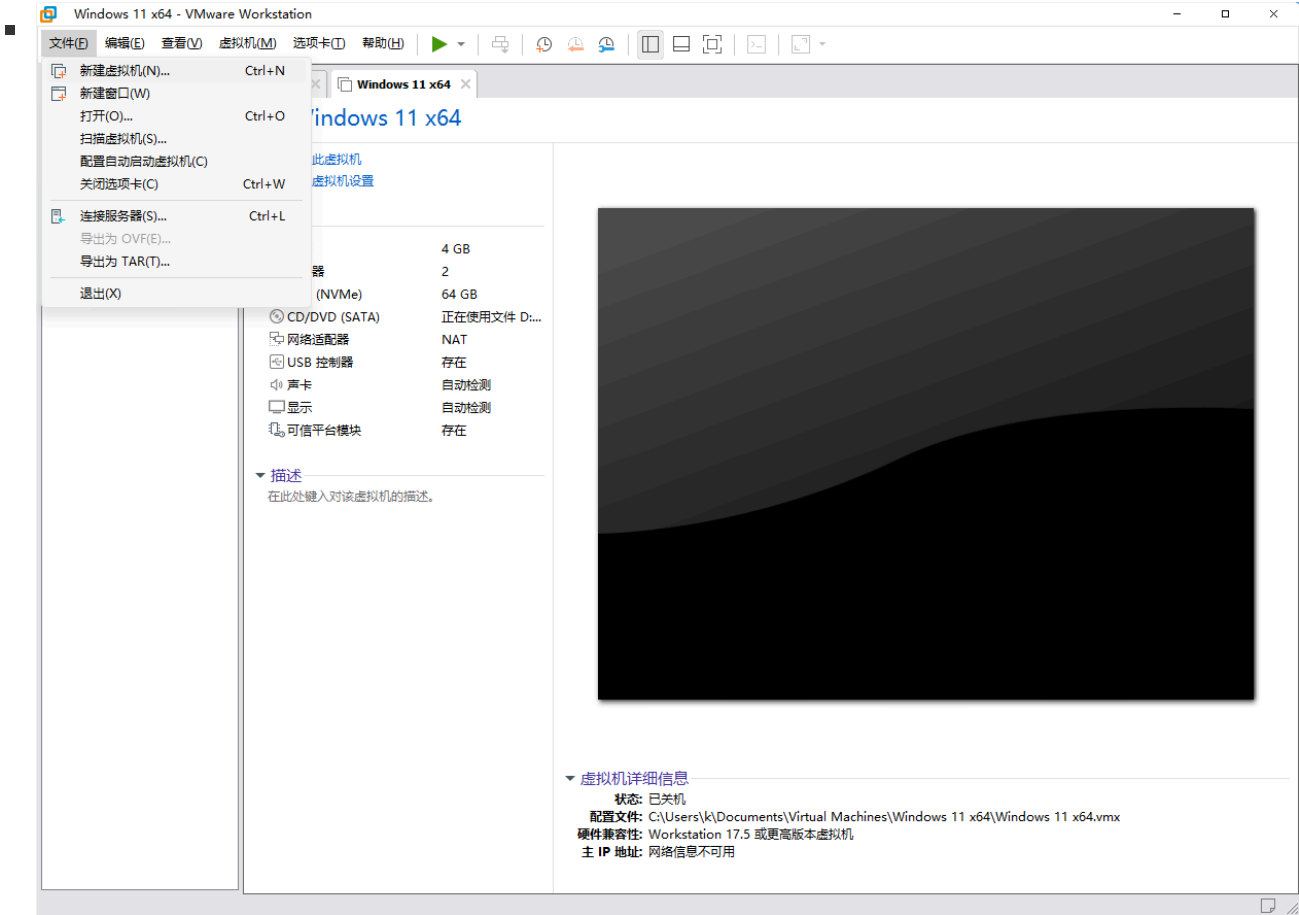


OpenEuler 实验报告

实验步骤

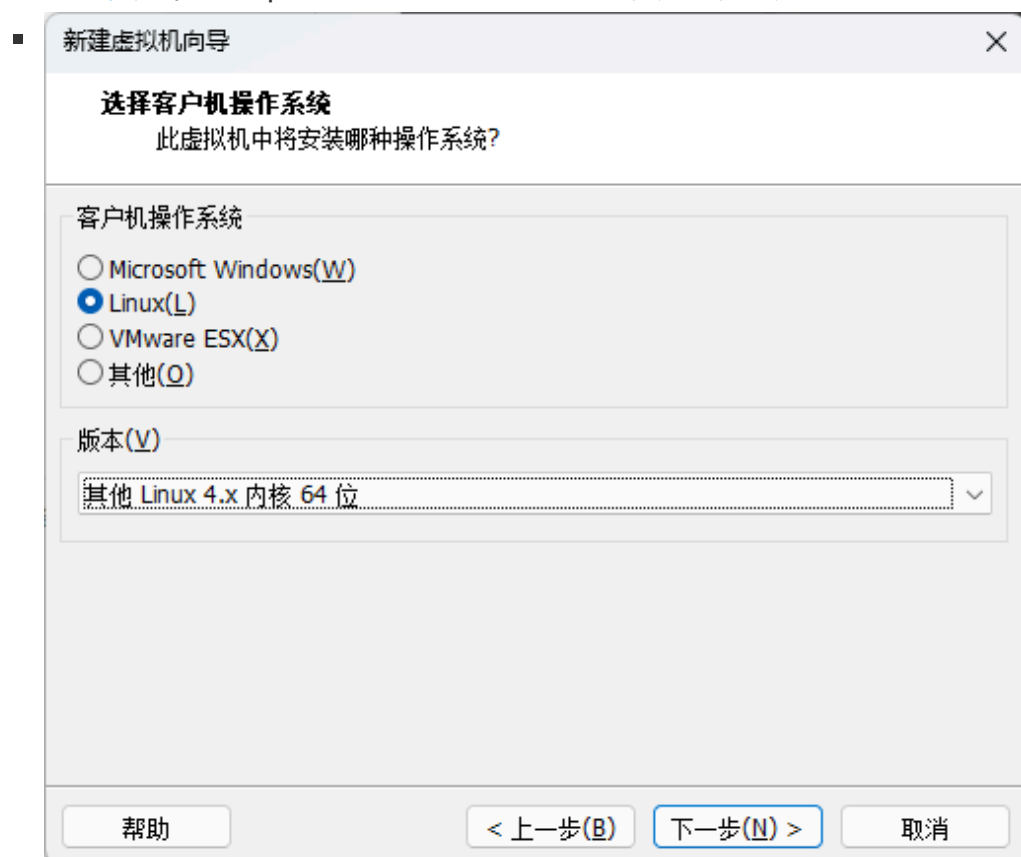
- 用虚拟机 VMware 完成 OpenEuler 操作系统的安装
 - 安装
 - 在 VMware 中新建虚拟机



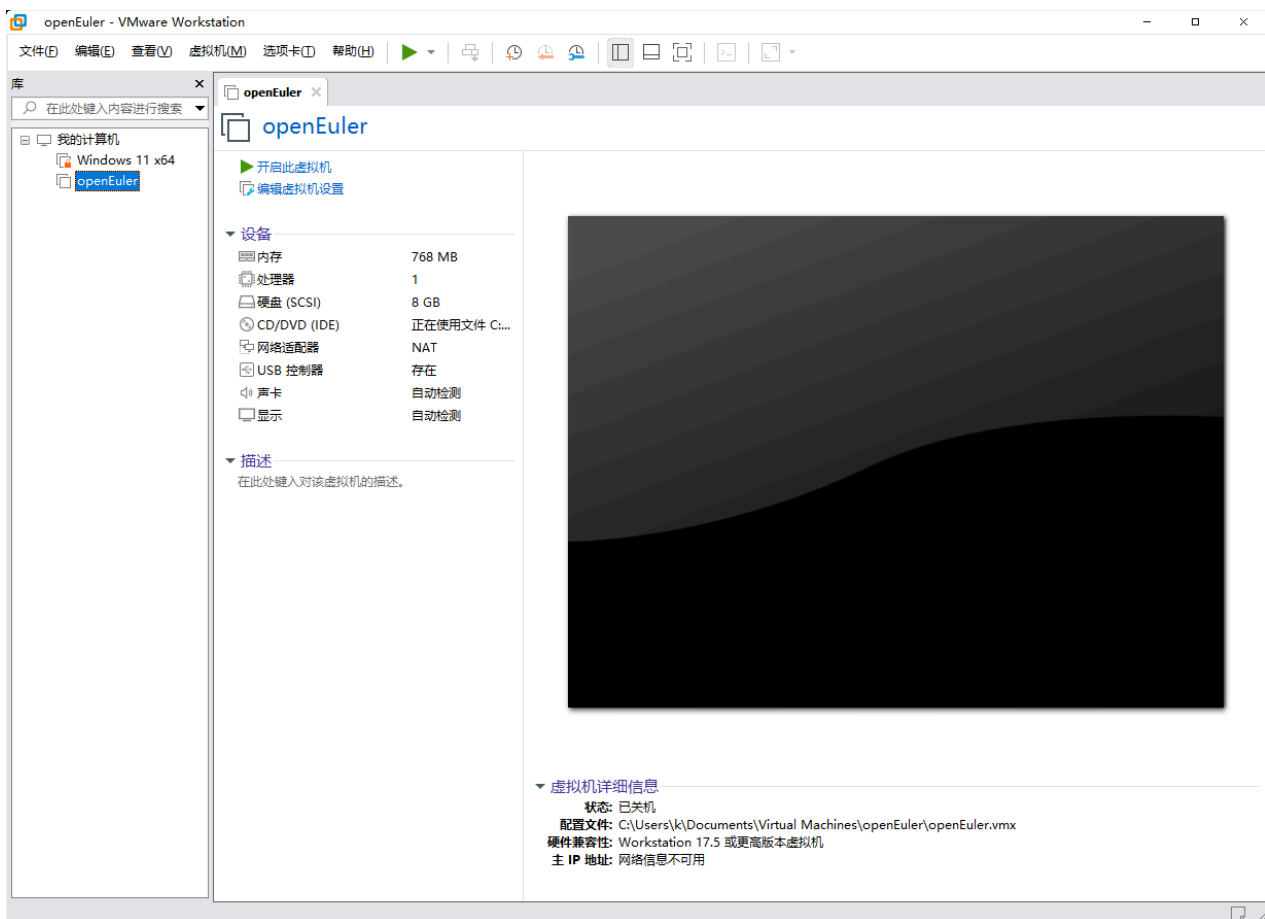
- 选择下载的 iso 文件



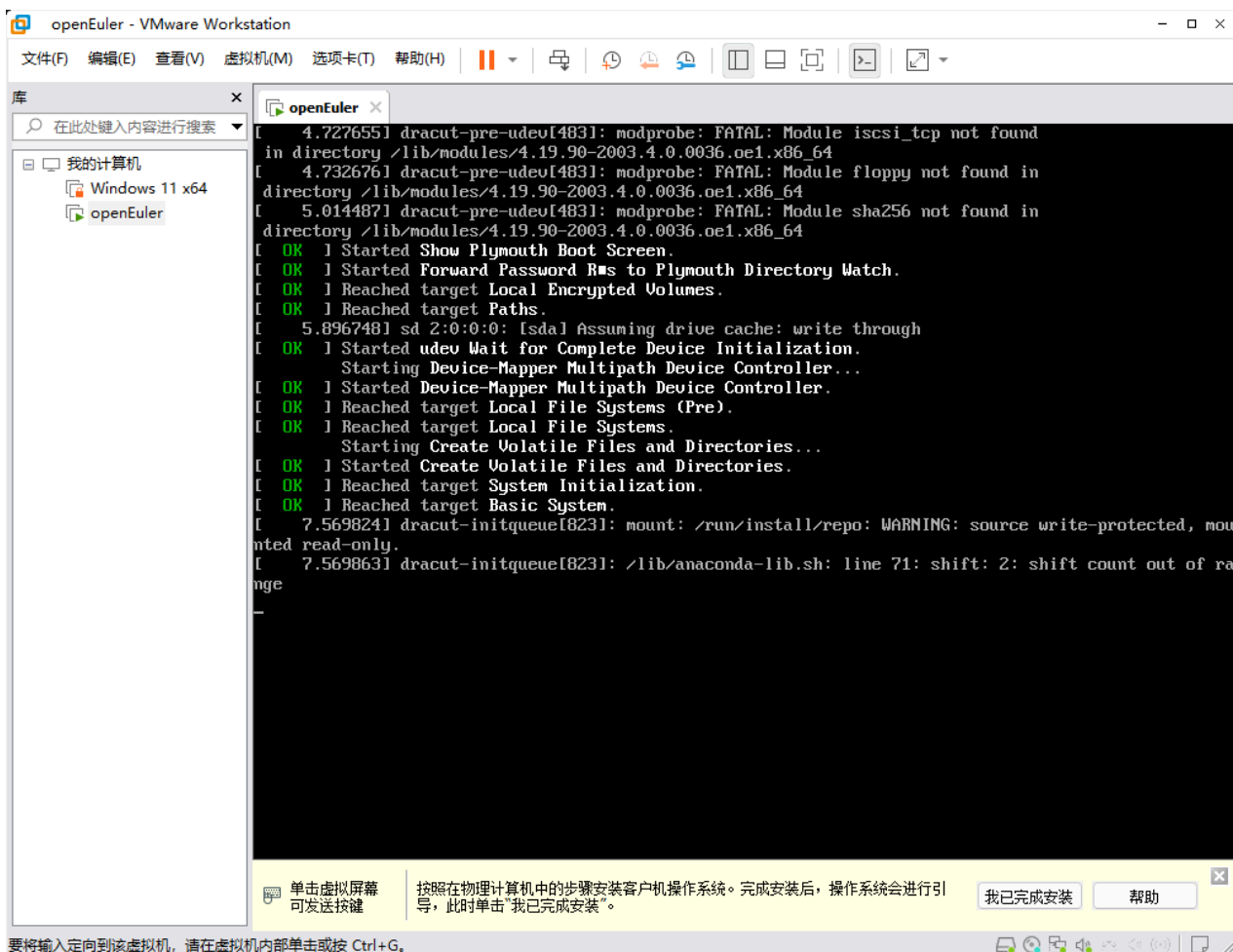
- [查询资料](#)得知 OpenEuler 20.03 LTS 在虚拟机安装时应选择 其他Linux 4.x 内核 64 位



