

一. 实验环境配置

1. 虚拟机: Virtual Box 6.1.36

(1) 官网下载 Virtual Box (<https://www.virtualbox.org/>)



(2) 安装 Virtual Box

按默认安装设定, 点击“下一步”, 即可成功安装。

(3) 新建虚拟机

a)



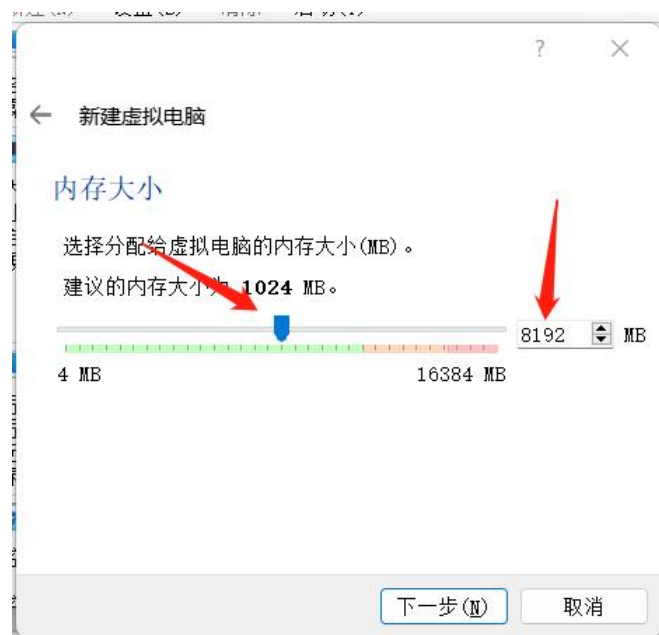
点击“新建”。

b)



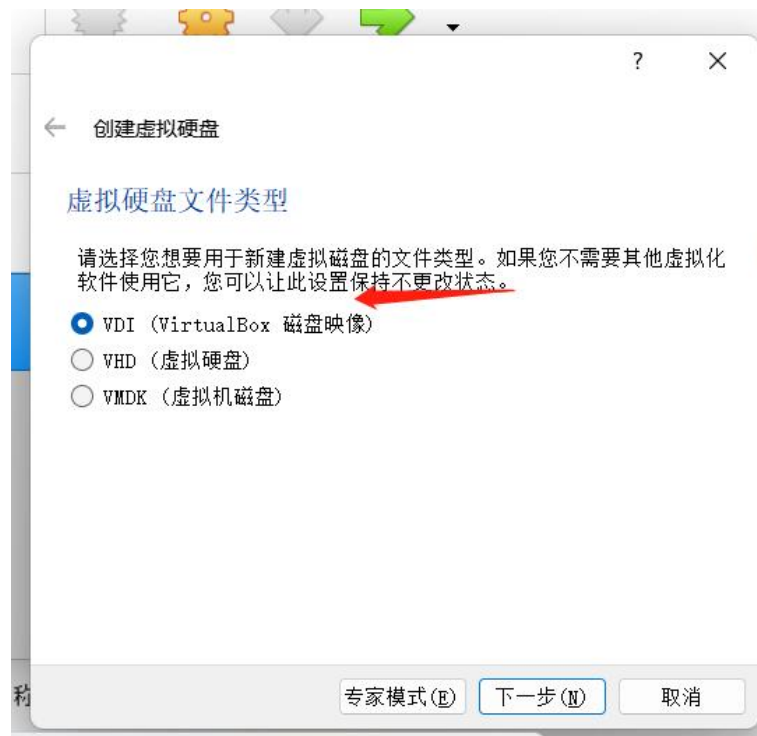
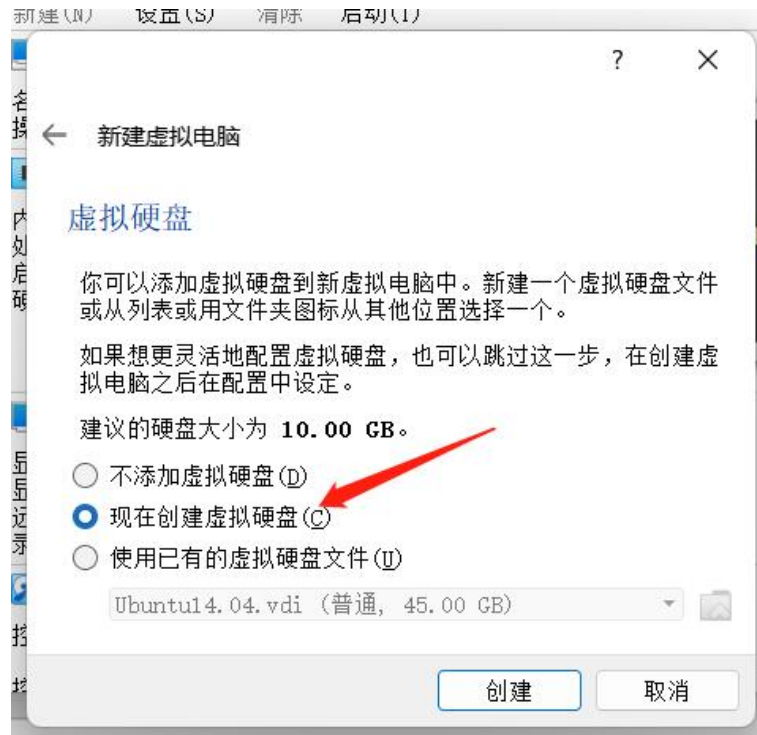
虚拟机名称可以自定义，存储文件夹路径可以自行选择，从这里也可以看出虚拟机是以一系列文件的形式存储在宿主机中，类型选择 Linux，版本根据宿主机位数进行选择，比如我的电脑系统是 windows64 位，版本就选择 Ubuntu(64-bit)。

