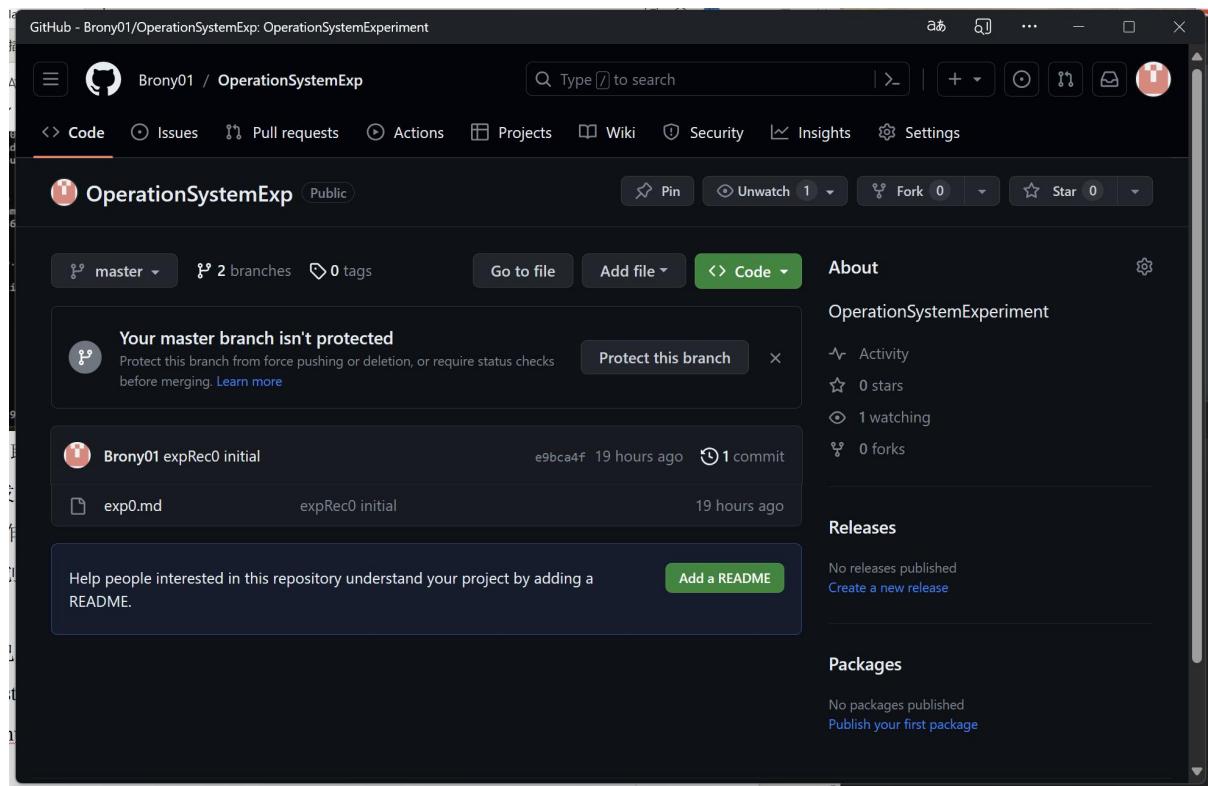
```
mkdir ~/GardenerOS/
```

可以看到已经出现 GardenerOS 目录



## 2.2 远程开发目录

在 GitHub 创建代码仓库 `git@github.com:Brony01/OperationSystemExp.git`



需要执行如下 git 相关的操作：

```
git init
```

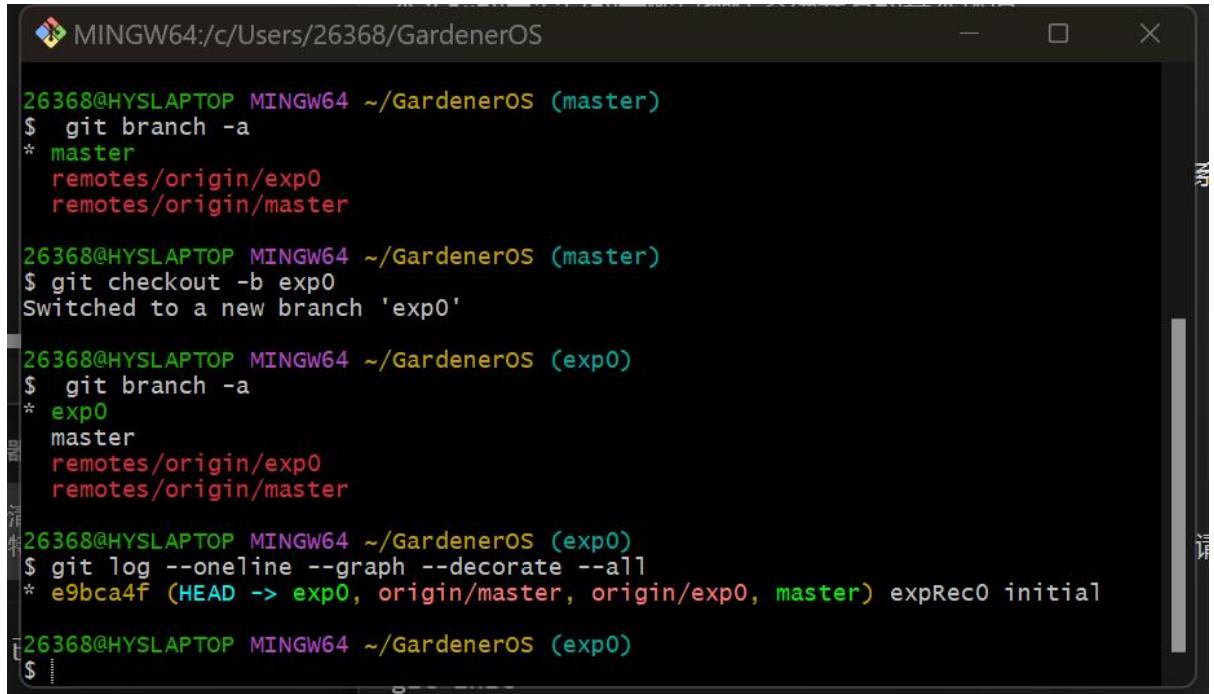
```
touch .gitignore (windows 下可用 echo 代替 touch)
```

```
git add .
```

```
git commit
```

```
git remote add origin git@github.com:Brony01/OperationSystemExp.git
```

```
git push --all
```



```
MINGW64:/c/Users/26368/GardenerOS
26368@HYSLAPTOP MINGW64 ~/GardenerOS (master)
$ git branch -a
* master
  remotes/origin/exp0
  remotes/origin/master

26368@HYSLAPTOP MINGW64 ~/GardenerOS (master)
$ git checkout -b exp0
Switched to a new branch 'exp0'

26368@HYSLAPTOP MINGW64 ~/GardenerOS (exp0)
$ git branch -a
* exp0
  master
  remotes/origin/exp0
  remotes/origin/master

26368@HYSLAPTOP MINGW64 ~/GardenerOS (exp0)
$ git log --oneline --graph --decorate --all
* e9bca4f (HEAD -> exp0, origin/master, origin/exp0, master) expRec0 initial

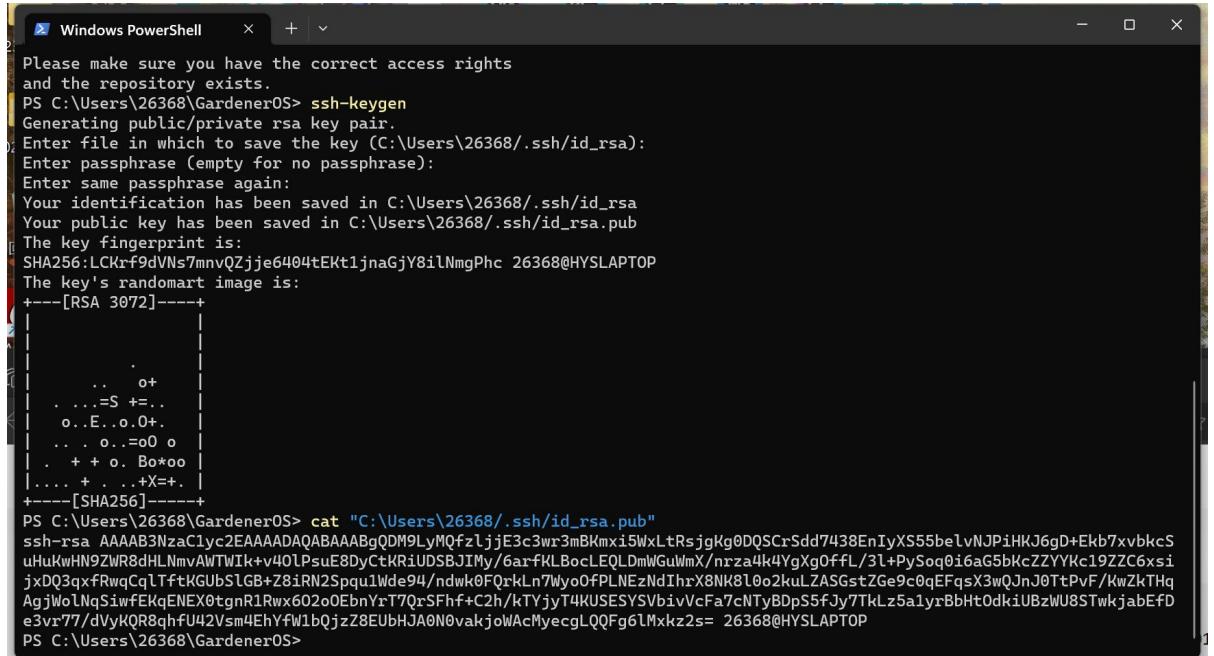
26368@HYSLAPTOP MINGW64 ~/GardenerOS (exp0)
$
```

