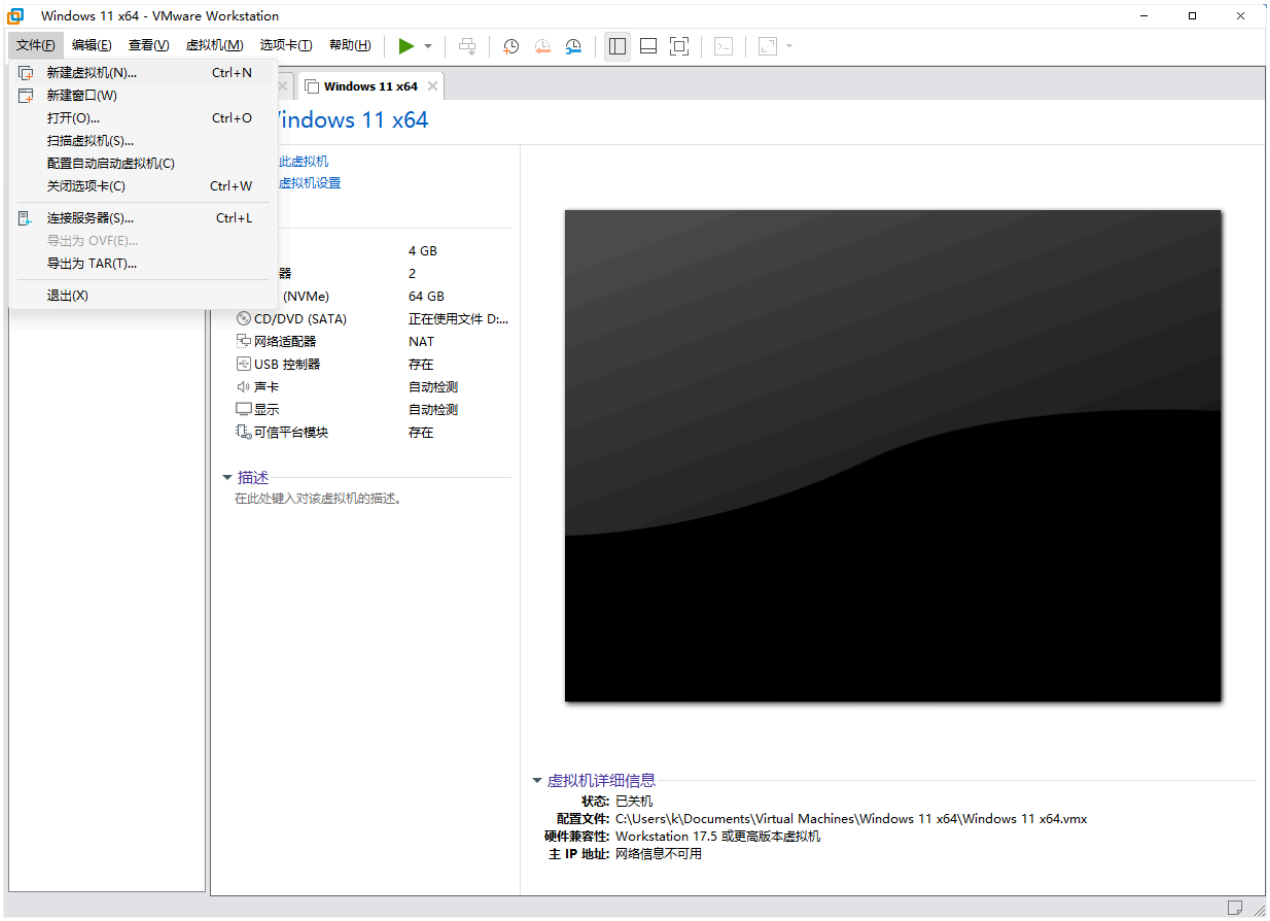
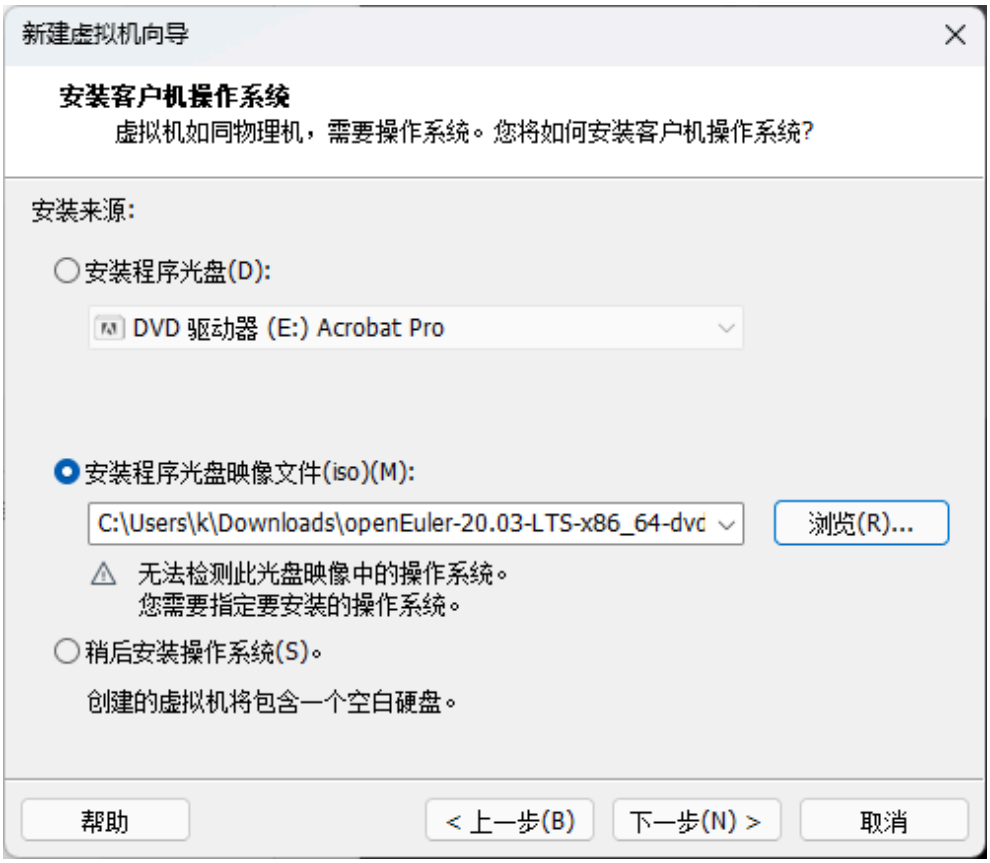
# OpenEuler 实验报告

## 实验步骤

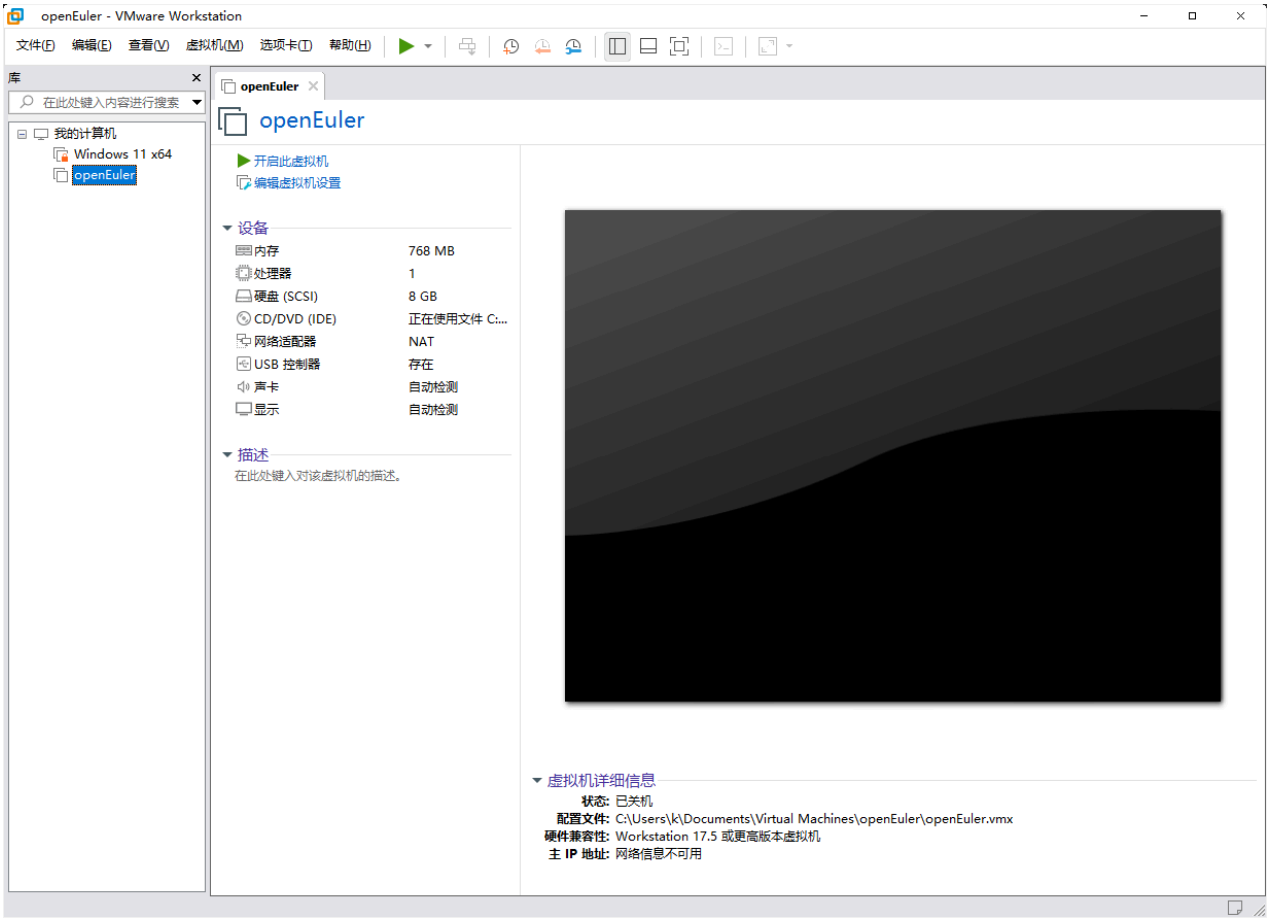
- 用虚拟机 VMware 完成 OpenEuler 操作系统的安装
  - 安装
    - 在 VMware 中新建虚拟机
    - 