- 其他 VMware 参数选择默认参数 成功添加虚拟机

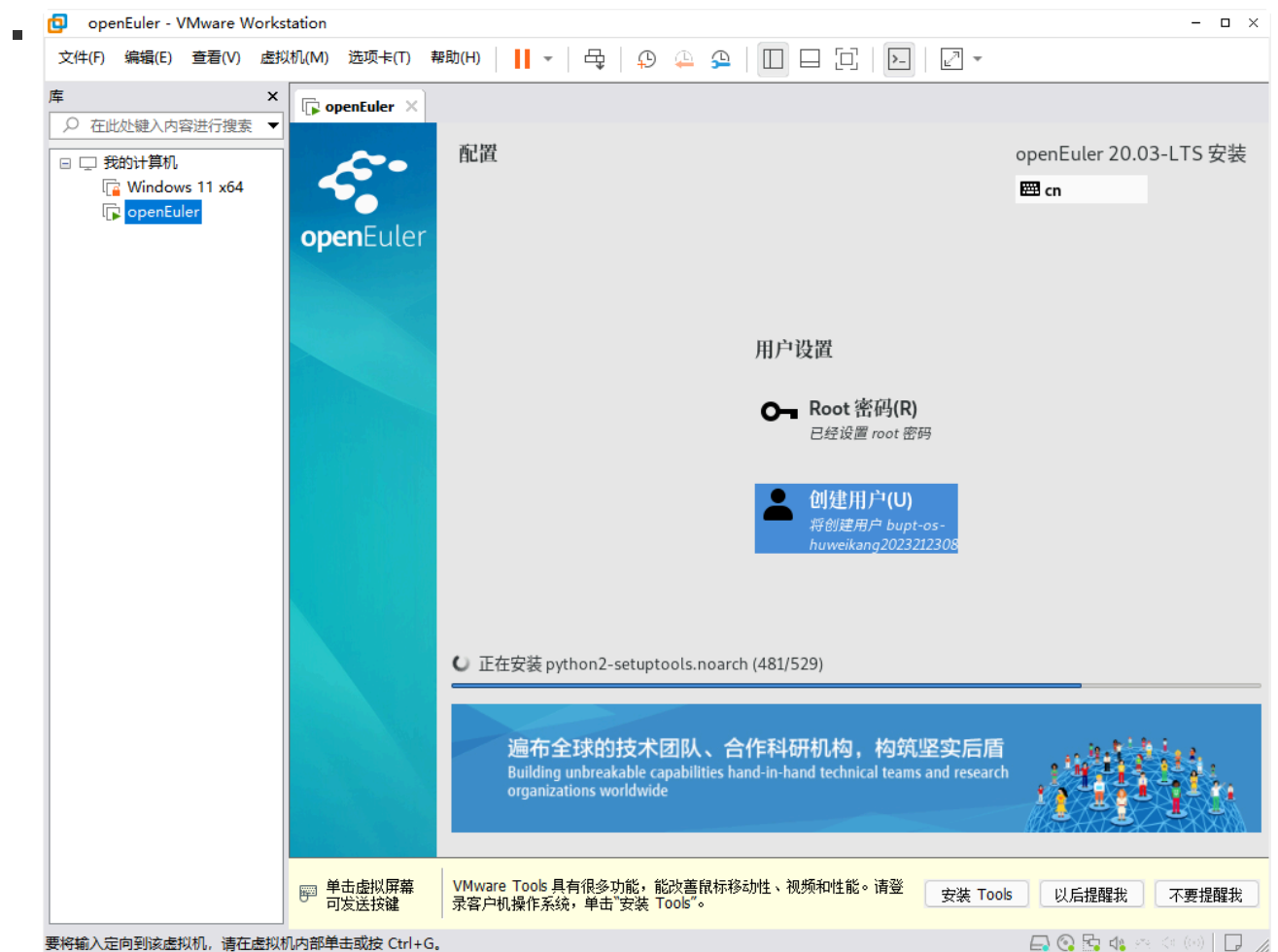


启动虚拟机 等待系统安装

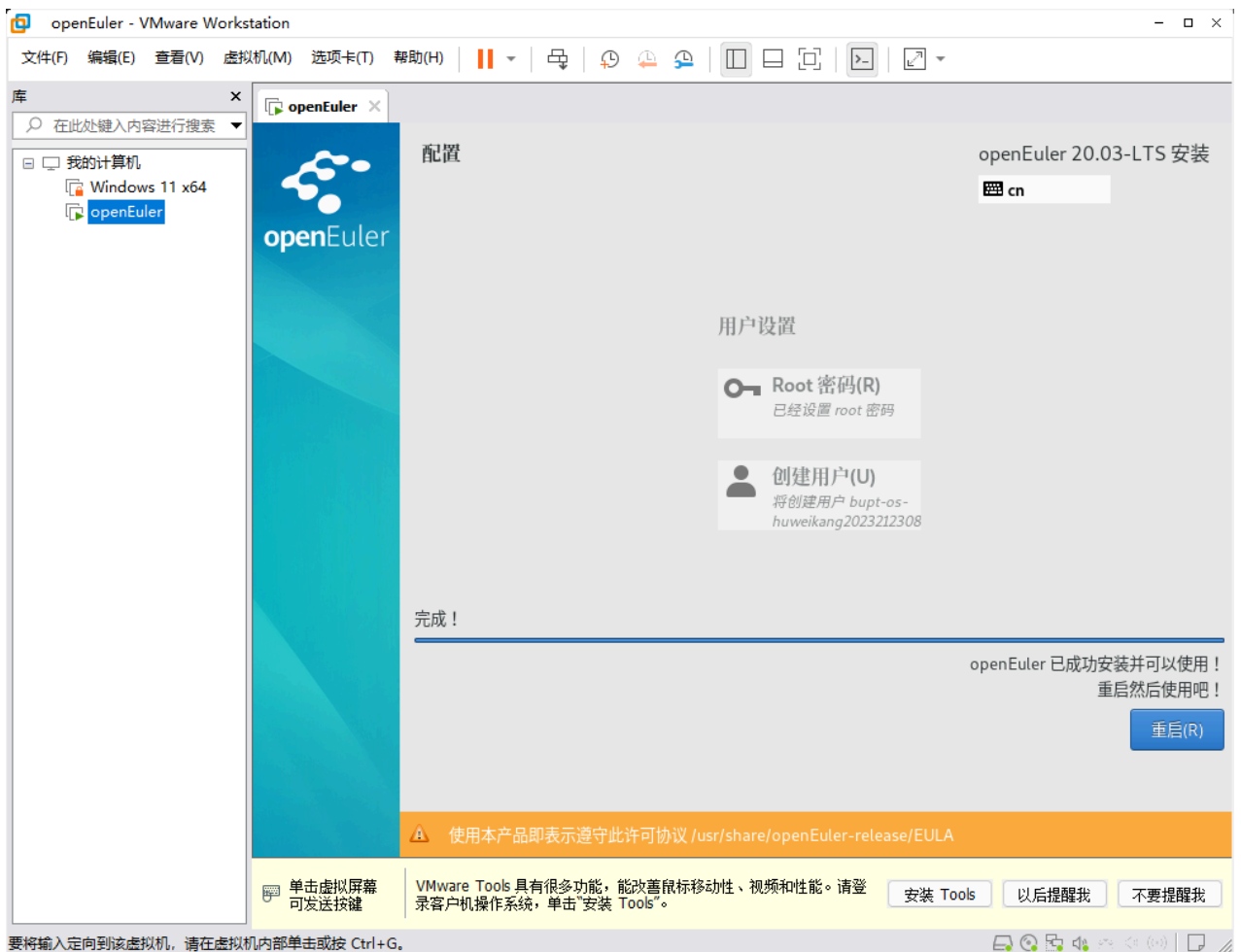


■ 设置 Root 密码 创建用户

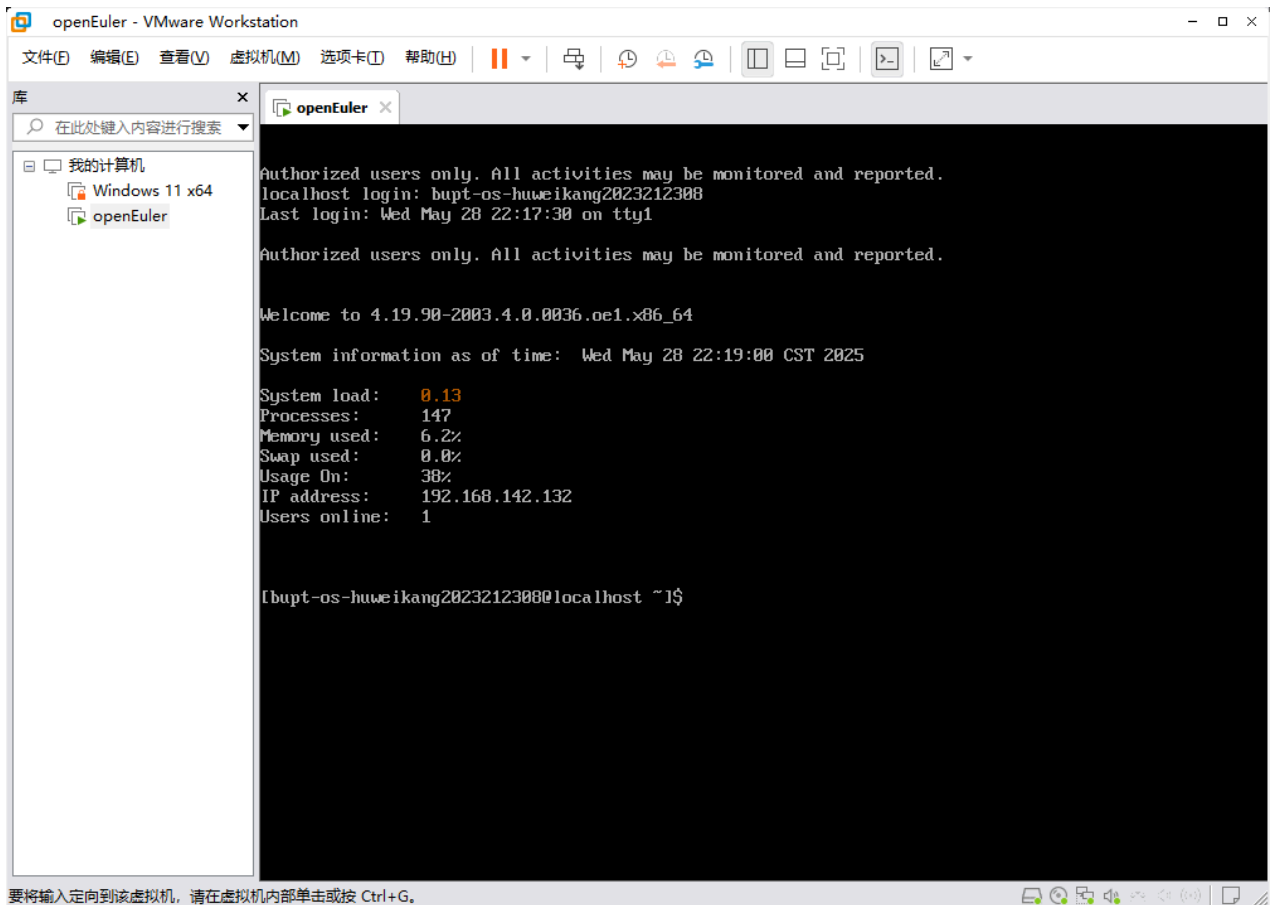




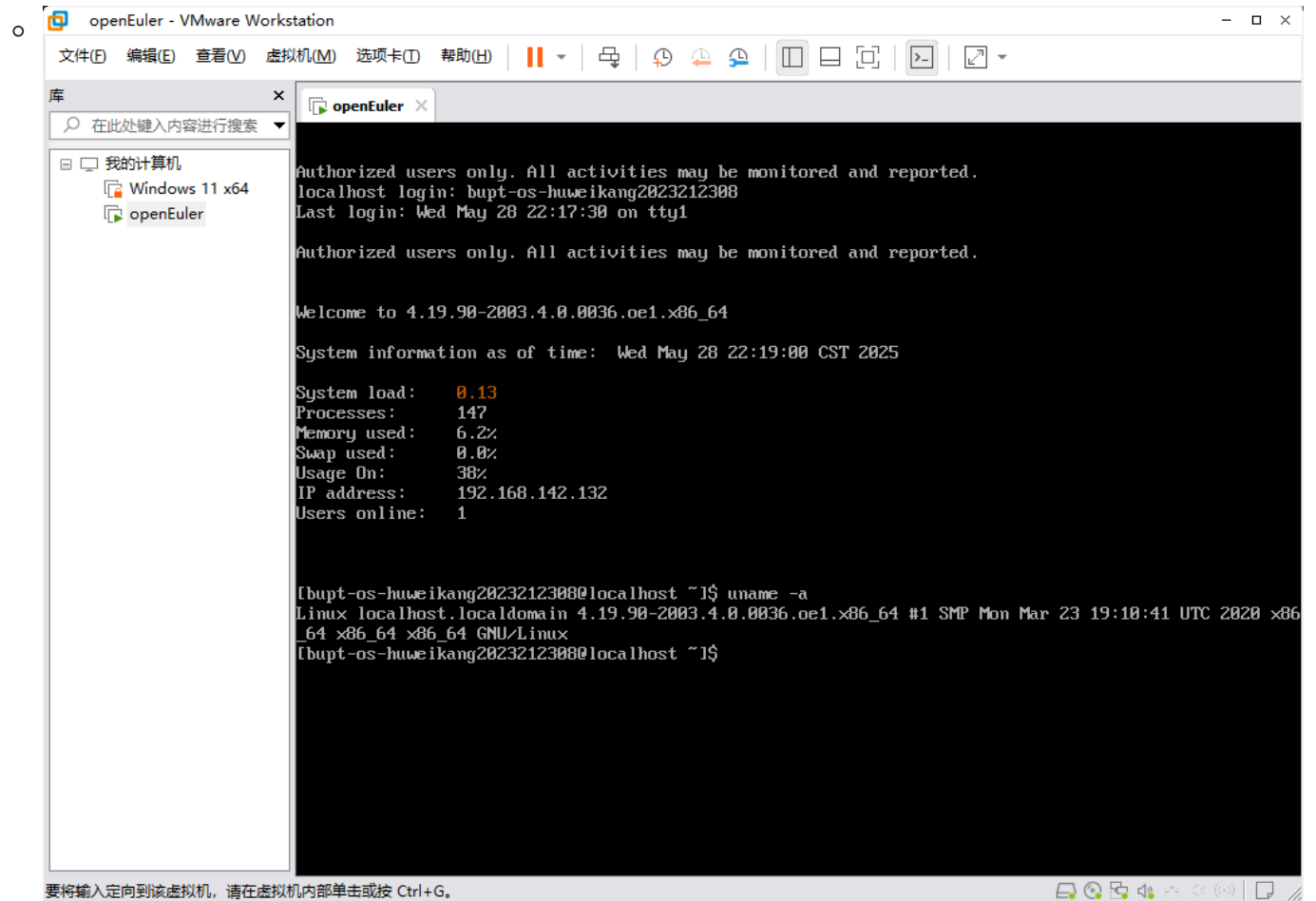
- 安装完毕 重启虚拟机



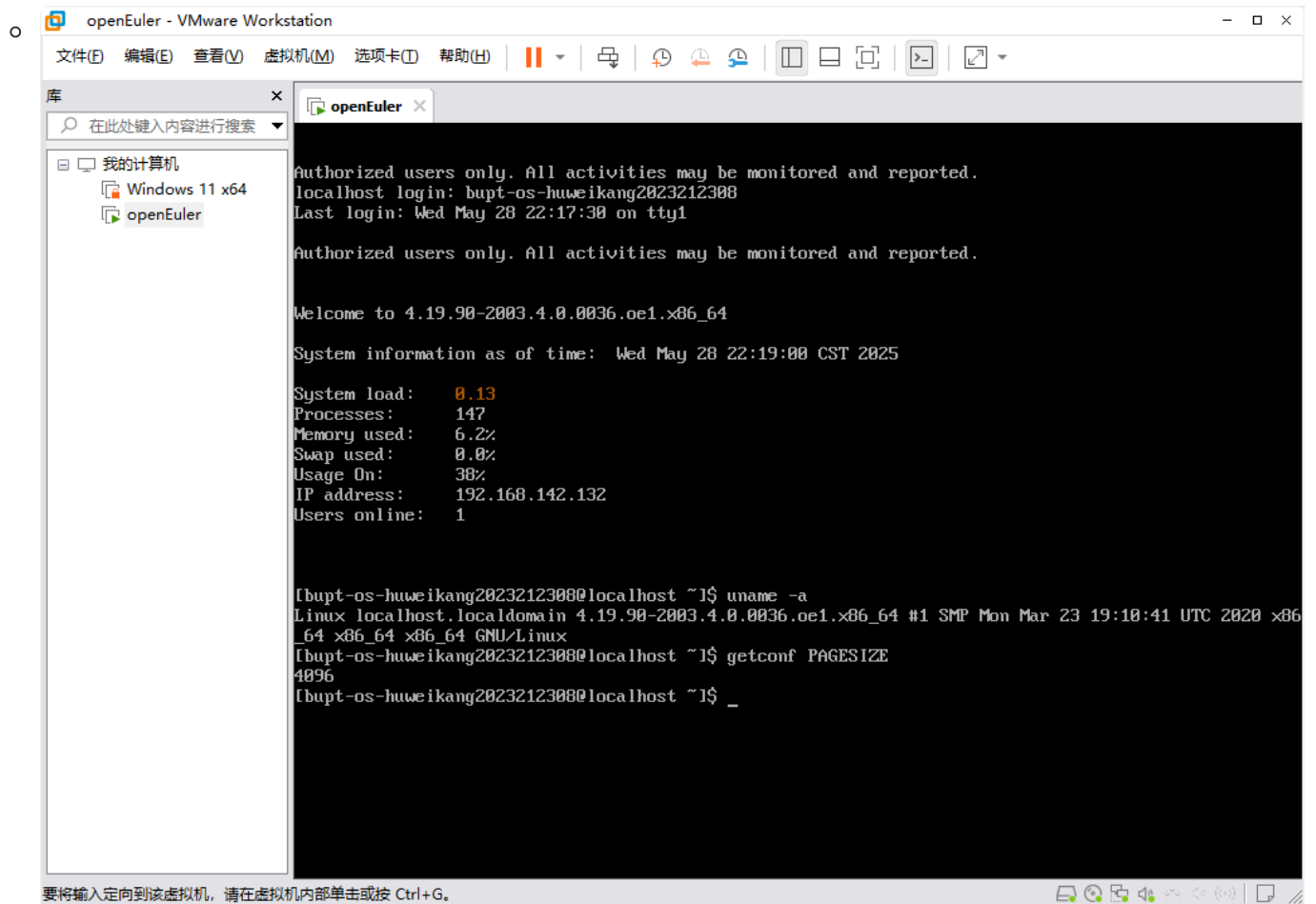
登入 bupt-os-huweikang2023212308



- `uname -a` 指令



- `getconf PAGESIZE` 指令



- 采用重新编译源代码的方式将内核更新至最新版
 - 登出 `bupt-os-huweikang2023212308` 登入 `root`
 - 指令
 - `logout`
 - 下载 `openEuler 20.03 update 4.19.90-2403.4.0` 对应 Source code
 - 将文件传入虚拟机中
 - 指令
 - `scp .\kernel-4.19.90-2403.4.0.tar.gz root@192.168.142.132:/root/`
 - IP 地址在登陆 OpenEuler 时可以得到


```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell, 了解新功能和改进! https://aka.ms/PSWindows

加载个人及系统配置文件用了 1924 毫秒。
PS C:\Users\k\Downloads> scp .\kernel-4.19.90-2403.4.0.tar.gz root@192.168.142.132:/root/
The authenticity of host '192.168.142.132 (192.168.142.132)' can't be established.
ED25519 key fingerprint is SHA256:RG1V2tftN7eeVWP5DDN4GZdPNic7Nnt83+t2BnrhVpU.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
Warning: Permanently added '192.168.142.132' (ED25519) to the list of known hosts.

Authorized users only. All activities may be monitored and reported.
root@192.168.142.132's password:
kernel-4.19.90-2403.4.0.tar.gz                                100% 157MB 147.1MB/s   00:01
PS C:\Users\k\Downloads> |
```