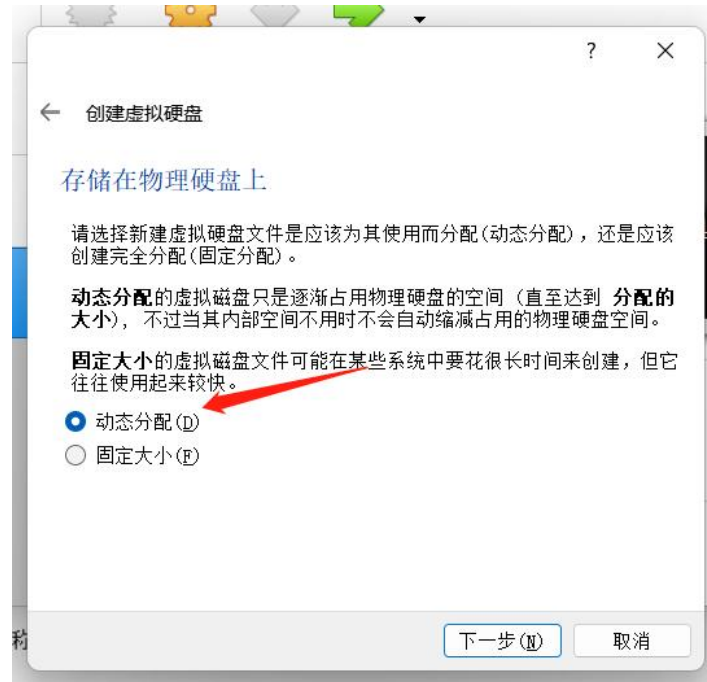
c)



选择分配给虚拟机的内存大小，可以直接拖动滑动条选择分配内存的大小，也可以通过键盘输入自定义分配内存的大小。

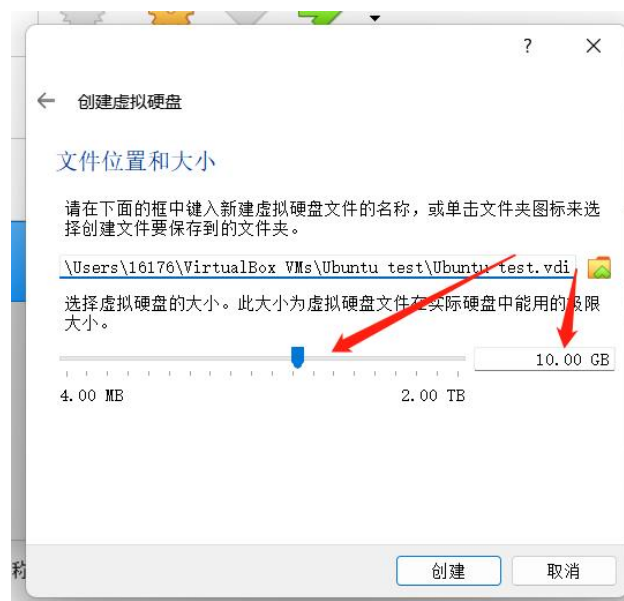
d)



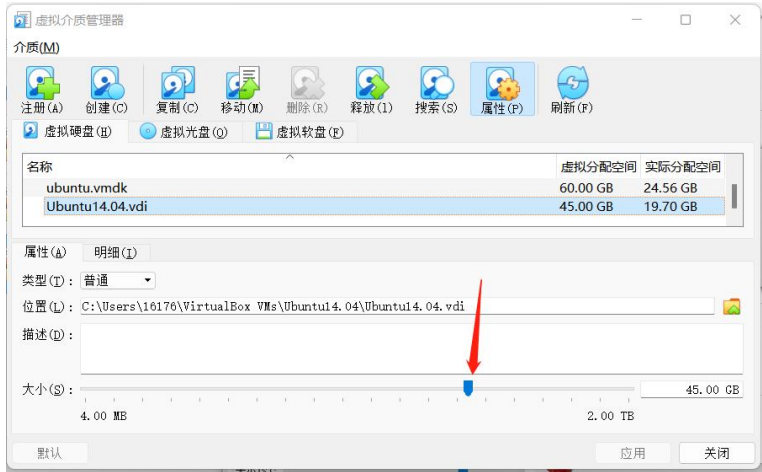


按照默认设定点击即可，这里动态分配指假设虚拟硬盘文件在实际硬盘中能用的极限大小是 10G，选择动态分配的话，虚拟硬盘在实际硬盘中占用的空间是 0-10G，固定大小指假设虚拟硬盘文件在实际硬盘中能用的极限大小是 10G，选择固定大小的话，虚拟硬盘占用实际硬盘的大小永远是 10G，无论空间是否全部使用。

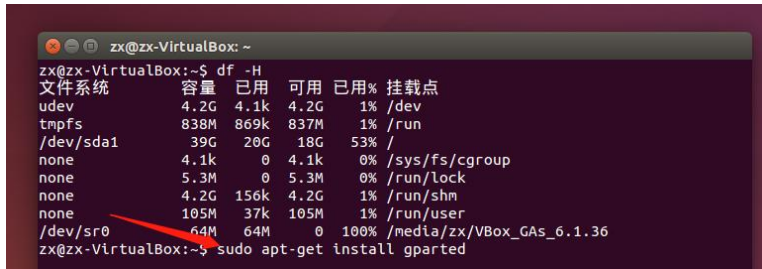
e)



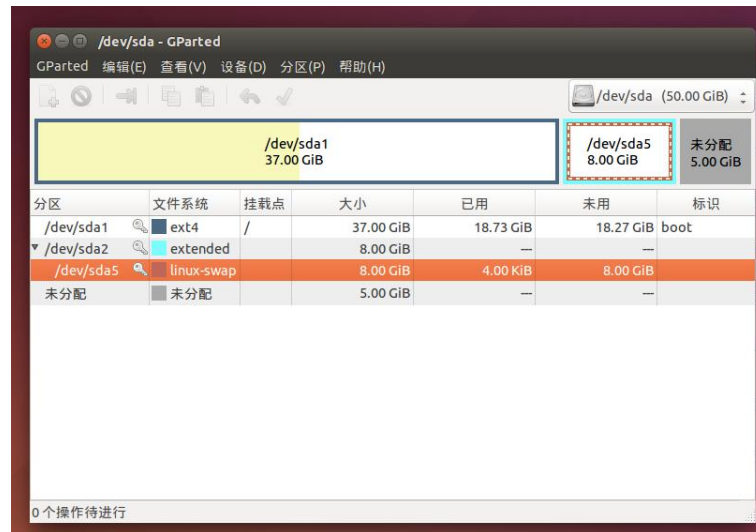
选择虚拟硬盘文件在实际硬盘中能用的极限大小，建议在创建虚拟机的时候便分配足够的磁盘空间，如果在虚拟机使用过程中出现存储容量不足的情况，可以参照下述流程进行扩容。



