

第一，编译时间很长，试错成本很高。

第二，安装的openEuler版本不一样，使用的linux内核不一样，对应的能编译的软件包不一样，因为这涉及到软件依赖的问题。

结论：必须使用名为 `openEuler-22.03-LTS-x86_64-dvd.iso` 的镜像文件，下载地址：<https://www.openeuler.org/zh/download/archive/detail/?version=openEuler%2022.03%20LTS>。你要编译群里面给的内核文件，使用的系统版本应该使用22.03，因为22.03使用的就是linux5.10的内核，和编译的5.10的内核文件是最匹配的，其他的都有风险

编译内核教程

准备：使用远程登录——解决无法复制粘贴问题

1 安装 finalshell 或者 xshell

参考资料：

2 查看虚拟机 ip地址

使用 `ifconfig` 命令，如下：

```
[root@localhost ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.88.146 netmask 255.255.255.0 broadcast 192.168.88.255
    inet6 fe80::5a51:ec1e:ef24:2bbf prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:73:9f:51 txqueuelen 1000 (Ethernet)
    RX packets 60 bytes 6217 (6.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 88 bytes 7515 (7.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
    ether 52:54:00:ee:b0:7c txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]#
```

使用 ip 地址如图中所示：

```
[root@localhost ~]# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.88.146 netmask 255.255.255.0 broadcast 192.168.88.255
    inet6 fe80::5a51:ec1e:ef24:2bbf prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:73:9f:51 txqueuelen 1000 (Ethernet)
    RX packets 60 bytes 6217 (6.0 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 88 bytes 7515 (7.3 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

virbr0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    inet 192.168.122.1 netmask 255.255.255.0 broadcast 192.168.122.255
    ether 52:54:00:ee:b0:7c txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

[root@localhost ~]#
```

3 使用 finalshell 或者 xshell 连接

首先安装软件依赖：`yum install openssh-server`

然后按如图所示步骤进行连接

FinalShell 4.3.10

同步状态

IP 192.168.88.146

复制

系统信息

运行 1:14

负载 0.22, 1.12, 1.16

CPU 11%

内存 29% 686M/2.3G

交换 100% 49M/49M

内存 CPU 命令

↑12K ↓2K ens33

98

68

38

0ms 本机

0

0

0

路径 可用/大小

/dev 1.2G/1.2G

/dev/shm 1.2G/1.2G

/run 379M/480M

/sys/fs/cgr... 4M/4M

/ 12.8G/35.2G

激活/升级

1 euler

连接管理器

显示已删除

SSH连接(Linux)

远程桌面连接(Windows)

新建文件夹

docker_learning CentOS 22 root

euler 192.168.88.146 22 root

meteor 192.168.88.135 22 root

openeuler 192.168.88.146 22 root

激活/升级

连接后关闭窗口

INSTALL sound/usb/caiaq/snd-usb-caiaq.ko

INSTALL sound/usb/hiface/snd-usb-hiface.ko

INSTALL sound/usb/line6/snd-usb-line6.ko

INSTALL sound/usb/line6/snd-usb-pod.ko

INSTALL sound/usb/line6/snd-usb-podhd.ko

INSTALL sound/usb/line6/snd-usb-toneport.ko

INSTALL sound/usb/line6/snd-usb-variax.ko

INSTALL sound/usb/misc/snd-ua101.ko

INSTALL sound/usb/snd-usb-audio.ko

INSTALL sound/usb/snd-usbmidi-lib.ko

INSTALL sound/usb/usx2y/snd-usb-us122l.ko

INSTALL sound/usb/usx2y/snd-usb-usx2y.ko

INSTALL sound/x86/snd-hdmi-lpe-audio.ko

INSTALL sound/xen/snd_xen_front.ko

INSTALL virt/lib/irqbypass.ko

DEPMOD 5.10.0

[root@localhost kernel-5.10.0-13.0.0]# make install

sh ./arch/x86/boot/install.sh 5.10.0 arch/x86/boot/bzImage \

System.map "/boot"

dracut-install: Failed to find module 'kvdo' /lib/modules/5.10.0//kernel/drivers/block/kvdo.ko

dracut-install: Failed to find module 'uds' /lib/modules/5.10.0//kernel/drivers/block/uds.ko