    - 选择下载的 iso 文件

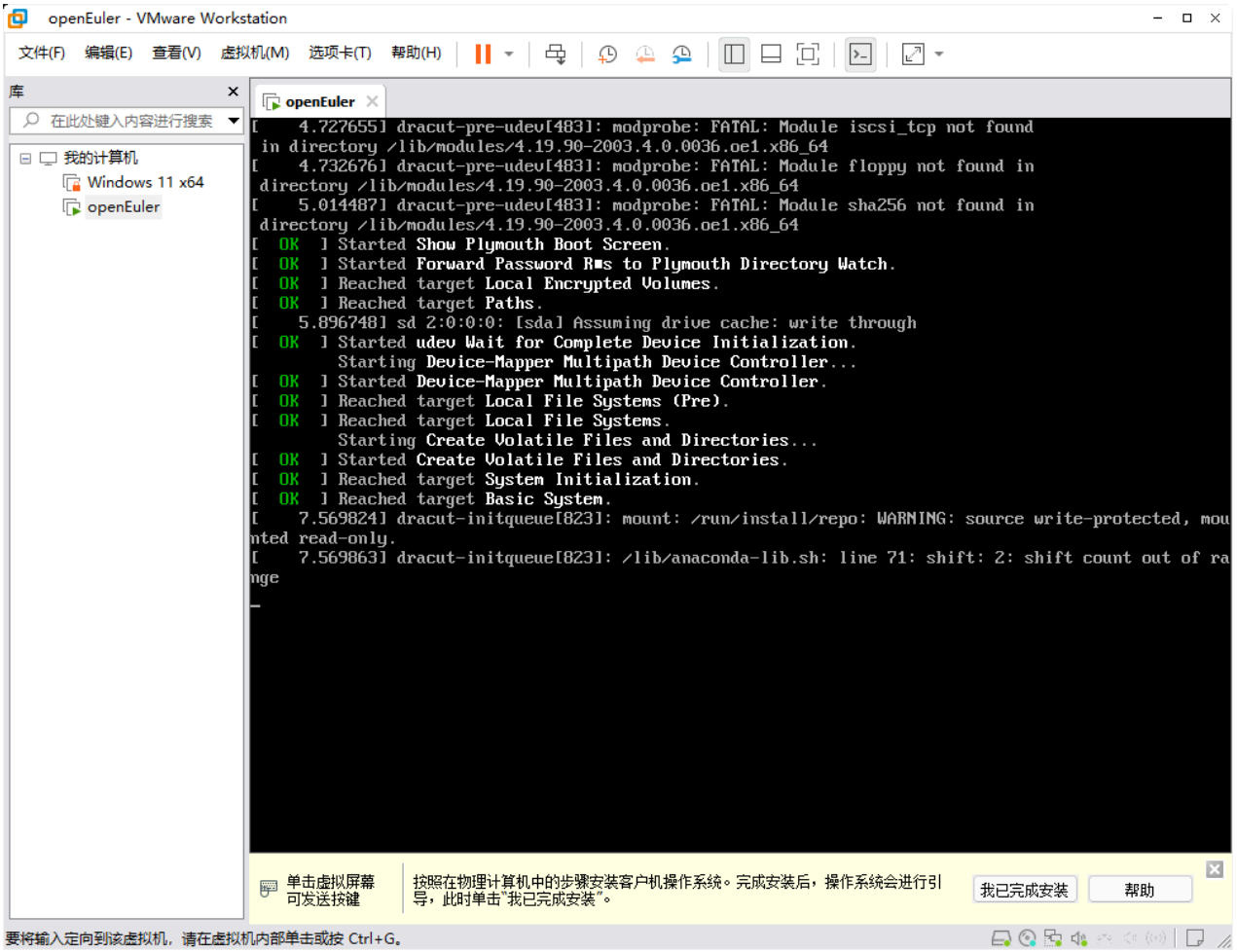
- 

- 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

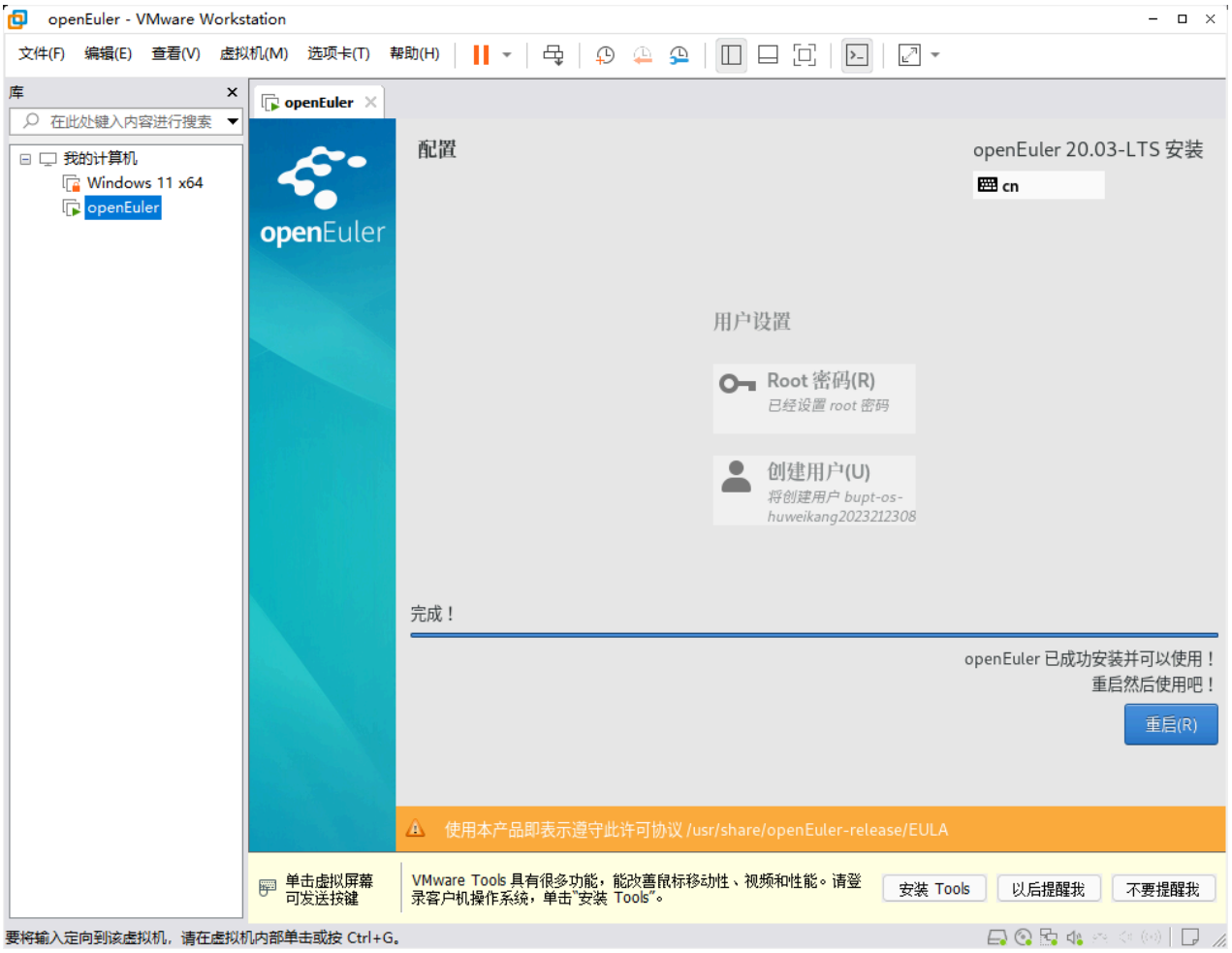



- 启动虚拟机 等待系统安装

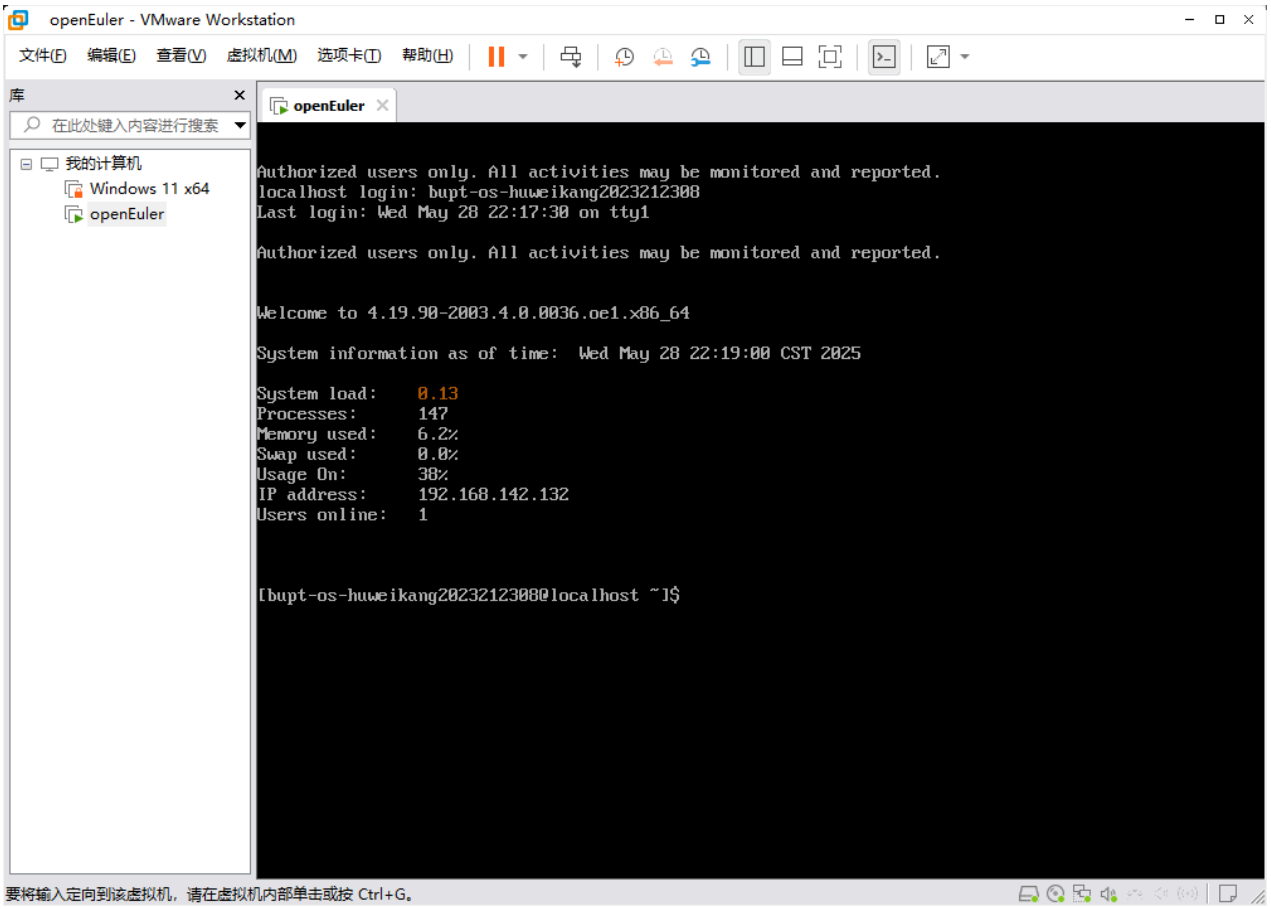


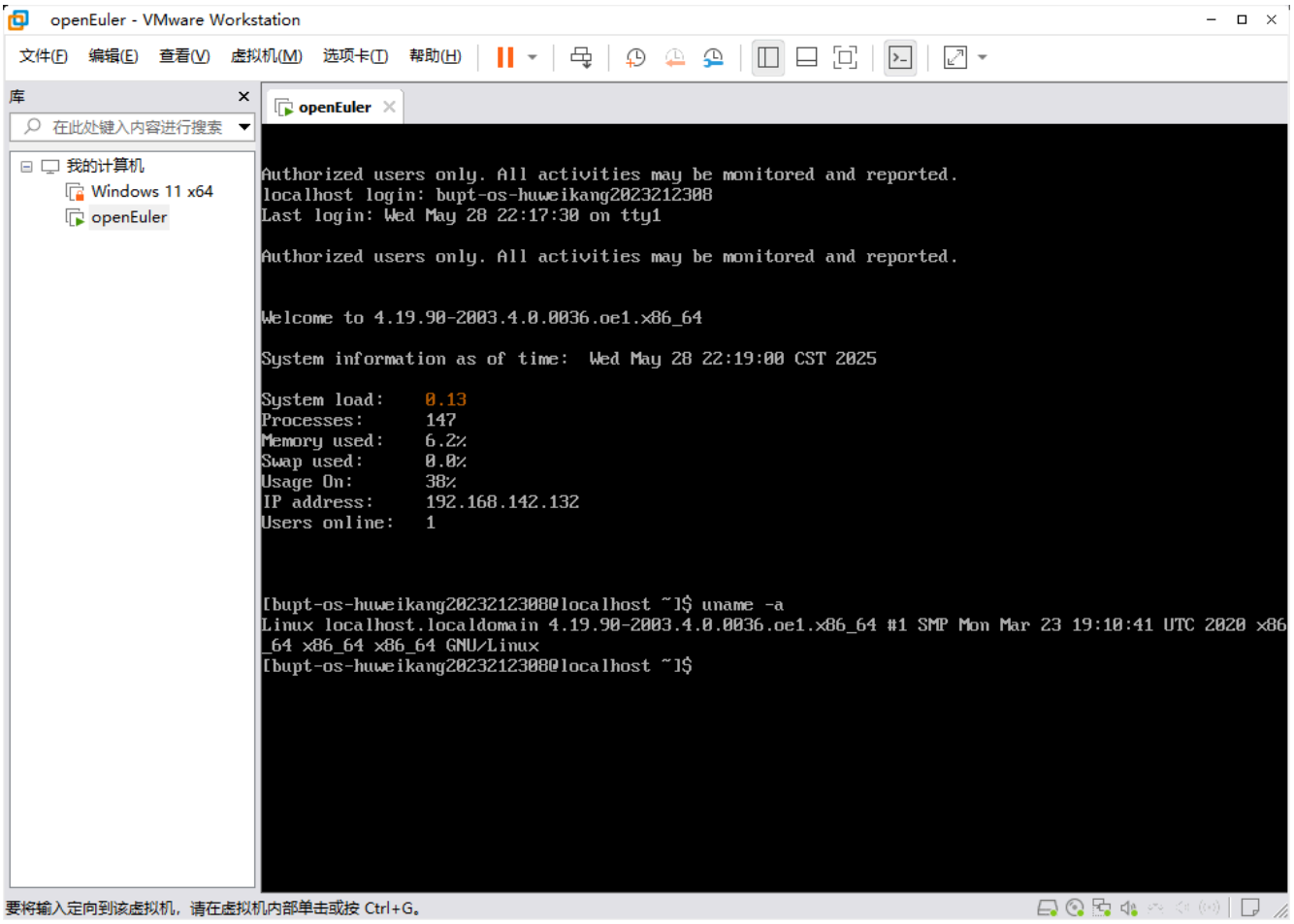
- 设置 Root 密码 创建用户
-

- 
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
openEuler
配置
openEuler 20.03-LTS 安装
cn
用户设置
Root 密码(R)
已经设置 root 密码
创建用户(U)
将创建用户 bupt-os-huweikang2023212308
正在安装 python2-setuptools.noarch (481/529)
遍布全球的技术团队、合作科研机构，构筑坚实后盾
Building unbreakable capabilities hand-in-hand technical teams and research organizations worldwide
单击虚拟屏幕可发送按键
VMware Tools 具有很多功能，能改善鼠标移动性、视频和性能。请登录客户机操作系统，单击"安装 Tools"。
安装 Tools
以后提醒我
不要提醒我
要将输入定向到该虚拟机，请在虚拟机内部单击或按 Ctrl+G。
- 安装完毕 重启虚拟机