指令

- ls

可以发现成功传入

```
openEuler - VMware Workstation
文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H)
库
  在此处键入内容进行搜索
  我的计算机
    Windows 11 x64
    openEuler

localhost login: root
Password:
Last login: Wed May 28 22:33:15 on tty1

Authorized users only. All activities may be monitored and reported.

Welcome to 4.19.90-2003.4.0.0036.oe1.x86_64

System information as of time: Wed May 28 22:41:06 CST 2025

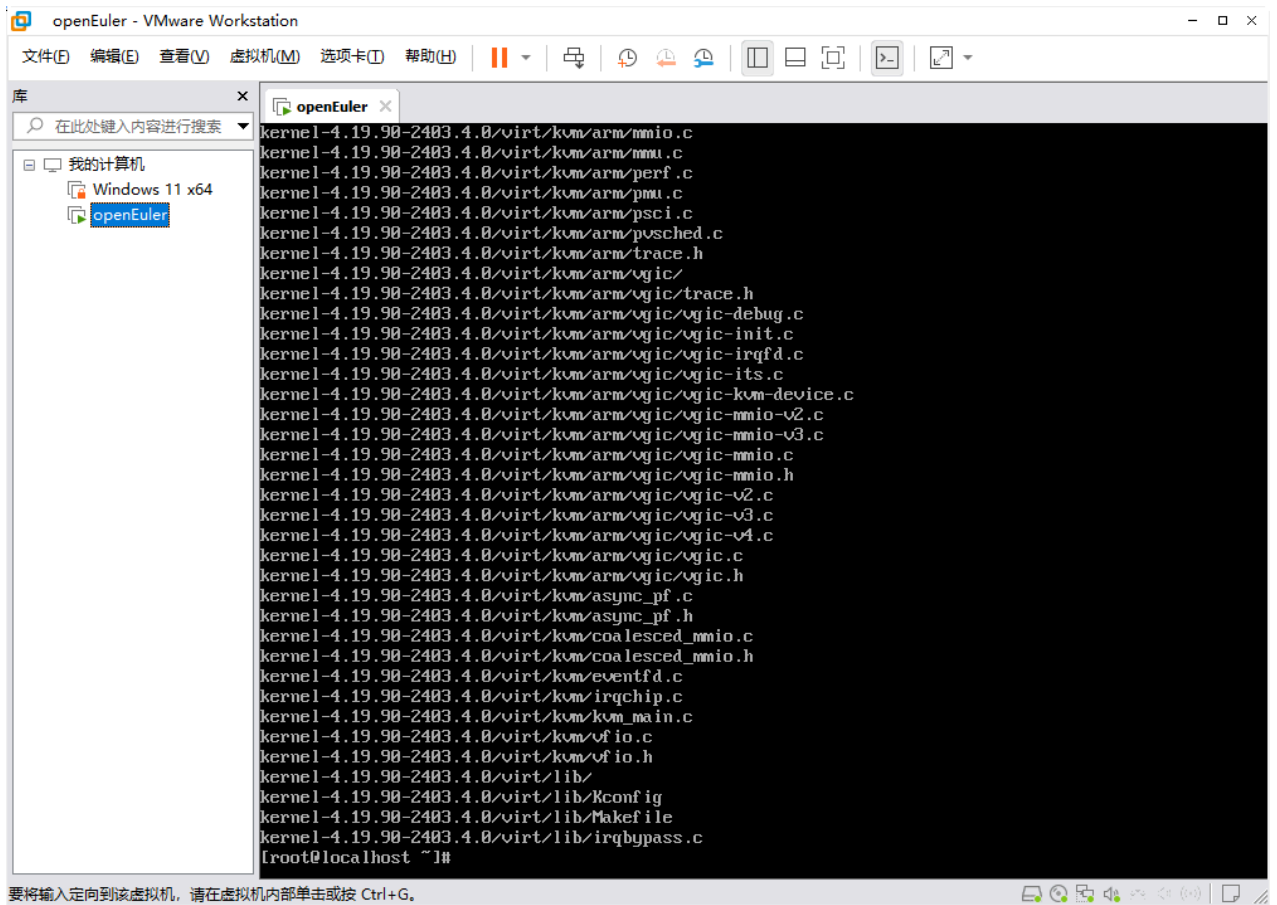
System load:  0.08
Processes:    140
Memory used:  6.2%
Swap used:    0.0%
Usage On:     41%
IP address:   192.168.142.132
Users online: 1

[root@localhost ~]# ls
anaconda-ks.cfg  kernel-4.19.90-2403.4.0.tar.gz
[root@localhost ~]#
```

解压文件

指令

- tar -xvf kernel-4.19.90-2403.4.0.tar.gz



编译源代码

指令

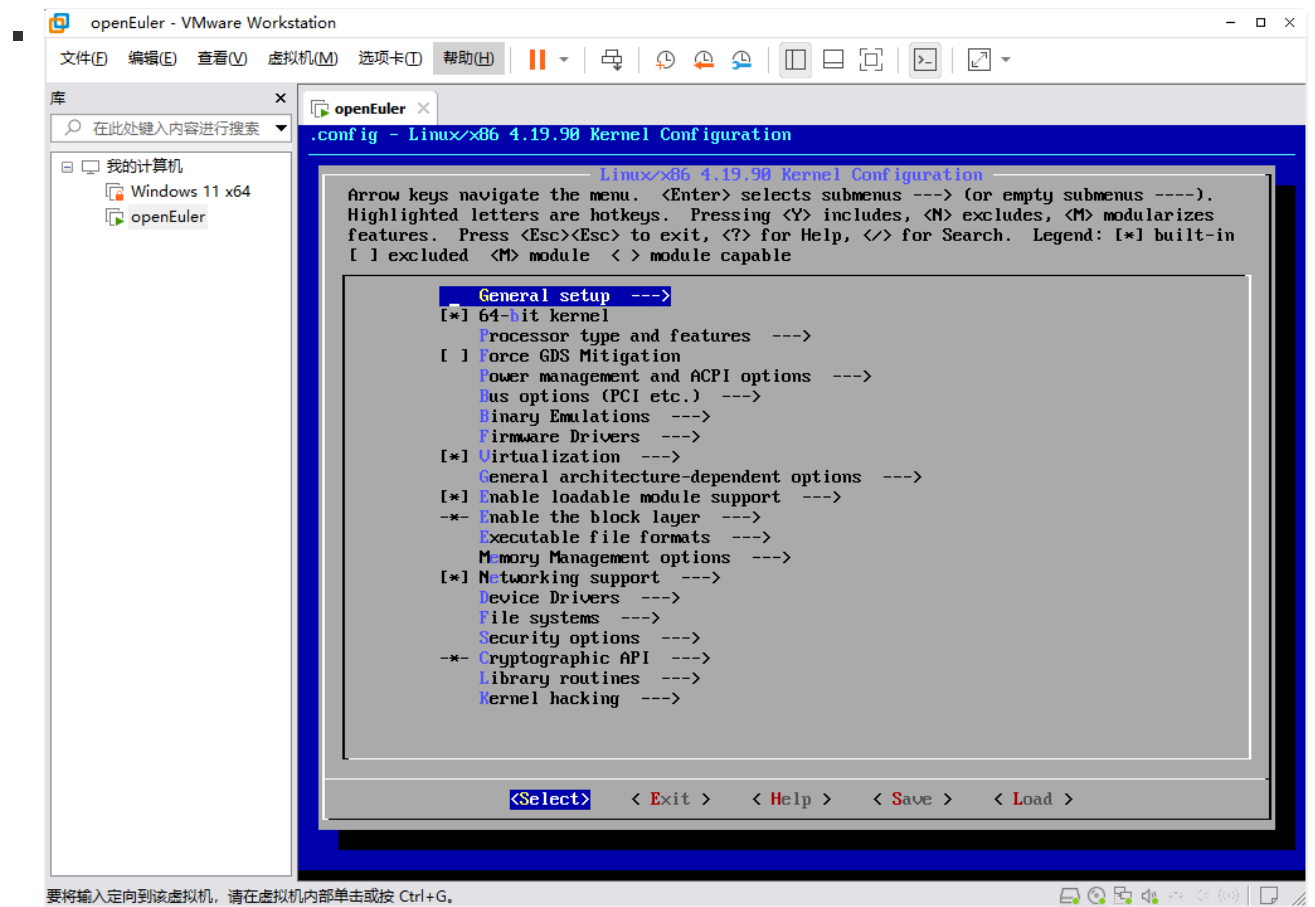
- cd kernel-4.19.90-2403.4.0

安装依赖

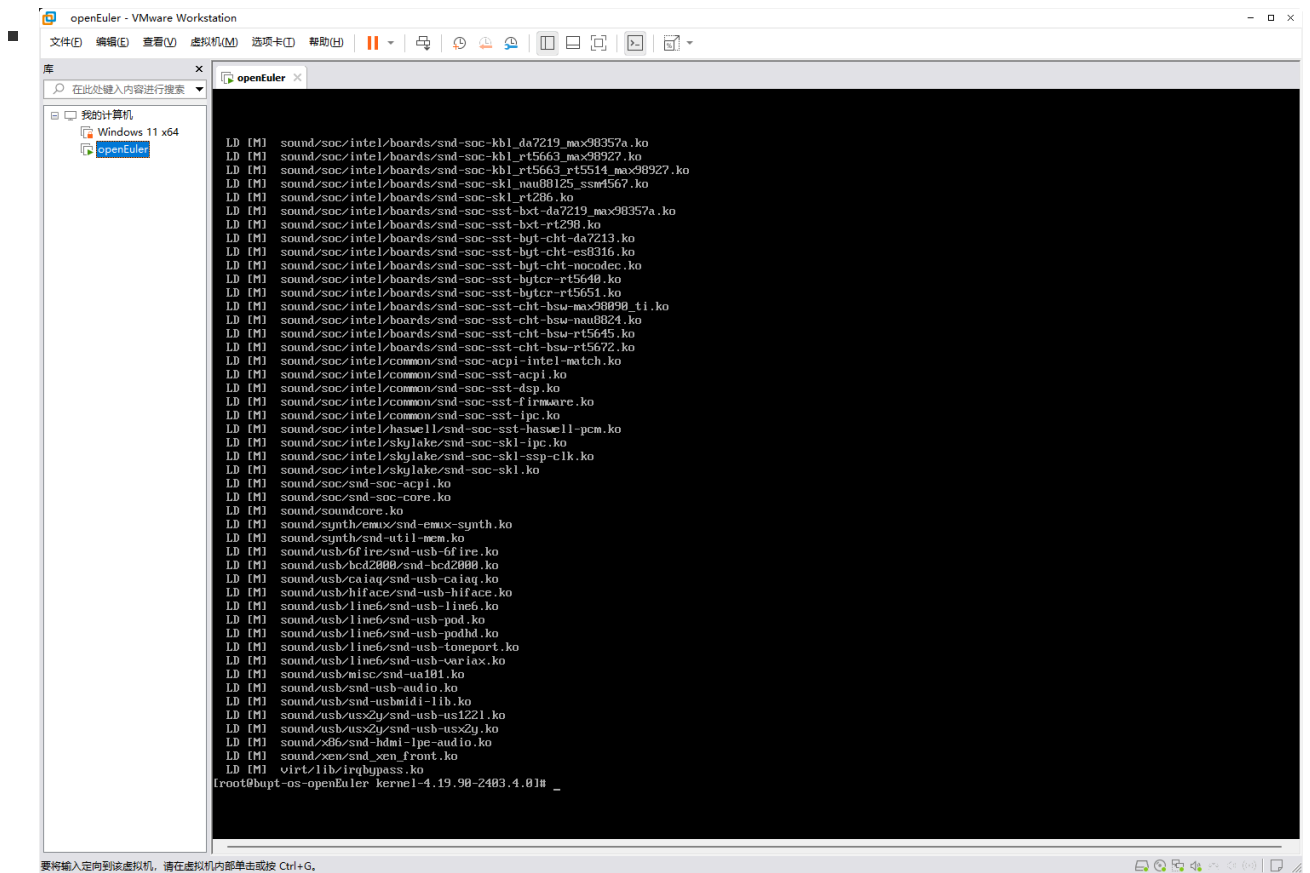
- yum install ncurses-devel
- yum install flex
- yum install bc
- yum install openssl-devel
- yum install elfutils-libelf-devel