[root@localhost kernel-5.10.0-13.0.0]#

连接断开

命令输入 (按ALT键提示历史,TAB键路径,ESC键返回,双击CTRL切换)

历史 选项

文件 命令

/root

历史

文件名列

大小

类型

修改时间

权限

用户

afs

bin

boot

dev

etc

home

lib

lih64

.config 文件夹 2025/04/13 12:03 drwx----- 0/0

.bash_logout 18 B Bash Log... 2019/12/24 08:00 -rw-r--r-- 0/0

.bash_profile 176 B Bash Prof... 2019/12/24 08:00 -rw-r--r-- 0/0

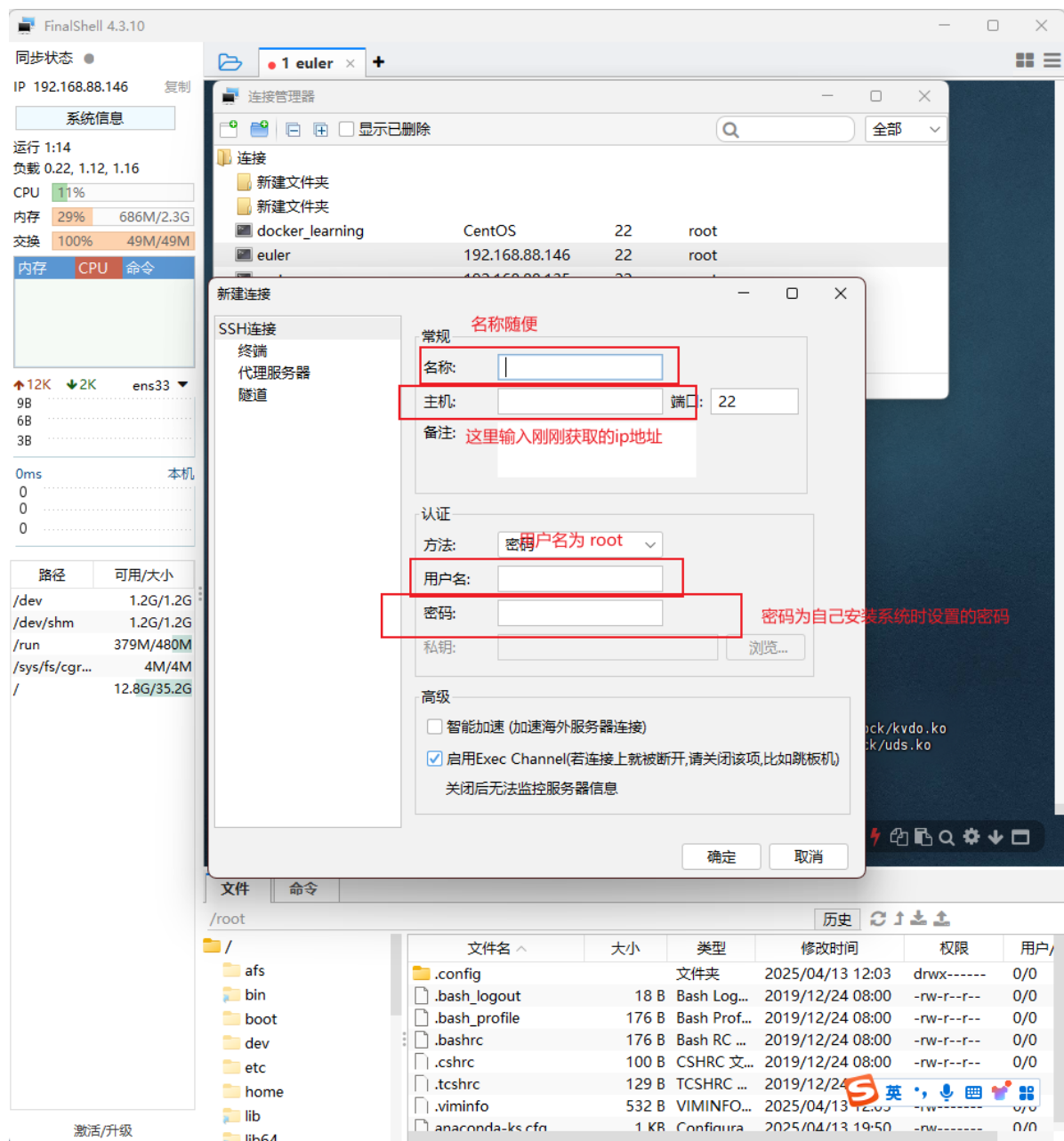
.bashrc 176 B Bash RC ... 2019/12/24 08:00 -rw-r--r-- 0/0