这里添加远程目录需要将本机加入 ssh key 否则没有访问权限

步骤如下

ssh-keygen

cat "C:\Users\26368\.ssh\id\_rsa.pub"



```
Windows PowerShell
Please make sure you have the correct access rights
and the repository exists.
PS C:\Users\26368\GardenerOS> ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (C:\Users\26368/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\26368/.ssh/id_rsa
Your public key has been saved in C:\Users\26368/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:LCKrf9dVNz7mmvQZjje6404tEkt1jnaGjY8ilNmgPhc 26368@HYSLAPTOP
The key's randomart image is:
+---[RSA 3072]---+
| . . . o+ |
| . . . =S +.. |
| o . E . o .0+ |
| . . . o . =oO o |
| . + + o . Bo*oo |
| . . . + . . +X=+. |
+---[SHA256]---+
PS C:\Users\26368\GardenerOS> cat "C:\Users\26368\.ssh\id_rsa.pub"
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQDM9lyM0fzljjE3c3wr3mBKmx5WxLrsgjKg0DQScRsd7438EhlyXS55belvNJPihKJ6gD+Ekb7xbkcs
uHukwHN9ZWR8dHLNmvaWTWIk+v40lPsuE8DyCtKRiUDSBJIMy/6arfKLBoeLEQLDmWGuUlmX/nrzau4k4YgXgOffl/3l+PySoq0i6aG5bkCZZYKc19ZZC6xsi
jxDQ3qxfrwqCqLTftKGUBsLGB+z8iRN2SpqulWde94/ndwk0FQrkLn7Wy0oPPLNEzNdIhrX8NK8l0o2kulZASGstZGe9c0qEFqsX3wQJnJ0TtPvF/KwZkTHq
AgjWolNqSiwfEKqENEX0tgnR1Rwx602oOEbnYrT7QrSFhf+C2h/kTYjyT4KUSeSYVbivVcFa7cNTyBDpS5fJy7TkLz5a1yrBbHtDkiUBzWU8STwkjabEfD
e3vr77/dVYKQR8qhFu42Vsm4EhYfW1bQjzz8EUbHJA0N0vakjoWAcMyecgLQQFg61Mxkz2s= 26368@HYSLAPTOP
PS C:\Users\26368\GardenerOS>
```

将得到的 ssh-key 复制到 GitHub 的 ssh-key 设置页面（选择 add ssh-key）

随后可以正常访问代码仓库。

```

Windows PowerShell

Enter file in which to save the key (C:\Users\26368\.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in C:\Users\26368\.ssh/id_rsa
Your public key has been saved in C:\Users\26368\.ssh/id_rsa.pub
The key fingerprint is:
SHA256:LCKrf9dVN87mnvQZjje6404tEkt1jnaGjY8ilNmgPhc 26368@HYSLAPTOP
The key's randomart image is:
+---[RSA 3072]---+
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
| . . . . . . . . . |
+---[SHA256]---+
PS C:\Users\26368\GardenerOS> cat "C:\Users\26368\.ssh/id_rsa.pub"
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQDM9lyMQfzljE3c3w3mBKmx5WxLrSjgKg0DQSCrSdd7438EnIyXS5belvNJPiHKJ6gD+Ekb7xbkcs
uHuKwHN9ZWR8dHNMvAWTlIk+v40lPsuE8DyCtKRiUDSBJIMy/6arfKLBoeLEQLDmWGuImX/nrzak4YgXgoFFl/3l+PySoq0i6aG5bkCZZYKc19ZZC6xsi
jxDQ3qxfrwqCqlTftKGubSlGB+Z8iRN2SpqU1Wde94/ndwk0FQrkLn7Wyo0fPLNEzNdIhrX8NK8l0o2kuLZASGstZGe9c0qEFqsX3wQJnJ0TtPvF/KwZkTHq
AgjWolNgSiwfEKqENEx0tgnR1Rwx6O2oOEbnYrTQcSFh+C2h/kTYjyT4KUSeSYSVbivVcfa7cNTyBDpS5fJy7TkLz5a1yrBbHtOdkiuBzWU8StTwkjabEfD
e3vr77/dVYkQR8qhFu42Vsm4EhYfW1bQjz8EUbhJIAON0vakjoWAcMyecgLQQFg61Mxkz2s= 26368@HYSLAPTOP
PS C:\Users\26368\GardenerOS> git push --all
No refs in common and none specified; doing nothing.
Perhaps you should specify a branch.
Everything up-to-date
PS C:\Users\26368\GardenerOS>

```

当新建的远程分支在本地不可见时，先用 fetch 命令更新 remote 索引。

git fetch

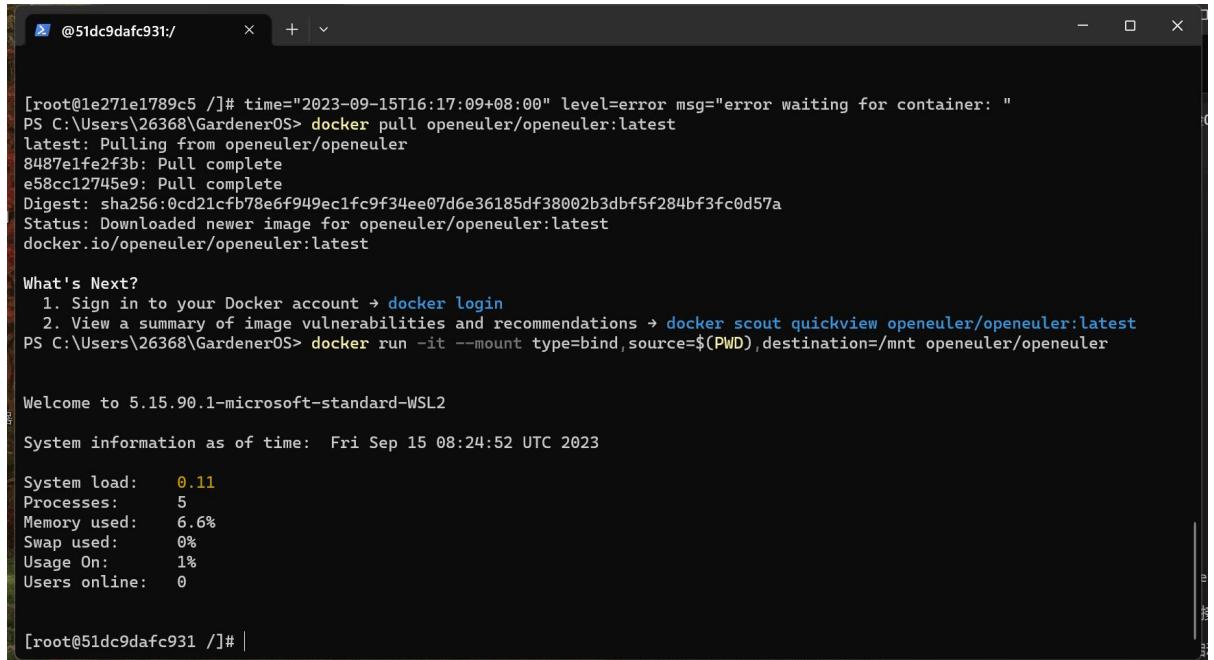
### 3. 创建自己的 docker 镜像

首先进入自己创建的实验目录

docker run -it --mount type=bind,source=\$(PWD),destination=/mnt openeuler/openeuler

通过如上命令启动容器后，本地创建的实验目录就可以直接通过容器里的/mnt 目录

直接访问了。



```
[root@1e271e1789c5 /]# time="2023-09-15T16:17:09+08:00" level=error msg="error waiting for container: "
PS C:\Users\26368\GardenerOS> docker pull openeuler/openeuler:latest
latest: Pulling from openeuler/openeuler
8487e1fe2f3b: Pull complete
e58cc12745e9: Pull complete
Digest: sha256:0cd21cfb78e6f949ec1fc9f34ee07d6e36185df38002b3dbf5f284bf3fc0d57a
Status: Downloaded newer image for openeuler/openeuler:latest
docker.io/openeuler/openeuler:latest

What's Next?
 1. Sign in to your Docker account → docker login
 2. View a summary of image vulnerabilities and recommendations → docker scout quickview openeuler/openeuler:latest
PS C:\Users\26368\GardenerOS> docker run -it --mount type=bind,source=$(PWD),destination=/mnt openeuler/openeuler

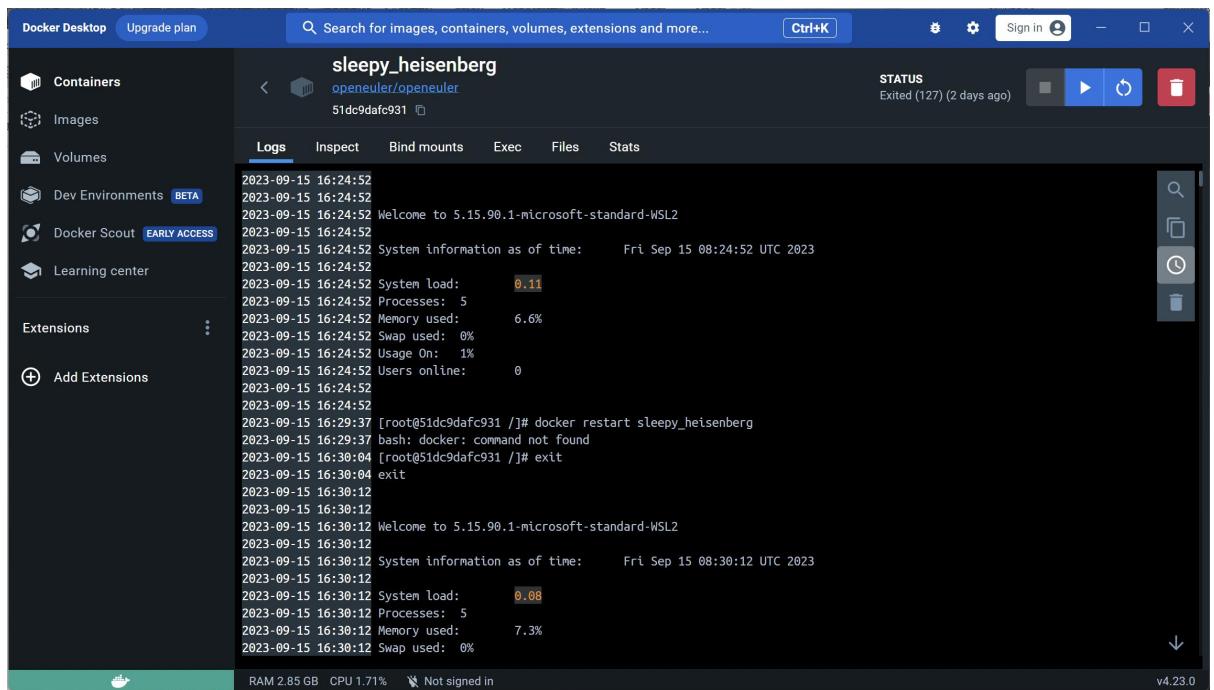
Welcome to 5.15.90.1-microsoft-standard-WSL2

System information as of time: Fri Sep 15 08:24:52 UTC 2023

System load: 0.11
Processes: 5
Memory used: 6.6%
Swap used: 0%
Usage On: 1%
Users online: 0

[root@51dc9dafc931 /]# |
```