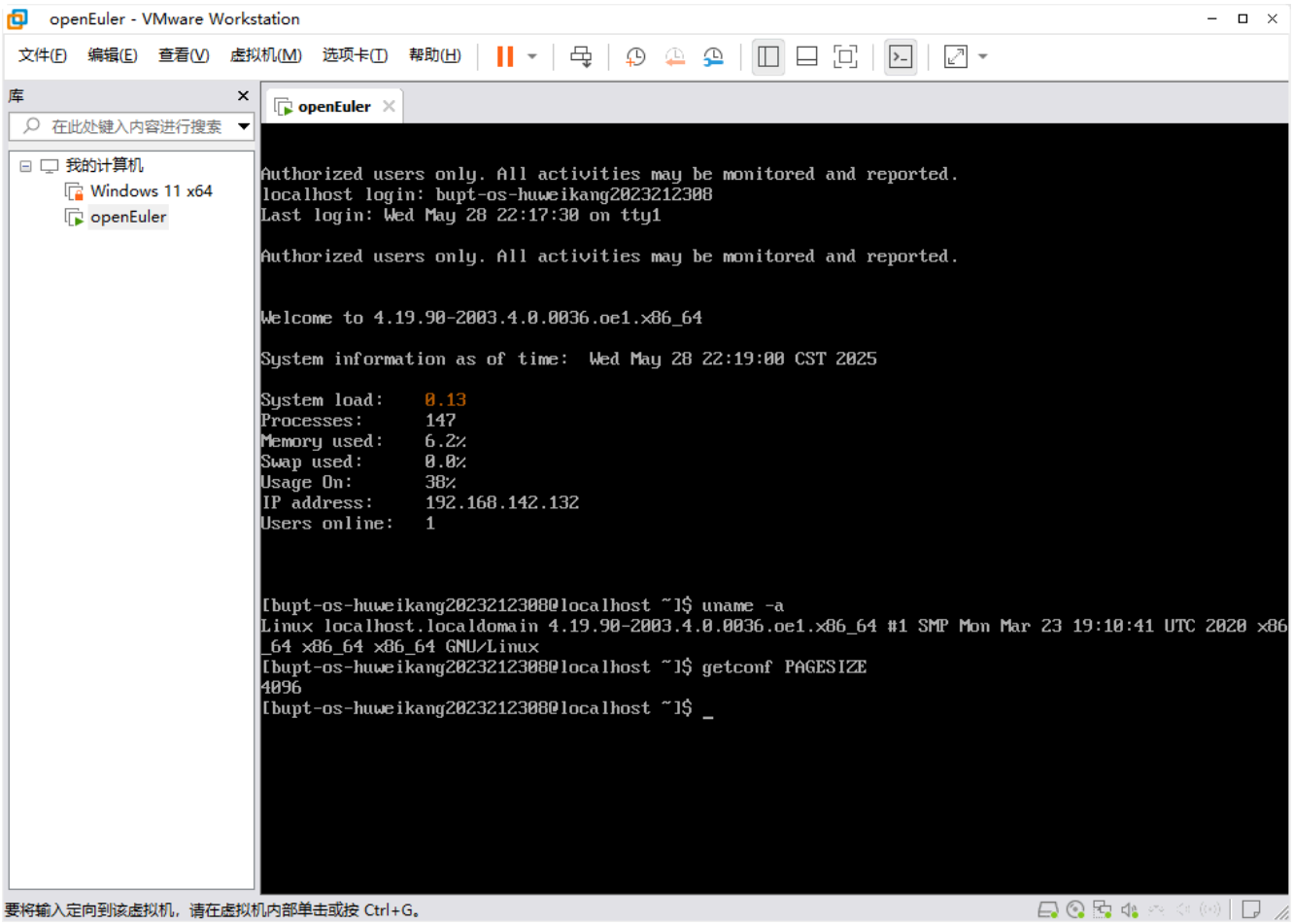



登入 bupt-os-huweikang2023212308

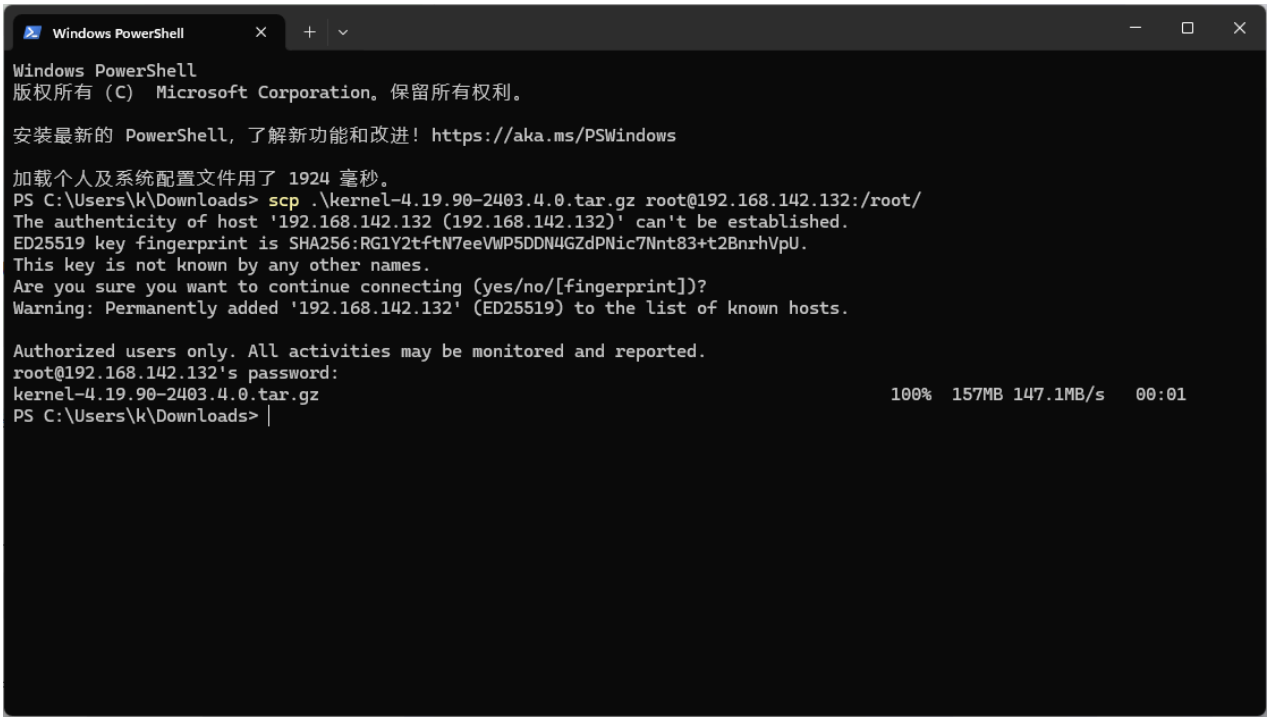


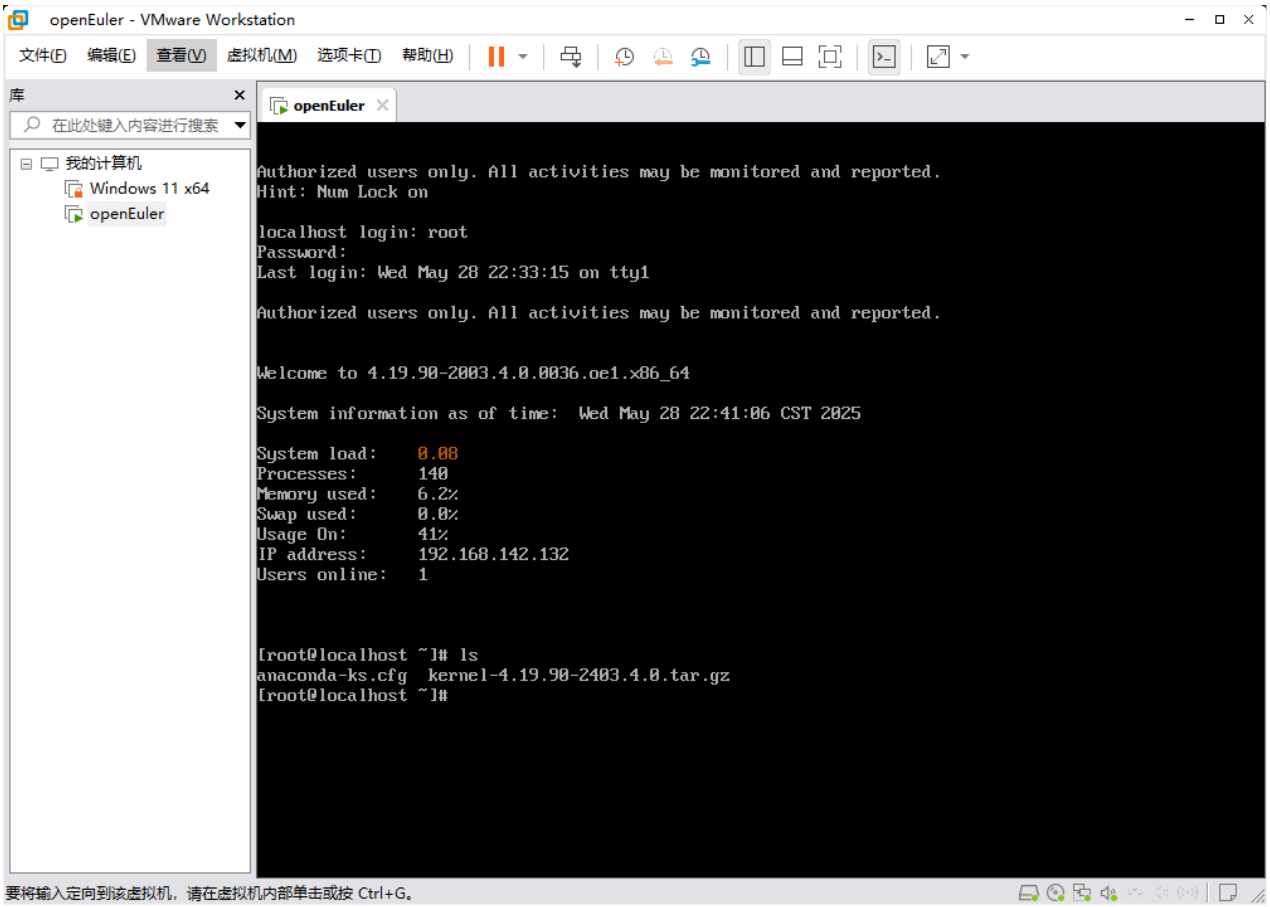
- uname -a 指令
- 

- getconf PAGESIZE 指令

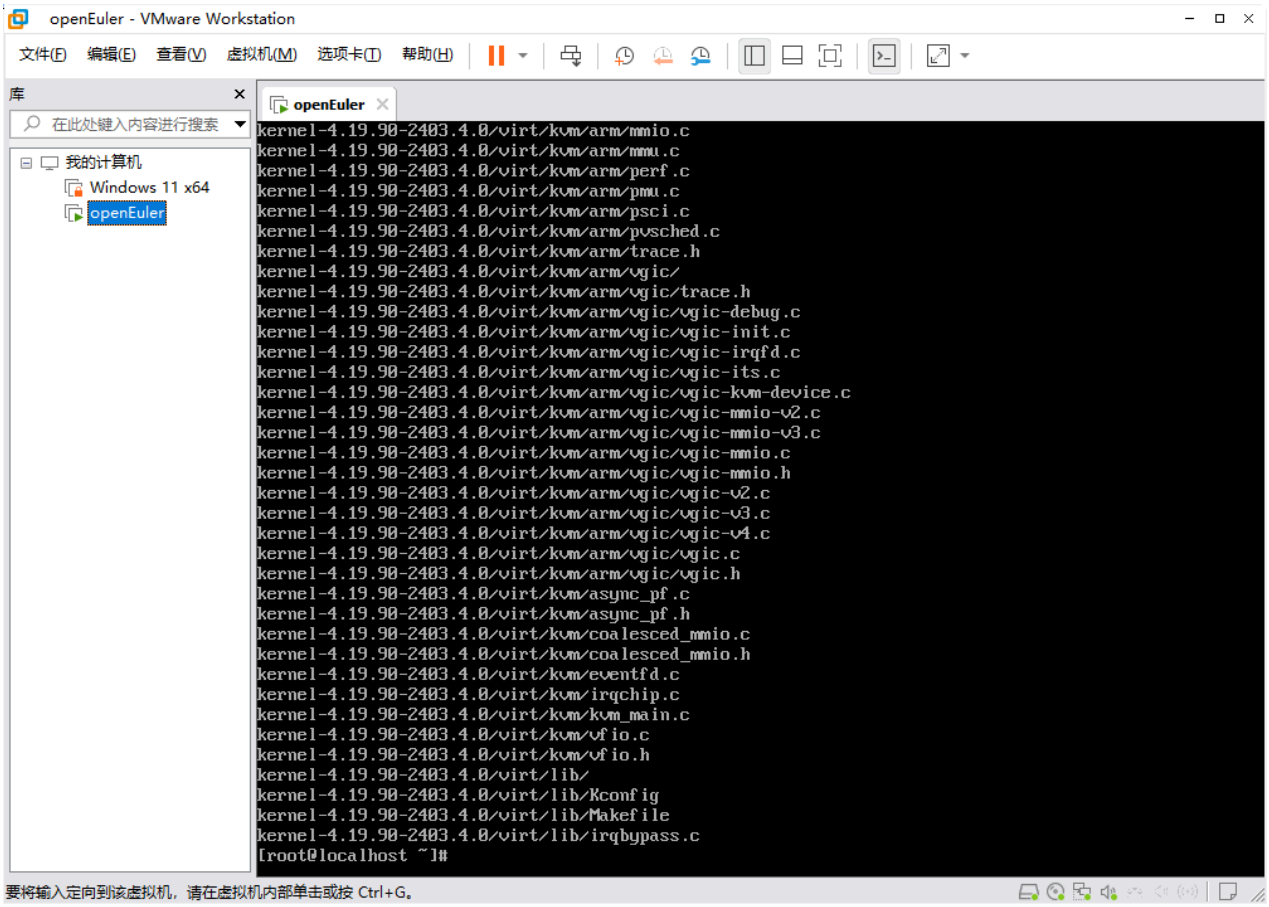
- 


- 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

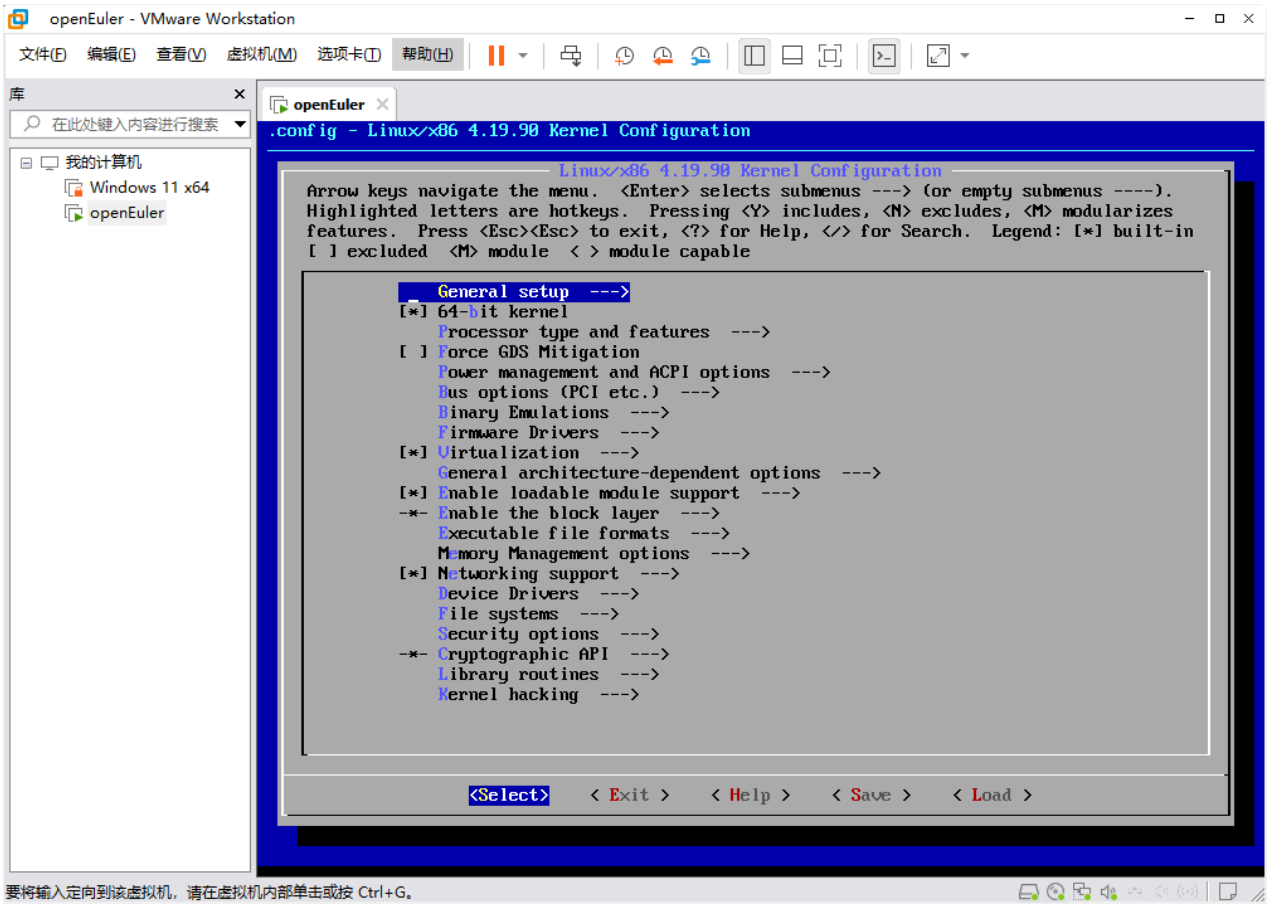
- 

  - 指令
    - ls
  - 可以发现成功传入
- 

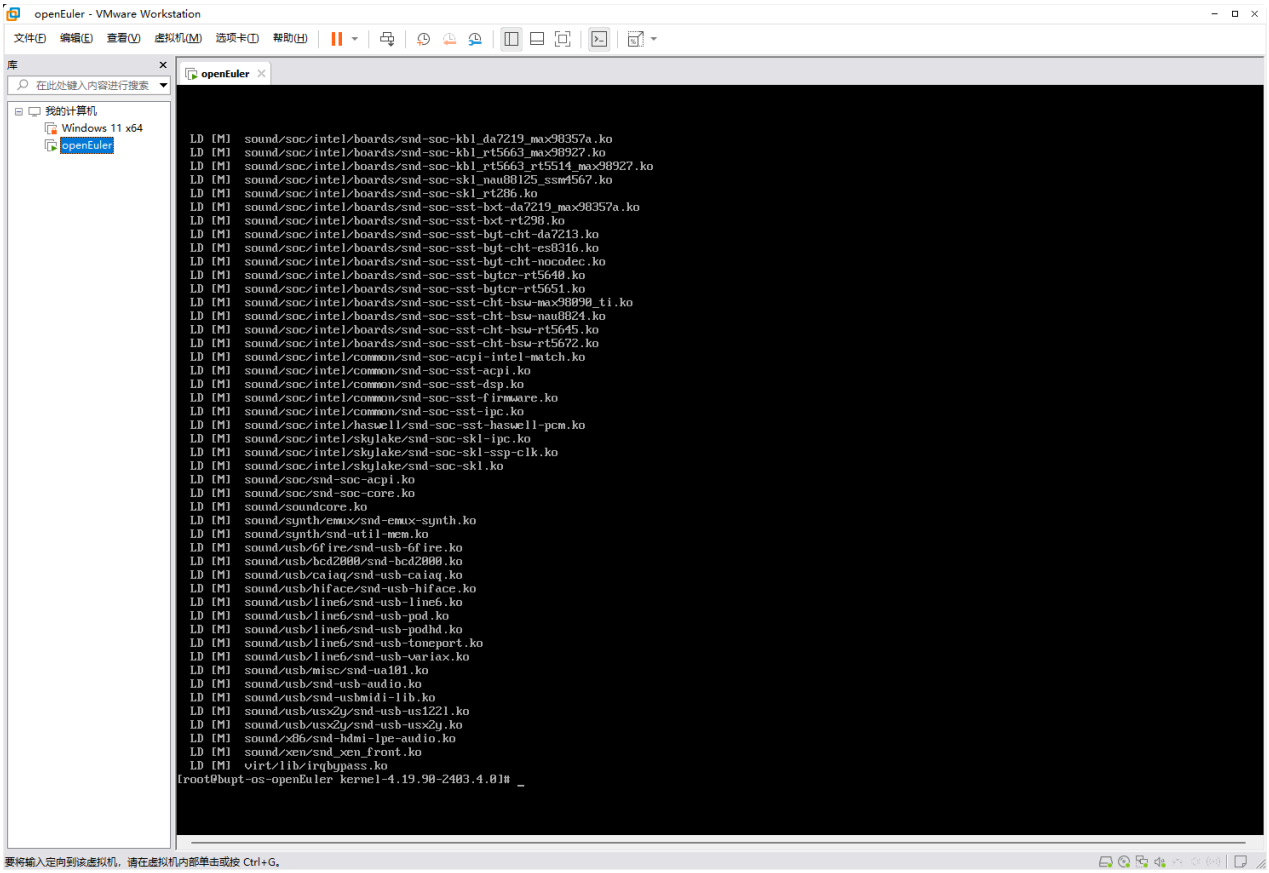

- 解压文件
  - 指令
    - tar -xvf kernel-4.19.90-2403.4.0.tar.gz

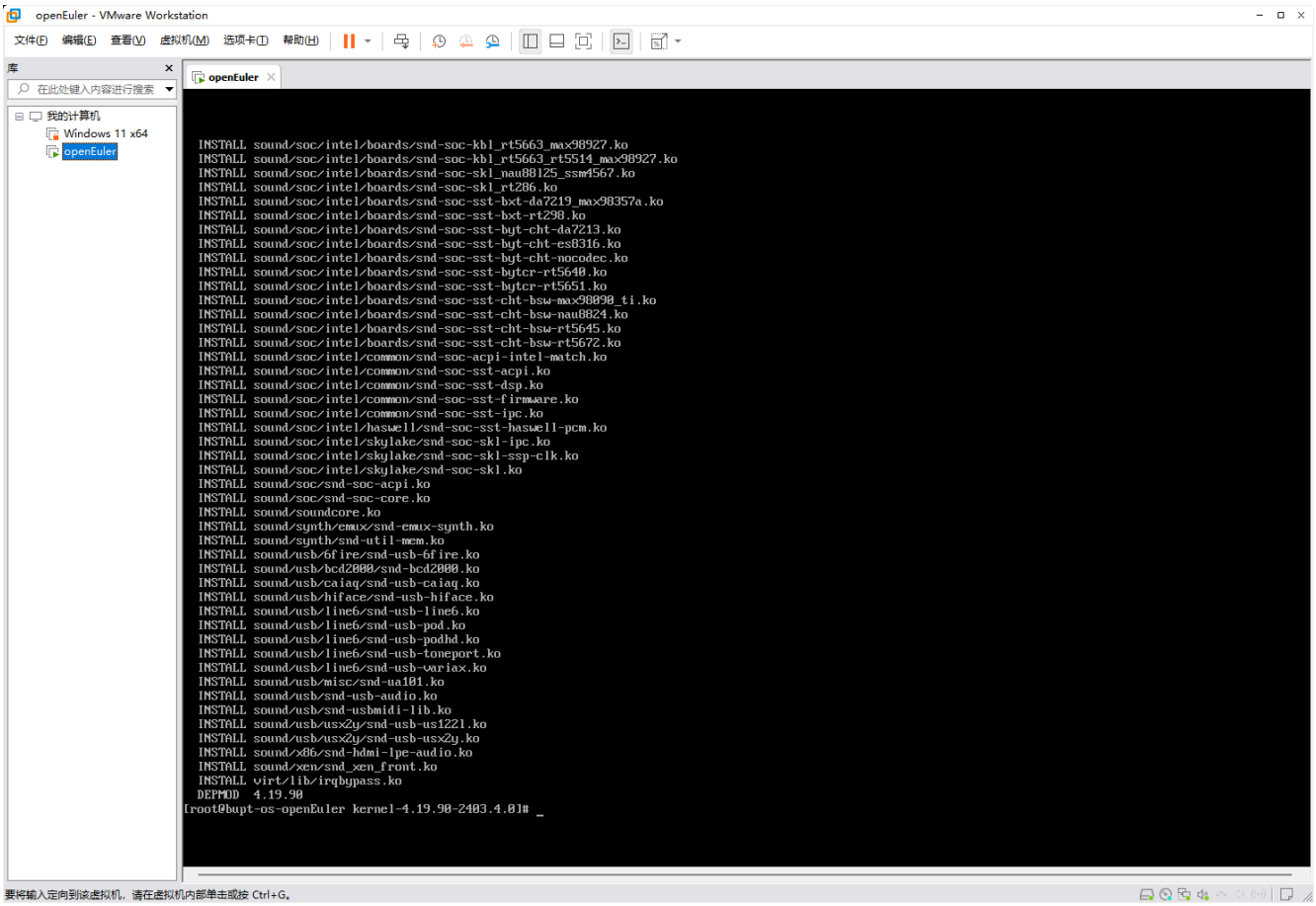
- 

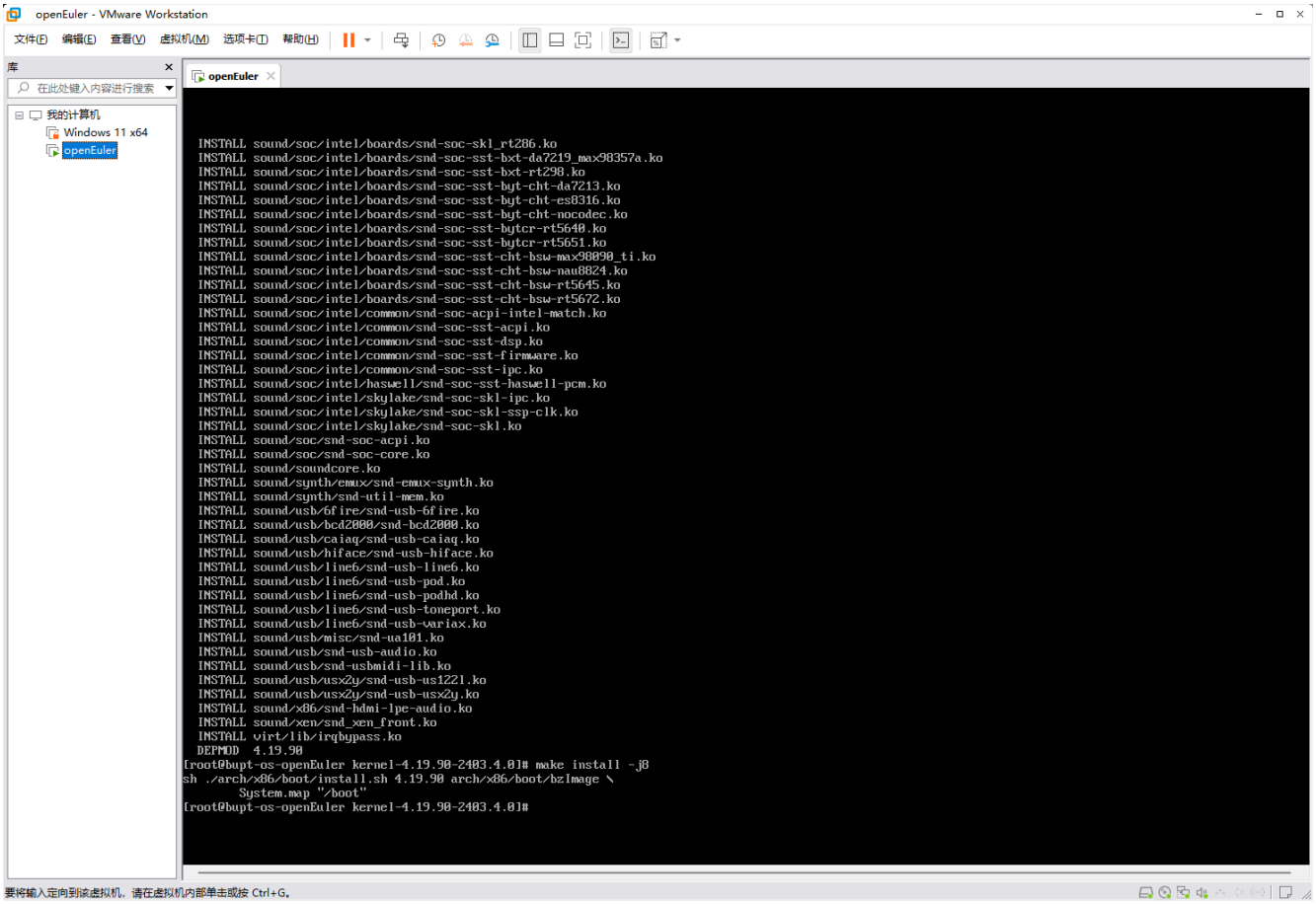

- 编译源代码
  - 指令
    - `cd kernel-4.19.90-2403.4.0`
  - 安装依赖
    - `yum install ncurses-devel`
    - `yum install flex`
    - `yum install bc`
    - `yum install openssl-devel`
    - `yum install elfutils-libelf-devel`
  - 指令
    - `make menuconfig`