指令

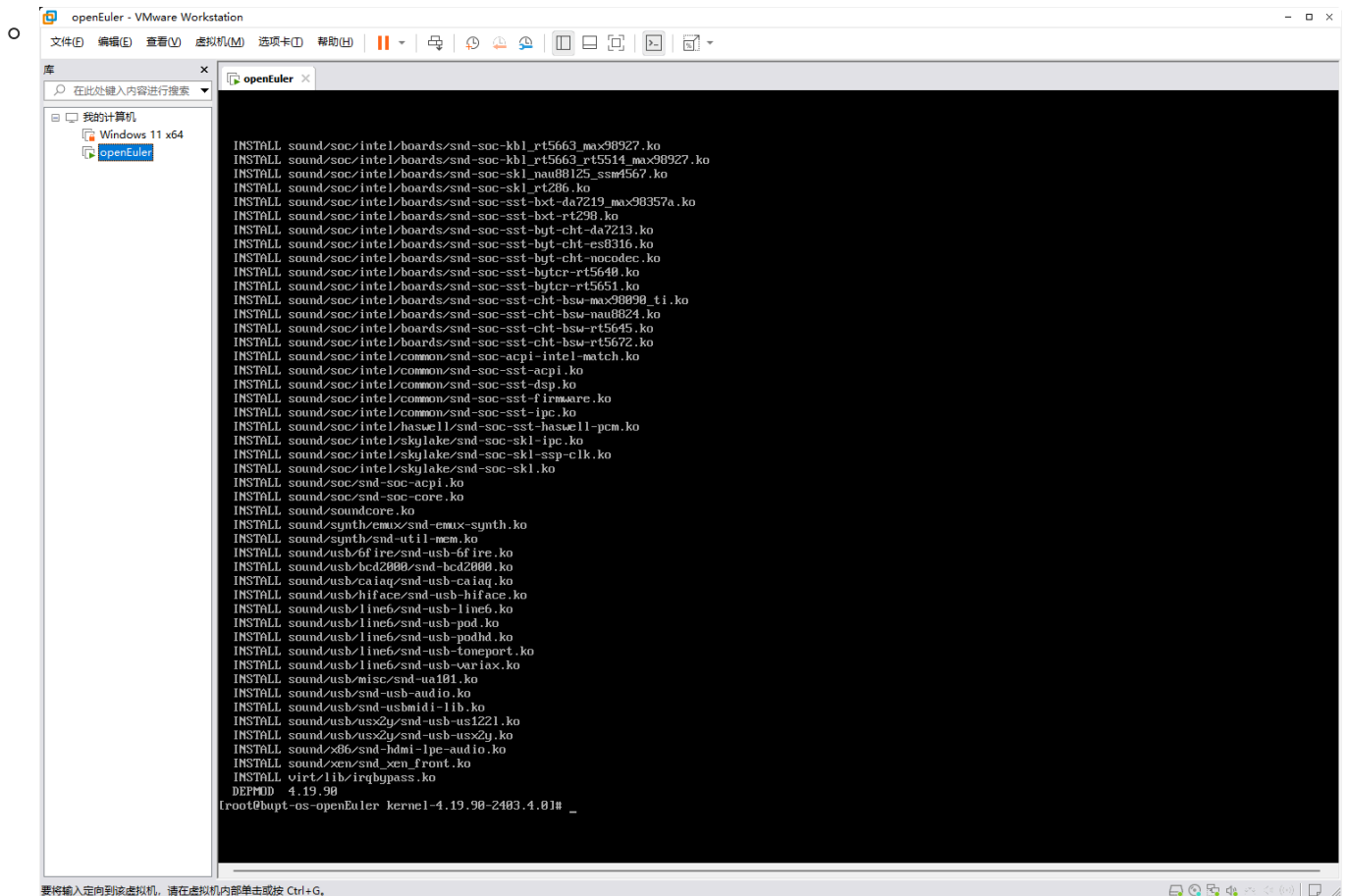
- make menuconfig



- 通过方向键选择 Save 并回车 保存默认配置
- 指令
 - make -j8
 - 8 线程加快编译速度

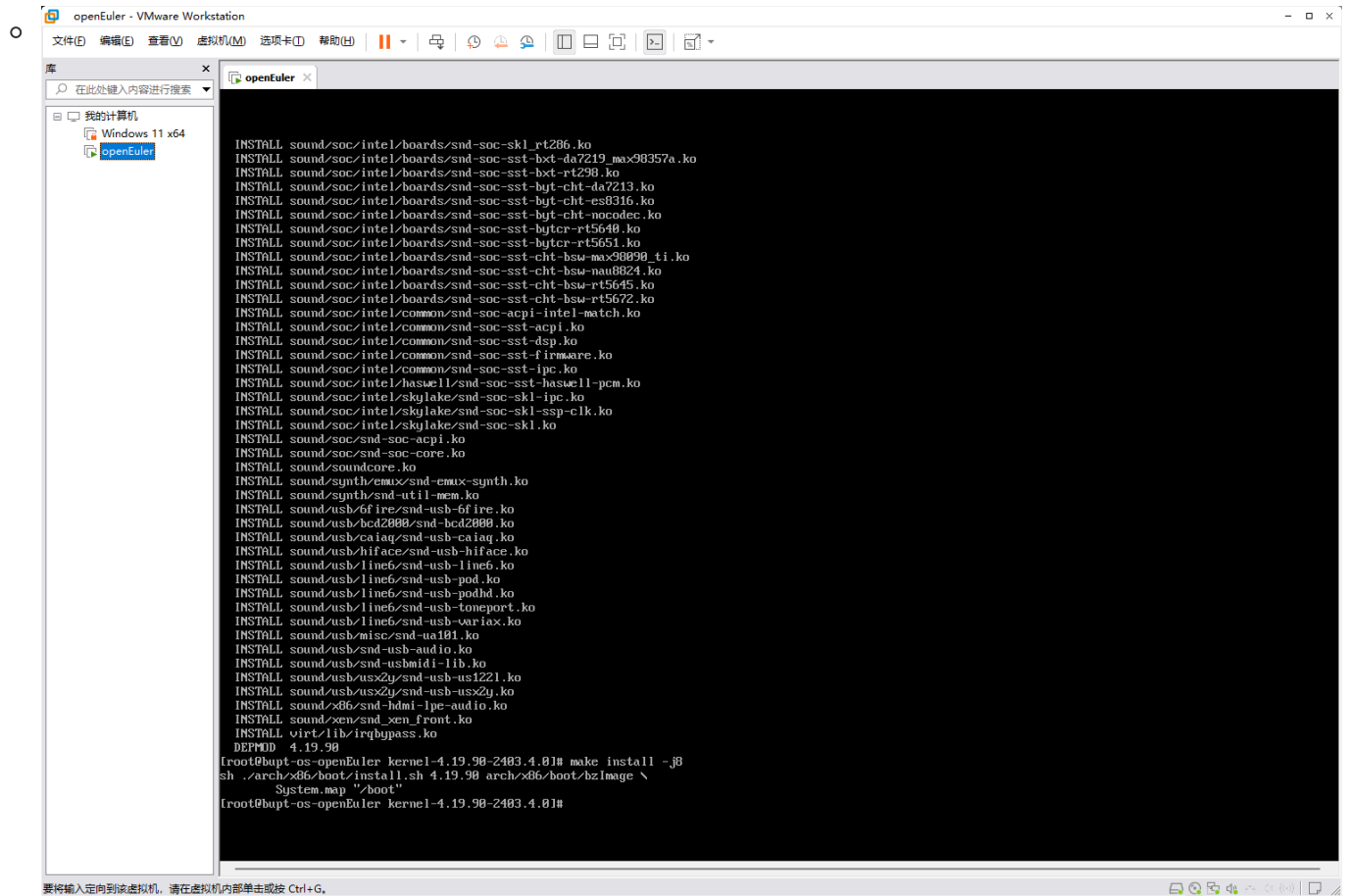


- 重新编译后，安装内核，执行 `uname -a` 指令
 - 指令
 - `make modules_install -j8`



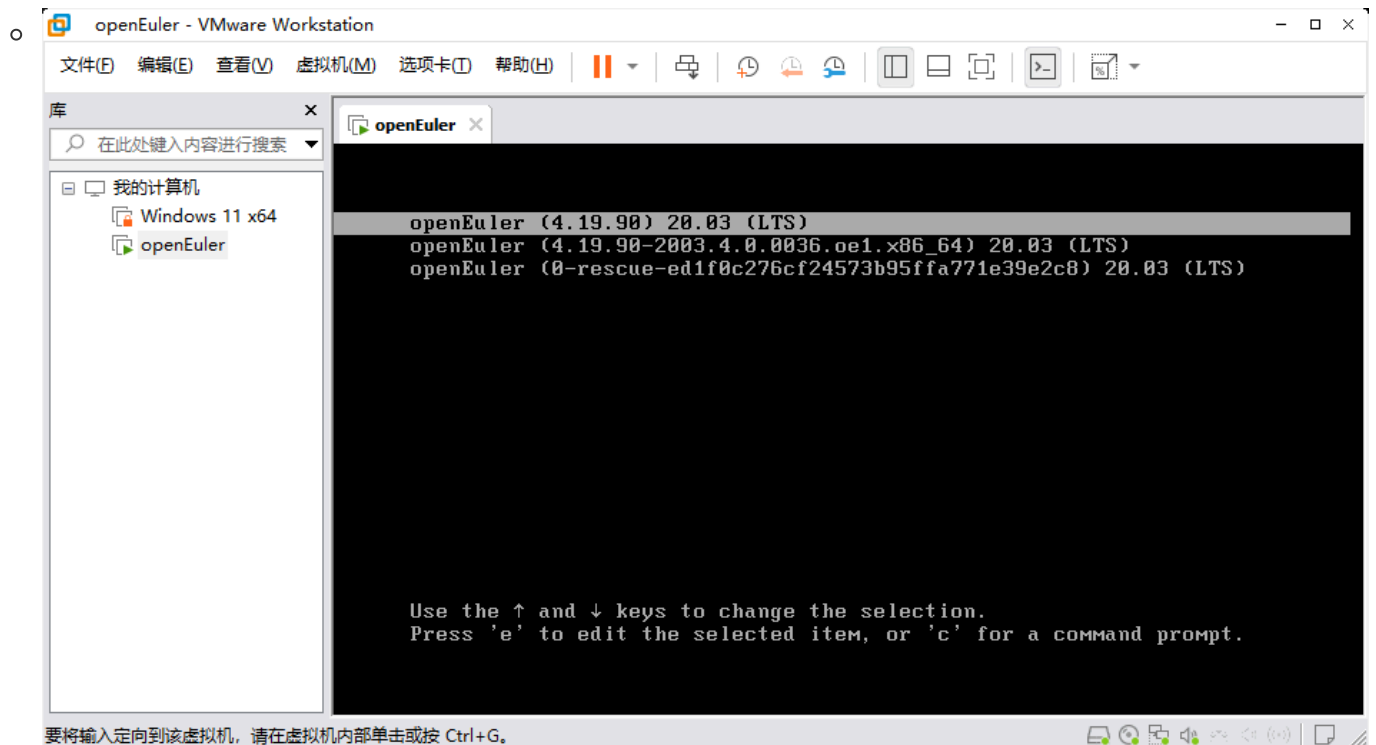
- 指令

- `make install -j8`

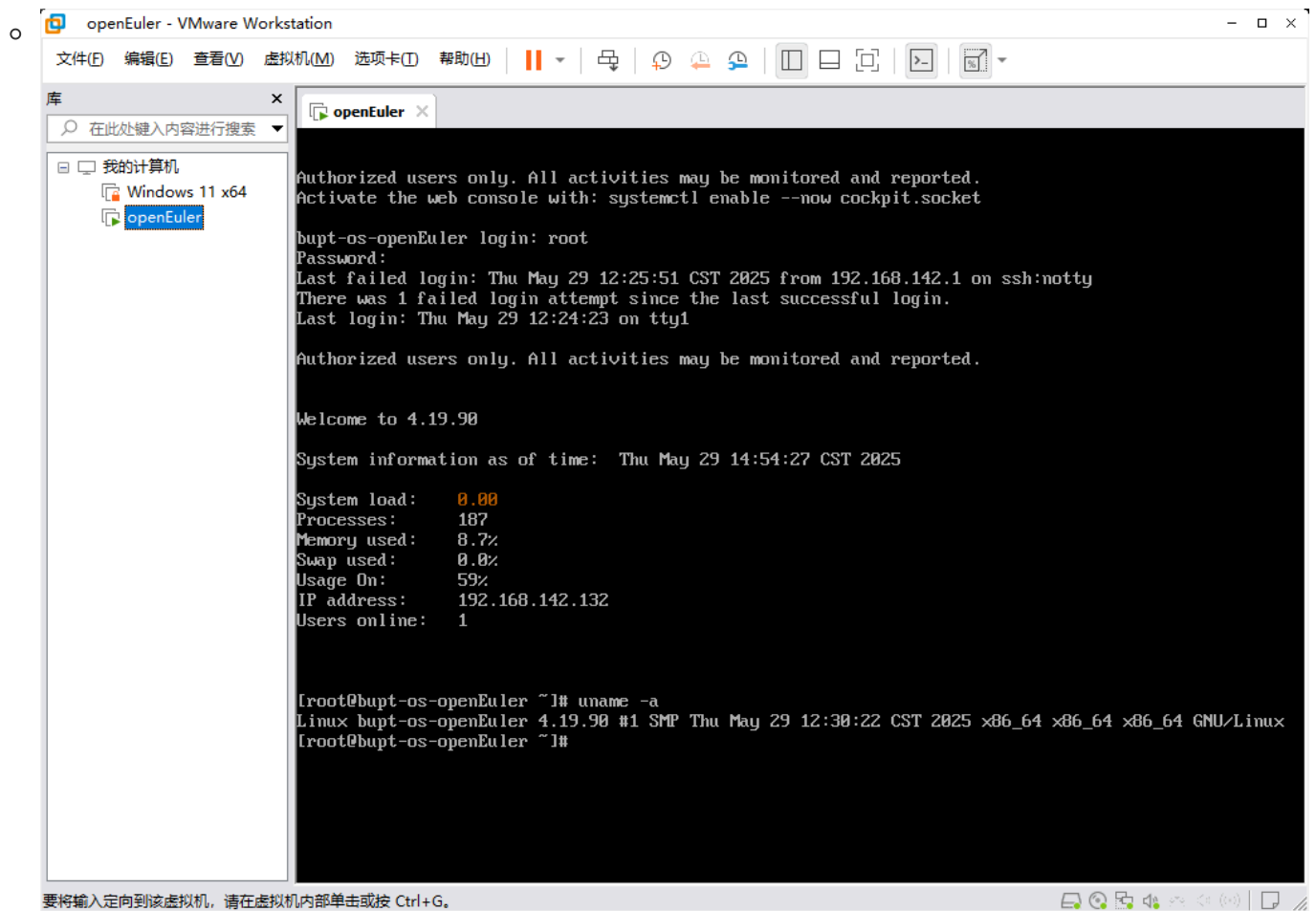


- 重启虚拟机

- 引导页面出现新的内核



- 执行 `uname -a`

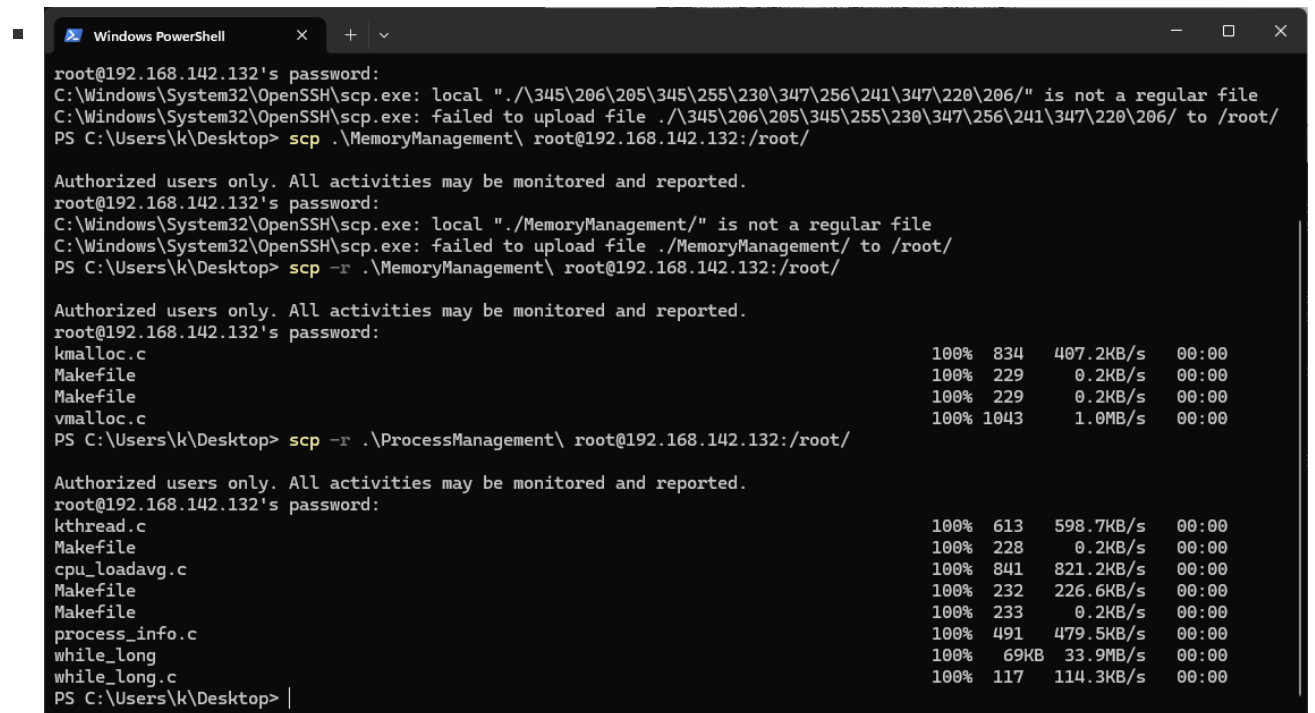


基础操作系统实验

将代码传入虚拟机中

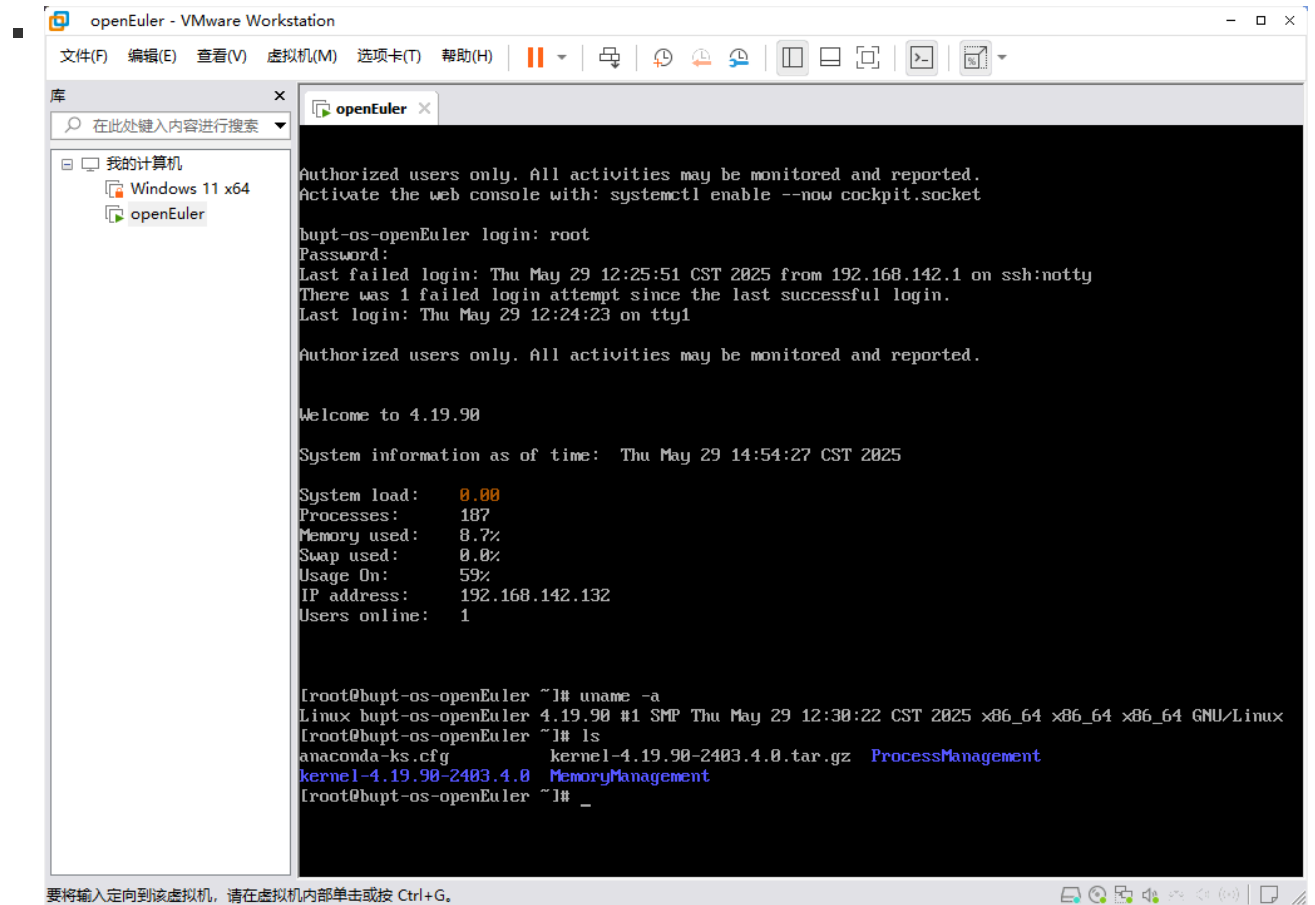
指令

- scp -r .\MemoryManagement\ root@192.168.142.132:/root/
- scp -r .\ProcessManagement\ root@192.168.142.132:/root/