在 docker desktop 中查看到容器名称为 sleepy\_heisenberg



重新启动以及访问容器命令如下。

docker restart sleepy\_heisenberg

docker attach sleepy\_heisenberg

```
Digest: sha256:0cd21cfb78e6f949ec1fc9f34ee07d6e36185df38002b3dbf5f284bf3fc0d57a
Status: Downloaded newer image for openeuler/openeuler:latest
docker.io/openeuler/openeuler:latest

What's Next?
 1. Sign in to your Docker account → docker login
 2. View a summary of image vulnerabilities and recommendations → docker scout quickview openeuler/openeuler:latest
PS C:\Users\26368\GardenerOS> docker run -it --mount type=bind,source=$(PWD),destination=/mnt openeuler/openeuler

Welcome to 5.15.90.1-microsoft-standard-WSL2

System information as of time: Fri Sep 15 08:24:52 UTC 2023

System load: 0.11
Processes: 5
Memory used: 6.6%
Swap used: 0%
Usage On: 1%
Users online: 0

[root@51dc9dafc931 /]# docker restart sleepy_heisenberg
bash: docker: command not found
[root@51dc9dafc931 /]# exit
exit
PS C:\Users\26368\GardenerOS> docker restart sleepy_heisenberg
sleepy_heisenberg
PS C:\Users\26368\GardenerOS> docker attach sleepy_heisenberg
[root@51dc9dafc931 /]#
```

安装一些必要的工具：

dnf install curl vim gcc

中间确认输入 y 确认

```
@51dc9dafc931:/ Verifying : gmp-c++-1:6.2.1-2.oe2203sp1.x86_64 7/21
Verifying : gmp-devel-1:6.2.1-2.oe2203sp1.x86_64 8/21
Verifying : gpm-libs-1.20.7-26.oe2203sp1.x86_64 9/21
Verifying : libmpc-1.2.0-3.oe2203sp1.x86_64 10/21
Verifying : libxcrypt-devel-4.4.26-4.oe2203sp1.x86_64 11/21
Verifying : isl-0.16.1-11.oe2203sp1.x86_64 12/21
Verifying : isl-devel-0.16.1-11.oe2203sp1.x86_64 13/21
Verifying : binutils-2.37-23.oe2203sp1.x86_64 14/21
Verifying : cpp-10.3.1-24.oe2203sp1.x86_64 15/21
Verifying : gcc-10.3.1-24.oe2203sp1.x86_64 16/21
Verifying : glibc-devel-2.34-112.oe2203sp1.x86_64 17/21
Verifying : kernel-headers-5.10.0-136.48.0.126.oe2203sp1.x86_64 18/21
Verifying : vim-common-2:9.0-15.oe2203sp1.x86_64 19/21
Verifying : vim-enhanced-2:9.0-15.oe2203sp1.x86_64 20/21
Verifying : vim-filesystem-2:9.0-15.oe2203sp1.noarch 21/21

Downgraded:
glibc-2.34-112.oe2203sp1.x86_64      glibc-common-2.34-112.oe2203sp1.x86_64      libxcrypt-4.4.26-4.oe2203sp1.x86_64

Installed:
binutils-2.37-23.oe2203sp1.x86_64      cpp-10.3.1-24.oe2203sp1.x86_64
gcc-10.3.1-24.oe2203sp1.x86_64      glibc-devel-2.34-112.oe2203sp1.x86_64
gmp-c++-1:6.2.1-2.oe2203sp1.x86_64      gmp-devel-1:6.2.1-2.oe2203sp1.x86_64
gpm-libs-1.20.7-26.oe2203sp1.x86_64      isl-0.16.1-11.oe2203sp1.x86_64
isl-devel-0.16.1-11.oe2203sp1.x86_64      kernel-headers-5.10.0-136.48.0.126.oe2203sp1.x86_64
libmpc-1.2.0-3.oe2203sp1.x86_64      libxcrypt-devel-4.4.26-4.oe2203sp1.x86_64
vim-common-2:9.0-15.oe2203sp1.x86_64      vim-enhanced-2:9.0-15.oe2203sp1.x86_64
vim-filesystem-2:9.0-15.oe2203sp1.noarch

Complete!
[root@51dc9dafc931 /]#
```

可以看到安装成功

## 4. 配置 Rust 开发环境

### 4.1 通过如下命令安装 rust。

cd ~

vim .bashrc

修改.bashrc 增加如下内容（增加位置如图）：

```
export RUSTUP_DIST_SERVER=https://mirrors.ustc.edu.cn/rust-static  
export RUSTUP_UPDATE_ROOT=https://mirrors.ustc.edu.cn/rust-static/rustup
```

然后，通过如下命令安装 rust。

```
curl https://sh.rustup.rs -sSf | sh
```

### 选择 1 (默认安装)

```
✖ @51dc9dafc931:~ x + v - □ ×

info: downloading component 'rust-docs'
13.7 MiB / 13.7 MiB (100 %) 4.9 MiB/s in 3s ETA: 0s
info: downloading component 'rust-std'
info: downloading component 'rustc'
63.7 MiB / 63.7 MiB (100 %) 2.7 MiB/s in 6s ETA: 0s
info: downloading component 'rustfmt'
info: installing component 'cargo'
info: installing component 'clippy'
info: installing component 'rust-docs'
13.7 MiB / 13.7 MiB (100 %) 13.3 MiB/s in 1s ETA: 0s
info: installing component 'rust-std'
26.8 MiB / 26.8 MiB (100 %) 18.0 MiB/s in 1s ETA: 0s
info: installing component 'rustc'
63.7 MiB / 63.7 MiB (100 %) 20.5 MiB/s in 3s ETA: 0s
info: installing component 'rustfmt'
info: default toolchain set to 'stable-x86_64-unknown-linux-gnu'

stable-x86_64-unknown-linux-gnu installed - rustc 1.72.0 (5680fa18f 2023-08-23)

Rust is installed now. Great!

To get started you may need to restart your current shell.
This would reload your PATH environment variable to include
Cargo's bin directory ($HOME/.cargo/bin).

To configure your current shell, run:
source "$HOME/.cargo/env"
[root@51dc9dafc931 ~]# source "$HOME/.cargo/env"
[root@51dc9dafc931 ~]#
```