.cshrc 100 B CSHRC 文... 2019/12/24 08:00 -rw-r--r-- 0/0

.tcshrc 129 B TCSHRC ... 2019/12/24 08:00 -rw-r--r-- 0/0

.viminfo 532 B VIMINFO... 2025/04/13 12:00 -rw-r--r-- 0/0

anaconda-ks.cfg 1 KB Configura 2025/04/13 19:50 -rw----- 0/0



然后选择接受并保存

安装系统并编译内核

1 镜像文件

使用：openEuler-22.03-LTS-x86_64-dvd.iso

下载地址：<https://www.openeuler.org/zh/download/archive/detail/?version=openEuler%2022.03%20LTS>

2 安装系统

建议提高内存和处理器核的设置，硬盘大小至少给30，保险给 40

3 编译之前的准备

按照老师给的一步都不能少。

过程如下：

安装软件依赖：

```
yum install elfutils-libelf-devel
yum install openssl-devel
yum install bc
yum install wget

yum install lrzsz      # rz和sz可以在终端下很方便的传输文件
```

下载内核包和软件依赖

```
#下载源码
wget https://gitee.com/openeuler/kernel/repository/archive/5.10.0-13.0.0.zip
#解压
unzip 5.10.0-13.0.0.zip
#进入源码根目录
cd kernel-5.10.0-13.0.0
三. 编译内核
#进入解压好的源码文件夹执行命令，清理过去内核编译产生的文件
cd kernel-5.10.0-13.0.0
make mrproper
#生成内核配置文件.config
cp -v /boot/config-$(uname -r) .config
#执行依赖安装
yum install ncurses-devel
#然后使用make menuconfig，直接sava，保存到.config，然后就退出，目的是使用默认的配置
make menuconfig
```

这些步骤的作用如下：

好的，我们以老师授课的方式逐句讲解这些命令的作用，并补充必要的前置知识：

第一步：进入内核源码目录

```
cd kernel-5.10.0-13.0.0
```

作用：就像进入一个工作间前要推开它的门一样，这个命令是让我们进入解压后的内核源码文件夹。

注意：如果文件夹名称和实际解压的不一致（如版本号不同），这个命令会报错，需要核对实际目录名。

第二步：清理编译环境

```
make mrproper
```

作用：相当于给你的工作间来一次"大扫除"。

- 内核源码编译过程中会产生很多临时文件和配置文件（如.config、.o文件等），如果之前编译过旧版本内核，这些残留文件可能导致新编译出错。
- make mrproper 会彻底清理所有编译生成的文件、配置文件及备份文件，确保编译环境干净。

对比： `make clean` 只是清理编译生成的中间文件（如.o文件），但保留配置文件，适用于小范围修改后重新编译。

第三步：生成基础配置文件

```
cp -v /boot/config-$(uname -r) .config
```

作用： 复制当前系统正在使用的内核配置文件作为新内核编译的起点。

- `/boot/config-$(uname -r)`： `uname -r` 会获取当前运行的内核版本（如5.10.0-13.0.0），因此实际复制的是类似 `/boot/config-5.10.0-13.0.0` 的文件。
- `-v` 参数表示显示复制过程，方便确认操作成功。

意义： 内核有数千个配置选项，手动配置极其耗时。使用现有配置能大幅提升效率，且保证新内核与当前系统兼容性。

第四步：安装文本界面依赖库

```
yum install ncurses-devel
```

作用： 安装 `make menuconfig` 所需的文本图形界面支持库。

- `ncurses-devel` 提供了在终端中绘制菜单、按钮等图形元素的能力，没有它后续的 `make menuconfig` 会报错。

延伸知识：

1. 不同系统包管理器命令不同（如Ubuntu用 `apt install libncurses5-dev`）。
 2. 如果编译过程中提示其他依赖缺失（如openssl、flex、bison等），需要按提示补充安装。
-