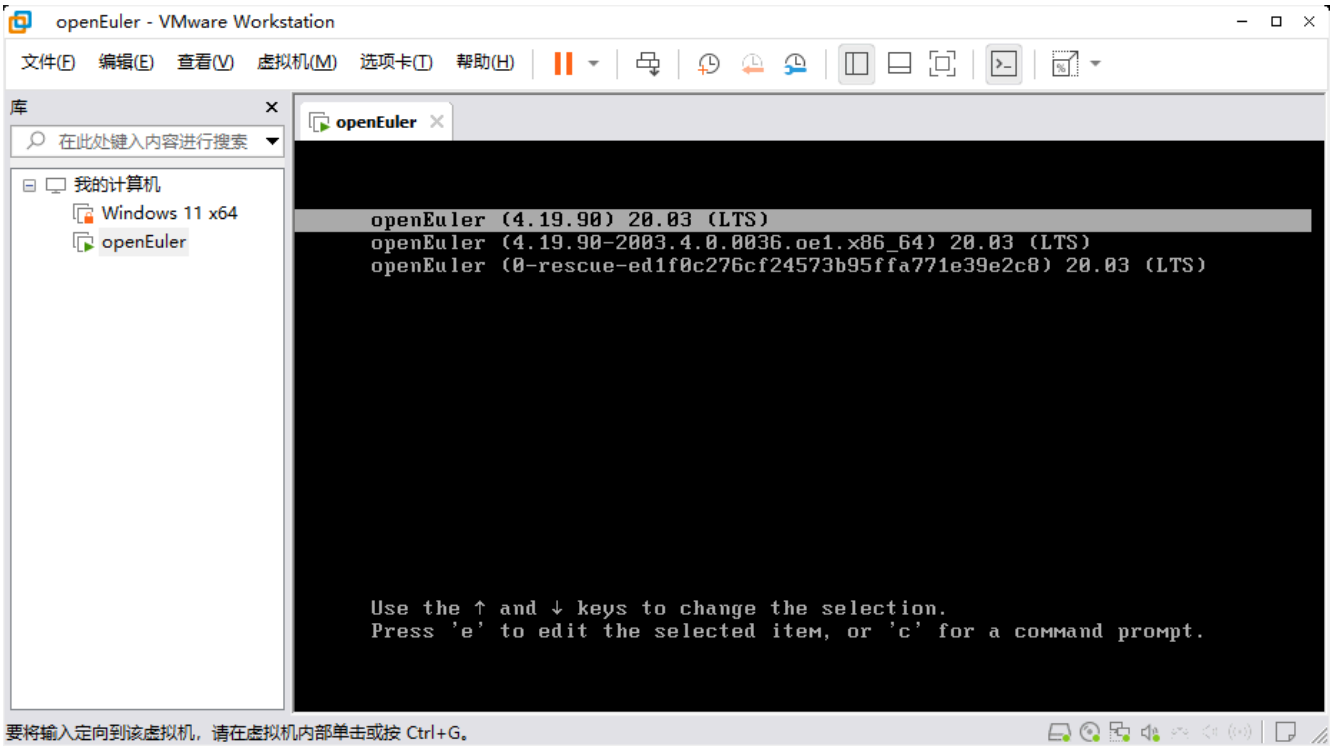
- 

- 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

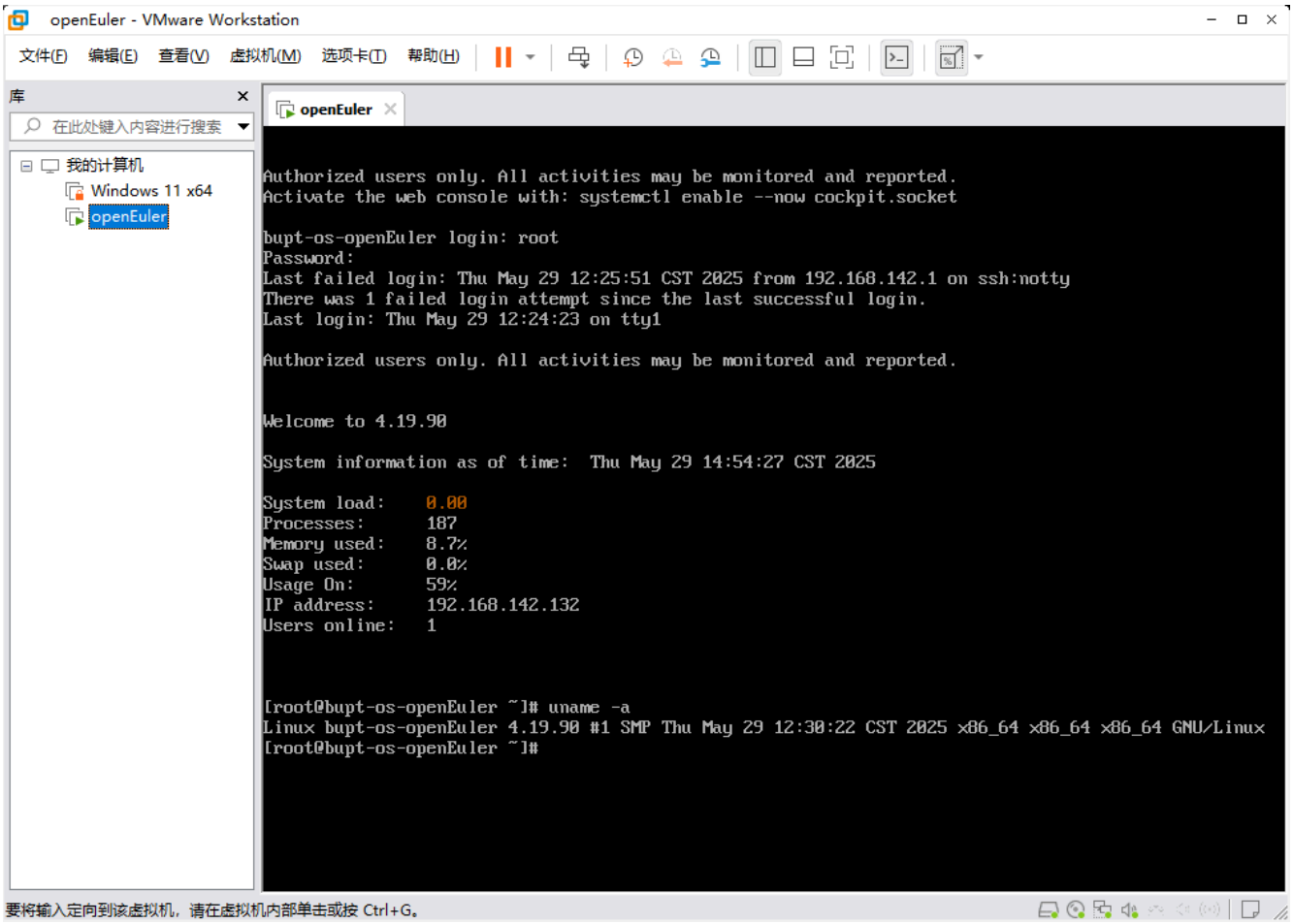


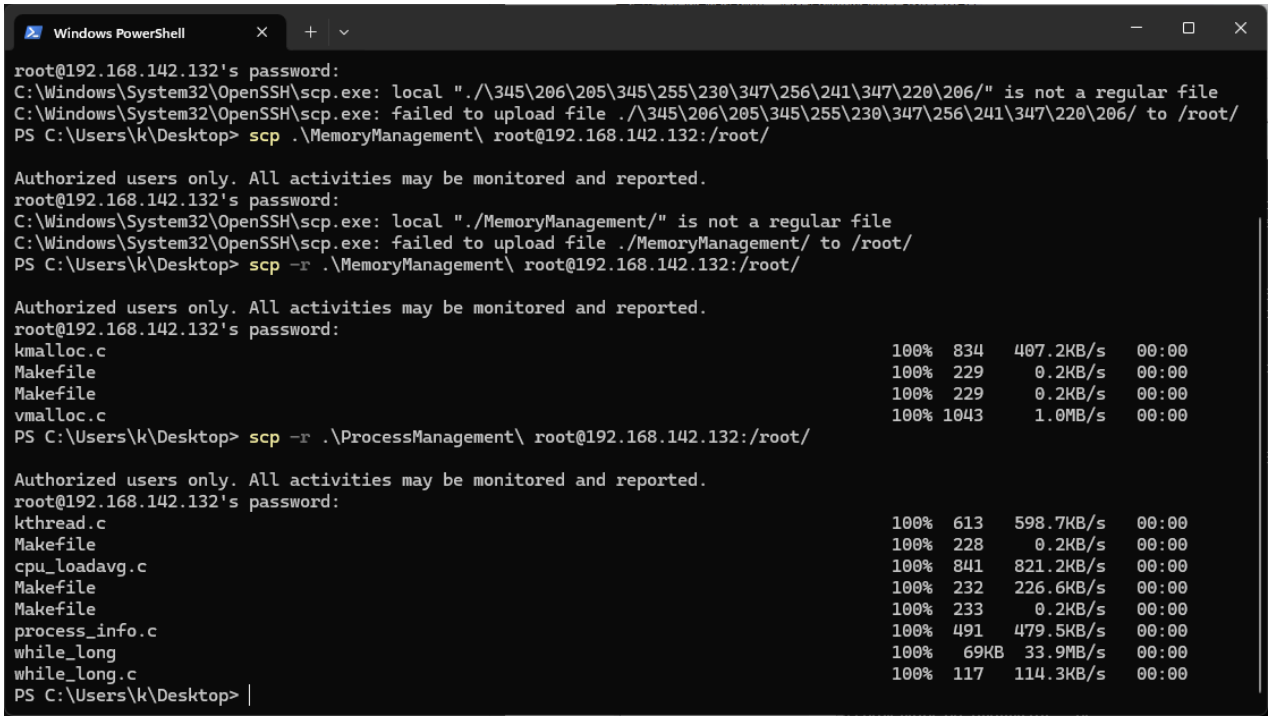

- 重新编译后，安装内核，执行 uname -a 指令
  - 指令
    - `make modules_install -j8`
  - 

- 指令
  - `make install -j8`
- 

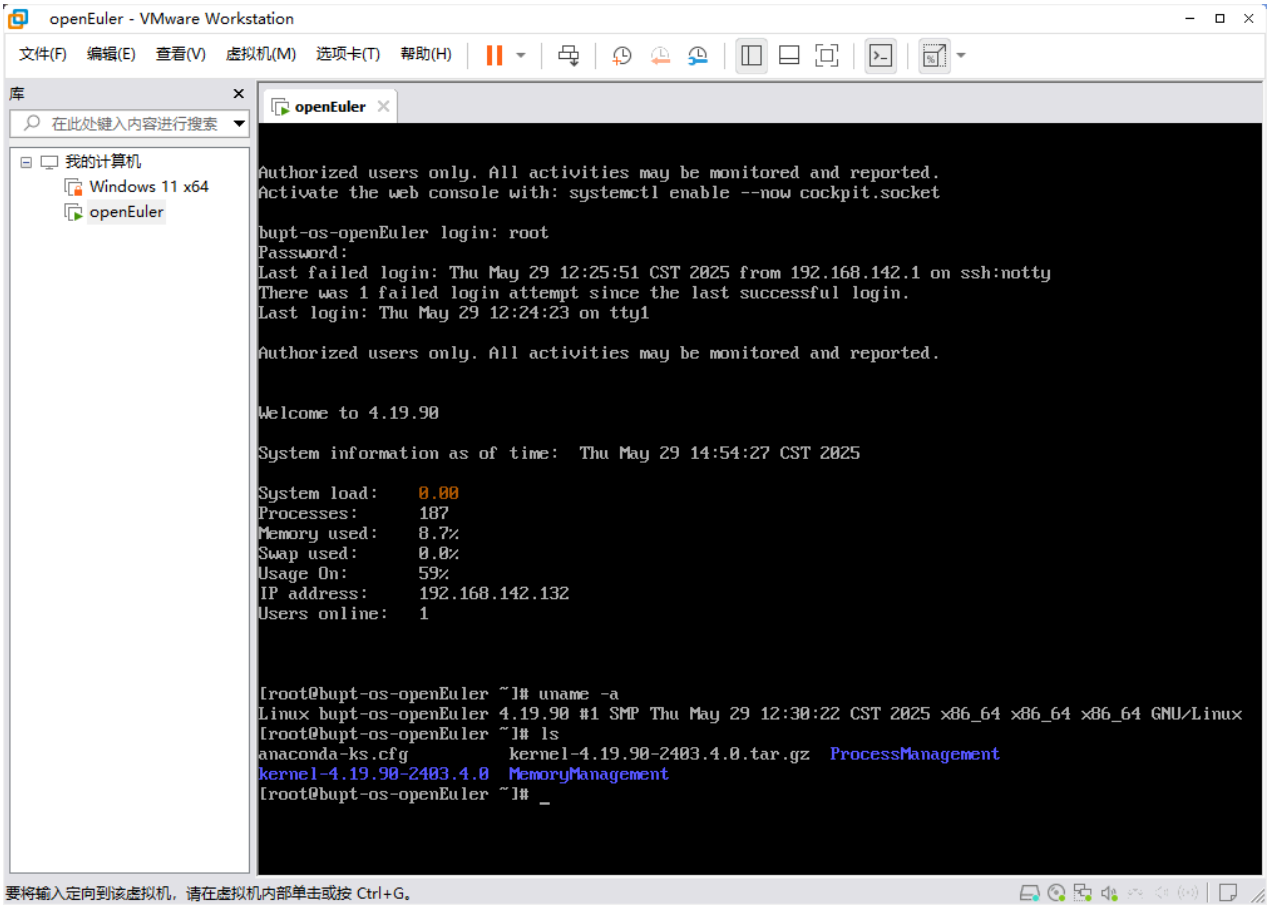
- 重启虚拟机
- 引导页面出现新的内核
- 

- 执行 `uname -a`

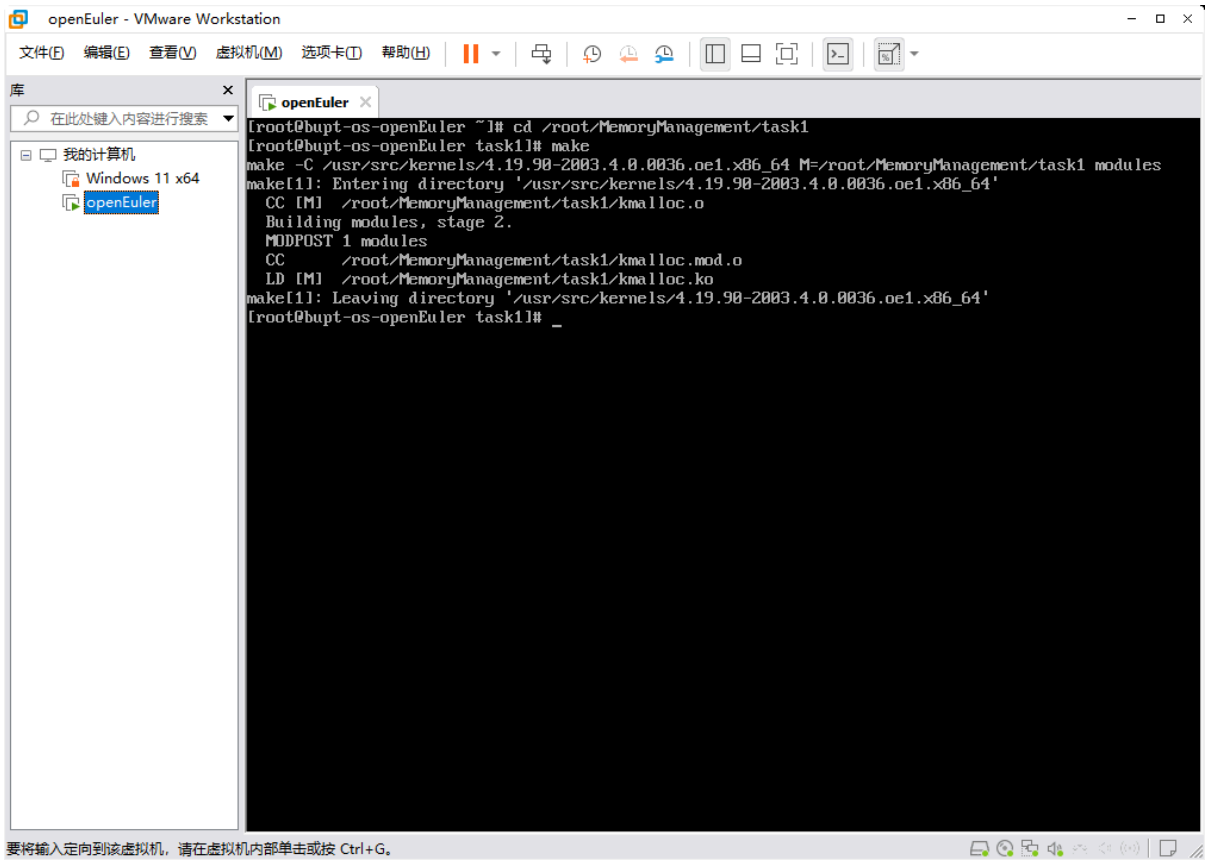
- 

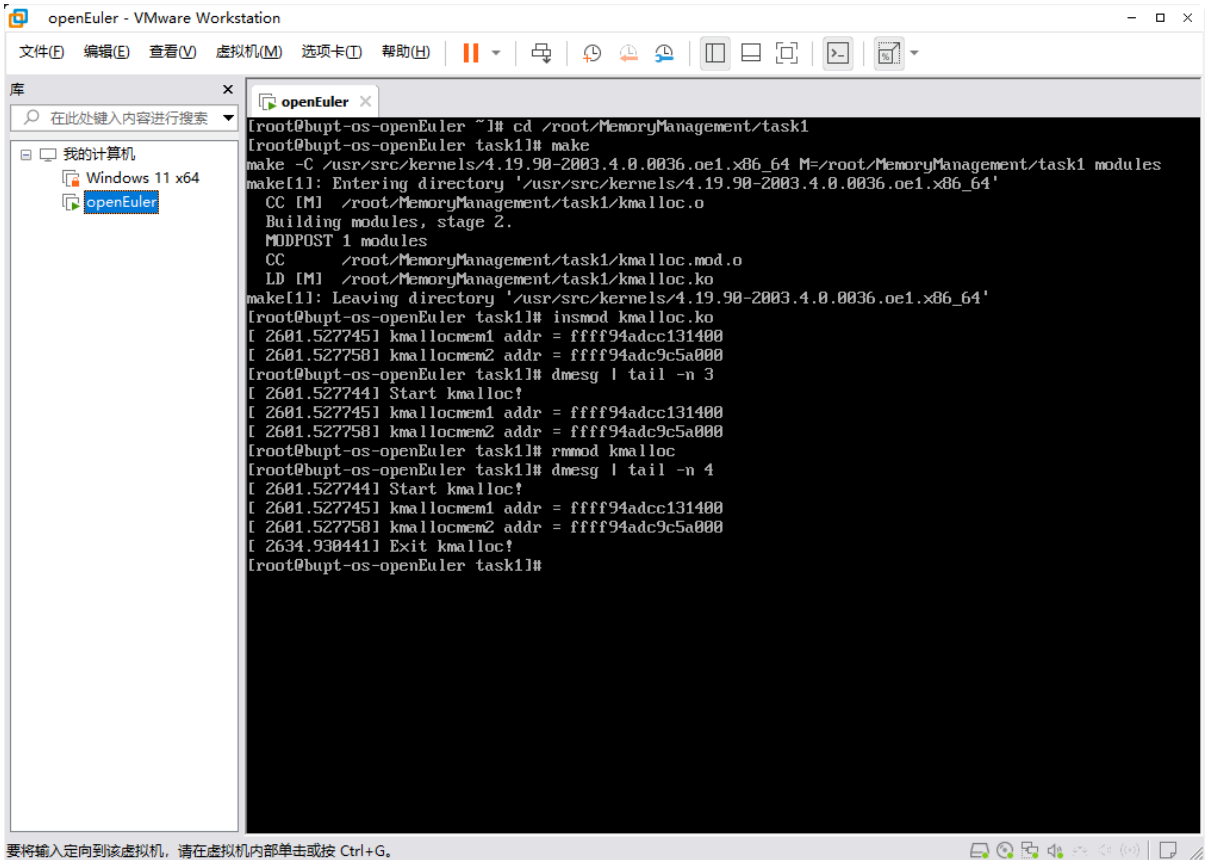