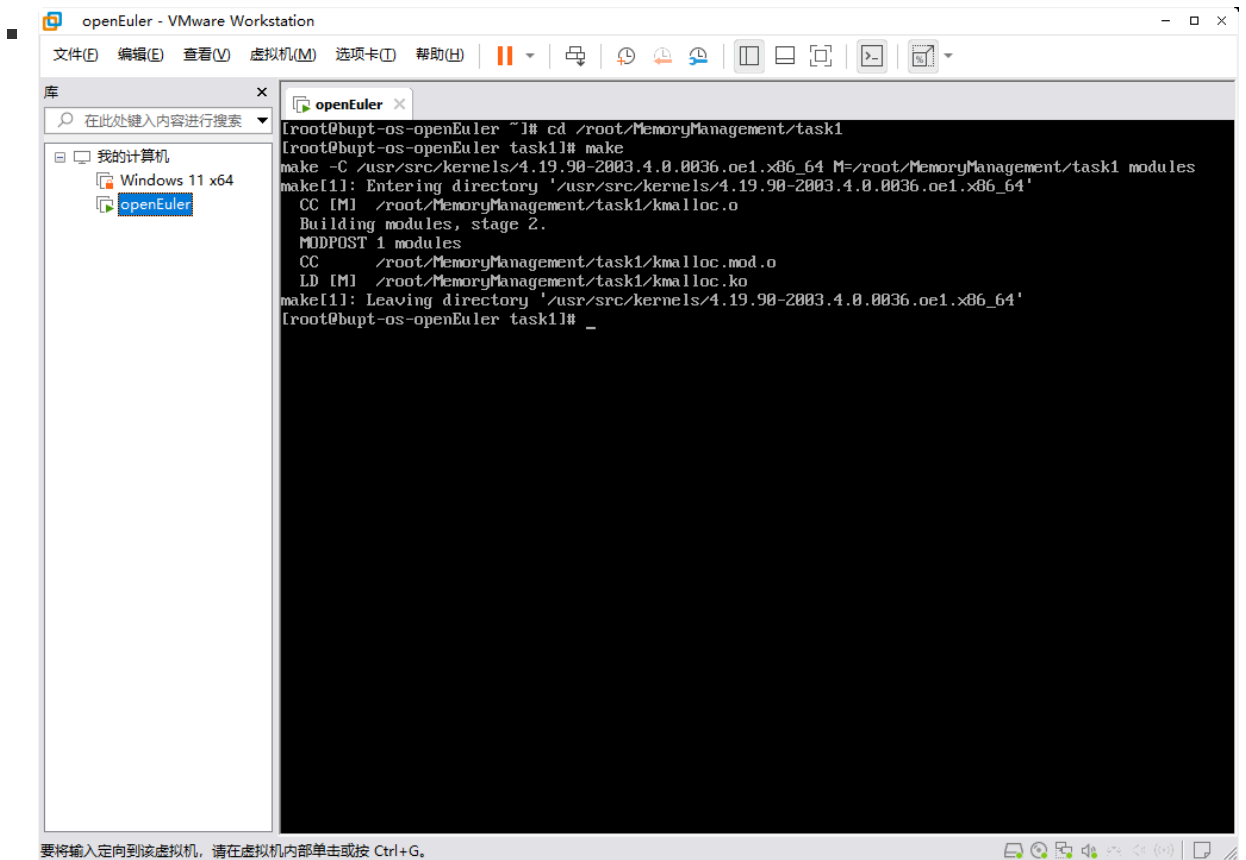
指令

- ls
- 可以发现成功传入



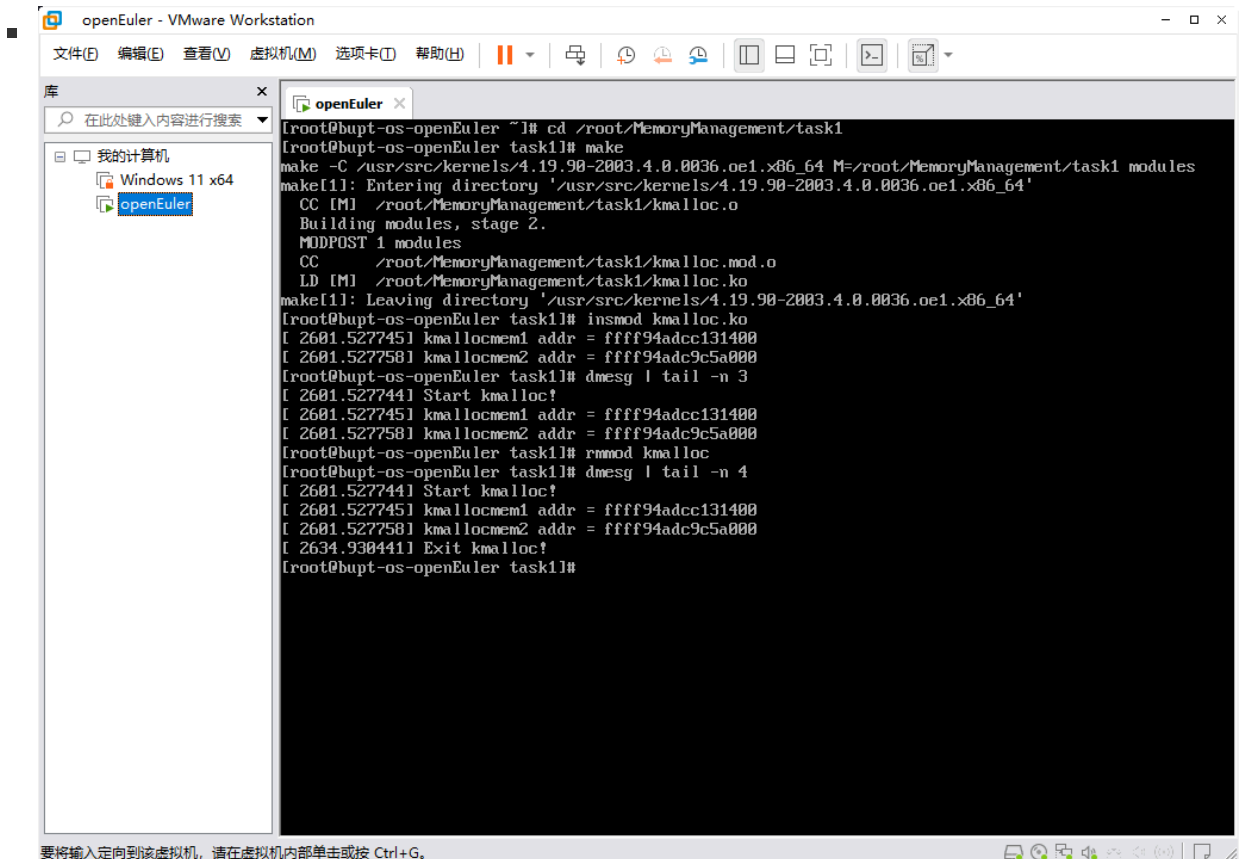
内存管理实验

- 使用 kmalloc 分配 1KB, 8KB的内存, 并打印指针地址
 - 指令
 - cd /root/MemoryManagement/task1
 - make