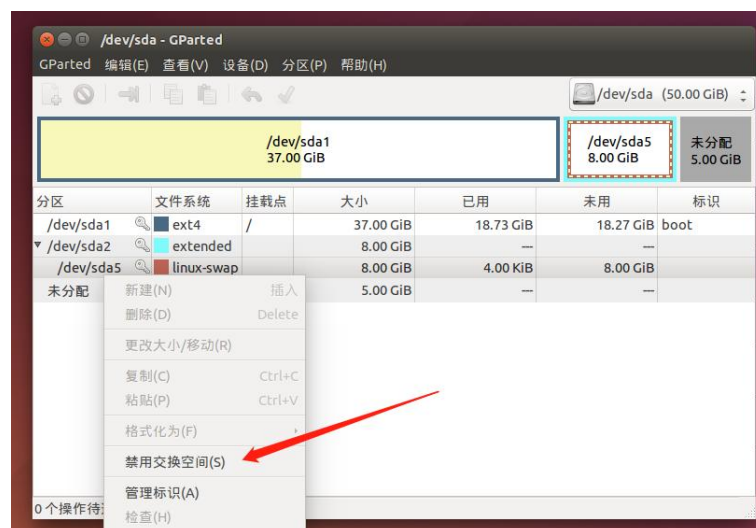
点击“管理”，选择“虚拟介质管理器”，拖动滑动条调整分配给虚拟机的磁盘空间大小。

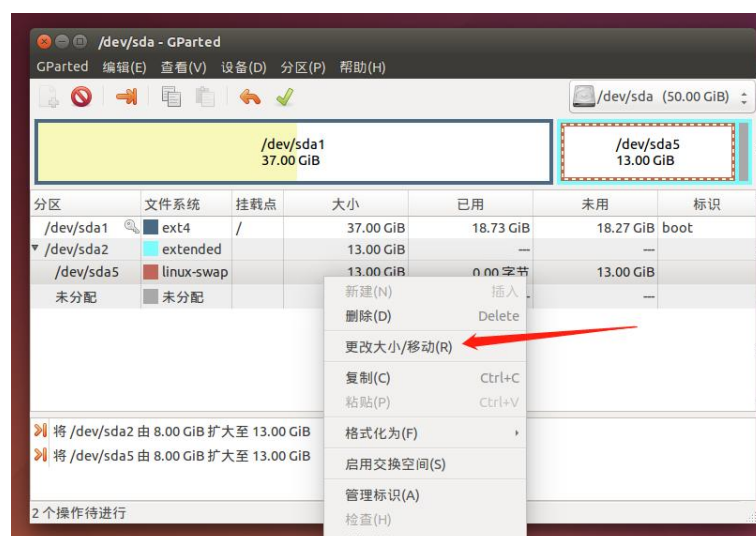
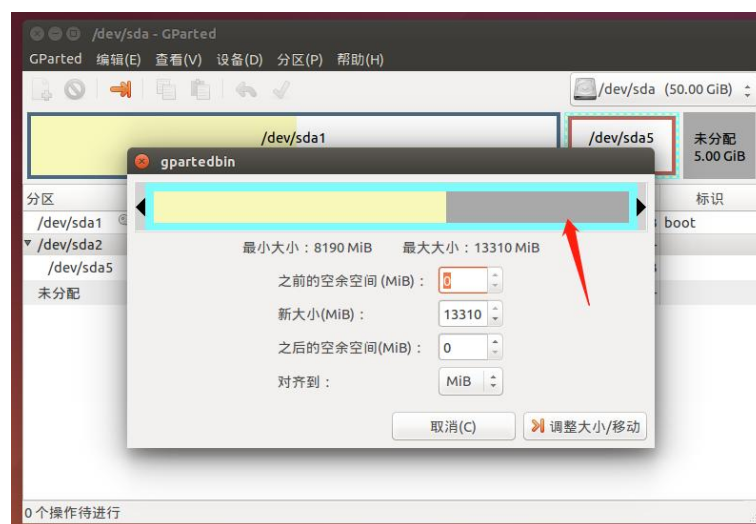
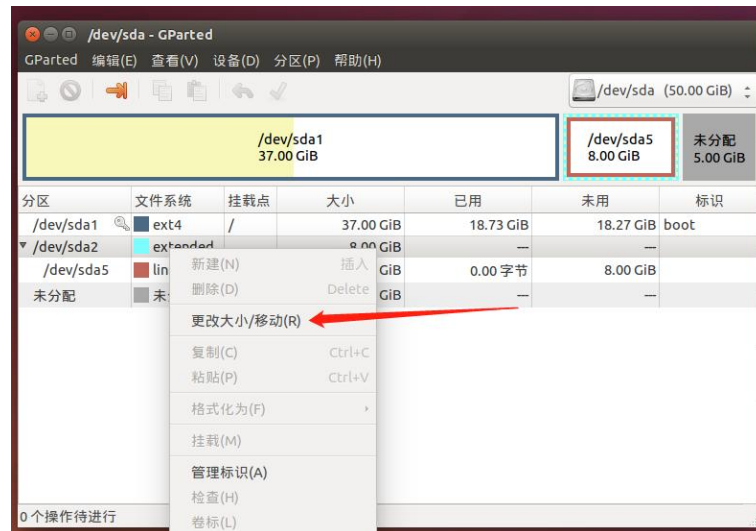