- 基础操作系统实验
  - 将代码传入虚拟机中
    - 指令
      - scp -r .\MemoryManagement\ root@192.168.142.132:/root/
      - scp -r .\ProcessManagement\ root@192.168.142.132:/root/
    - 

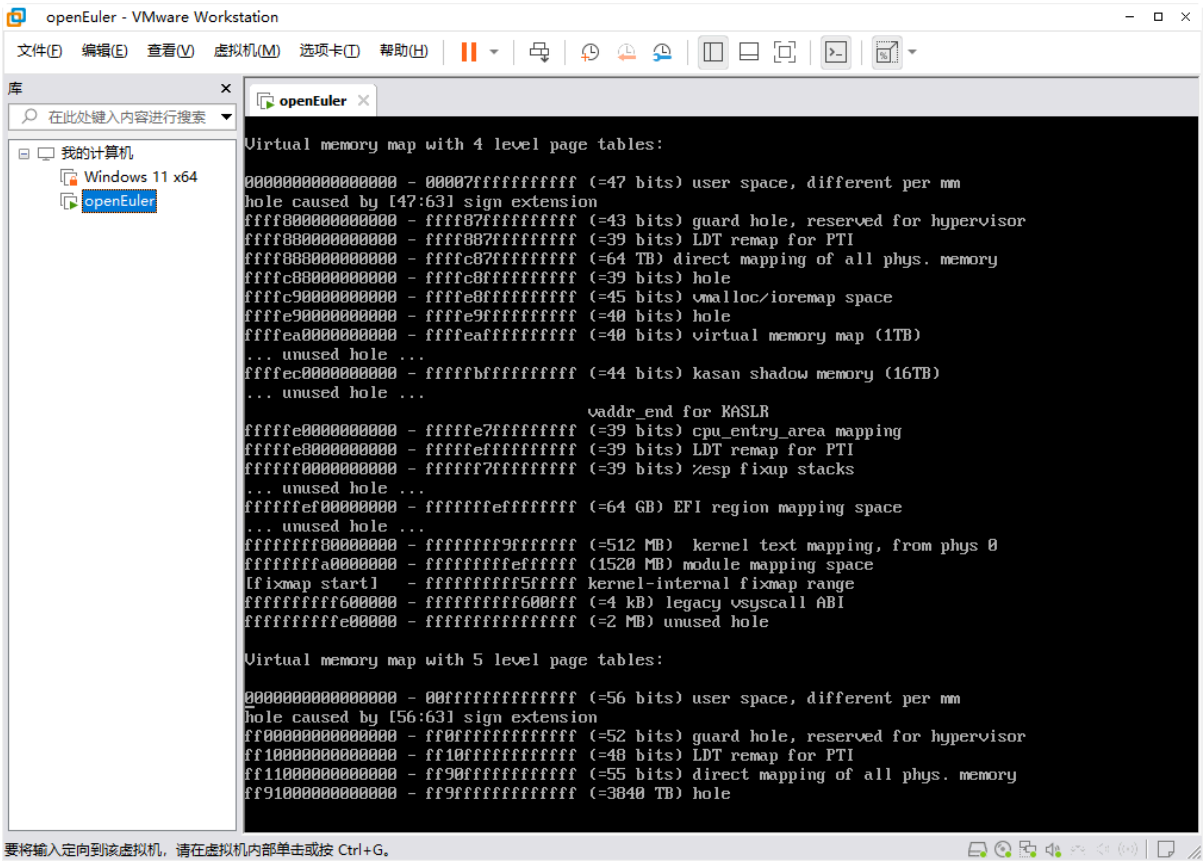
    - 指令

        - `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`

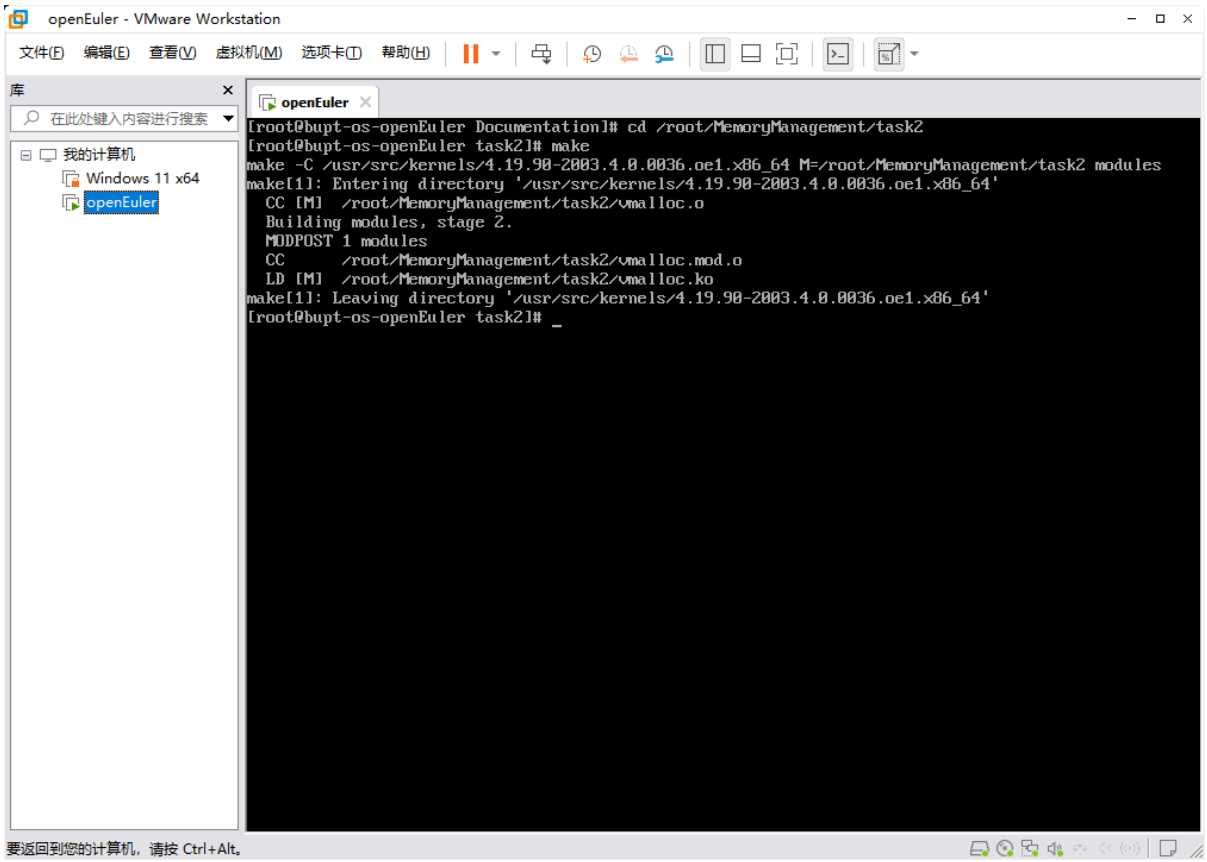
- 

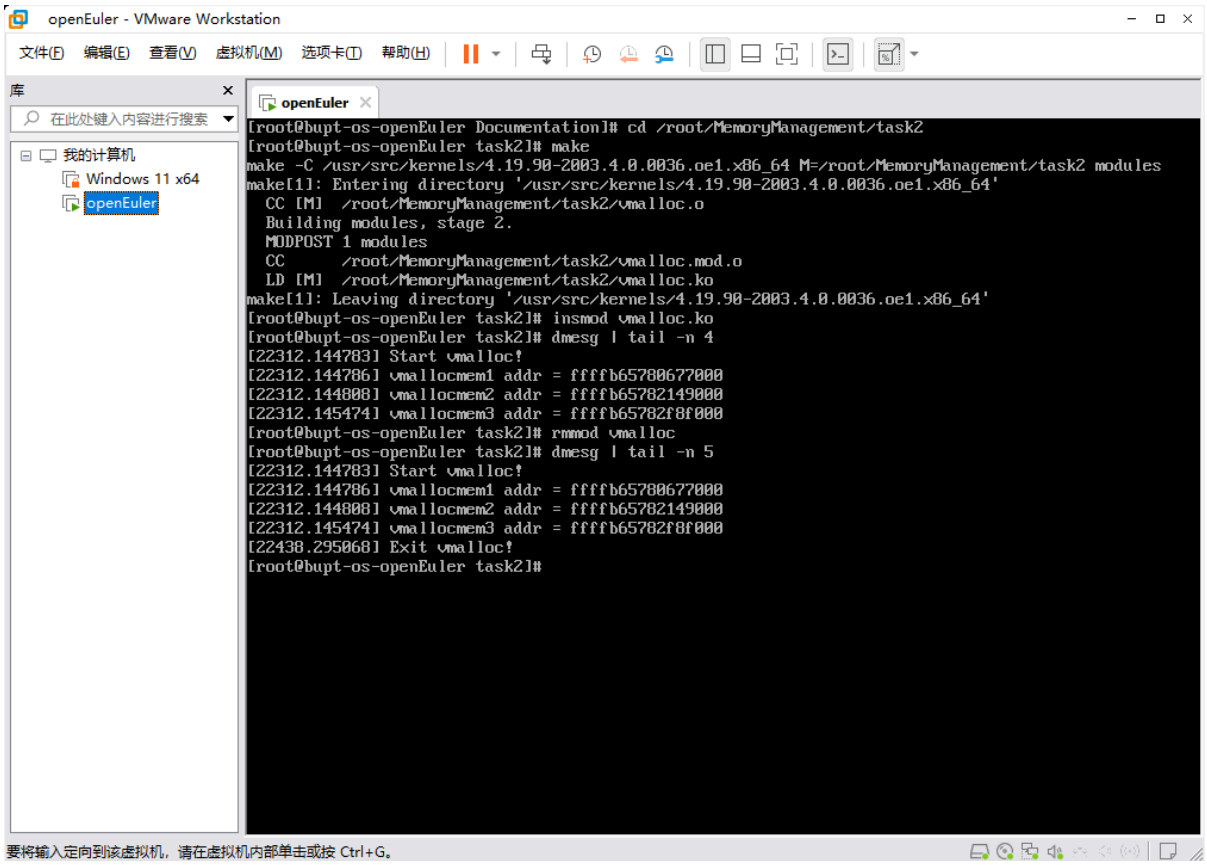
- 指令
  - insmod kmalloc.ko
  - dmesg | tail -n 3
  - rmmod kmalloc
  - dmesg | tail -n 4
- 