第五步：启动内核配置界面

```
make menuconfig
```

作用： 进入一个类似下图的文本图形界面，允许你修改内核配置：

Linux Kernel Configuration

```
| Arrow keys navigate the menu. <Enter> selects submenus --->
|
|
| Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,
|
|
| <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help.
|
|
|
|
┌─┴─┐
| |
| |      [*] 64-bit kernel
| |
| |
| |      [*] Enable loadable module support --->
| |
| |
| |      [*] Enable the block layer --->
| |
| |
| |      *- Networking support --->
| |
|
└─┴─┐
|
└─┴─┐
|
```

操作指南：

1. 用方向键导航，回车进入子菜单。
2. 按 **Y** 编译进内核，**M** 编译为模块（动态加载），**N** 不编译。
3. 按两次 **Esc** 返回上级，**/** 搜索配置项。

建议：

- 新手可先直接退出保存，使用默认配置确保编译通过，后续再学习定制。
- 若添加了新硬件支持（如新显卡驱动），需在此界面启用对应选项。

常见问题答疑：

- ## 1. 为什么要复制旧配置文件？不能从头配置吗？

可以，但手动配置需要处理数千个选项，极易出错。基于现有配置修改更高效可靠。

- ## 2. 执行 `make menuconfig` 时报错无法打开终端?

确保：① 已安装 `ncurses-devel`；② 在纯终端环境执行（不支持图形终端模拟器）。

- ### 3. 配置完成后如何保存?

在界面中依次选择 `< Save >` -> 回车确认文件名（默认 `.config`） -> `< Ok >`。

- #### 4. 后续编译还需要什么步骤?

通常继续执行 `make -jN`（编译内核）、`make modules_install`（安装模块）、`make install`（安装内核），但本示例未涉及。

通过这五步操作，你已完成内核编译前的准备工作：清理环境、获取基础配置、安装必要工具、定制个性化选项。这是内核编译的关键基础阶段，后续的编译过程将依赖这里的配置结果。