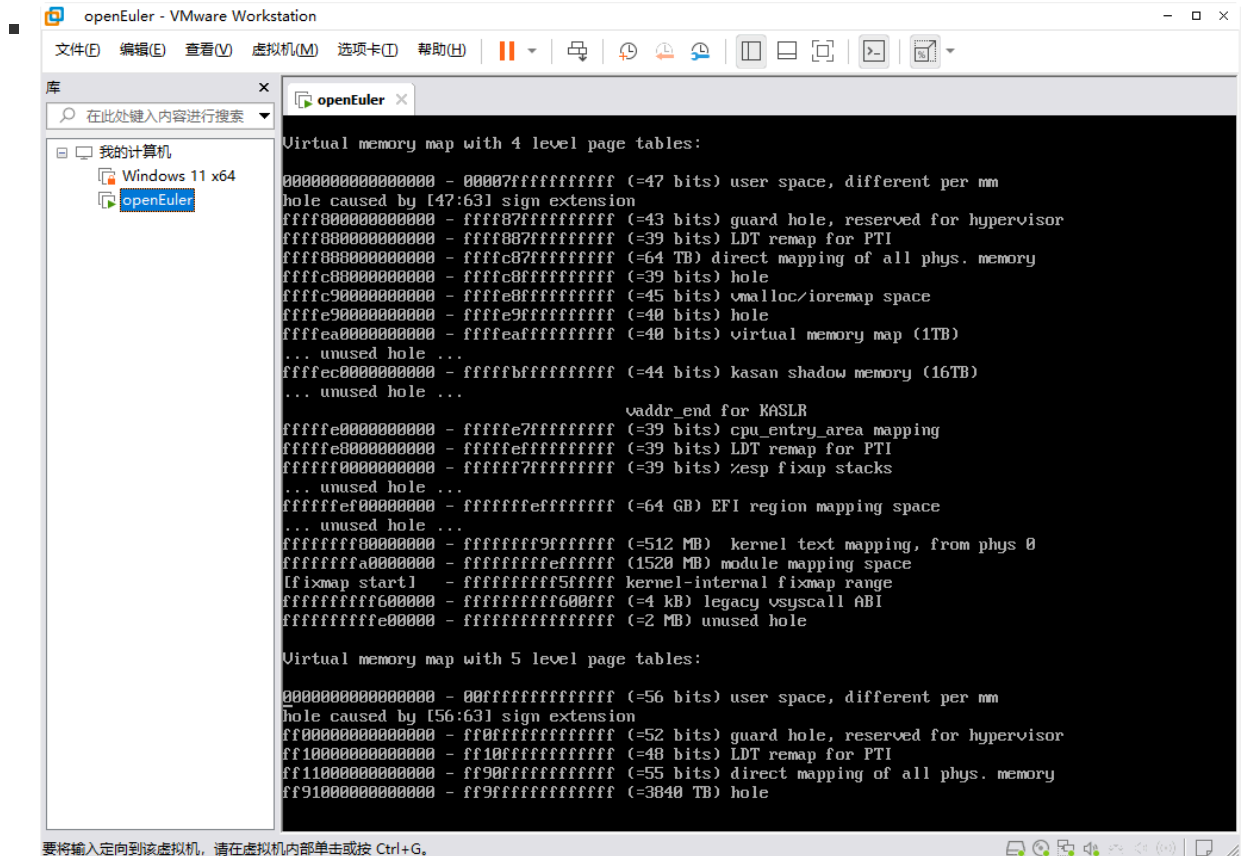
■ 指令

- insmod kmalloc.ko
- dmesg | tail -n 3
- rmmod kmalloc
- dmesg | tail -n 4



- 指令

- vi /root/kernel-4.19.90-2403.4.0/Documentation/x86/x86_64/mm.txt

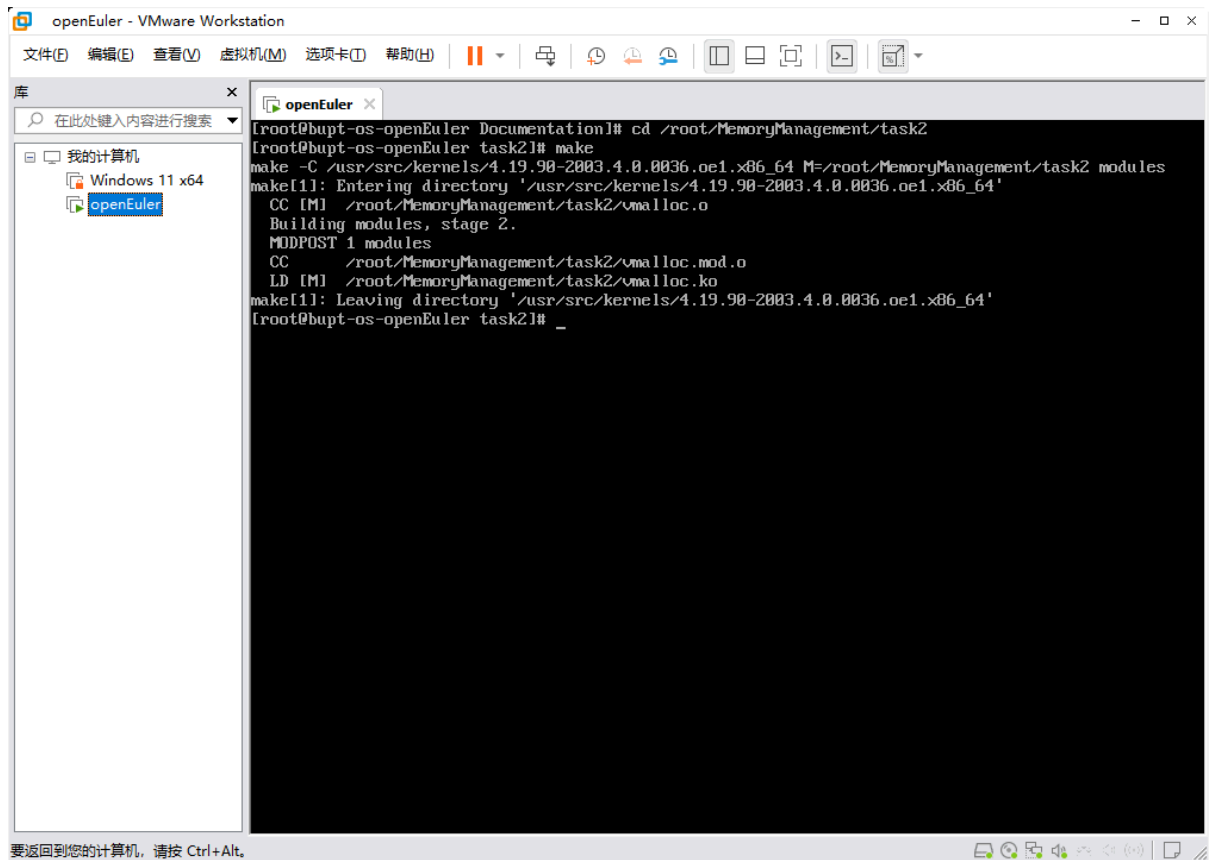


- 可知 kmalloc 分配的内存地址位于内核空间

- 使用 vmalloc 分别分配8KB、1MB、64MB的内存，打印指针地址

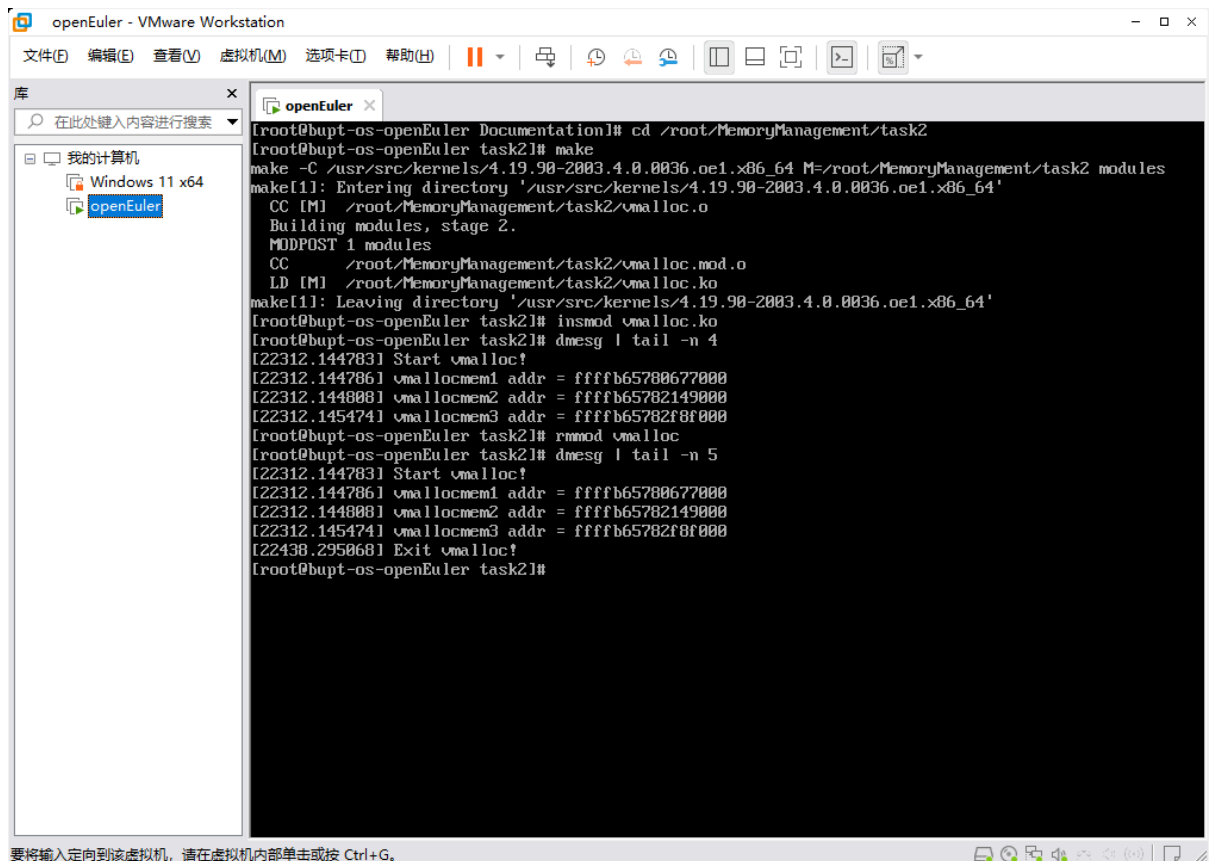
- 指令

- cd /root/MemoryManagement/task2
 - make

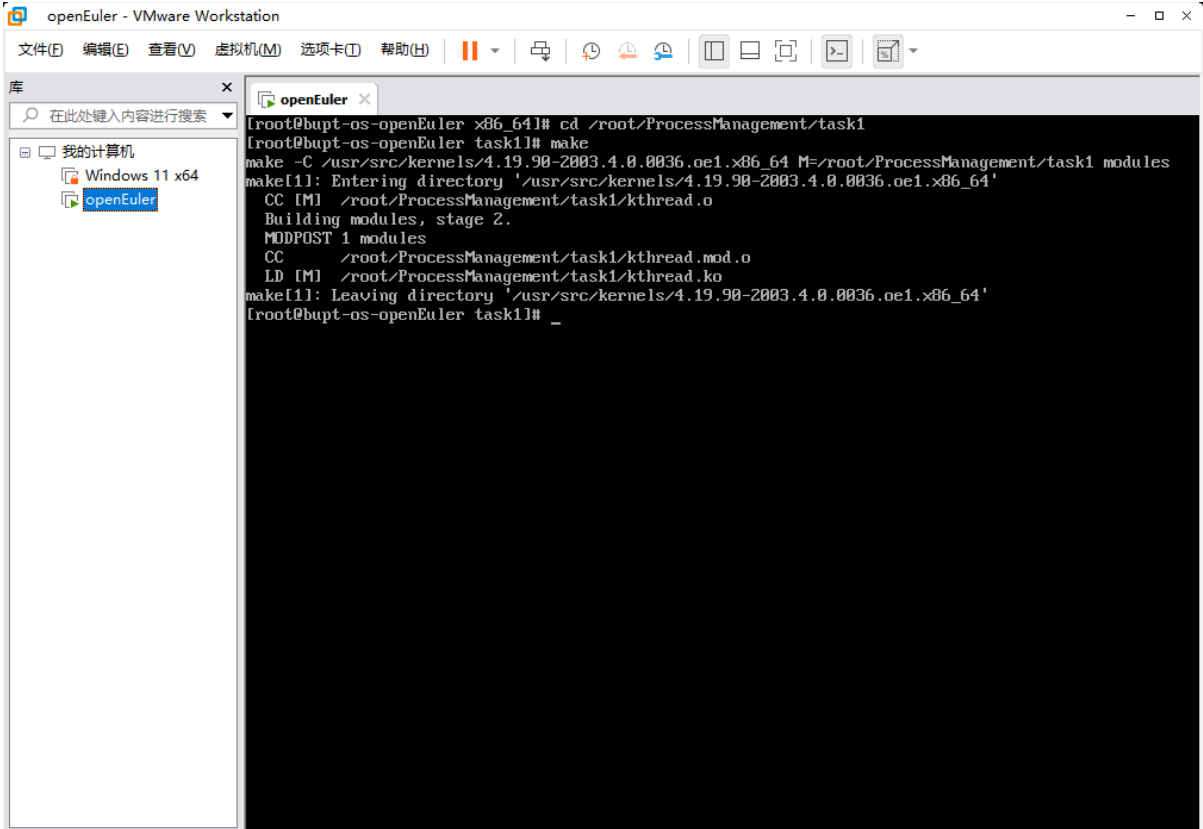


■ 指令

- insmod vmalloc.ko
- dmesg | tail -n 4
- rmmod vmalloc
- dmesg | tail -n 5



- 可知 vmalloc 分配的内存地址位于内核空间
- 进程管理实验
 - 创建并运行内核线程
 - 指令
 - `cd /root/ProcessManagement/task1`
 - `make`

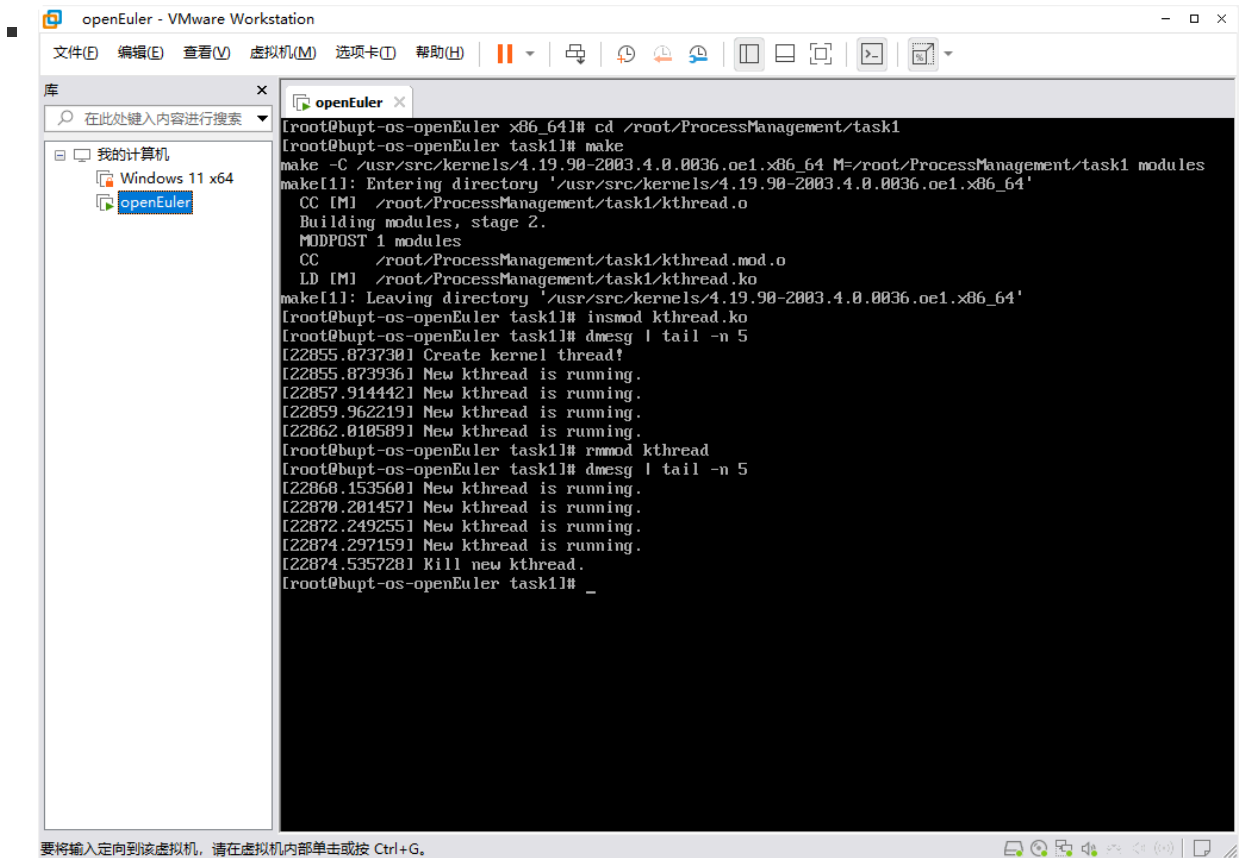


The screenshot shows a VMware Workstation window titled "openEuler - VMware Workstation". The main window displays a terminal session within a virtual machine named "openEuler". The terminal output shows the user navigating to the directory `/root/ProcessManagement/task1` and running the `make` command. The make process compiles the kernel module `kthread.o` and links it into `kthread.ko`. The terminal output is as follows:

```
[root@bupt-os-openEuler x86_64]# cd /root/ProcessManagement/task1
[root@bupt-os-openEuler task1]# make
make -C /usr/src/kernels/4.19.90-2003.4.0.0036.oe1.x86_64 M=/root/ProcessManagement/task1 modules
make[1]: Entering directory '/usr/src/kernels/4.19.90-2003.4.0.0036.oe1.x86_64'
CC [M] /root/ProcessManagement/task1/kthread.o
Building modules, stage 2.
MODPOST 1 modules
CC      /root/ProcessManagement/task1/kthread.mod.o
LD [M]  /root/ProcessManagement/task1/kthread.ko
make[1]: Leaving directory '/usr/src/kernels/4.19.90-2003.4.0.0036.oe1.x86_64'
[root@bupt-os-openEuler task1]# _
```

At the bottom of the window, a status bar contains the text: "要将输入定向到该虚拟机，请在虚拟机内部单击或按 Ctrl+G."

- 指令
 - `insmod kthread.ko`
 - `dmesg | tail -n 5`
 - `rmmod kthread`
 - `dmesg | tail -n 5`