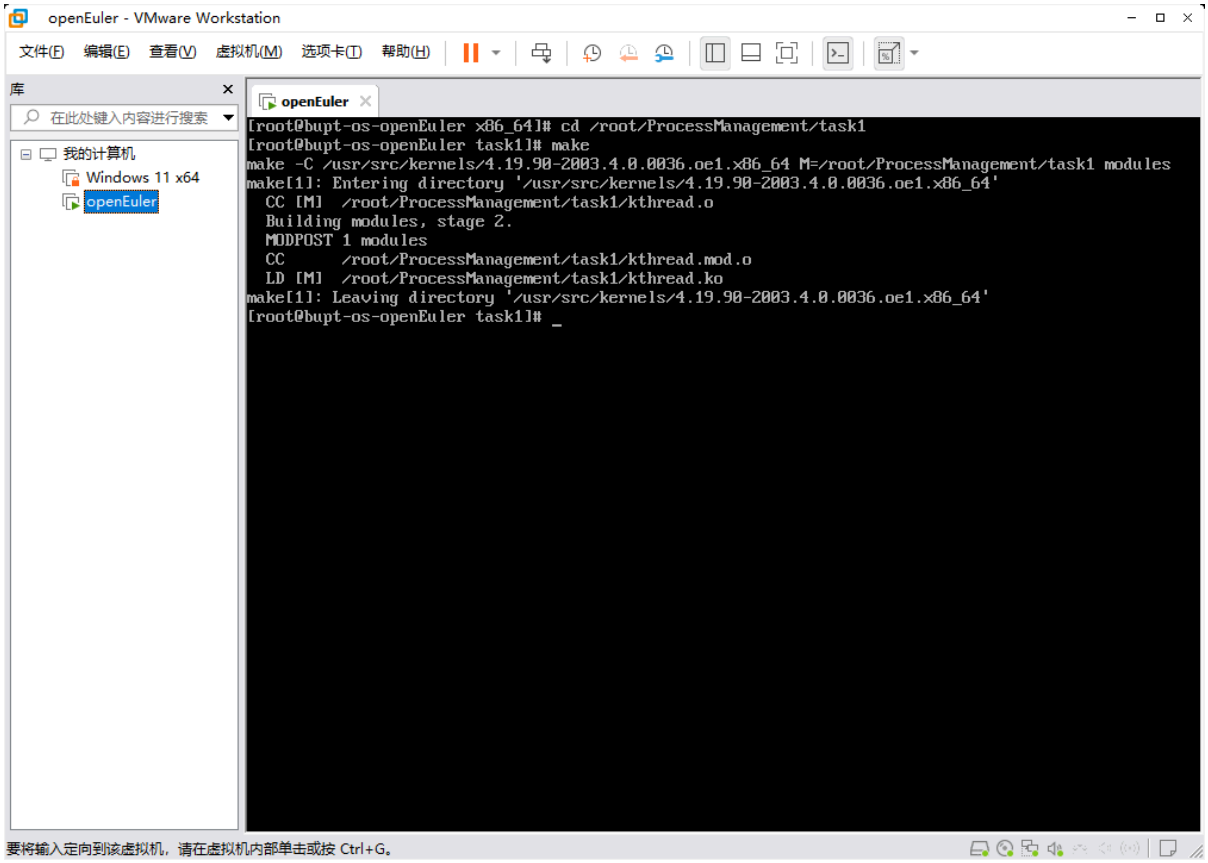
- 指令
  - `vi /root/kernel-4.19.90-2403.4.0/Documentation/x86/x86_64/mm.txt`
- 

- 可知 kmalloc 分配的内存地址位于内核空间

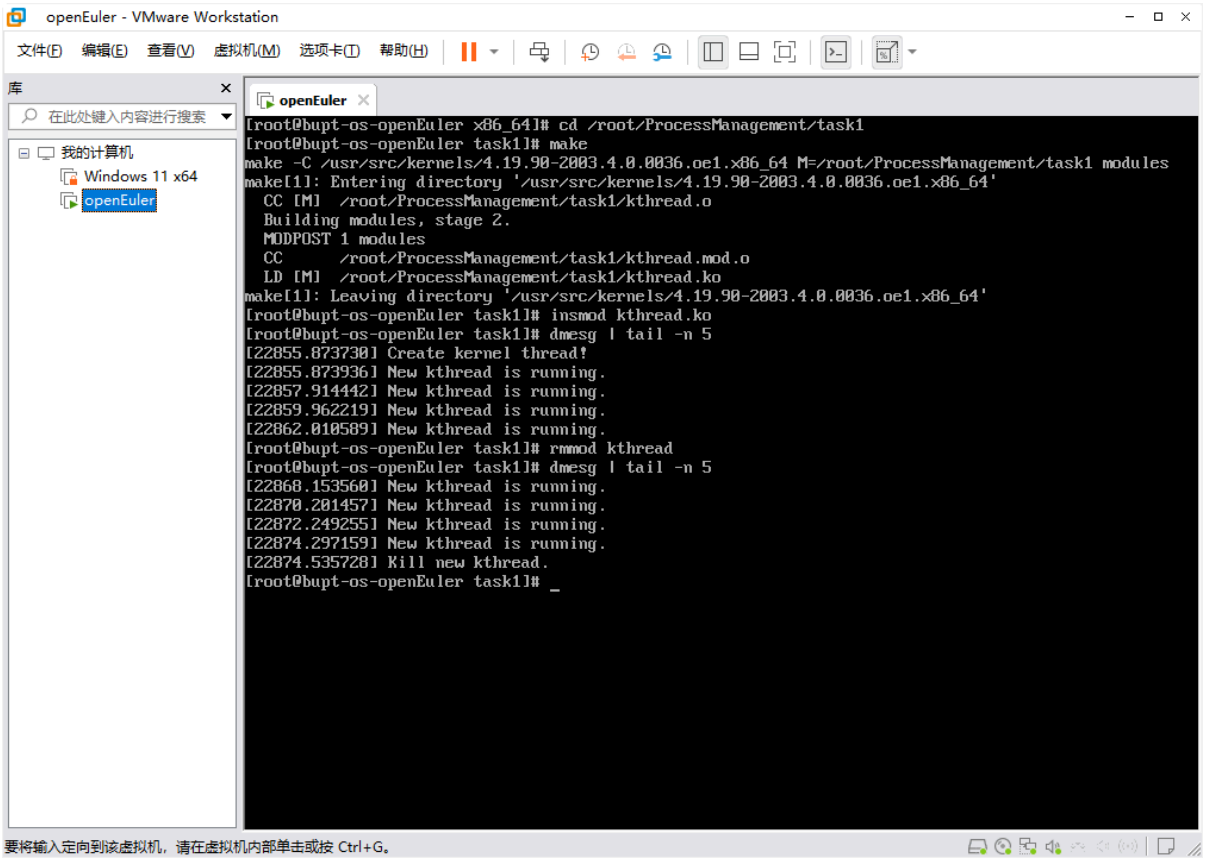
- 使用 vmalloc 分别分配8KB、1MB、64MB的内存，打印指针地址
  - 指令
    - `cd /root/MemoryManagement/task2`
    - `make`

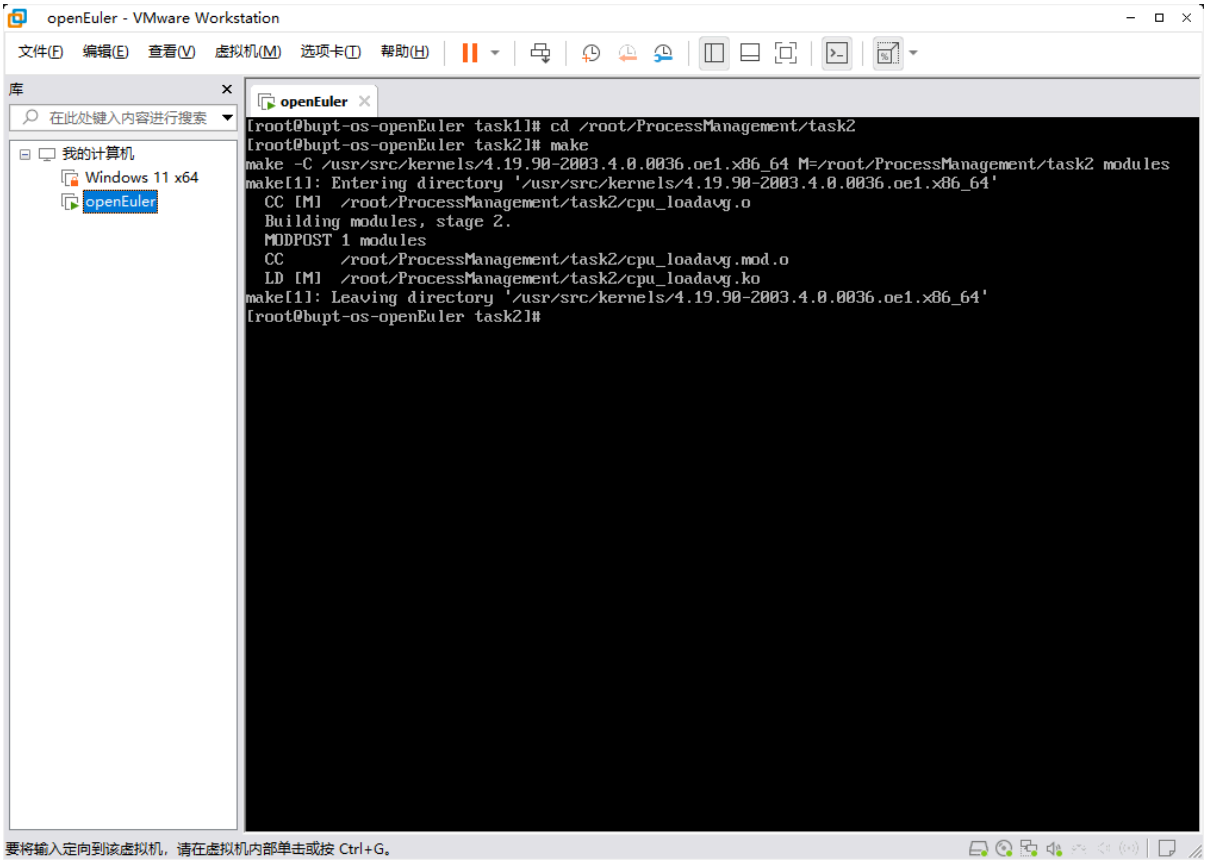
- 

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

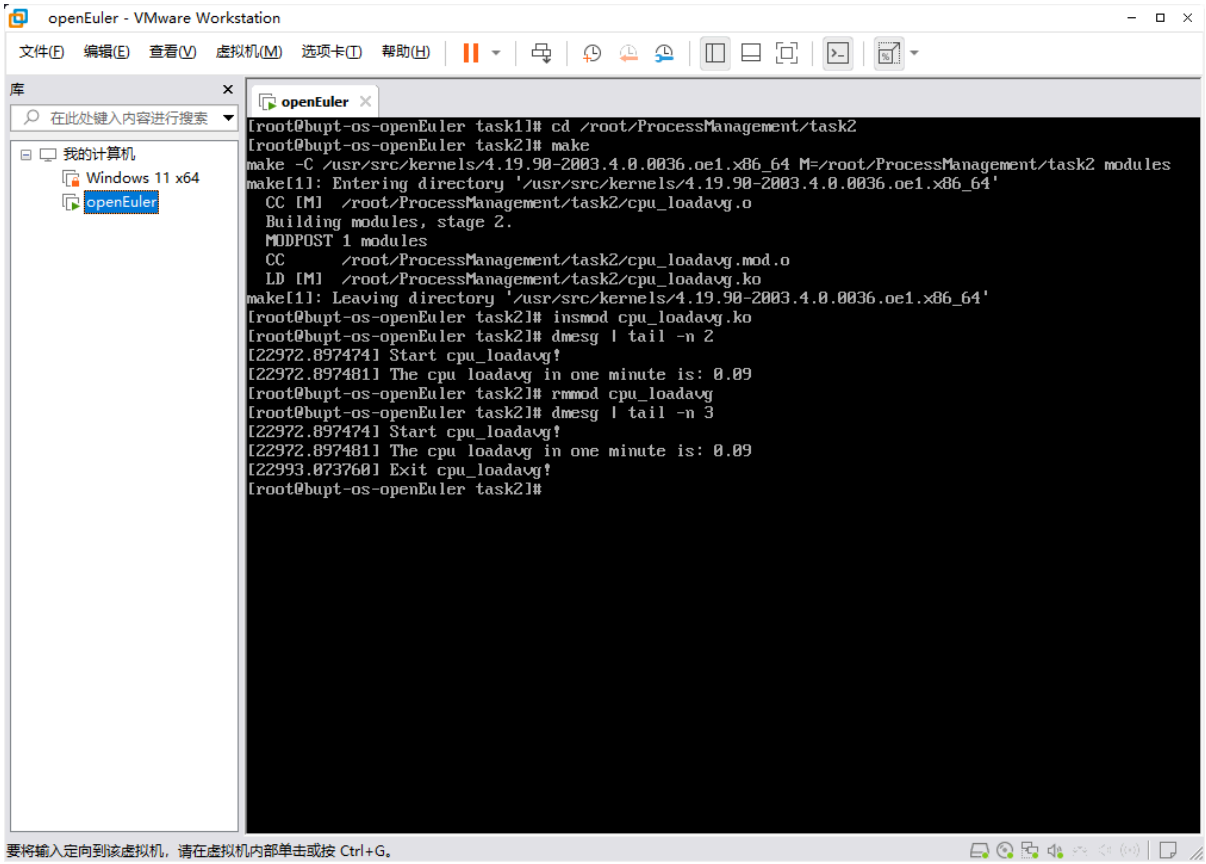
        - 可知 vmalloc 分配的内存地址位于内核空间
- 进程管理实验
    - 创建并运行内核线程
        - 指令
            - `cd /root/ProcessManagement/task1`
            - `make`
        - 