可以看到安装完成

执行如下命令，使得安装环境启用。

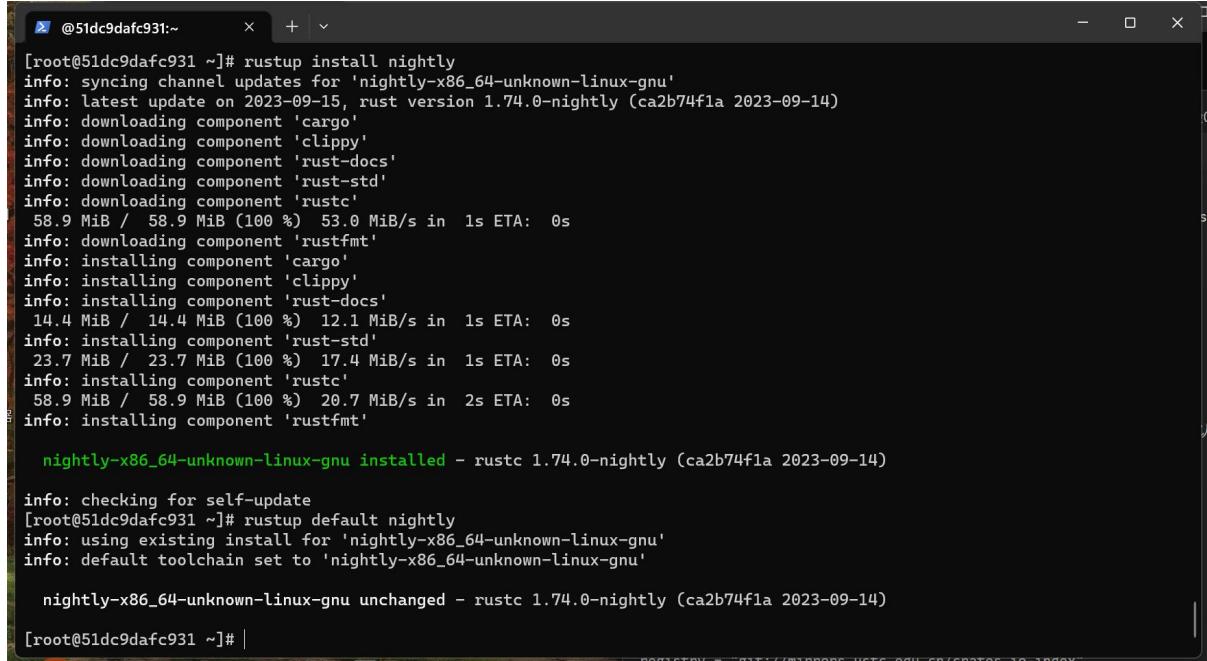
```
source "$HOME/.cargo/env"
```

操作系统实验依赖于 nightly 版本的 rust，因此需要安装 nightly 版本，并将 rust 默认设置为 nightly。

执行如下命令：

```
rustup install nightly
```

```
rustup default nightly
```



The screenshot shows a terminal window with the following command history:

```
[root@51dc9dafc931 ~]# rustup install nightly
info: syncing channel updates for 'nightly-x86_64-unknown-linux-gnu'
info: latest update on 2023-09-15, rust version 1.74.0-nightly (ca2b74f1a 2023-09-14)
info: downloading component 'cargo'
info: downloading component 'clippy'
info: downloading component 'rust-docs'
info: downloading component 'rust-std'
info: downloading component 'rustc'
58.9 MiB / 58.9 MiB (100 %) 53.0 MiB/s in 1s ETA: 0s
info: downloading component 'rustfmt'
info: installing component 'cargo'
info: installing component 'clippy'
info: installing component 'rust-docs'
14.4 MiB / 14.4 MiB (100 %) 12.1 MiB/s in 1s ETA: 0s
info: installing component 'rust-std'
23.7 MiB / 23.7 MiB (100 %) 17.4 MiB/s in 1s ETA: 0s
info: installing component 'rustc'
58.9 MiB / 58.9 MiB (100 %) 20.7 MiB/s in 2s ETA: 0s
info: installing component 'rustfmt'

nightly-x86_64-unknown-linux-gnu installed - rustc 1.74.0-nightly (ca2b74f1a 2023-09-14)

info: checking for self-update
[root@51dc9dafc931 ~]# rustup default nightly
info: using existing install for 'nightly-x86_64-unknown-linux-gnu'
info: default toolchain set to 'nightly-x86_64-unknown-linux-gnu'

nightly-x86_64-unknown-linux-gnu unchanged - rustc 1.74.0-nightly (ca2b74f1a 2023-09-14)

[root@51dc9dafc931 ~]# |
```

## 4.2 切换 cargo 软件包镜像为 tuna

```
vim ~/.cargo/config
```

文件内容如下：

```
[source.crates-io]
replace-with = 'ustc'
```

```
[source.ustc]
registry = "git://mirrors.ustc.edu.cn/crates.io-index"
```

## 4.3 接下来安装一些 Rust 相关的软件包

```
rustup target add riscv64gc-unknown-none-elf
```

```
cargo install cargo-binutils
```

```
✖ @51dc9d9fc931:~ x + ▾ - □ ×
info: downloading component 'rustc'
info: downloading component 'rustc'
  58.9 MiB /  58.9 MiB (100 %)  53.0 MiB/s in  1s ETA:  0s
info: downloading component 'rustfmt'
info: installing component 'cargo'
info: installing component 'clippy'
info: installing component 'rust-docs'
  14.4 MiB /  14.4 MiB (100 %)  12.1 MiB/s in  1s ETA:  0s
info: installing component 'rustc'
  23.7 MiB /  23.7 MiB (100 %)  17.4 MiB/s in  1s ETA:  0s
info: installing component 'rustc'
  58.9 MiB /  58.9 MiB (100 %)  20.7 MiB/s in  2s ETA:  0s
info: installing component 'rustfmt'

nightly-x86_64-unknown-linux-gnu installed - rustc 1.74.0-nightly (ca2b74f1a 2023-09-14)

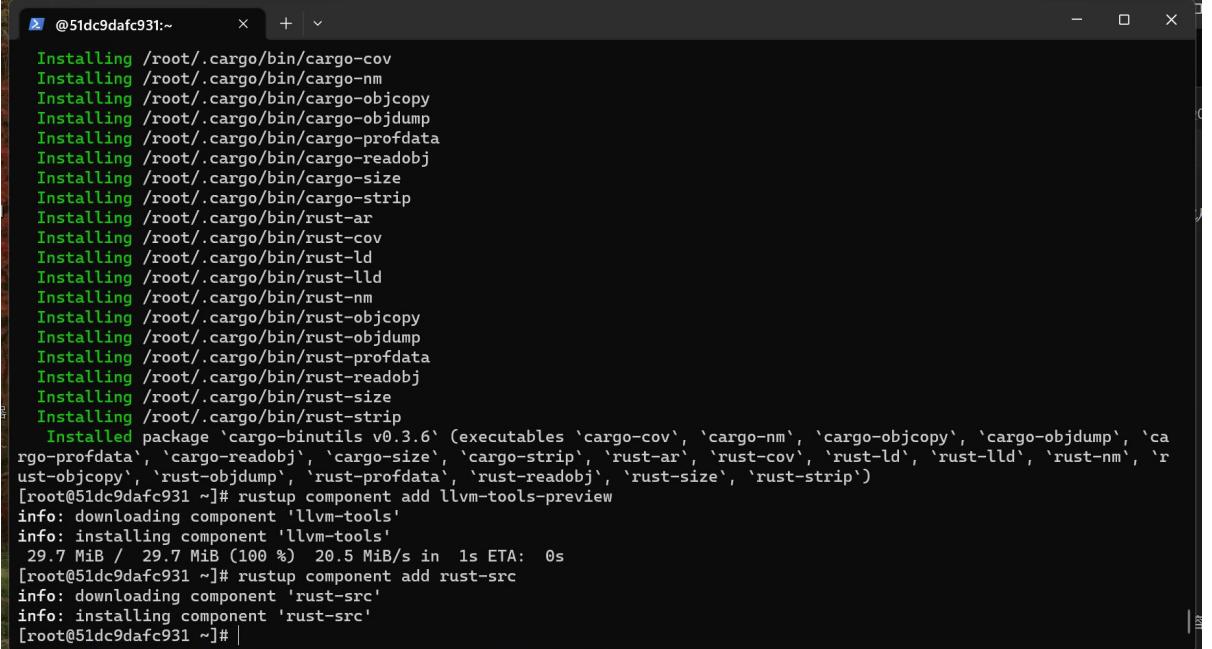
info: checking for self-update
[root@51dc9d9fc931 ~]# rustup default nightly
info: using existing install for 'nightly-x86_64-unknown-linux-gnu'
info: default toolchain set to 'nightly-x86_64-unknown-linux-gnu'

nightly-x86_64-unknown-linux-gnu unchanged - rustc 1.74.0-nightly (ca2b74f1a 2023-09-14)

[root@51dc9d9fc931 ~]# vim ~/.cargo/config
[root@51dc9d9fc931 ~]# rustup target add riscv64gc-unknown-none-elf
info: downloading component 'rustc' for 'riscv64gc-unknown-none-elf'
info: installing component 'rustc' for 'riscv64gc-unknown-none-elf'
[root@51dc9d9fc931 ~]# cargo install cargo-binutils
  Updating 'ustc' index
```

```
rustup component add llvm-tools-preview
```

```
rustup component add rust-src
```