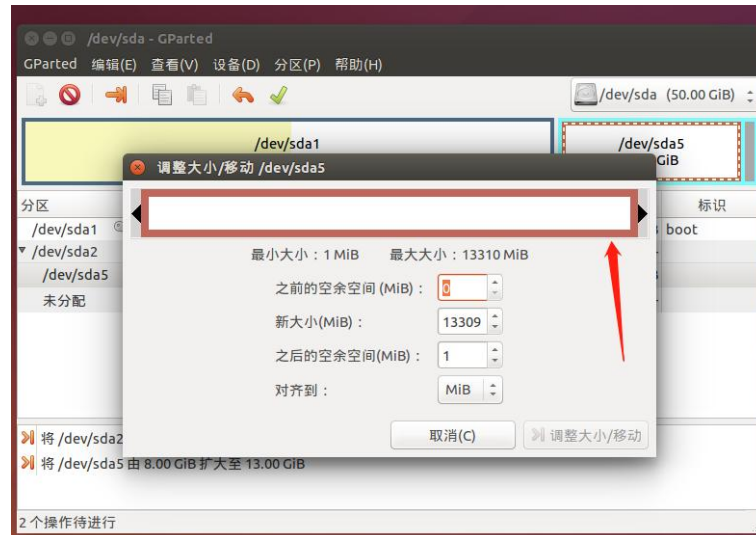
```
正在处理用于 libgdkmm-2.4-1c2a:amd64 (1:2.24.4-1ubuntu1) ...  
正在设置 libgdkmm-2.4-1c2a:amd64 (1:2.24.4-1ubuntu1) ...  
正在设置 gparted (0.18.0-1) ...  
正在处理用于 libc-bin (2.19-0ubuntu6.15) 的触发器 ...  
zx@zx-VirtualBox:~$ sudo gparted
```



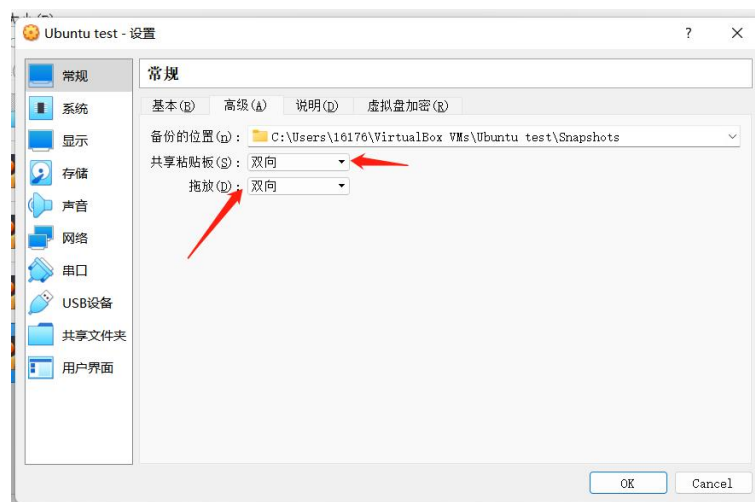
下载并运行 GParted，出现如上界面，但此时 /dev/sda2/ 分区以及 /dev/sda5/ 分区旁边有个“小钥匙”，无法对其大小进行更改，此时需要右键点击“linux-swap”，然后点击“禁用交换空间”，这时“小钥匙”便消失啦！然后通过右键点击“需要调整大小的分区”，然后点击“更换大小”，来对分区大小进行调整啦！







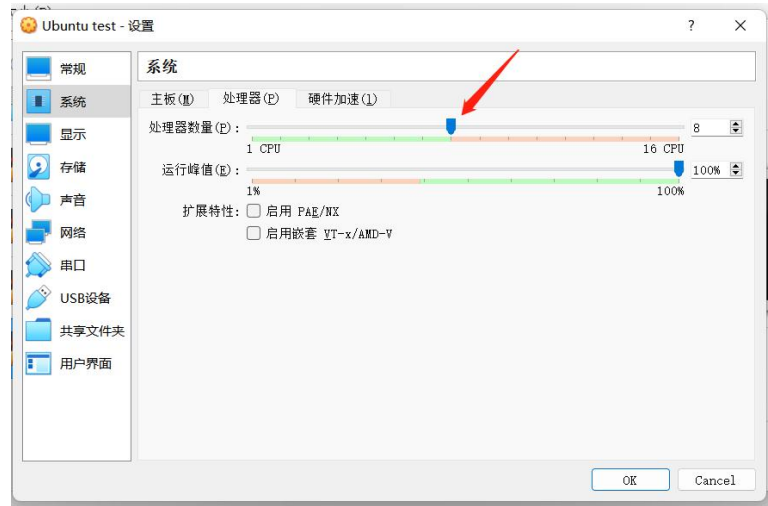
f)



共享粘贴板和拖放选择“双向”，然后在虚拟机中点击“设备”，安装增强功能，之后宿主机和虚拟机便可以共享粘贴板啦！同时安装增强功能后，点击“视

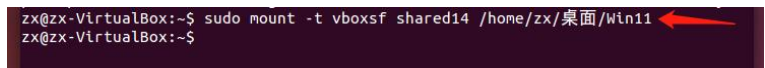
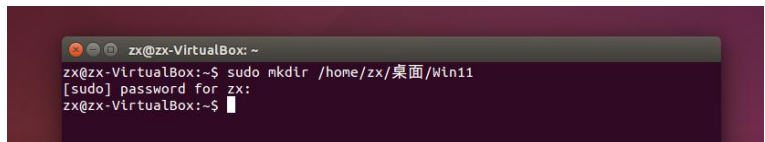
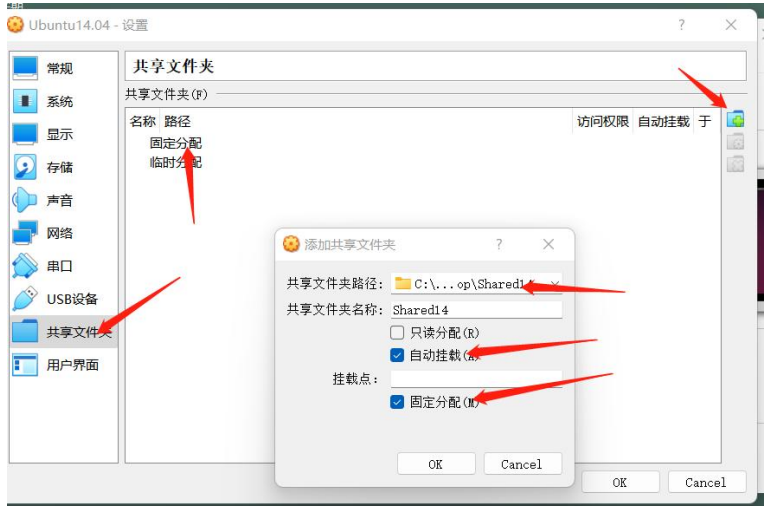
图”，可以选择多种显示模式，更有利于用户使用虚拟机。

g)