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

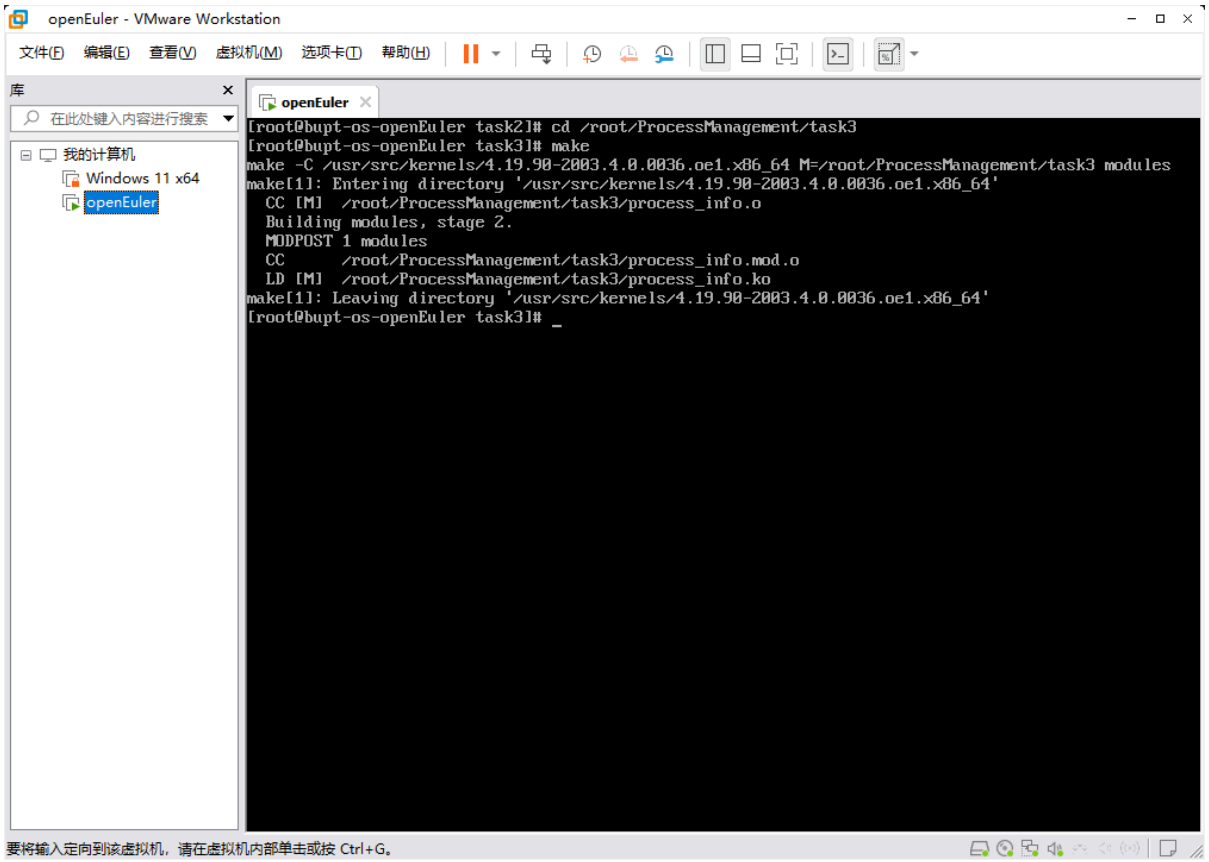
- 

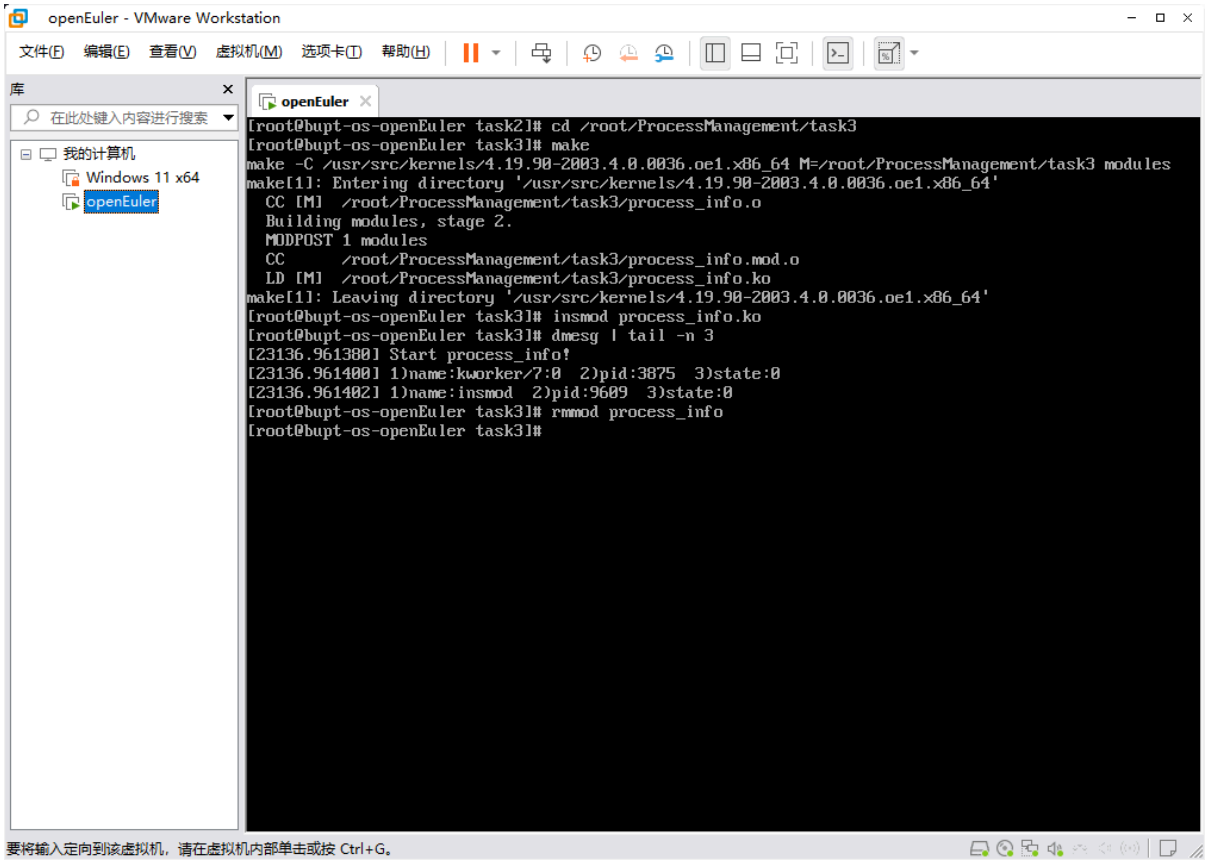

- 打印输出当前系统 CPU 负载情况
  - 指令
    - `cd /root/ProcessManagement/task2`
    - `make`
  - 

  - 指令

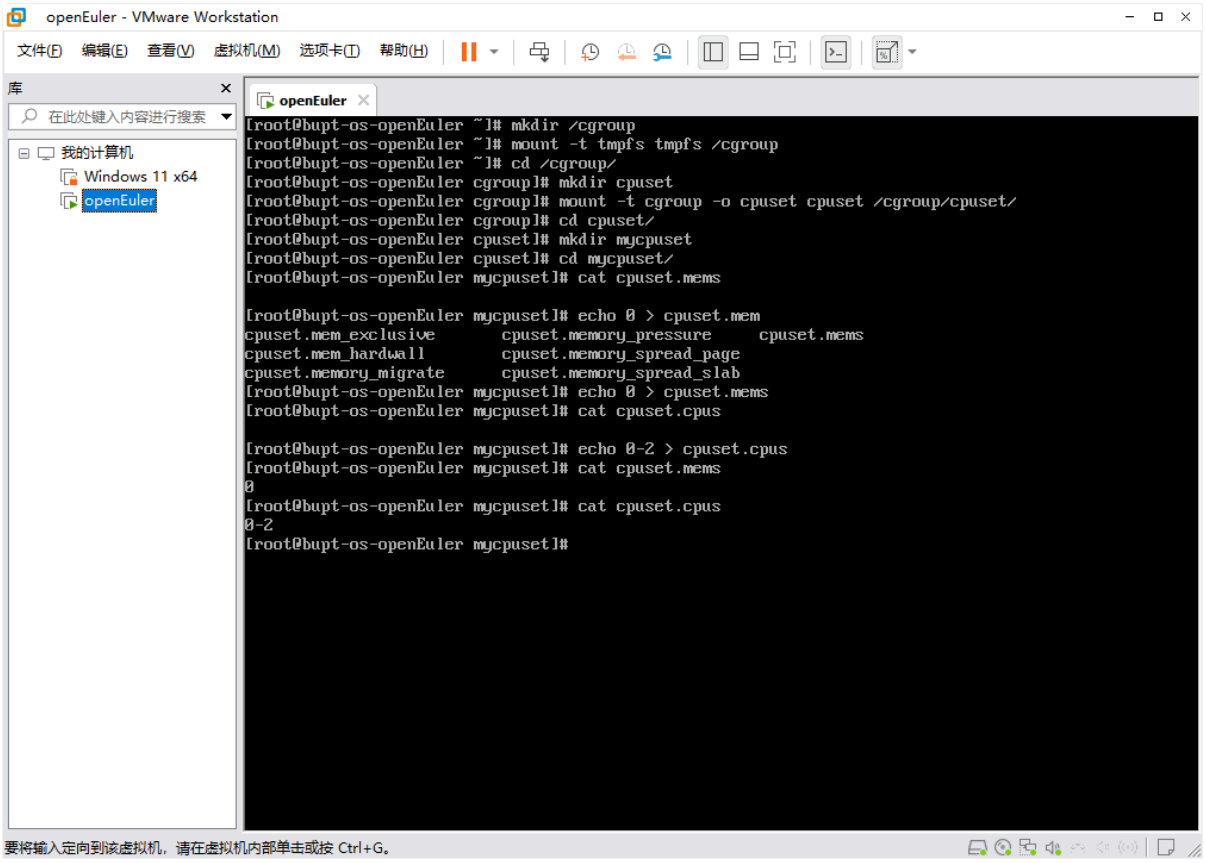
        - `insmod cpu_loadavg.ko`
        - `dmesg | tail -n 2`
        - `rmmod cpu_loadavg`
        - `dmesg | tail -n 3`
    - 

- 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`

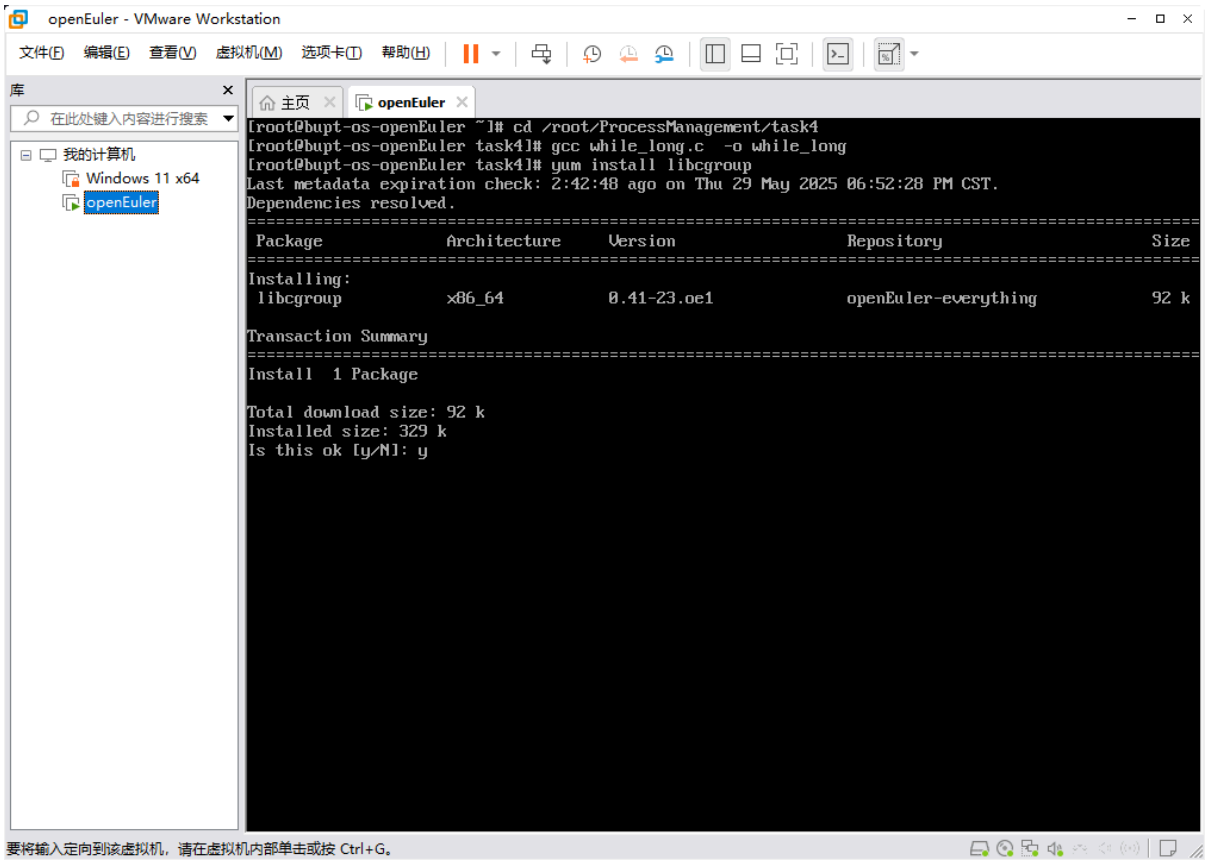
- 

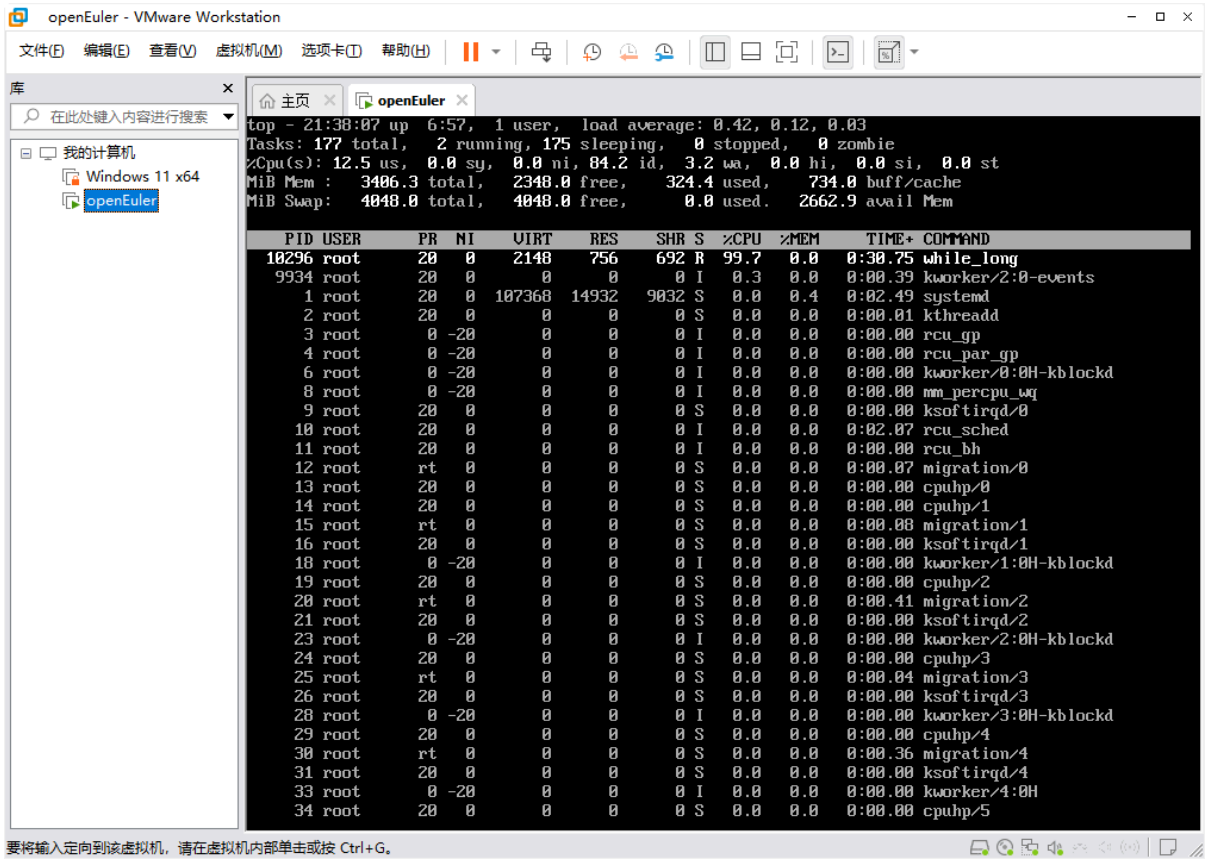
- 指令
  - `insmod process_info.ko`
  - `dmesg | tail -n 3`
  - `rmmod process_info`
- 