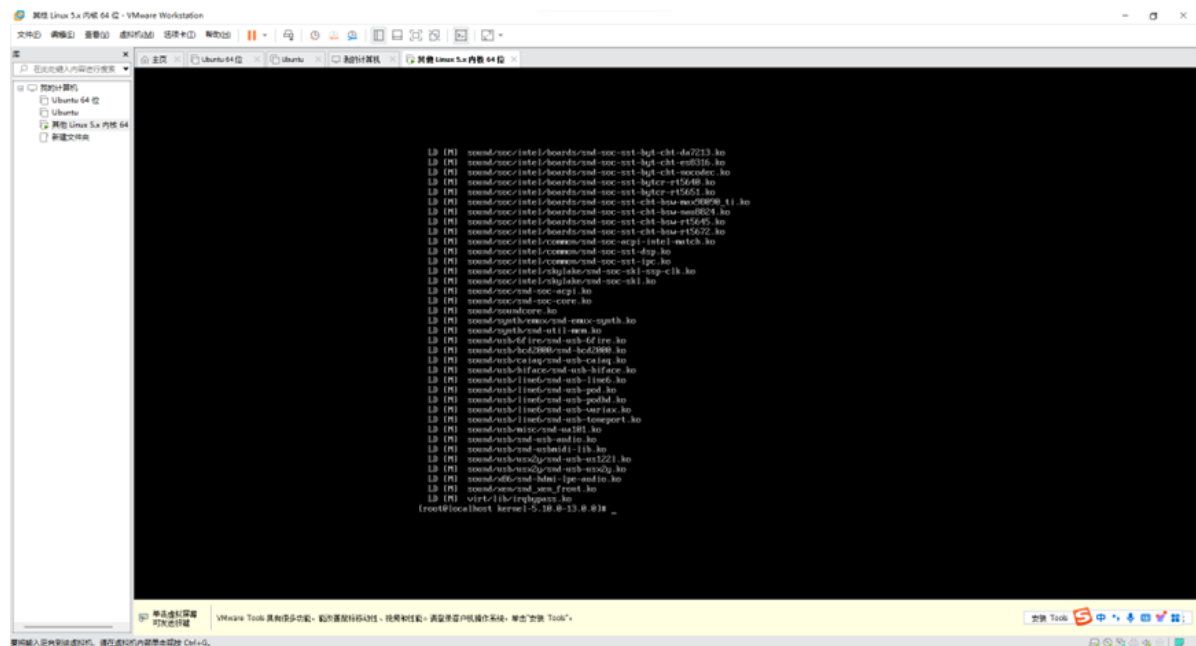
4 开始编译——这里建议在 VMware 中执行，会快一些

注意：这里要安装一下软件依赖，否则后面可能会出现问题

```
yum install dwarves
```

```
# 后面的参数是使用的cpu内核数，根据自己之前的配置选择，建议：越大越快  
make -j4
```

成功如下所：



这里可能出现如下报错

```
BTF: .tmp_vmlinux.btf: pahole (pahole) is not available  
Failed to generate BTF for vmlinux  
Try to disable CONFIG_DEBUG_INFO_BTF  
make: *** [Makefile:1161: vmlinux] Error 1
```

解决方法：

安装软件 dwarves

```
yum install dwarves
```

然后安装内核模块

```
make modules install
```

成功如下所示：

FinalShell 4.3.10

同步状态 ●

IP 192.168.88.146 复制

系统信息

运行 1:07

负载 0.67, 0.60, 0.87

CPU 2%

内存 24% 581M/2.3G

交换 3% 97M/3G

内存	CPU	命令
2.7M	1	sshd
460K	0.3	sshd
0	0.3	kworker/...
3.5M	0.3	bash

↑5K ↓542B ens33 ▼

16K

11K

5K

0ms

本机

路径	可用/大小
/dev	1.2G/1.2G
/dev/shm	1.2G/1.2G
/run	455M/480M
/sys/fs/cgr...	4M/4M
/	12.9G/35.2G
/tmp	1.2G/1.2G
/boot	598M/973M

激活/升级

```
INSTALL sound/soc/intel/boards/snd-soc-kbl_rt5663_rt5514_max98927.ko
INSTALL sound/soc/intel/boards/snd-soc-skl_nau88l25_ssm4567.ko
INSTALL sound/soc/intel/boards/snd-soc-skl_rt286.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-bxt-da7219_max98357a.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-bxt-rt298.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-byt-cht-da7213.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-byt-cht-es8316.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-byt-cht-nocodec.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-bytcr-rt5640.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-bytcr-rt5651.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-max98090_ti.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-nau8824.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-rt5645.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-rt5672.ko
INSTALL sound/soc/intel/common/snd-soc-acpi-intel-match.ko
INSTALL sound/soc/intel/common/snd-soc-sst-dsp.ko
INSTALL sound/soc/intel/common/snd-soc-sst-ipc.ko
INSTALL sound/soc/intel/skylake/snd-soc-skl-ssp-clk.ko
INSTALL sound/soc/intel/skylake/snd-soc-skl.ko
INSTALL sound/soc/snd-soc-acpi.ko
INSTALL sound/soc/snd-soc-core.ko
INSTALL sound/soundcore.ko
INSTALL sound/synth/emux/snd-emux-synth.ko
INSTALL sound/synth/snd-util-mem.ko
INSTALL sound/usb/6fire/snd-usb-6fire.ko
INSTALL sound/usb/bcd2000/snd-bcd2000.ko
INSTALL sound/usb/caiaq/snd-usb-caiaq.ko
INSTALL sound/usb/hiface/snd-usb-hiface.ko
INSTALL sound/usb/line6/snd-usb-line6.ko
INSTALL sound/usb/line6/snd-usb-pod.ko
INSTALL sound/usb/line6/snd-usb-podhd.ko
INSTALL sound/usb/line6/snd-usb-toneport.ko
INSTALL sound/usb/line6/snd-usb-variax.ko
INSTALL sound/usb/misc/snd-ua101.ko
INSTALL sound/usb/snd-usb-audio.ko
INSTALL sound/usb/snd-usbmidi-lib.ko
INSTALL sound/usb/usx2y/snd-usb-us122l.ko
INSTALL sound/usb/usx2y/snd-usb-usx2y.ko
INSTALL sound/x86/snd-hdmi-lpe-audio.ko
INSTALL sound/xen/snd_xen_front.ko
INSTALL virt/lib/irqbypass.ko
DEPMOD 5.10.0
[root@localhost kernel-5.10.0-13.0.0]#
```