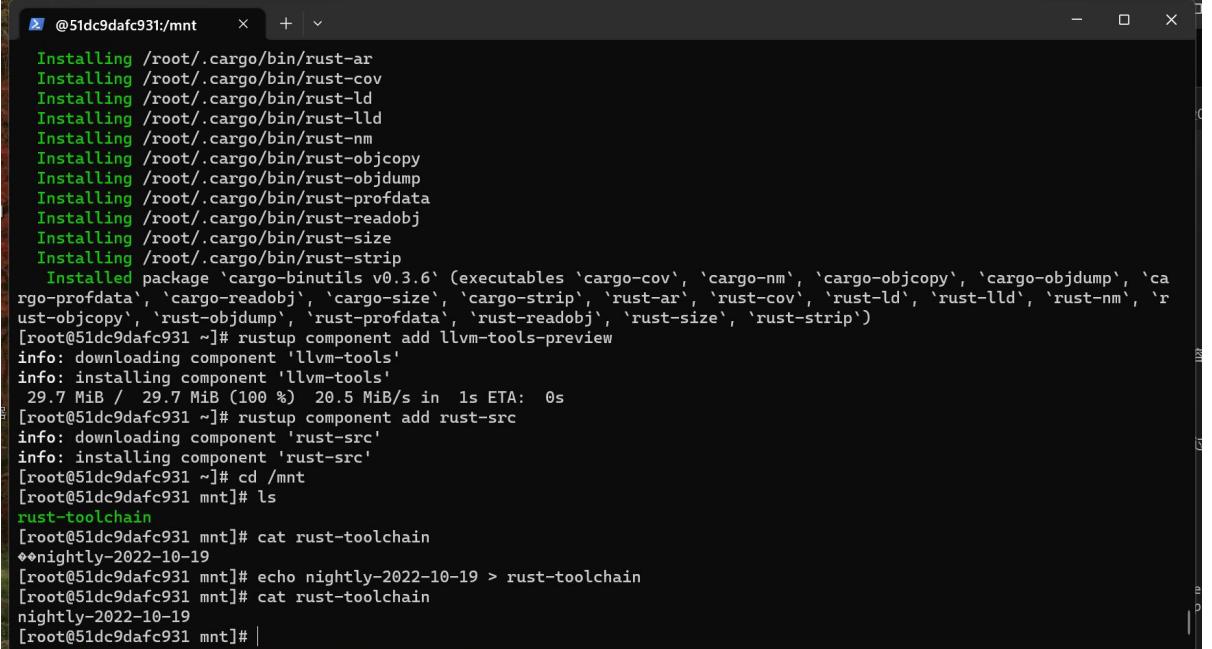
```
Installing /root/.cargo/bin/cargo-cov
Installing /root/.cargo/bin/cargo-nm
Installing /root/.cargo/bin/cargo-objcopy
Installing /root/.cargo/bin/cargo-objdump
Installing /root/.cargo/bin/cargo-profdata
Installing /root/.cargo/bin/cargo-readobj
Installing /root/.cargo/bin/cargo-size
Installing /root/.cargo/bin/cargo-strip
Installing /root/.cargo/bin/rust-ar
Installing /root/.cargo/bin/rust-cov
Installing /root/.cargo/bin/rust-ld
Installing /root/.cargo/bin/rust-lld
Installing /root/.cargo/bin/rust-nm
Installing /root/.cargo/bin/rust-objcopy
Installing /root/.cargo/bin/rust-objdump
Installing /root/.cargo/bin/rust-profdata
Installing /root/.cargo/bin/rust-readobj
Installing /root/.cargo/bin/rust-size
Installing /root/.cargo/bin/rust-strip
    Installed package `cargo-binutils v0.3.6` (executables `cargo-cov`, `cargo-nm`, `cargo-objcopy`, `cargo-objdump`, `cargo-profdata`, `cargo-readobj`, `cargo-size`, `cargo-strip`, `rust-ar`, `rust-cov`, `rust-ld`, `rust-lld`, `rust-nm`, `rust-objcopy`, `rust-objdump`, `rust-profdata`, `rust-readobj`, `rust-size`, `rust-strip`)
[root@51dc9dafc931 ~]# rustup component add llvm-tools-preview
info: downloading component 'llvm-tools'
info: installing component 'llvm-tools'
29.7 MiB / 29.7 MiB (100%) 20.5 MiB/s in 1s ETA: 0s
[root@51dc9dafc931 ~]# rustup component add rust-src
info: downloading component 'rust-src'
info: installing component 'rust-src'
[root@51dc9dafc931 ~]#
```

#### 4.4 限制 rust 的版本

在工作目录下创建一个名为 `rust-toolchain` 的文件，以限制 `rust` 的版本，文件内容如下：

`nightly-2022-10-19`

命令为 `echo nightly-2022-10-19 > rust-toolchain`



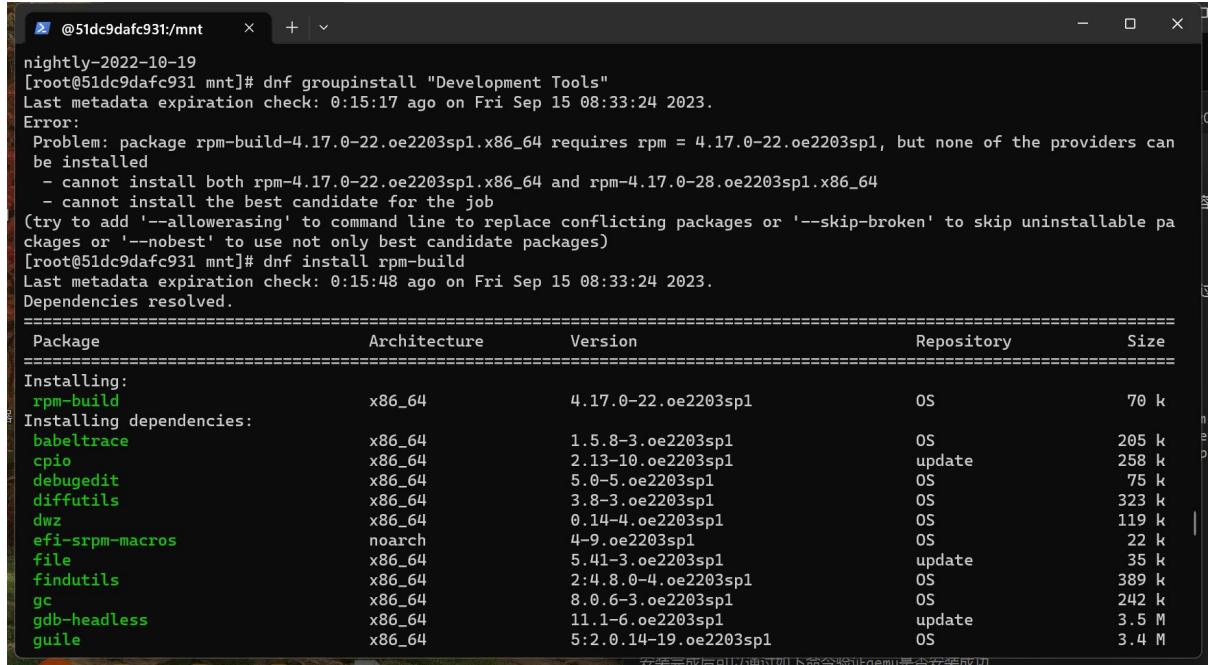
```
Installing /root/.cargo/bin/rust-ar
Installing /root/.cargo/bin/rust-cov
Installing /root/.cargo/bin/rust-ld
Installing /root/.cargo/bin/rust-lld
Installing /root/.cargo/bin/rust-nm
Installing /root/.cargo/bin/rust-objcopy
Installing /root/.cargo/bin/rust-objdump
Installing /root/.cargo/bin/rust-profdata
Installing /root/.cargo/bin/rust-readobj
Installing /root/.cargo/bin/rust-size
Installing /root/.cargo/bin/rust-strip
    Installed package `cargo-binutils v0.3.6` (executables `cargo-cov`, `cargo-nm`, `cargo-objcopy`, `cargo-objdump`, `cargo-profdata`, `cargo-readobj`, `cargo-size`, `cargo-strip`, `rust-ar`, `rust-cov`, `rust-ld`, `rust-lld`, `rust-nm`, `rust-objcopy`, `rust-objdump`, `rust-profdata`, `rust-readobj`, `rust-size`, `rust-strip`)
[root@51dc9dafc931 ~]# rustup component add llvm-tools-preview
info: downloading component 'llvm-tools'
info: installing component 'llvm-tools'
29.7 MiB / 29.7 MiB (100%) 20.5 MiB/s in 1s ETA: 0s
[root@51dc9dafc931 ~]# rustup component add rust-src
info: downloading component 'rust-src'
info: installing component 'rust-src'
[root@51dc9dafc931 ~]# cd /mnt
[root@51dc9dafc931 mnt]# ls
rust-toolchain
[root@51dc9dafc931 mnt]# cat rust-toolchain
◆◆nightly-2022-10-19
[root@51dc9dafc931 mnt]# echo nightly-2022-10-19 > rust-toolchain
[root@51dc9dafc931 mnt]# cat rust-toolchain
nightly-2022-10-19
[root@51dc9dafc931 mnt]#
```

说明：访问工作目录有两种方式：在本地通过`~/GardenerOS/`目录访问；在容器内通过`/mnt/`目录访问。

## 5. 安装 qemu

### 5.1 执行如下命令安装基本的软件包

dnf groupinstall "Development Tools" 未成功——需先执行 dnf install rpm-build 然后  
再执行



```
nightly-2022-10-19
[root@51dc9dafc931 mnt]# dnf groupinstall "Development Tools"
Last metadata expiration check: 0:15:17 ago on Fri Sep 15 08:33:24 2023.
Error:
  Problem: package rpm-build-4.17.0-22.oe2203sp1.x86_64 requires rpm = 4.17.0-22.oe2203sp1, but none of the providers can
  be installed
    - cannot install both rpm-4.17.0-22.oe2203sp1.x86_64 and rpm-4.17.0-28.oe2203sp1.x86_64
    - cannot install the best candidate for the job
  (try to add '--allowerasing' to command line to replace conflicting packages or '--skip-broken' to skip un/installable pa
ckages or '--nobest' to use not only best candidate packages)
[root@51dc9dafc931 mnt]# dnf install rpm-build
Last metadata expiration check: 0:15:48 ago on Fri Sep 15 08:33:24 2023.
Dependencies resolved.
=====
 Package          Architecture      Version       Repository      Size
 =====
Installing:
  rpm-build        x86_64          4.17.0-22.oe2203sp1   OS            70 k
Installing dependencies:
  babeltrace      x86_64          1.5.8-3.oe2203sp1   OS            205 k
  cpio             x86_64          2.13-10.oe2203sp1  update         258 k
  debugedit        x86_64          5.0-5.oe2203sp1    OS            75 k
  diffutils        x86_64          3.8-3.oe2203sp1    OS            323 k
  dwz              x86_64          0.14-4.oe2203sp1   OS            119 k
  efi-srpm-macros noarch          4-9.oe2203sp1     OS            22 k
  file             x86_64          5.41-3.oe2203sp1  update         35 k
  findutils        x86_64          2:4.8.0-4.oe2203sp1 OS            389 k
  gc               x86_64          8.0.6-3.oe2203sp1   OS            242 k
  gdb-headless     x86_64          11.1-6.oe2203sp1  update         3.5 M
  guile            x86_64          5:2.0.14-19.oe2203sp1 OS            3.4 M
```

dnf install autoconf automake gcc gcc-c++ kernel-devel curl libmpc-devel mpfr-devel