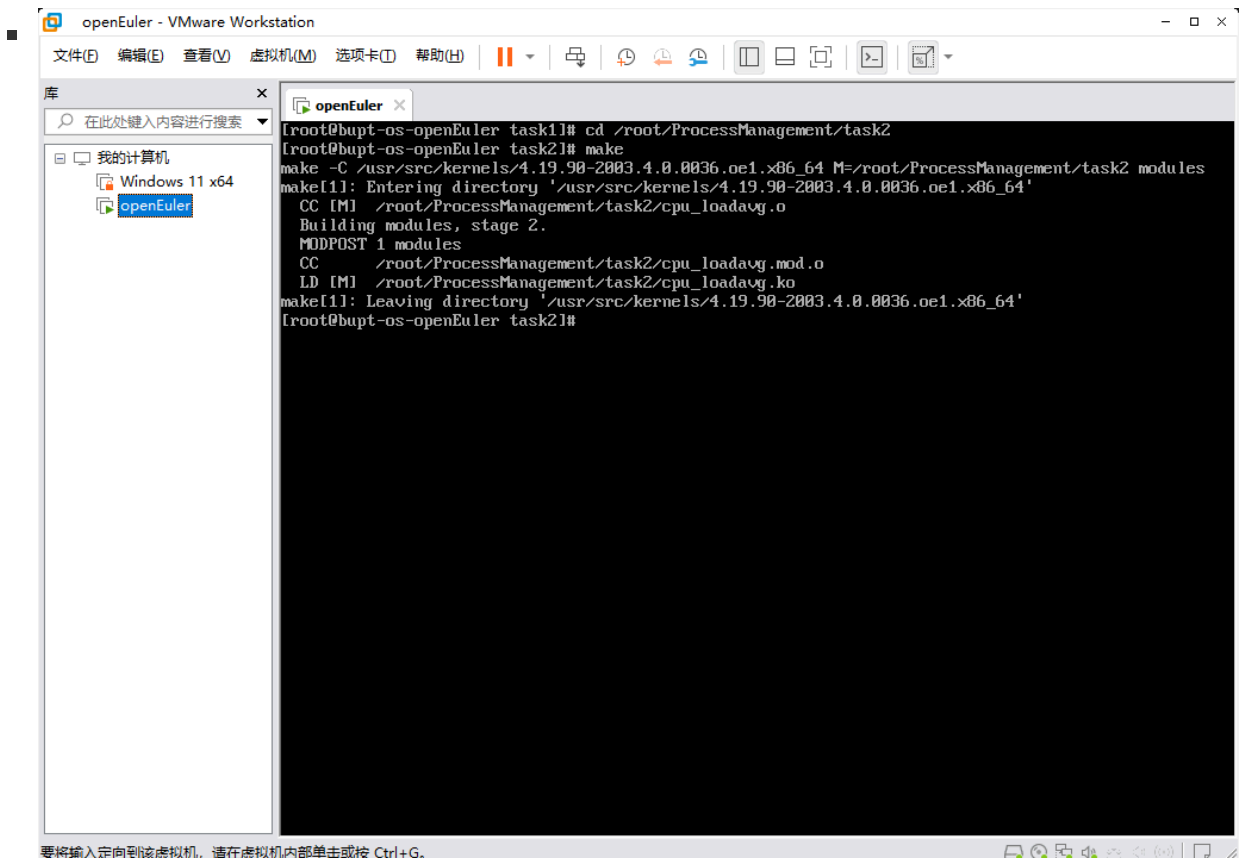


■ 打印输出当前系统 CPU 负载情况

■ 指令

- `cd /root/ProcessManagement/task2`

- `make`

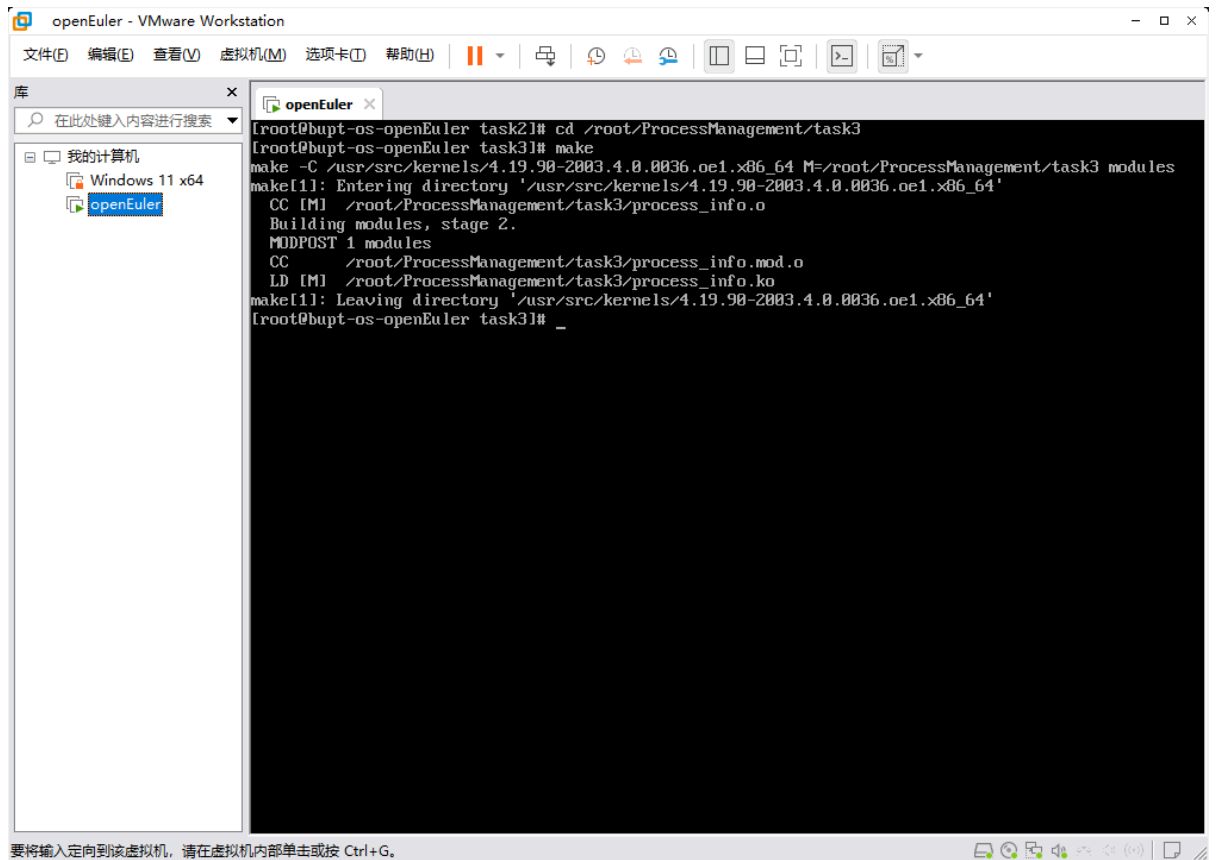


■ 指令

- insmod cpu_loadavg.ko
- dmesg | tail -n 2
- rmmod cpu_loadavg
- dmesg | tail -n 3

```
[root@bupt-os-openEuler task1]# cd /root/ProcessManagement/task2
[root@bupt-os-openEuler task2]# make
make -C /usr/src/kernels/4.19.90-2003.4.0.0036.oe1.x86_64 M=/root/ProcessManagement/task2 modules
make[1]: Entering directory '/usr/src/kernels/4.19.90-2003.4.0.0036.oe1.x86_64'
CC [M] /root/ProcessManagement/task2/cpu_loadavg.o
Building modules, stage 2.
MODPOST 1 modules
CC /root/ProcessManagement/task2/cpu_loadavg.mod.o
LD [M] /root/ProcessManagement/task2/cpu_loadavg.ko
make[1]: Leaving directory '/usr/src/kernels/4.19.90-2003.4.0.0036.oe1.x86_64'
[root@bupt-os-openEuler task2]# insmod cpu_loadavg.ko
[root@bupt-os-openEuler task2]# dmesg | tail -n 2
[22972.897474] Start cpu_loadavg!
[22972.897481] The cpu loadavg in one minute is: 0.09
[root@bupt-os-openEuler task2]# rmmod cpu_loadavg
[root@bupt-os-openEuler task2]# dmesg | tail -n 3
[22972.897474] Start cpu_loadavg!
[22972.897481] The cpu loadavg in one minute is: 0.09
[22993.073760] Exit cpu_loadavg!
[root@bupt-os-openEuler task2]#
```

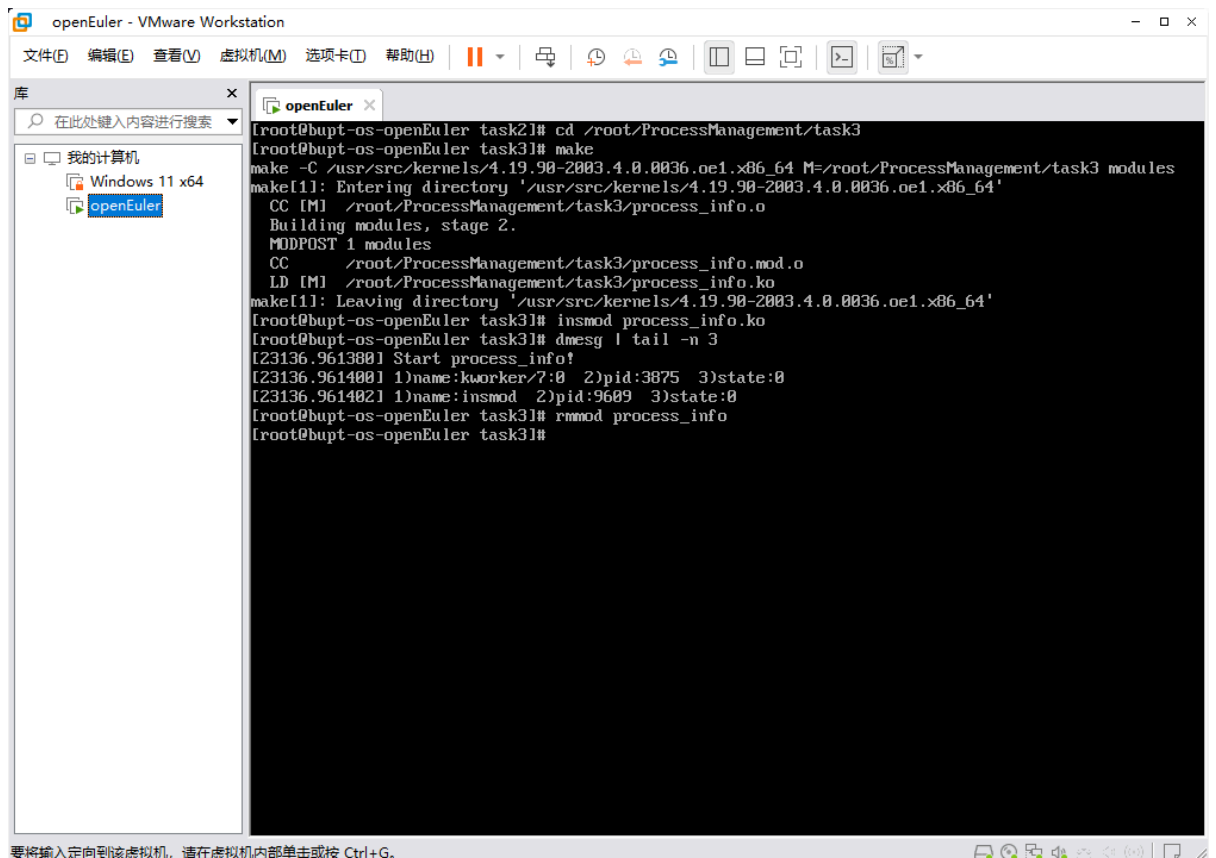
- 打印输出当前处于运行状态的进程的 PID 和名字
 - 指令
 - cd /root/ProcessManagement/task3
 - make



要将输入定向到该虚拟机，请在虚拟机内部单击或按 Ctrl+G。

指令

- insmod process_info.ko
- dmesg | tail -n 3
- rmmod process_info



要将输入定向到该虚拟机，请在虚拟机内部单击或按 Ctrl+G。

- 使用 cgroup 实现限制 CPU 核数

■ 指令

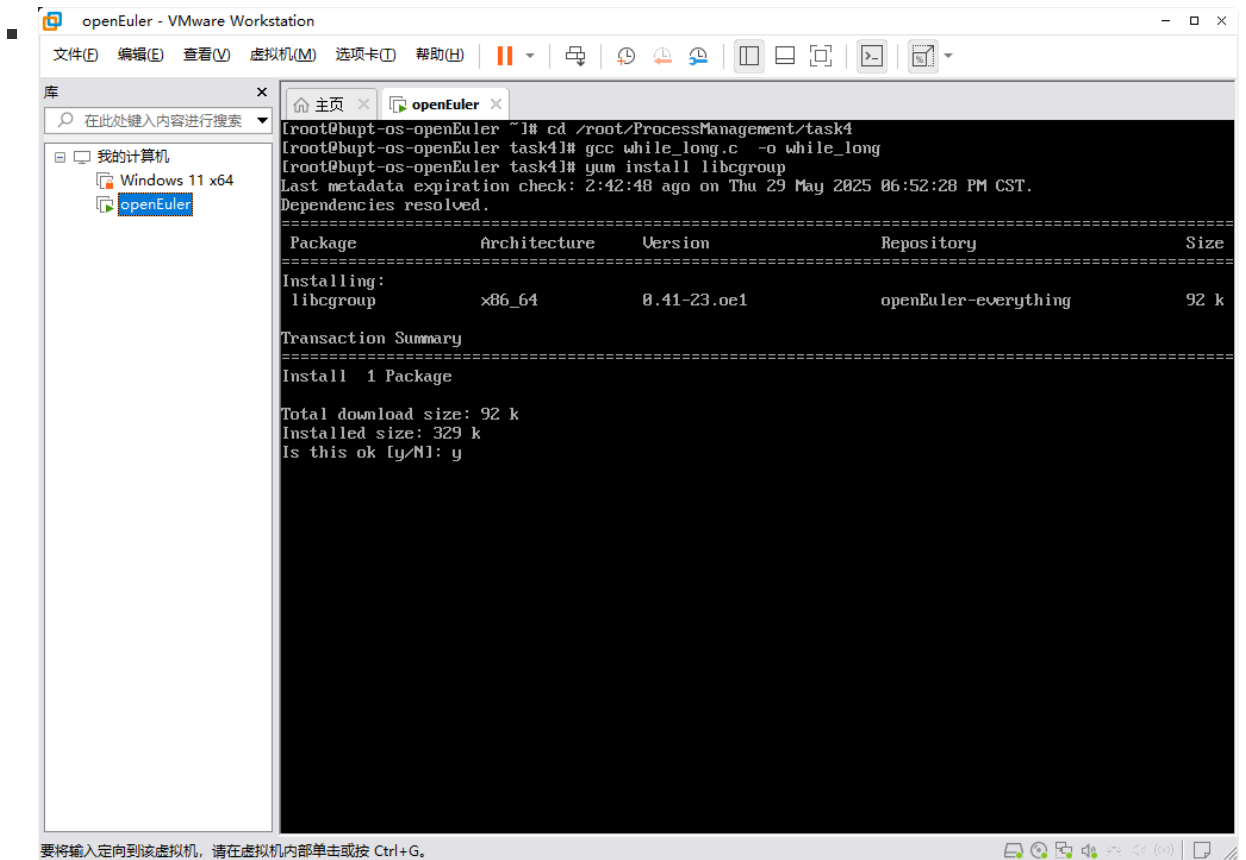
- `mkdir /cgroup`
- `mount -t tmpfs tmpfs /cgroup`
- `cd /cgroup`
- `mkdir cpuset`
- `mount -t cgroup -o cpuset cpuset /cgroup/cpuset`
- `cd cpuset`
- `mkdir mycpuset`
- `cd mycpuset`
- `cat cpuset.mems`
- `echo 0 > cpuset.mems`
- `cat cpuset.cpus`
- `echo 0-2 > cpuset.cpus`
- `cat cpuset.mems`
- `cat cpuset.cpus`

```
openEuler - VMware Workstation
文件(F) 编辑(E) 查看(V) 虚拟机(M) 选项卡(T) 帮助(H)
库
在此处键入内容进行搜索
我的计算机
Windows 11 x64
openEuler
[root@bupt-os-openEuler ~]# mkdir /cgroup
[root@bupt-os-openEuler ~]# mount -t tmpfs tmpfs /cgroup
[root@bupt-os-openEuler ~]# cd /cgroup/
[root@bupt-os-openEuler cgroup]# mkdir cpuset
[root@bupt-os-openEuler cgroup]# mount -t cgroup -o cpuset cpuset /cgroup/cpuset/
[root@bupt-os-openEuler cgroup]# cd cpuset/
[root@bupt-os-openEuler cpuset]# mkdir mycpuset
[root@bupt-os-openEuler cpuset]# cd mycpuset/
[root@bupt-os-openEuler mycpuset]# cat cpuset.mems

[root@bupt-os-openEuler mycpuset]# echo 0 > cpuset.mem
cpuset.mem_exclusive      cpuset.memory_pressure  cpuset.mems
cpuset.mem_hardwall       cpuset.memory_spread_page
cpuset.memory_migrate     cpuset.memory_spread_slab
[root@bupt-os-openEuler mycpuset]# echo 0 > cpuset.mems
[root@bupt-os-openEuler mycpuset]# cat cpuset.cpus
0
0-2
[root@bupt-os-openEuler mycpuset]# cat cpuset.cpus
0-2
[root@bupt-os-openEuler mycpuset]#
要将输入定向到该虚拟机，请在虚拟机内部单击或按 Ctrl+G.
```

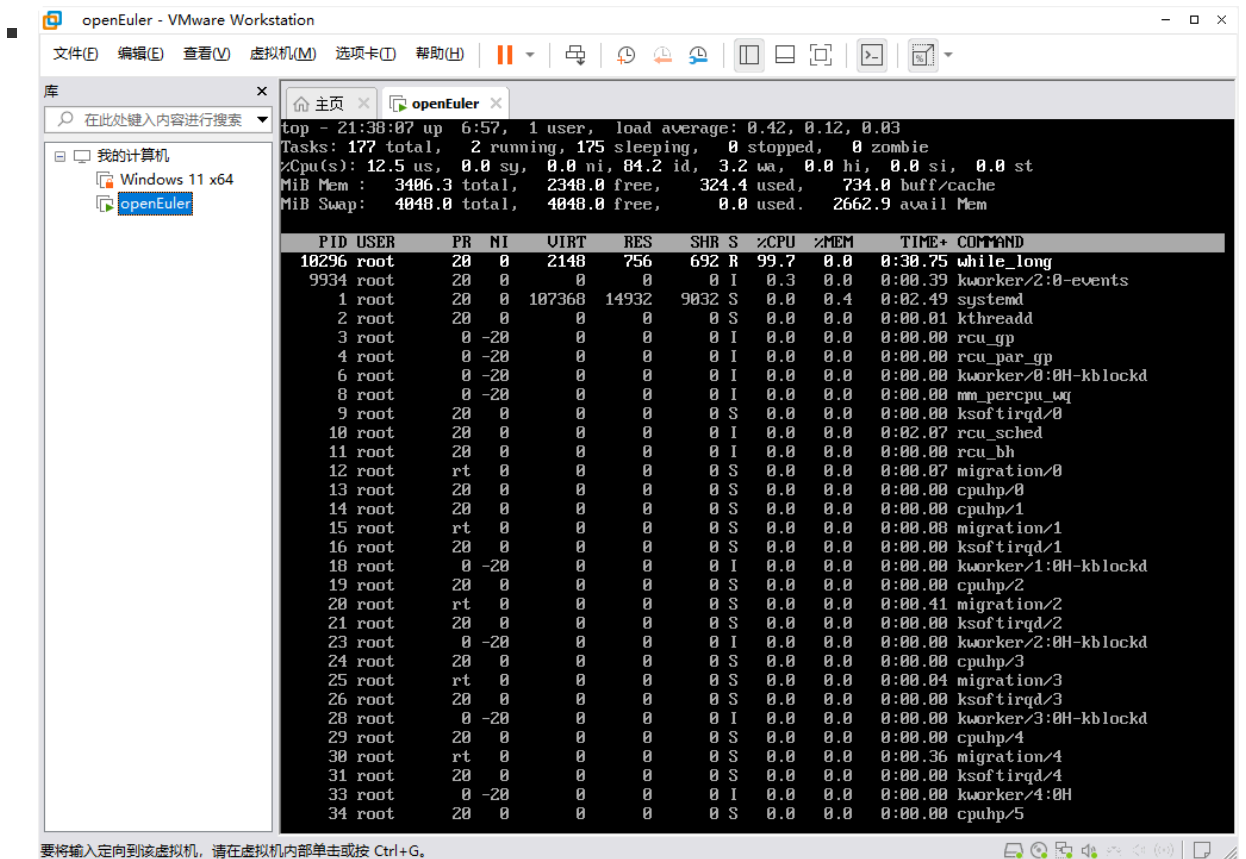
■ 指令

- `cd /root/ProcessManagement/task4`
- `gcc while_long.c -o while_long`
- `yum install libcgroup`