命令输入 (按ALT键提示历史,TAB键路径ESC键返回,双击CTRL切换)

文件 命令

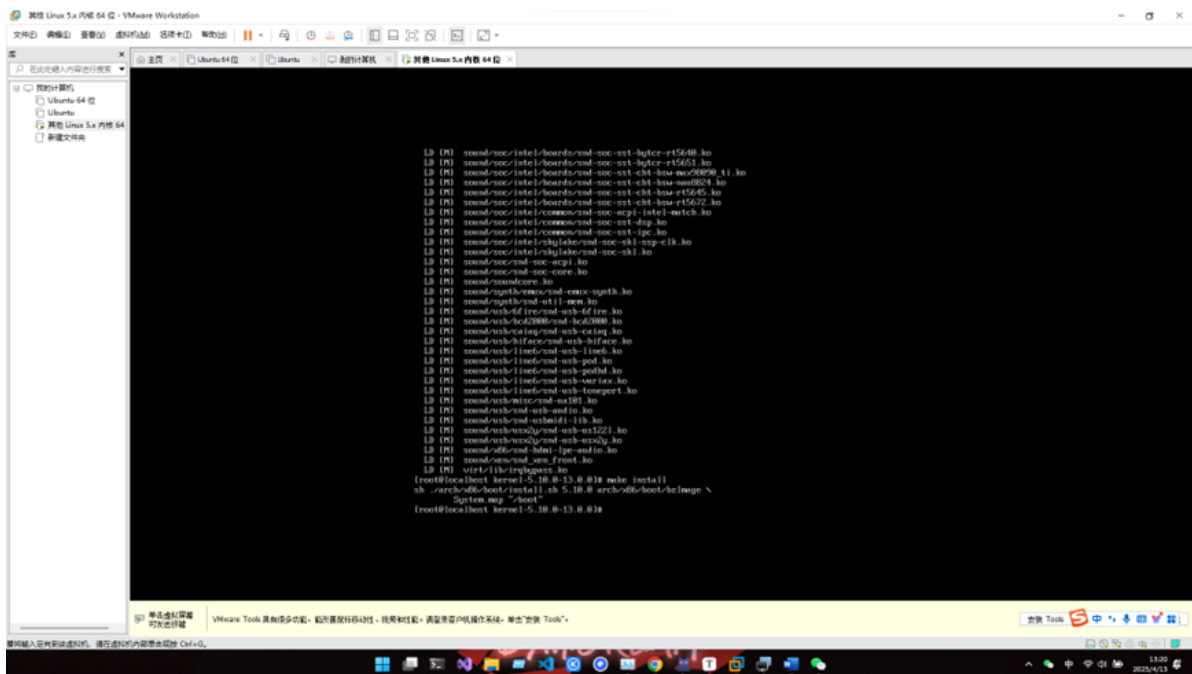
/root

文件名 ^	大小	类型	修改时间	权限	用户/
./config		文件夹	2025/04/13 12:03	drwx-----	0/0
./bash_logout	18 B	Bash Log...	2019/12/24 08:00	-rw-r--r--	0/0
./bash_profile	176 B	Bash Prof...	2019/12/24 08:00	-rw-r--r--	0/0
./bashrc	176 B	Bash RC ...	2019/12/24 08:00	-rw-r--r--	0/0
./cshrc	100 B	CSHRC 文...	2019/12/24 08:00	-rw-r--r--	0/0
./tcshrc	129 B	TCSHRC ...	2019/12/24		
./viminfo	532 B	VIMINFO...	2025/04/13 12:03	-rw-r--r--	0/0
./anaconda-ks.cfg	1 KB	Configura	2025/04/13 19:50	-rw-----	0/0

然后安装内核

```
make install
```

成功如下所示:



这里我也有一点问题，如下：

FinalShell 4.3.10

同步状态 ●

IP 192.168.88.146 复制

系统信息

运行 1:10

负载 3.16, 2.05, 1.39

CPU 2%

内存 25% 594M/2.3G

交换 3% 97M/3G

内存	CPU	命令
2.7M	1	sshd
4.2M	0.3	systemd-j...
0	0.3	kworker/...
0	0.3	kworker/...