gmp-devel \

glib2 glib2-devel make cmake gawk bison flex texinfo gperf libtool

patchutils bc \

python3 ninja-build wget xz

```
@51dc9dafc931:/mnt      x  +  v
a libglvnd-1:1.3.4-3.oe2203sp1.x86_64
a libglvnd-devel-1:1.3.4-3.oe2203sp1.x86_64
a libglvnd-gles-1:1.3.4-3.oe2203sp1.x86_64
a libglvnd-opengl-1:1.3.4-3.oe2203sp1.x86_64
a libmetalink-0.1.3-10.oe2203sp1.x86_64
a libpciaccess-0.16-3.oe2203sp1.x86_64
a libpng-devel-2:1.6.38-1.oe2203sp1.x86_64
a libsepol-devel-3.3-4.oe2203sp1.x86_64
a libthai-devel-0.1.29-1.oe2203sp1.x86_64
a libuv-1:1.42.0-5.oe2203sp1.x86_64
a libxkbcommon-devel-1.3.1-3.oe2203sp1.x86_64
a libxshmfence-1.3-9.oe2203sp1.x86_64
a mesa-libEGL-21.3.1-3.oe2203sp1.x86_64
a mesa-libgbm-21.3.1-3.oe2203sp1.x86_64
a mpfr-devel-4.1.0-2.oe2203sp1.x86_64
a pango-devel-1.50.7-1.oe2203sp1.x86_64
a pcre2-devel-10.39-9.oe2203sp1.x86_64
a perl-File-ShareDir-1.118-2.oe2203sp1.noarch
a perl-Unicode-EastAsianWidth-12.0-2.oe2203sp1.noarch
a pixman-devel-0.40.0-3.oe2203sp1.x86_64
a polkit-help-0.120-8.oe2203sp1.noarch
a sysprof-3.38.1-3.oe2203sp1.x86_64
a sysprof-devel-3.38.1-3.oe2203sp1.x86_64
a util-linux-devel-2.37.2-13.oe2203sp1.x86_64
a wayland-protocols-devel-1.25-1.oe2203sp1.noarch
a xorg-x11proto-devel-2021.5-2.oe2203sp1.noarch
a zlib-devel-1.2.11-22.oe2203sp1.x86_64

Complete!
[root@51dc9dafc931 mnt]#
```

## 5.2 通过源码安装 qemu 5.2 并验证安装是否成功

```
wget https://download.qemu.org/qemu-5.2.0.tar.xz
```

```
tar xvJf qemu-5.2.0.tar.XZ
```

```
@51dc9dafc931:/mnt      x  +  v
a qemu-5.2.0/pc-bios/efi-vmxnet3.rom
a qemu-5.2.0/pc-bios/vgabios-virtio.bin
a qemu-5.2.0/pc-bios/skiboot.lid
a qemu-5.2.0/pc-bios/hppa-firmware.img
a qemu-5.2.0/pc-bios/pxe-e1000.rom
a qemu-5.2.0/pc-bios/efi-e1000e.rom
a qemu-5.2.0/pc-bios/bios-256k.bin
a qemu-5.2.0/pc-bios/npcm7xx_bootrom.bin
a qemu-5.2.0/pc-bios/multiboot.bin
a qemu-5.2.0/pc-bios/qemu-nsis.bmp
a qemu-5.2.0/pc-bios/descriptors/
a qemu-5.2.0/pc-bios/descriptors/60-edk2-x86_64.json
a qemu-5.2.0/pc-bios/descriptors/60-edk2-aarch64.json
a qemu-5.2.0/pc-bios/descriptors/50-edk2-x86_64-secure.json
a qemu-5.2.0/pc-bios/descriptors/60-edk2-arm.json
a qemu-5.2.0/pc-bios/descriptors/meson.build
a qemu-5.2.0/pc-bios/descriptors/50-edk2-i386-secure.json
a qemu-5.2.0/pc-bios/descriptors/60-edk2-i386.json
a qemu-5.2.0/pc-bios/vgabios-ati.bin
a qemu-5.2.0/pc-bios/vgabios-vmware.bin
a qemu-5.2.0/pc-bios/efi-eepro100.rom
a qemu-5.2.0/pc-bios/s390-csw.img
a qemu-5.2.0/pc-bios/opensbi-riscv32-generic-fw_dynamic.bin
a qemu-5.2.0/pc-bios/edk2-arm-code.fd.bz2
a qemu-5.2.0/pc-bios/linuxboot.bin
a qemu-5.2.0/configure
a qemu-5.2.0/qemu.nsi
a qemu-5.2.0/kconfig
a qemu-5.2.0/.gdbinit
[root@51dc9dafc931 mnt]#
```

```
cd qemu-5.2.0
```

```
@51dc9d9fc931:/mnt/qemu + | v
a qemu-5.2.0/pc-bios/vgabios-virtio.bin
qemu-5.2.0/pc-bios/skiboot.lid
l qemu-5.2.0/pc-bios/hppa-firmware.img
qemu-5.2.0/pc-bios/pxe-e1000.rom
qemu-5.2.0/pc-bios/efi-e1000e.rom
v qemu-5.2.0/pc-bios/bios-256k.bin
qemu-5.2.0/pc-bios/npcm7xx-bootrom.bin
qemu-5.2.0/pc-bios/multiboot.bin
c qemu-5.2.0/pc-bios/qemu-nsis.bmp
qemu-5.2.0/pc-bios/descriptors/
a qemu-5.2.0/pc-bios/descriptors/60-edk2-x86_64.json
qemu-5.2.0/pc-bios/descriptors/60-edk2-aarch64.json
qemu-5.2.0/pc-bios/descriptors/50-edk2-x86_64-secure.json
qemu-5.2.0/pc-bios/descriptors/60-edk2-arm.json
p qemu-5.2.0/pc-bios/descriptors/meson.build
qemu-5.2.0/pc-bios/descriptors/50-edk2-i386-secure.json
qemu-5.2.0/pc-bios/descriptors/60-edk2-i386.json
c qemu-5.2.0/pc-bios/vgabios-ati.bin
qemu-5.2.0/pc-bios/vgabios-vmware.bin
qemu-5.2.0/pc-bios/efi-eepro100.rom
qemu-5.2.0/pc-bios/s390-ccw.img
qemu-5.2.0/pc-bios/opensbi-riscv32-generic-fw_dynamic.bin
qemu-5.2.0/pc-bios/edk2-arm-code.fd.bz2
qemu-5.2.0/pc-bios/linuxboot.bin
qemu-5.2.0/configure
qemu-5.2.0/qemu.nsi
qemu-5.2.0/kconfig
qemu-5.2.0/.gdbinit
[root@51dc9d9fc931 mnt]# cd qemu-5.2.0
[root@51dc9d9fc931 qemu-5.2.0]# |
2022-09-15 17:22:34 qemu-5.2.0/qemu_nsi
```

./configure --target-list=riscv64-softmmu,riscv64-linux-user

```
@51dc9d9fc931:/mnt/qemu + | v
a          zstd support: NO
NUMA host support: NO
        libxml2: YES
memory allocator: system
avx2 optimization: YES
avx512f optimization: NO
replication support: YES
bochs support: YES
cloop support: YES
dmg support: YES
qcow v1 support: YES
vdi support: YES
vvfat support: YES
qed support: YES
parallels support: YES
sheepdog support: NO
        capstone: internal
libpmem support: NO
libdaxctl support: NO
        libudev: NO
default devices: YES
plugin support: NO
fuzzing support: NO
        gdb: /usr/bin/gdb
thread sanitizer: NO
        rng-none: NO
Linux keyring: YES

Found ninja-1.10.2 at /usr/bin/ninja
[root@51dc9d9fc931 qemu-5.2.0]# |
2022-09-15 18:00:50
```

make -j\$(nproc) install

```

@51dc9d9fc931:/mnt/qemu x + v
a Installing /mnt/qemu-5.2.0/pc-bios/u-boot.e500 to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/pc-bios/u-boot-sam460-20100605.bin to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/pc-bios/qemu_vga.ndrv to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/pc-bios/edk2-licenses.txt to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/build/po/bg.gmo to /usr/local/share/locale/bg/LC_MESSAGES/qemu.mo
v Installing /mnt/qemu-5.2.0/build/po/de_DE.gmo to /usr/local/share/locale/de_DE/LC_MESSAGES/qemu.mo
Installing /mnt/qemu-5.2.0/build/po/fr_FR.gmo to /usr/local/share/locale/fr_FR/LC_MESSAGES/qemu.mo
Installing /mnt/qemu-5.2.0/build/po/hu.gmo to /usr/local/share/locale/hu/LC_MESSAGES/qemu.mo
Installing /mnt/qemu-5.2.0/build/po/it.gmo to /usr/local/share/locale/it/LC_MESSAGES/qemu.mo
Installing /mnt/qemu-5.2.0/build/po/sv.gmo to /usr/local/share/locale/sv/LC_MESSAGES/qemu.mo
a Installing /mnt/qemu-5.2.0/build/po/tr.gmo to /usr/local/share/locale/tr/LC_MESSAGES/qemu.mo
Installing /mnt/qemu-5.2.0/build/po/zh_CN.gmo to /usr/local/share/locale/zh_CN/LC_MESSAGES/qemu.mo
Installing /mnt/qemu-5.2.0/pc-bios/hppa-firmware.img to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/pc-bios/opensbi-riscv32-generic-fw_dynamic.bin to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/pc-bios/opensbi-riscv64-generic-fw_dynamic.bin to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/pc-bios/opensbi-riscv32-generic-fw_dynamic.elf to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/pc-bios/opensbi-riscv64-generic-fw_dynamic.elf to /usr/local/share/qemu
c Installing /mnt/qemu-5.2.0/pc-bios/npcm7xx_bootrom.bin to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/build/pc-bios/descriptors/50-edk2-i386-secure.json to /usr/local/share/qemu/firmware
Installing /mnt/qemu-5.2.0/build/pc-bios/descriptors/50-edk2-x86_64-secure.json to /usr/local/share/qemu/firmware
Installing /mnt/qemu-5.2.0/build/pc-bios/descriptors/60-edk2-aarch64.json to /usr/local/share/qemu/firmware
Installing /mnt/qemu-5.2.0/build/pc-bios/descriptors/60-edk2-arm.json to /usr/local/share/qemu/firmware
Installing /mnt/qemu-5.2.0/build/pc-bios/descriptors/60-edk2-i386.json to /usr/local/share/qemu/firmware
Installing /mnt/qemu-5.2.0/build/pc-bios/descriptors/60-edk2-x86_64.json to /usr/local/share/qemu/firmware
Installing /mnt/qemu-5.2.0/build/pc-bios/keymaps/sl to /usr/local/share/qemu/keymaps
Installing /mnt/qemu-5.2.0/build/pc-bios/keymaps/sv to /usr/local/share/qemu/keymaps
Running custom install script '/usr/bin/python3 /mnt/qemu-5.2.0/meson/meson.py --internal gettext install --subdir=po --
localedir=share/locale --pkgnname=qemu'
make[1]: Leaving directory '/mnt/qemu-5.2.0/build'
[root@51dc9d9fc931 qemu-5.2.0]#