配置虚拟机的处理器数量。

h)

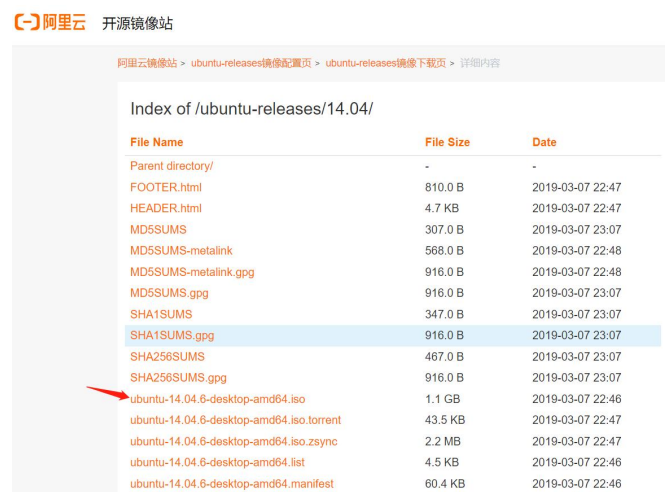


设置共享文件夹，“Shared14”是宿主机的共享文件夹，“Win11”是虚拟机中的共享文件夹，通过“`sudo mount -t vboxsf shared14 /home/zx/桌面/Win11`”将宿主机中的共享文件夹与虚拟机中的共享文件夹连接起来，注意在关闭虚拟机后连接便会自动断开，下一次启动虚拟机需要重新输入上述命令才能将宿主机与虚拟机中的共享文件夹连接起来。

2. 操作系统：Ubuntu 14.04

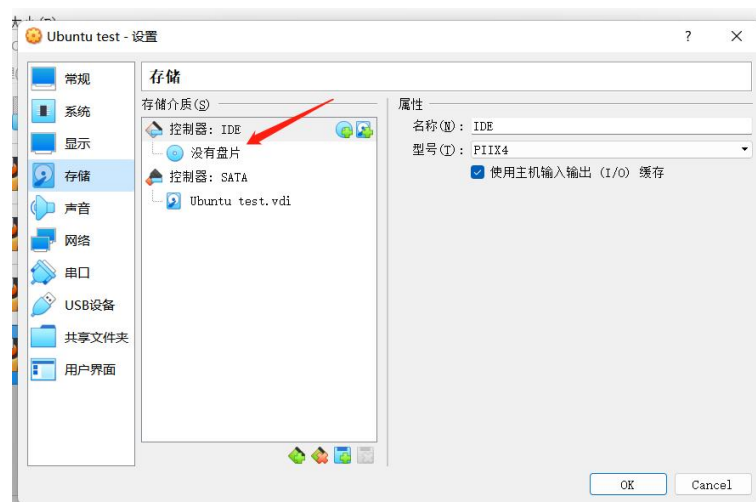
(1) 阿里云镜像网站下载 Ubuntu 14.04

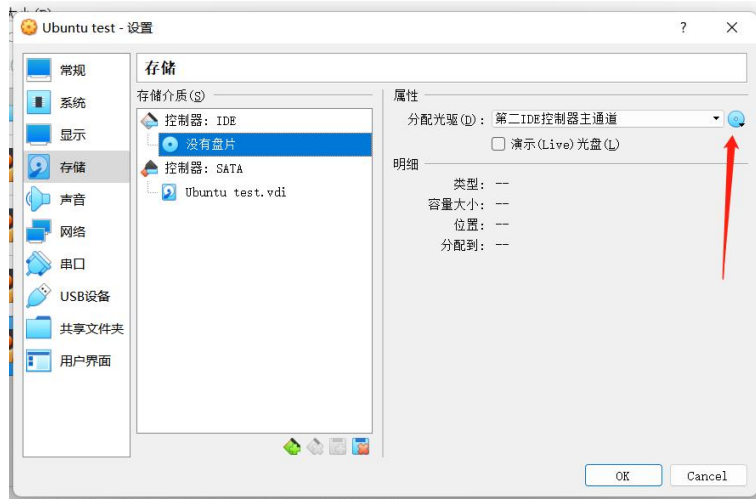
网址：<http://mirrors.aliyun.com/ubuntu-releases/14.04/>



(2) 安装 Ubuntu 14.04 操作系统

a)

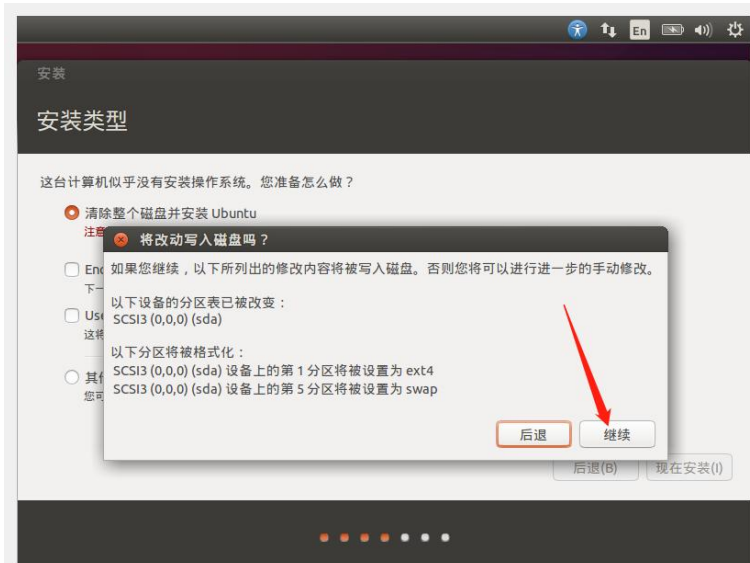
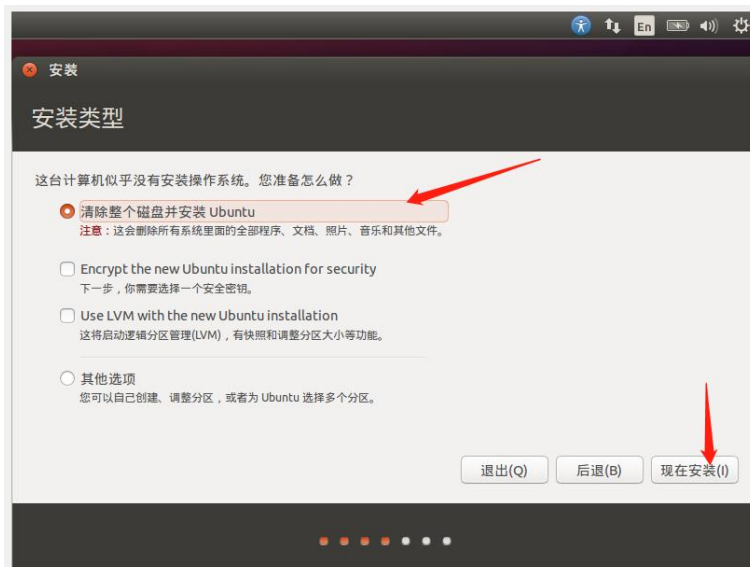