↑8388 ↓1808 ens33

26K

18K

9K

0ms

本机

路径	可用/大小
/dev	1.2G/1.2G
/dev/shm	1.2G/1.2G
/run	449M/480M
/sys/fs/cgr...	4M/4M
/	12.8G/35.2G
/tmp	1.2G/1.2G
/boot	507M/973M

激活/升级

1 euler x +

```

INSTALL sound/soc/intel/boards/snd-soc-sst-byt-cht-da7213.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-byt-cht-es8316.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-byt-cht-nocodec.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-bytcr-rt5640.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-bytcr-rt5651.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-max98090_ti.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-nau8824.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-rt5645.ko
INSTALL sound/soc/intel/boards/snd-soc-sst-cht-bsw-rt5672.ko
INSTALL sound/soc/intel/common/snd-soc-acpi-intel-match.ko
INSTALL sound/soc/intel/common/snd-soc-sst-dsp.ko
INSTALL sound/soc/intel/common/snd-soc-sst-ipc.ko
INSTALL sound/soc/intel/skylake/snd-soc-skl-ssp-clk.ko
INSTALL sound/soc/intel/skylake/snd-soc-skl.ko
INSTALL sound/soc/snd-soc-acpi.ko
INSTALL sound/soc/snd-soc-core.ko
INSTALL sound/soundcore.ko
INSTALL sound/synth/emux/snd-emux-synth.ko
INSTALL sound/synth/snd-util-mem.ko
INSTALL sound/usb/6fire/snd-usb-6fire.ko
INSTALL sound/usb/bcd2000/snd-bcd2000.ko
INSTALL sound/usb/caiaq/snd-usb-caiaq.ko
INSTALL sound/usb/hiface/snd-usb-hiface.ko
INSTALL sound/usb/line6/snd-usb-line6.ko
INSTALL sound/usb/line6/snd-usb-pod.ko
INSTALL sound/usb/line6/snd-usb-podhd.ko
INSTALL sound/usb/line6/snd-usb-toneport.ko
INSTALL sound/usb/line6/snd-usb-variax.ko
INSTALL sound/usb/misc/snd-ua101.ko
INSTALL sound/usb/snd-usb-audio.ko
INSTALL sound/usb/snd-usbmidi-lib.ko
INSTALL sound/usb/usx2y/snd-usb-us122l.ko
INSTALL sound/usb/usx2y/snd-usb-usx2y.ko
INSTALL sound/x86/snd-hdmi-lpe-audio.ko
INSTALL sound/xen/snd_xen_front.ko
INSTALL virt/lib/irqbypass.ko
DEPMOD 5.10.0
[root@localhost kernel-5.10.0-13.0.0]# make install
sh ./arch/x86/boot/install.sh 5.10.0 arch/x86/boot/bzImage \
System.map "/boot"
dracut-install: Failed to find module 'kvd' /lib/modules/5.10.0/kernel/drivers/block/kvd.ko
dracut-install: Failed to find module 'uds' /lib/modules/5.10.0/kernel/drivers/block/uds.ko
[root@localhost kernel-5.10.0-13.0.0]#

```

命令输入 (按ALT键提示历史,TAB键路径,ESC键返回,双击CTRL切换)