```

安装完成后可以通过如下命令验证 qemu 是否安装成功。

`qemu-system-riscv64 --version`

`qemu-riscv64 --version`

```

sleepy_heisenberg
@51dc9d9fc931:/mnt/qemu x + v
Installing /mnt/qemu-5.2.0/build/po/fr_FR.gmo to /usr/local/share/locale/fr_FR/LC_MESSAGES/qemu.mo
Installing /mnt/qemu-5.2.0/build/po/hu.gmo to /usr/local/share/locale/hu/LC_MESSAGES/qemu.mo
Installing /mnt/qemu-5.2.0/build/po/it.gmo to /usr/local/share/locale/it/LC_MESSAGES/qemu.mo
Installing /mnt/qemu-5.2.0/build/po/sv.gmo to /usr/local/share/locale/sv/LC_MESSAGES/qemu.mo
Installing /mnt/qemu-5.2.0/build/po/tr.gmo to /usr/local/share/locale/tr/LC_MESSAGES/qemu.mo
Installing /mnt/qemu-5.2.0/build/po/zh_CN.gmo to /usr/local/share/locale/zh_CN/LC_MESSAGES/qemu.mo
Installing /mnt/qemu-5.2.0/pc-bios/hppa-firmware.img to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/pc-bios/opensbi-riscv32-generic-fw_dynamic.bin to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/pc-bios/opensbi-riscv64-generic-fw_dynamic.bin to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/pc-bios/opensbi-riscv32-generic-fw_dynamic.elf to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/pc-bios/opensbi-riscv64-generic-fw_dynamic.elf to /usr/local/share/qemu
c Installing /mnt/qemu-5.2.0/pc-bios/npcm7xx_bootrom.bin to /usr/local/share/qemu
Installing /mnt/qemu-5.2.0/build/pc-bios/descriptors/50-edk2-i386-secure.json to /usr/local/share/qemu/firmware
Installing /mnt/qemu-5.2.0/build/pc-bios/descriptors/50-edk2-x86_64-secure.json to /usr/local/share/qemu/firmware
Installing /mnt/qemu-5.2.0/build/pc-bios/descriptors/60-edk2-aarch64.json to /usr/local/share/qemu/firmware
Installing /mnt/qemu-5.2.0/build/pc-bios/descriptors/60-edk2-arm.json to /usr/local/share/qemu/firmware
Installing /mnt/qemu-5.2.0/build/pc-bios/descriptors/60-edk2-i386.json to /usr/local/share/qemu/firmware
Installing /mnt/qemu-5.2.0/build/pc-bios/descriptors/60-edk2-x86_64.json to /usr/local/share/qemu/firmware
Installing /mnt/qemu-5.2.0/build/pc-bios/keymaps/sl to /usr/local/share/qemu/keymaps
Installing /mnt/qemu-5.2.0/build/pc-bios/keymaps/sv to /usr/local/share/qemu/keymaps
Running custom install script '/usr/bin/python3 /mnt/qemu-5.2.0/meson/meson.py --internal gettext install --subdir=po --
localedir=share/locale --pkgnname=qemu'
make[1]: Leaving directory '/mnt/qemu-5.2.0/build'
[root@51dc9d9fc931 qemu-5.2.0]# qemu-system-riscv64 --version
QEMU emulator version 5.2.0
Copyright (c) 2003-2020 Fabrice Bellard and the QEMU Project developers
[root@51dc9d9fc931 qemu-5.2.0]# qemu-riscv64 --version
qemu-riscv64 version 5.2.0
Copyright (c) 2003-2020 Fabrice Bellard and the QEMU Project developers
[root@51dc9d9fc931 qemu-5.2.0]#

```

通过源码安装qemu 5.2。  
`wget https://download.qemu.org/qemu-5.2.0.tar.xz`  
`tar xvJf qemu-5.2.0.tar.xz`  
`cd qemu-5.2.0`  
`./configure --target-list=riscv64-softmmu,riscv64-linux-gnu`  
`make -j$(nproc) install`

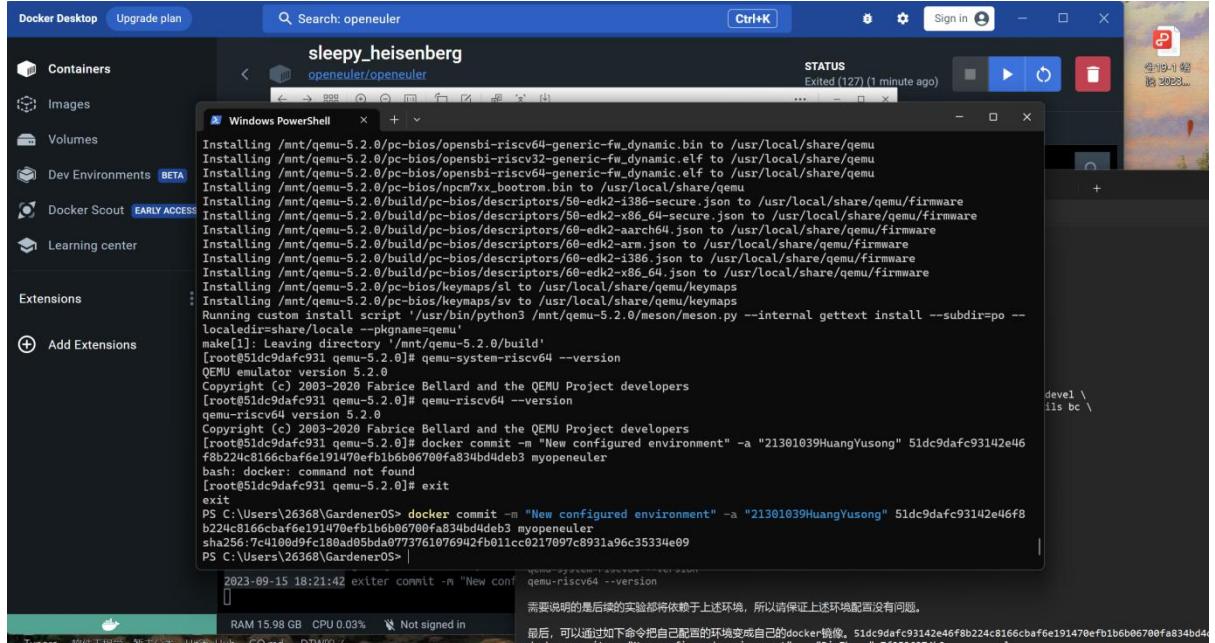
安装完成后可以通过如下命令验证qemu是否安装成功。  
`qemu-system-riscv64 --version`  
`qemu-riscv64 --version`