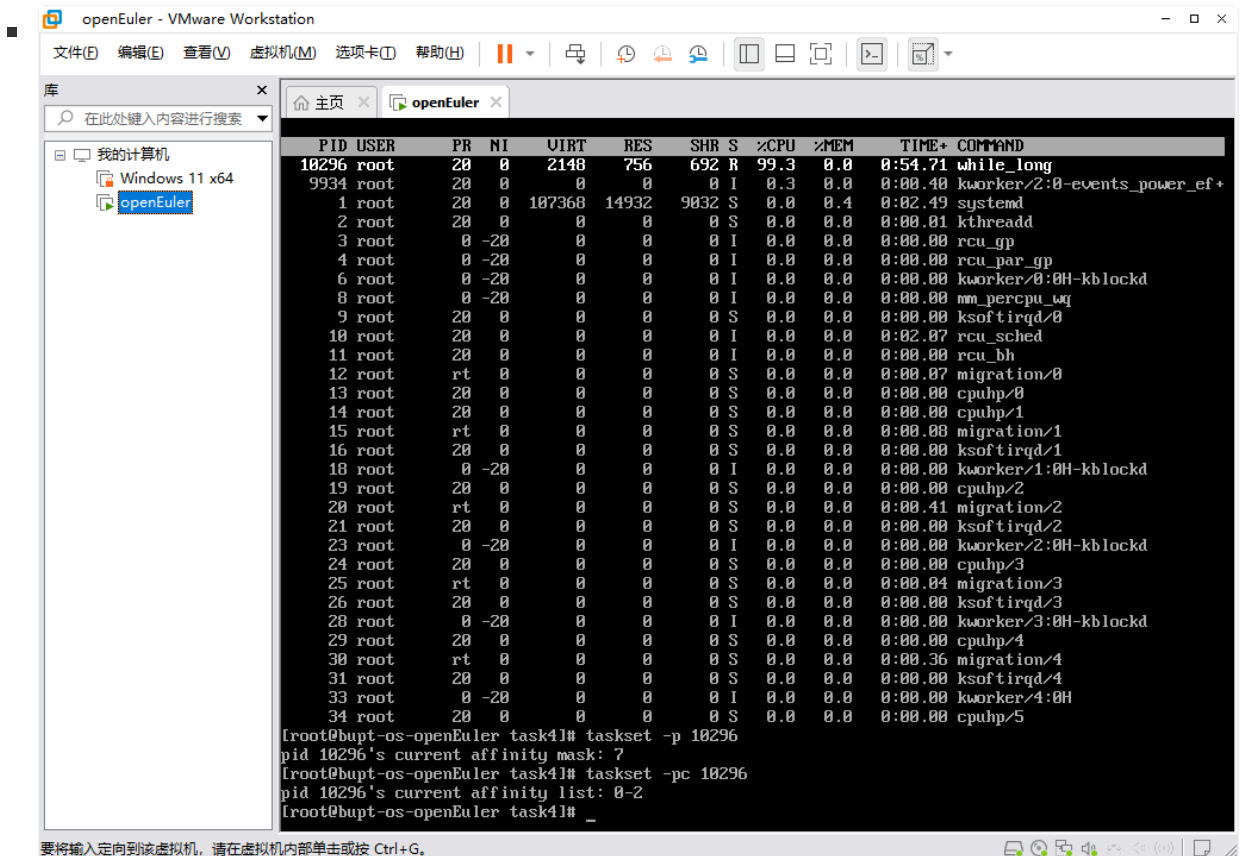
指令

- top



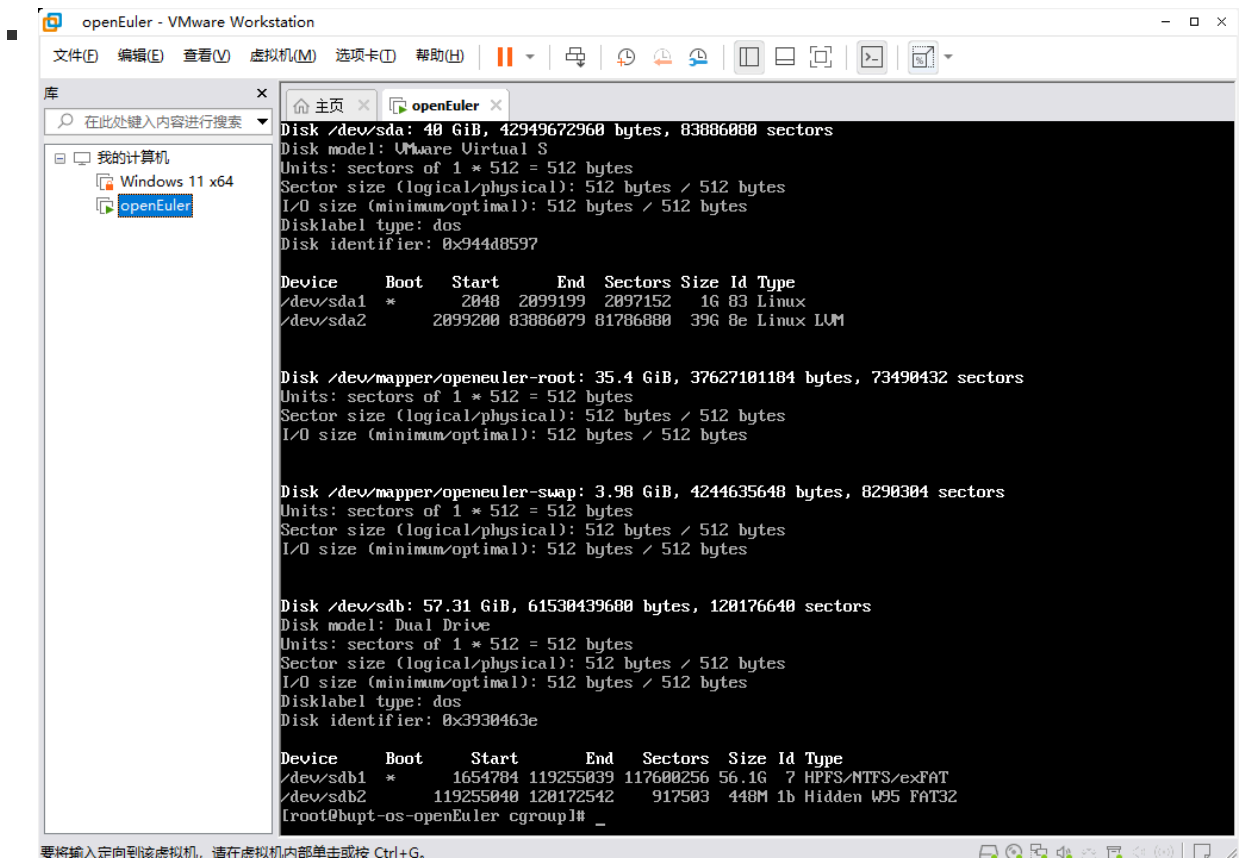
指令

- taskset -p 10296
- taskset -pc 10296



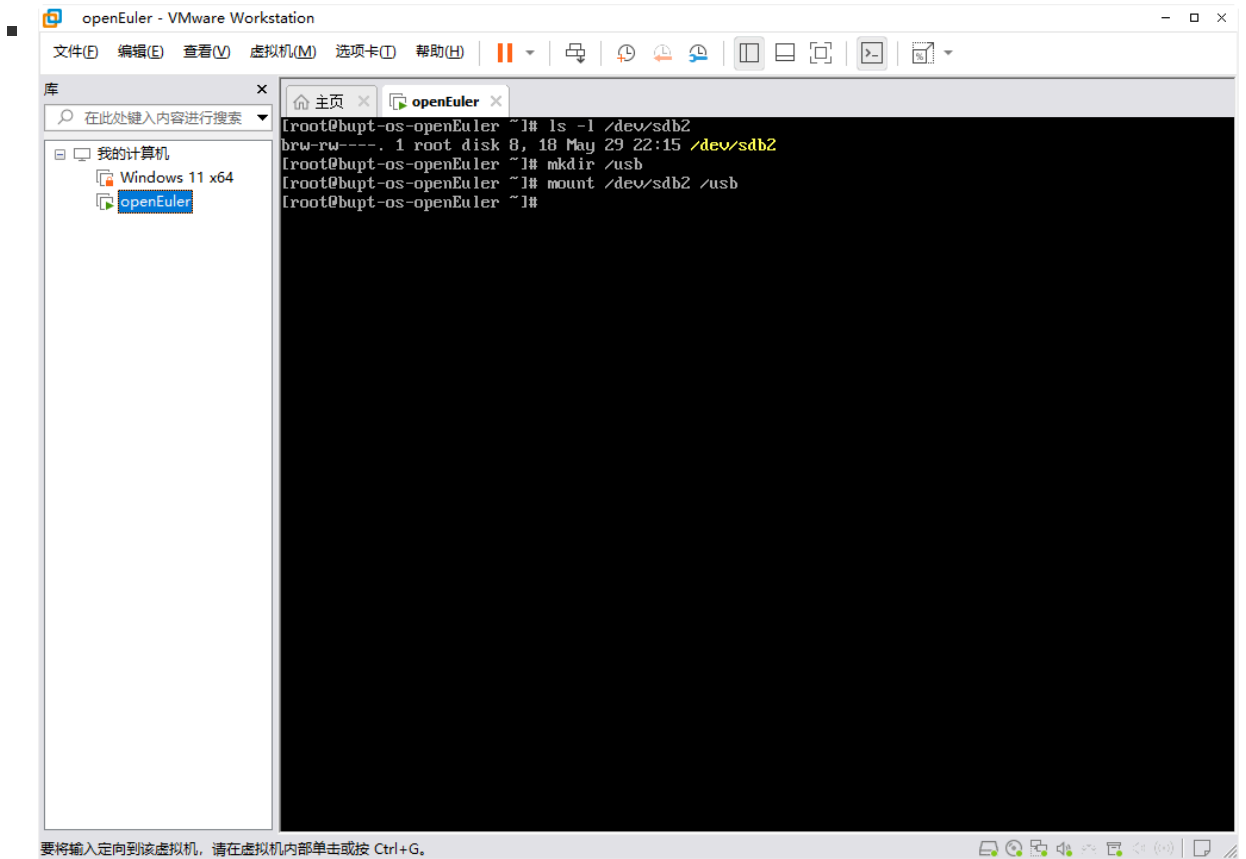
- 测试限制 cpu 核数成功
- 使用 cgroup 实现不允许访问U盘
- 指令

fdisk -l



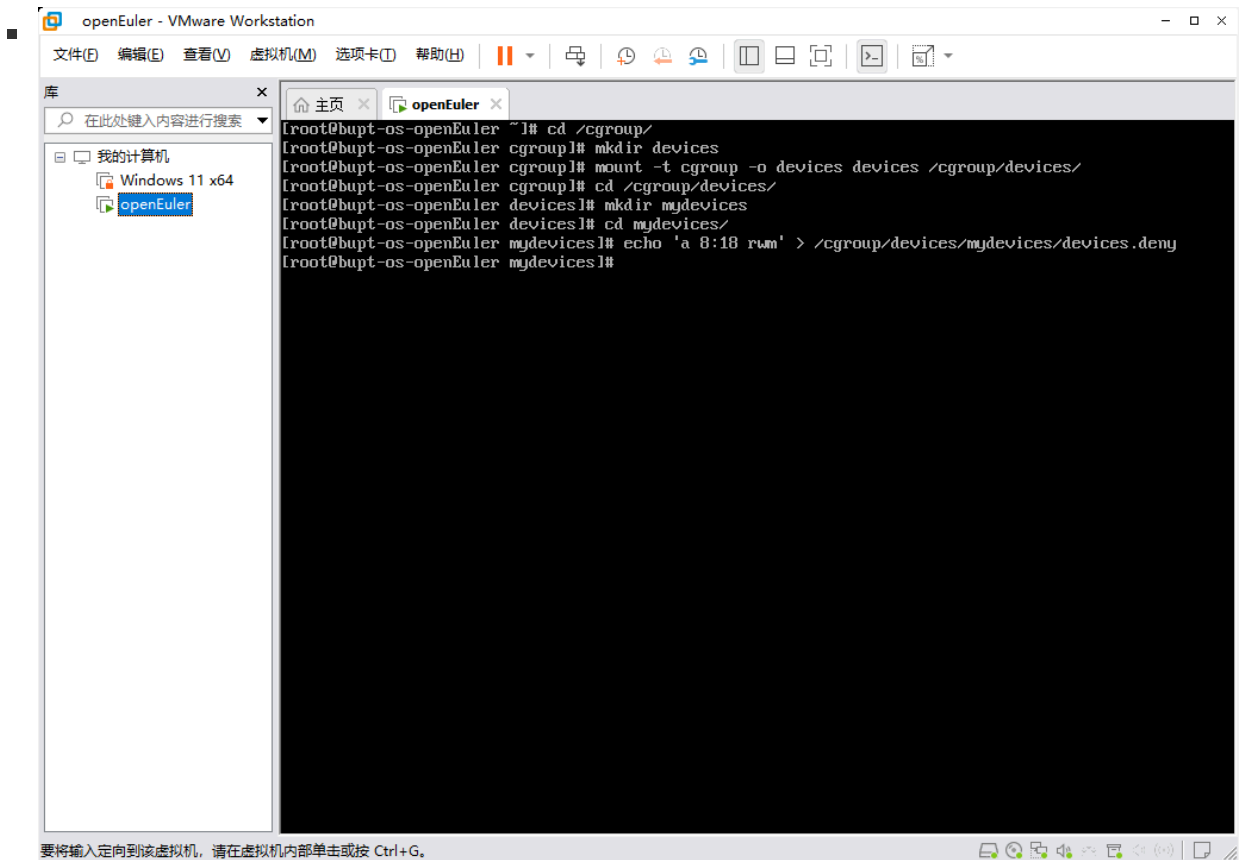
- 指令

- `ls -l /dev/sdb2`
- `mkdir /usb`
- `mount /dev/sdb2 /usb`



■ 指令

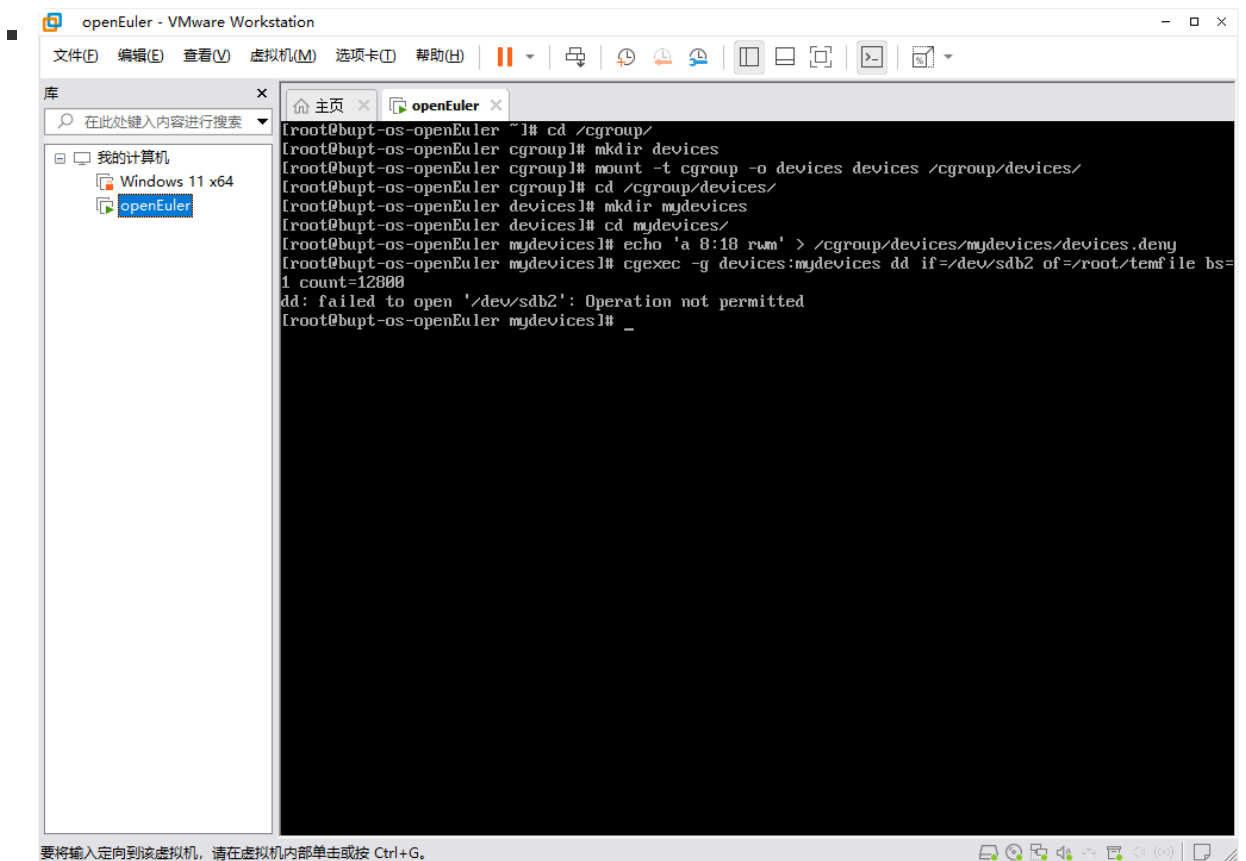
- `cd /cgroup/`
- `mkdir devices`
- `mount -t cgroup -o devices devices /cgroup/devices`
- `cd /cgroup/devices`
- `mkdir mydevices`
- `cd mydevices`
- `echo 'a 8:18 rwm' > /cgroup/devices/mydevices/devices.deny`



要将输入定向到该虚拟机，请在虚拟机内部单击或按 Ctrl+G。

■ 指令

- `cgexec -g devices:mydevices dd if=/dev/sdb2 of=/root/temfile bs=1 count=12800`



要将输入定向到该虚拟机，请在虚拟机内部单击或按 Ctrl+G。

- 可见终端显示 Operation not permitted
- 说明不允许访问u盘设置成功