历史 选项

文件 命令

/root

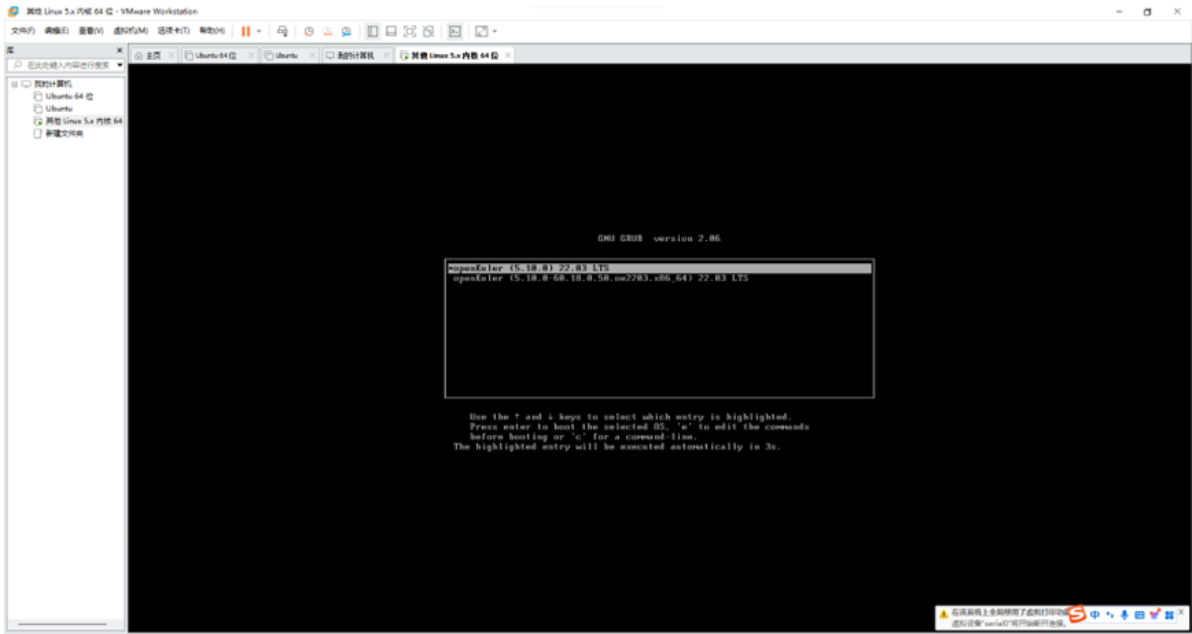
历史

文件名	大小	类型	修改时间	权限	用户
.config		文件夹	2025/04/13 12:03	drwx-----	0/0
.bash_logout	18 B	Bash Log...	2019/12/24 08:00	-rw-r--r--	0/0
.bash_profile	176 B	Bash Prof...	2019/12/24 08:00	-rw-r--r--	0/0
.bashrc	176 B	Bash RC ...	2019/12/24 08:00	-rw-r--r--	0/0
.cshrc	100 B	CSHRC文...	2019/12/24 08:00	-rw-r--r--	0/0
.tcshrc	129 B	TCSHRC ...	2019/12/24 08:00	-rw-r--r--	0/0
.viminfo	532 B	VIMINFO...	2025/04/13 12:03	-rw-r--r--	0/0
anaconda-ks.cfg	1 KB	Configura...	2025/04/13 19:50	-rw-----	0/0

建议在 vmware 中执行命令

最后重启检查

1. grub 引导界面



2. uname -r 查看

FinalShell 4.3.10

同步状态

IP 192.168.88.146 复制

系统信息

运行 21 分

负载 0.74, 0.67, 0.32

CPU 2%

内存 25% 594M/2.4G

交换 0% 0/3G

内存	CPU	命令
5.6M	1	sshd
3.5M	0.7	bash
15.7M	0	systemd
0	0	kthreadd

↑ 928B ↓ 378B ens33

24K

16K

8K

0ms 本机

0

0

0

路径	可用/大小
/dev	1.1G/1.1G
/dev/shm	1.2G/1.2G
/run	474M/481M
/sys/fs/cgr...	4M/4M
/	12.8G/35.2G
/tmp	1.2G/1.2G
/boot	425M/973M

激活/升级

1 meteor x 2 euler x +

[root@localhost ~]# uname -r

5.10.0

[root@localhost ~]#

命令输入 (按ALT键提示历史,TAB键路径,ESC键返回,双击CTRL切换) 历史 选项

文件 命令

/root

文件

大小

类型

修改时间

权限

用户

文件夹

2025/04/13 12:03

drwx-----

0/0

文件夹

2025/04/13 12:24

drwxr-xr-x

0/0

文件夹

2025/04/13 13:18

drwxr-xr-x

0/0

404 B

BASH_HI...

2025/04/13 13:35

-rw-----

0/0

18 B

Bash Log...

2019/12/24 08:00

-rw-r--r--

0/0

176 B

Bash Prof...

2019/12/24 08:00

-rw-r--r--

0/0

176 B

Bash RC ...

2019/12/24 08:00

-rw-r--r--

0/0

100 B

CSHRC 文

2019/12/24 08:00

-rw-r--r--

0/0