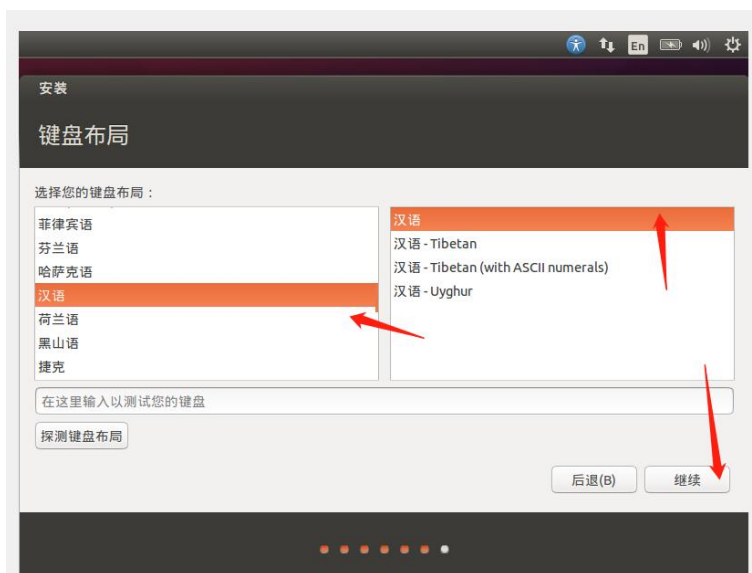
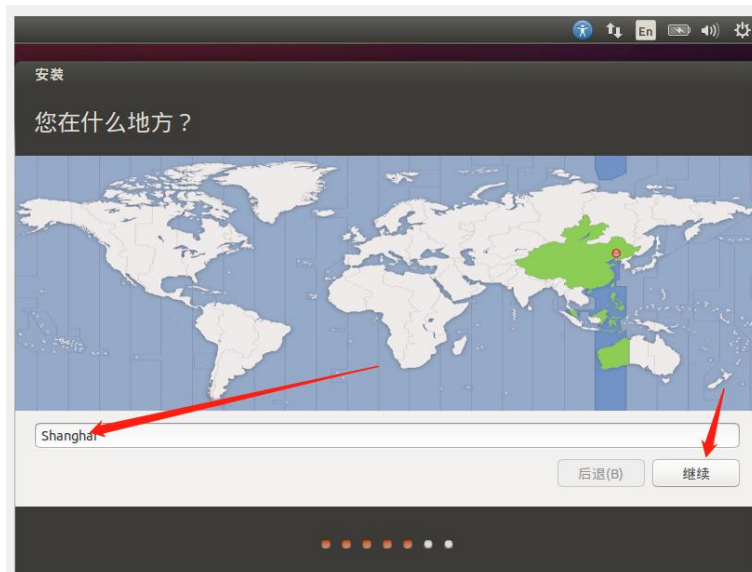


选择上一步在阿里云镜像网站下载好的 Ubuntu 14.04 光盘镜像文件。

b)







按照默认设定点击，即可成功安装 Ubuntu 14.04 操作系统。

c) 配置清华源

- 网址: <https://mirrors.tuna.tsinghua.edu.cn/help/ubuntu/>

Ubuntu 镜像使用帮助

本镜像仅包含 32/64 位 x86 架构处理器的软件包。在 ARM(arm64, armhf)、PowerPC(ppc64el)、RISC-V(riscv64) 和 S390x 等架构的设备上 (对应官方源为ports.ubuntu.com) 请使用 [ubuntu-ports 镜像](#)。

手动替换

Ubuntu 的软件源配置文件是 `/etc/apt/sources.list`。将系统自带的该文件做个备份，将该文件替换为下面内容，即可使用 TUNA 的软件源镜像。

选择你的ubuntu版本: 14.04 LTS

```
# 默认注释了源码镜像以提高 apt update 速度，如有需要可自行取消注释
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ trusty main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ trusty main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ trusty-updates main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ trusty-updates main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ trusty-backports main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ trusty-backports main restricted universe multiverse
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ trusty-security main restricted universe multiverse
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ trusty-security main restricted universe multiverse
```

- 备份源文件

```
zx@zx-VirtualBox:~$ cd /etc/apt/
zx@zx-VirtualBox:/etc/apt$ sudo cp sources.list sources.list.bak
[sudo] password for zx:
zx@zx-VirtualBox:/etc/apt$
```

- 配置清华源

```
zx@zx-VirtualBox:/etc/apt$ sudo gedit sources.list
```

将上述网址中对应 Ubuntu 版本的镜像源码粘贴至源配置文件。

- 更新清华源

```
zx@zx-VirtualBox:/etc/apt$ sudo apt-get update
获取: 1 https://mirrors.tuna.tsinghua.edu.cn trusty InRelease
忽略 https://mirrors.tuna.tsinghua.edu.cn trusty InRelease
命中 http://security.ubuntu.com trusty-security InRelease
忽略 http://extras.ubuntu.com trusty InRelease
获取: 2 https://mirrors.tuna.tsinghua.edu.cn trusty-backports InRelease
获取: 3 https://mirrors.tuna.tsinghua.edu.cn trusty-security InRelease
```