- 使用 cgroup 实现限制 CPU 核数

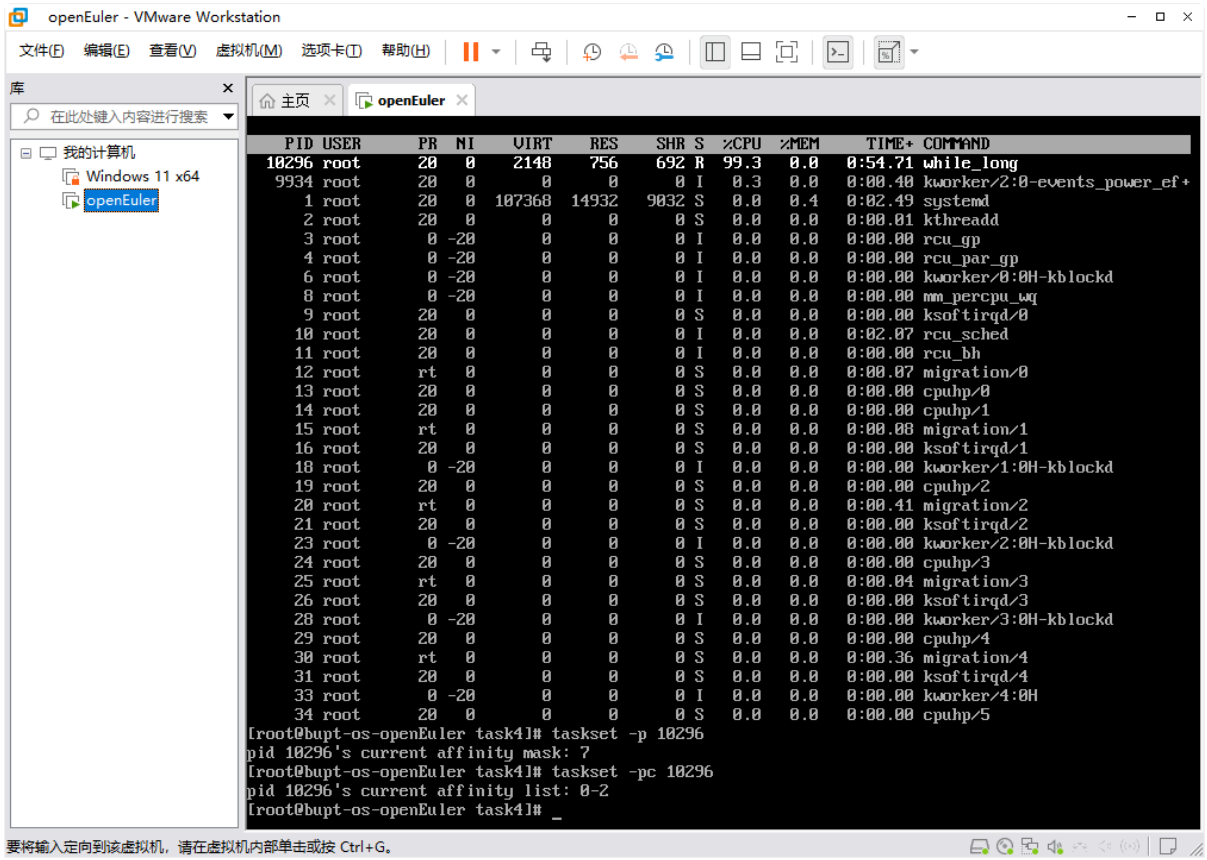
- 指令
  - mkdir /cgroup
  - mount -t tmpfs tmpfs /cgroup
  - cd /cgroup
  - mkdir cpuset
  - mount -t cgroup -o cpuset cpuset /cgroup/cpuset
  - cd cpuset
  - mkdir mycpuset
  - cd mycpuset
  - cat cpuset.mems
  - echo 0 > cpuset.mems
  - cat cpuset.cpus
  - echo 0-2 > cpuset.cpus
  - cat cpuset.mems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

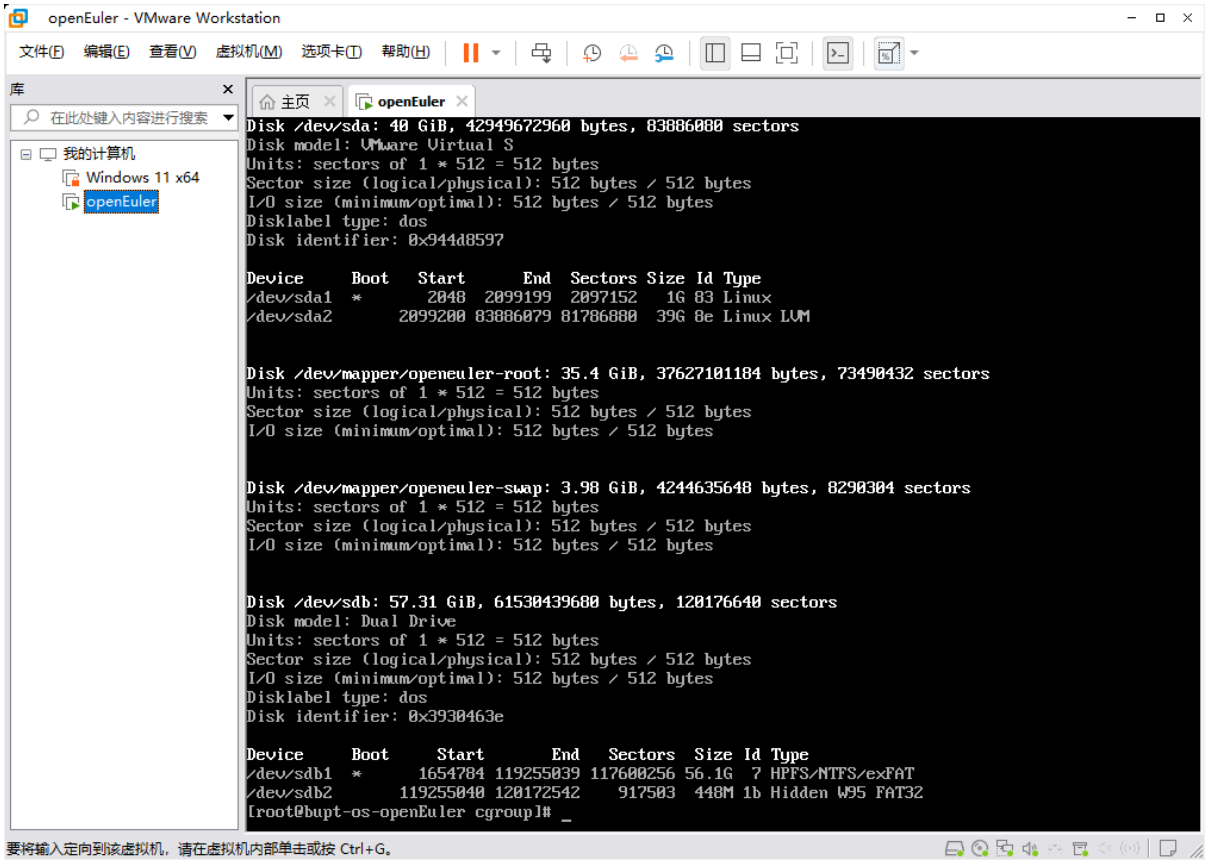
- 指令
  - cd /root/ProcessManagement/task4
  - gcc while_long.c -o while_long
  - yum install libcgroup

- 

- 指令
  - top
- 

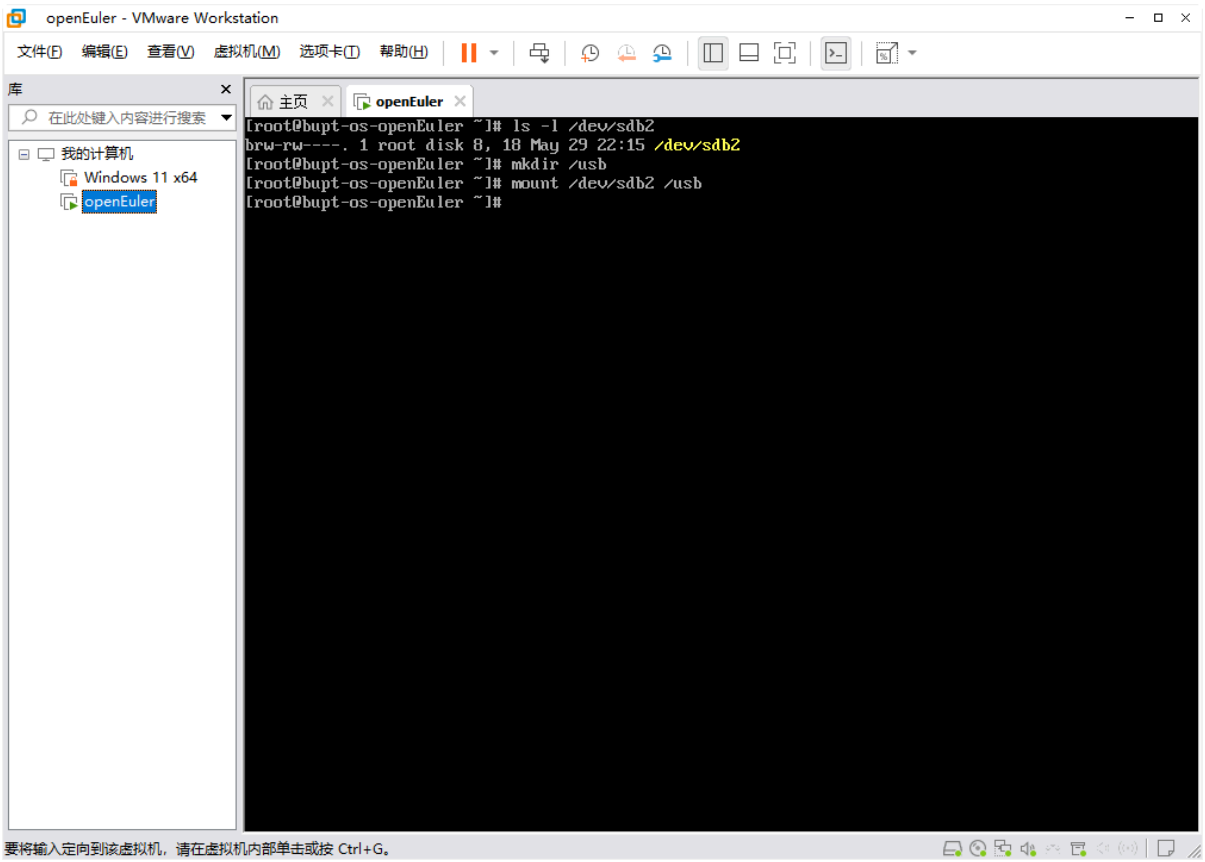
- 指令
  - taskset -p 10296
  - taskset -pc 10296

- 

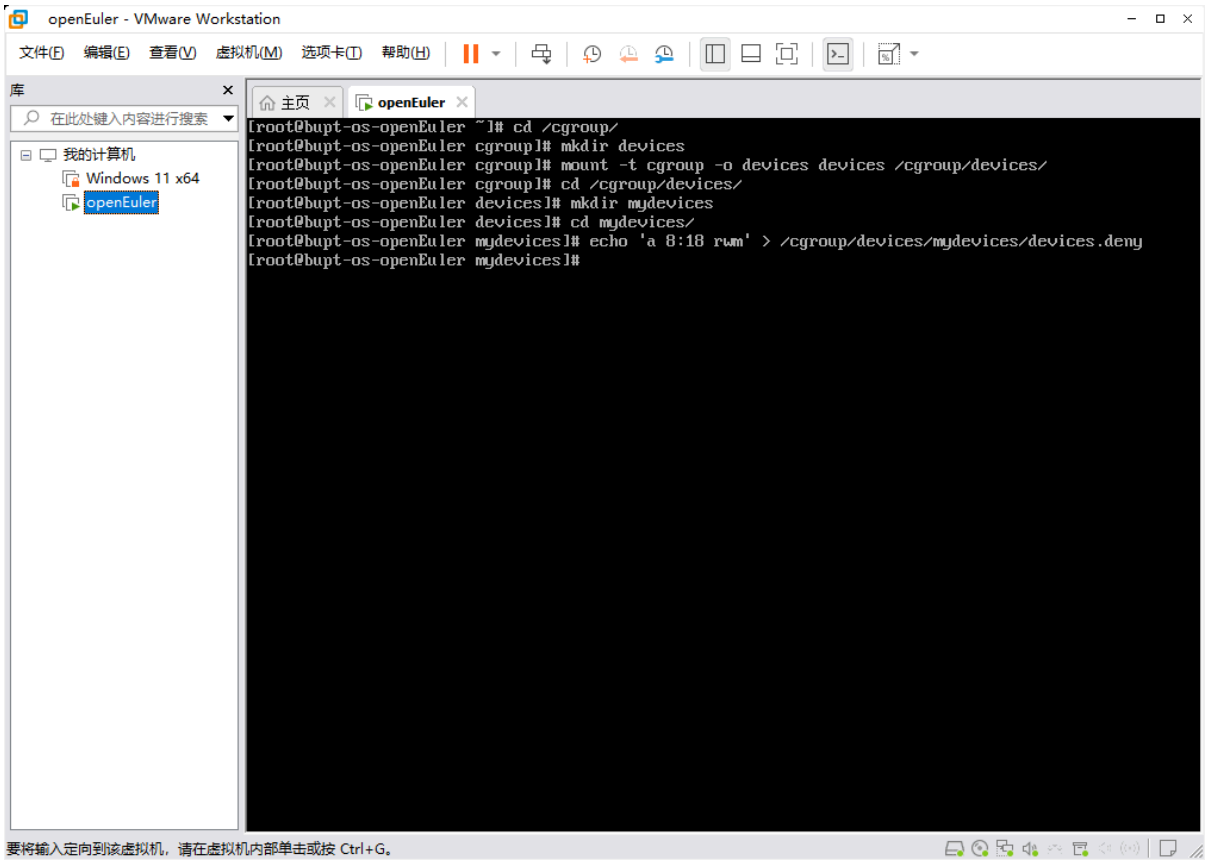
  - 测试限制 cpu 核数成功
- 使用 cgroup 实现不允许访问U盘
  - 指令
    - `fdisk -l`
  - 

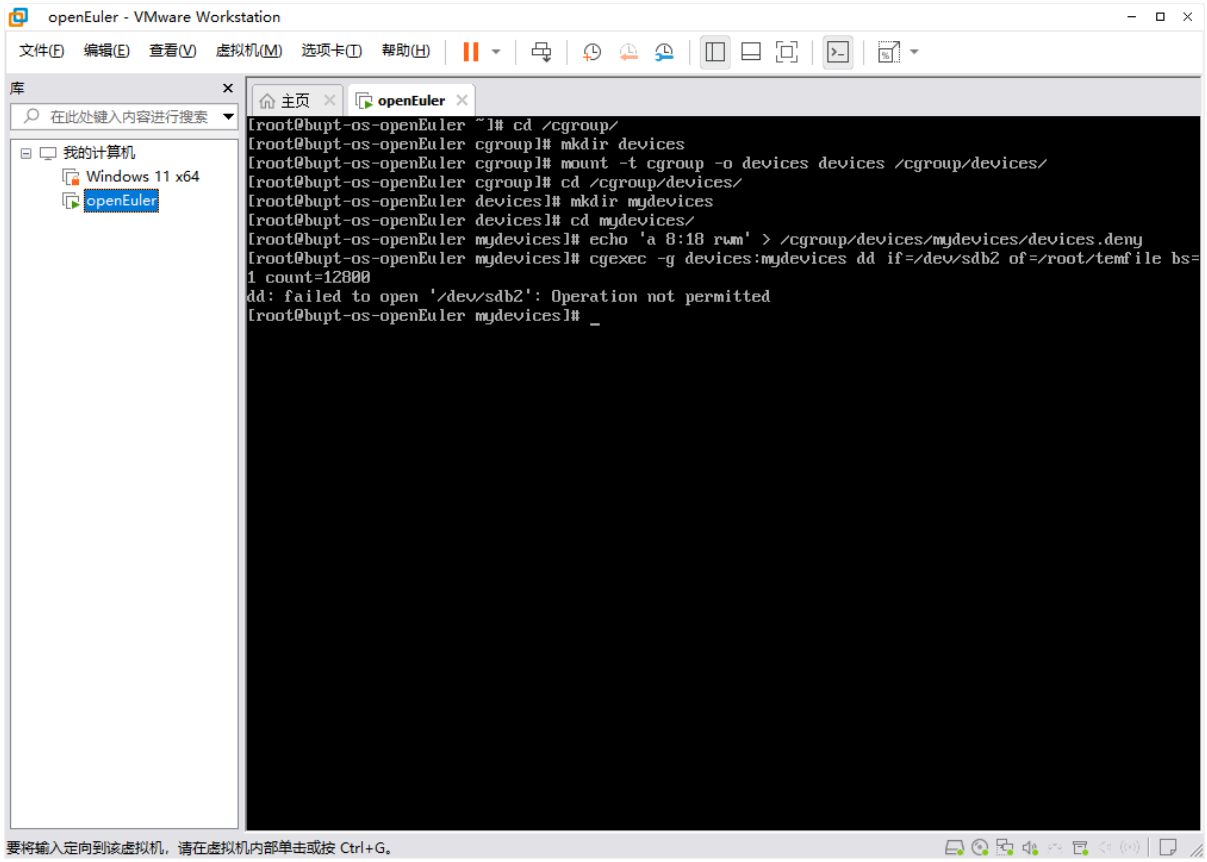
  - 指令

- 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

- 

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

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