最后，可以通过如下命令把自己配置的环境变成自己的 docker 镜像。

`docker commit -m "New configured environment" -a "21301039HuangYusong"`

51dc9dafc93142e46f8b224c8166cbaf6e191470efb1b6b06700fa834bd4deb3

myopeneuler



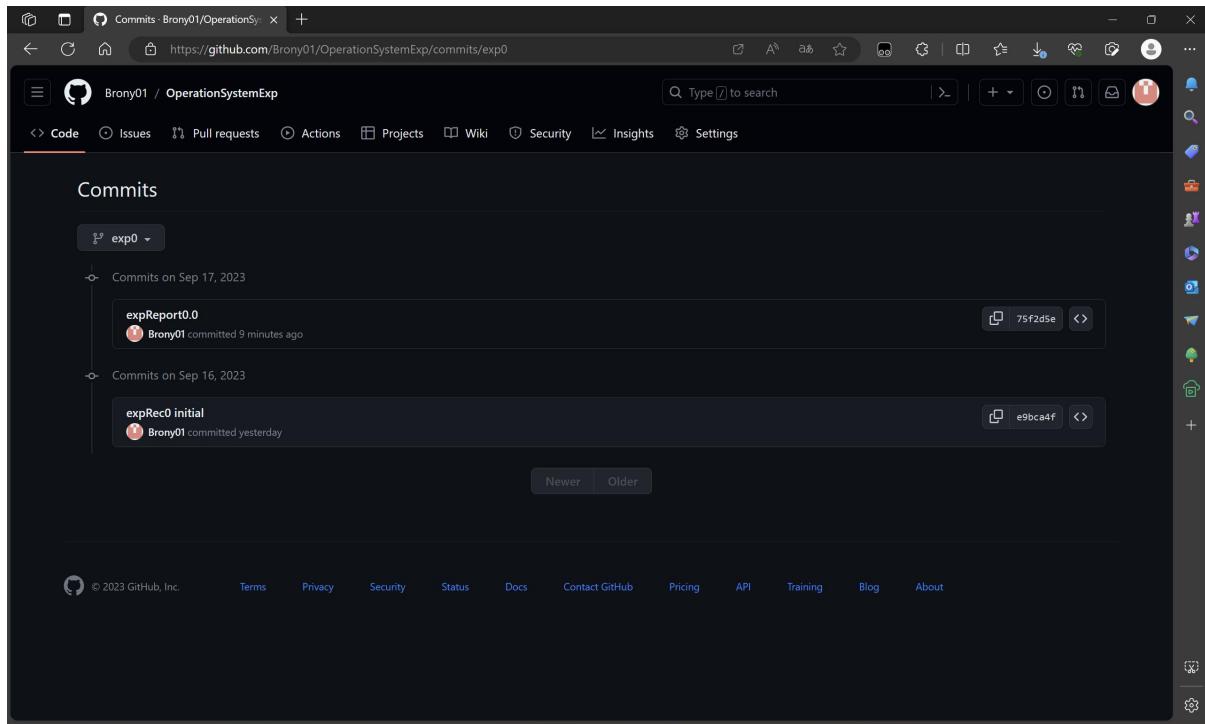
## 二、思考问题

## 三、Git 提交截图

```
Counting objects: 100% (2/2), done.
Delta compression using up to 20 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (2/2), done.
Total 2 (delta 0), reused 0 (delta 0), pack-reused 0
warning: There are too many unreachable loose objects; run 'git prune' to remove them.
[exp0 75f2d5e] expReport0.0
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 ExperimentReport0.docx

26368@HYSLAPTOP MINGW64 ~/GardenerOS (exp0)
$ git push origin exp0
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 20 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 4.33 MiB | 1.41 MiB/s, done.
Total 3 (delta 0), reused 3 (delta 0), pack-reused 0
To github.com:Brony01/OperationSystemExp.git
 e9bca4f..75f2d5e  exp0 -> exp0

26368@HYSLAPTOP MINGW64 ~/GardenerOS (exp0)
$ |
```



#### 四、 其他说明

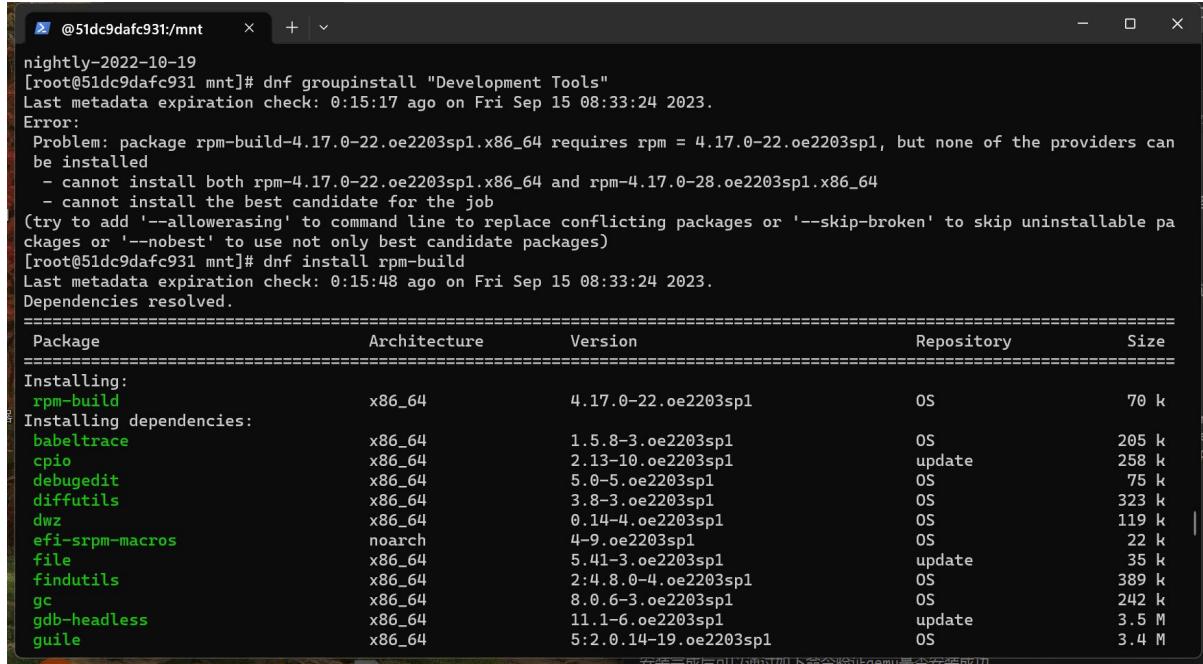
遇到过如下问题（按步骤顺序）

1. 拉取 openeuler 镜像时由于之前配置过 docker 镜像源的原因，直接执行 docker pull openeuler/openeuler 拉取到了较老版本的镜像，这导致后续步骤安装基本软件包执行 dnf groupinstall "Development Tools" 时会报如下错误 Segmentation fault (如下图) 并且一时无法找到其他报错信息。

```
[fab00efbea45] @fab00efbea45/ - + v
[SKIPPED] gettext-devel-0.21-2.oe1.x86_64.rpm: Already downloaded
[SKIPPED] ghostscript-9.52-4.oe1.x86_64.rpm: Already downloaded
[SKIPPED] giflib-5.2.1-1.oe1.x86_64.rpm: Already downloaded
[SKIPPED] google-droid-sans-fonts-20200215-1.oe1.noarch.rpm: Already downloaded
[SKIPPED] graphviz-2.44.0-1.oe1.x86_64.rpm: Already downloaded
[SKIPPED] gstreamer1-plugins-base-1.16.2-1.oe1.x86_64.rpm: Already downloaded
[SKIPPED] httpd-2.4.43-4.oe1.x86_64.rpm: Already downloaded
[SKIPPED] intltowl-0.51.0-14.oe1.noarch.rpm: Already downloaded
[SKIPPED] libtirpc-devel-1.2.6-1.oe1.x86_64.rpm: Already downloaded
[SKIPPED] libtool-2.4.6-32.oe1.x86_64.rpm: Already downloaded
[SKIPPED] ltrace-0.7.91-31.oe1.x86_64.rpm: Already downloaded
[SKIPPED] patchutils-0.3.4-13.oe1.x86_64.rpm: Already downloaded
[SKIPPED] perl-generators-1.10-9.oe1.noarch.rpm: Already downloaded
[SKIPPED] pesign-0.113-4.oe1.x86_64.rpm: Already downloaded
[SKIPPED] source-highlight-3.1.8-24.oe1.x86_64.rpm: Already downloaded
[SKIPPED] subversion-1.12.2-1.oe1.x86_64.rpm: Already downloaded
[SKIPPED] systemtap-4.3-1.oe1.x86_64.rpm: Already downloaded
[SKIPPED] systemtap-client-4.3-1.oe1.x86_64.rpm: Already downloaded
[SKIPPED] systemtap-runtime-4.3-1.oe1.x86_64.rpm: Already downloaded
[SKIPPED] valgrind-devel-3.13.0-29.oe1.x86_64.rpm: Already downloaded
[SKIPPED] webkit2gtk3-2.22.2-10.oe1.x86_64.rpm: Already downloaded
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing   : httpd-2.4.43-4.oe1.x86_64 [=====]
Segmentation fault
[root@fab00efbea45 ~]# ] 1/39S
```

在多次重新配置后用 docker pull openeuler/openeuler:latest 拉取到了最新镜像，上述问题解决。

2. 第一遍执行 dnf groupinstall "Development Tools" 未成功，问题确认为缺乏依赖项，执行如下命令安装依赖项 dnf install rpm-build 后再执行 dnf groupinstall "Development Tools" 即可安装。



```
nightly-2022-10-19
[root@51dc9dafc931 mnt]# dnf groupinstall "Development Tools"
Last metadata expiration check: 0:15:17 ago on Fri Sep 15 08:33:24 2023.
Error:
Problem: package rpm-build-4.17.0-22.oe2203sp1.x86_64 requires rpm = 4.17.0-22.oe2203sp1, but none of the providers can be installed
- cannot install both rpm=4.17.0-22.oe2203sp1.x86_64 and rpm=4.17.0-28.oe2203sp1.x86_64
- cannot install the best candidate for the job
(try to add '--allow-erasing' to command line to replace conflicting packages or '--skip-broken' to skip uninstallable packages or '--nobest' to use not only best candidate packages)
[root@51dc9dafc931 mnt]# dnf install rpm-build
Last metadata expiration check: 0:15:48 ago on Fri Sep 15 08:33:24 2023.
Dependencies resolved.
=====
Package           Architecture   Version        Repository  Size
=====
Installing:
rpm-build        x86_64        4.17.0-22.oe2203sp1    OS          70 k
Installing dependencies:
babeltrace       x86_64        1.5.8-3.oe2203sp1    OS          205 k
cpio              x86_64        2.13-10.oe2203sp1   update      258 k
debugedit         x86_64        5.0-5.oe2203sp1     OS          75 k
diffutils         x86_64        3.8-3.oe2203sp1     OS          323 k
dwz               x86_64        0.14-4.oe2203sp1    OS          119 k
efi-srpm-macros noarch        4-9.oe2203sp1      OS          22 k
file              x86_64        5.41-3.oe2203sp1   update      35 k
findutils         x86_64        2:4.8.0-4.oe2203sp1 OS          389 k
gc                x86_64        8.0.6-3.oe2203sp1    OS          242 k
gdb-headless      x86_64        11.1-6.oe2203sp1   update      3.5 M
guile             x86_64        5:2.0.14-19.oe2203sp1 OS          3.4 M
```