- 更新软件

```
zx@zx-VirtualBox:/etc/apt$ sudo apt-get upgrade
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
正在对升级进行计算... 完成
```

3. Linux 内核: 3.19.1

(1) 下载 Linux 3.19.1 内核

- 网址: <https://mirror.bjtu.edu.cn/kernel/linux/kernel/v3.0/>

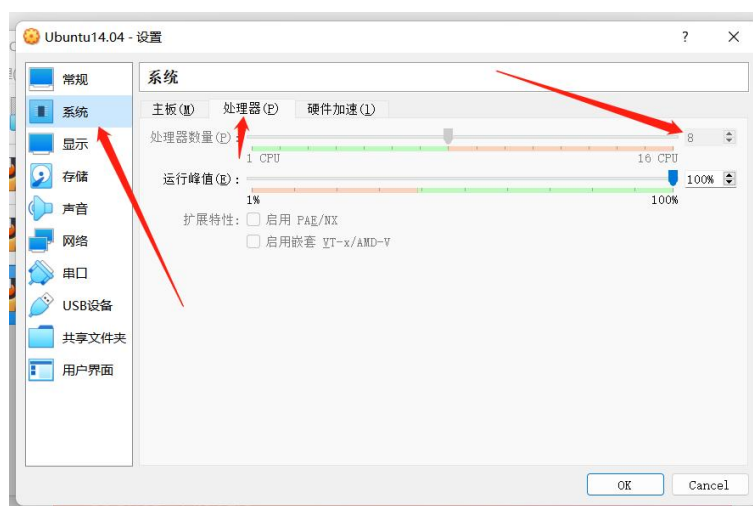
linux-3.18.tar.gz	08-Dec-2014 08:01	117M
linux-3.18.tar.sign	08-Dec-2014 08:01	473
linux-3.18.tar.xz	08-Dec-2014 08:01	77M
linux-3.19.1.tar.gz	07-Mar-2015 21:46	117M
linux-3.19.1.tar.sign	07-Mar-2015 21:46	819
linux-3.19.1.tar.xz	07-Mar-2015 21:46	78M
linux-3.19.2.tar.gz	18-Mar-2015 22:08	117M
linux-3.19.2.tar.sign	18-Mar-2015 22:08	819

```

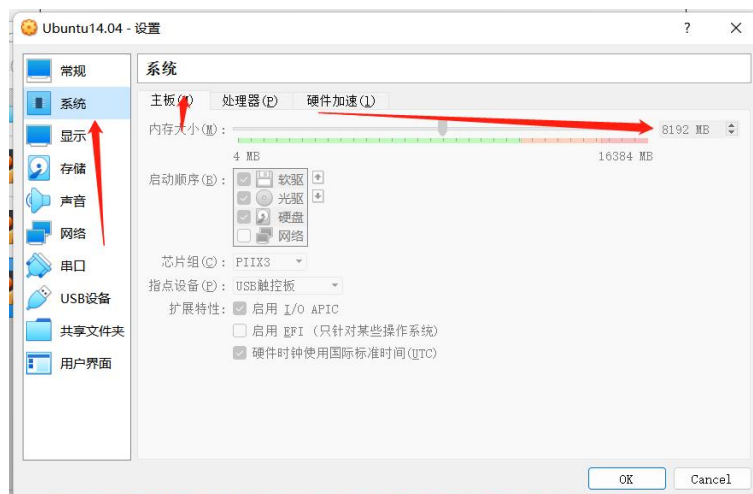
zx@zx-VirtualBox:~/桌面$ ls
gnome-terminal.desktop  linux-3.19.1.tar.xz
zx@zx-VirtualBox:~/桌面$ sudo cp ./linux-3.19.1.tar.xz /usr/src
[sudo] password for zx:
zx@zx-VirtualBox:~/桌面$ ls
gnome-terminal.desktop  linux-3.19.1.tar.xz
zx@zx-VirtualBox:~/桌面$ cd /usr/src
zx@zx-VirtualBox:/usr/src$ ls
linux-3.19.1.tar.xz  linux-headers-4.4.0-142  linux-headers-4.4.0-142-generic
zx@zx-VirtualBox:/usr/src$

```

4. 处理器：8 核(Virtual Box 分配) Intel(R) Core(TM) i7-10875H



5. 内存：8192MB



二. 实验方法步骤

(1) 解压压缩包

```
zx@zx-VirtualBox:/usr/src$ sudo tar xvfj ./linux-3.19.1.tar.xz
linux-3.19.1/virt/kvm/eventfd.c
linux-3.19.1/virt/kvm/iodev.h
linux-3.19.1/virt/kvm/irqchip.c
linux-3.19.1/virt/kvm/kvm_main.c
linux-3.19.1/virt/kvm/vfio.c
linux-3.19.1/virt/kvm/vfio.h
zx@zx-VirtualBox:/usr/src$ ls
linux-3.19.1      linux-headers-4.4.0-142
linux-3.19.1.tar.xz  linux-headers-4.4.0-142-generic
```

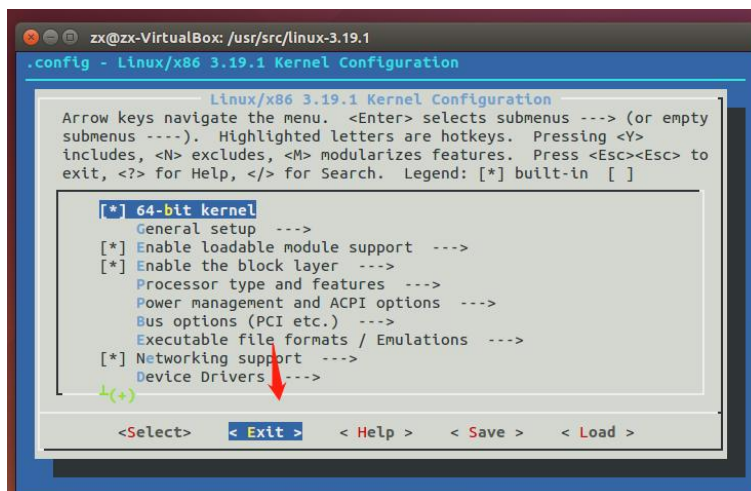
(2) 安装第三方库

```
zx@zx-VirtualBox:/usr/src/linux-3.19.1$ sudo apt-get install libncurses5-dev build-essential kernel-package
```

- libncurses5-dev 是为之后配置内核能运行 make menuconfig 程序做准备
- build-essential 为编译工具
- kernel-package 是编译内核工具

(3) 配置文件

```
zx@zx-VirtualBox:/usr/src/linux-3.19.1$ sudo make menuconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/mconf.o
SHIPPED scripts/kconfig/zconf.tab.c
SHIPPED scripts/kconfig/zconf.lex.c
SHIPPED scripts/kconfig/zconf.hash.c
HOSTCC scripts/kconfig/zconf.tab.o
HOSTCC scripts/kconfig/lxdialog/checklist.o
HOSTCC scripts/kconfig/lxdialog/util.o
```



打开配置菜单后，直接关闭即可，不进行改动。

(5) 编译内核

```
zx@zx-VirtualBox:/usr/src/linux-3.19.1$ sudo make -j8
```

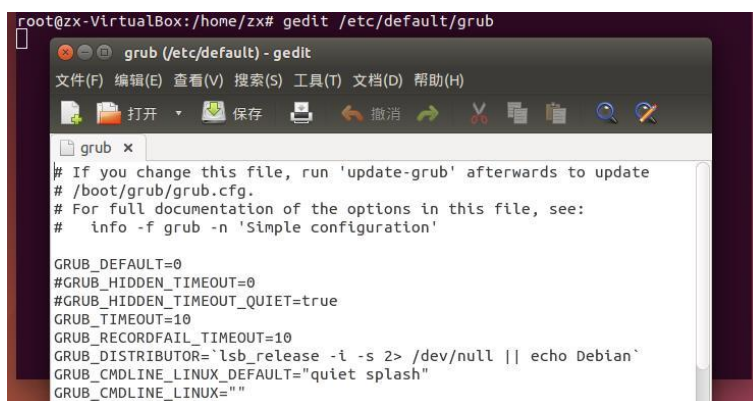
(6) 安装内核模块

```
zx@zx-VirtualBox:/usr/src/linux-3.19.1$ sudo make modules_install
```

```
zx@zx-VirtualBox:/usr/src/linux-3.19.1$ sudo make install
sh ./arch/x86/boot/install.sh 3.19.1 arch/x86/boot/bzImage \
    System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 3.19.1 /boot/vmlinu
z-3.19.1
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 3.19.1 /boot/vmlinuz
-3.19.1
update-initramfs: Generating /boot/initrd.img-3.19.1
run-parts: executing /etc/kernel/postinst.d/pm-utils 3.19.1 /boot/vmlinuz-3.19.1
run-parts: executing /etc/kernel/postinst.d/update-notifier 3.19.1 /boot/vmlinuz
-3.19.1
```

(7) 修改引导菜单配置

- `gedit /etc/default/grub` 按下图所示编辑/etc/default/grub 文件



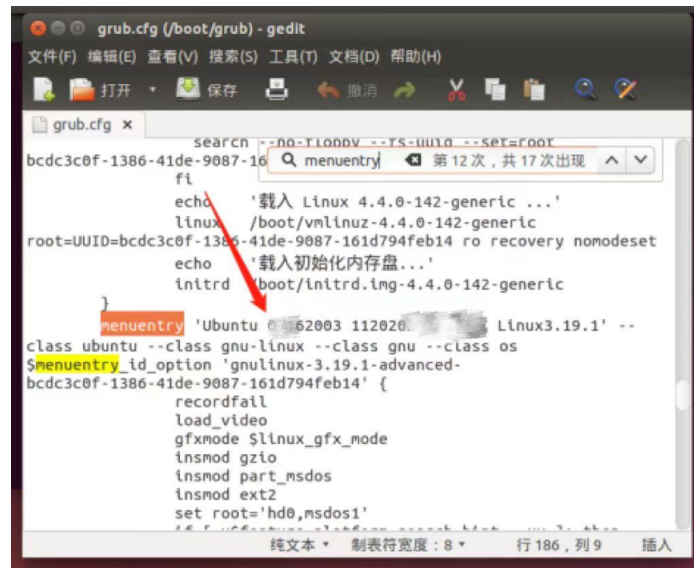
```
root@zx-VirtualBox:/home/zx# gedit /etc/default/grub
grub (/etc/default) - gedit
文件(F) 编辑(E) 查看(V) 搜索(S) 工具(T) 文档(D) 帮助(H)
打开 保存 撤消 重做 剪切 粘贴 查找 替换
grub x
# If you change this file, run 'update-grub' afterwards to update
# /boot/grub/grub.cfg.
# For full documentation of the options in this file, see:
#   info -f grub -n 'Simple configuration'

GRUB_DEFAULT=0
#GRUB_HIDDEN_TIMEOUT=0
#GRUB_HIDDEN_TIMEOUT_QUIET=true
GRUB_TIMEOUT=10
GRUB_RECORDFAIL_TIMEOUT=10
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
```

- `sudo update-grub` 更新启动选项

```
root@zx-VirtualBox:/home/zx# sudo update-grub
```

- `gedit /boot/grub/grub.cfg` 按下图所示，找到新编译的内核对应的启动选项，按要求修改其名称



```
grub.cfg (/boot/grub) - gedit
文件(F) 编辑(E) 查看(V) 搜索(S) 工具(T) 文档(D) 帮助(H)
grub.cfg x
search --no-floppy --fs-uuid --set=root
bcd3c0f-1386-41de-9087-16
ft
echo '载入 Linux 4.4.0-142-generic ...'
linux /boot/vmlinuz-4.4.0-142-generic
root=UUID=bcd3c0f-1386-41de-9087-161d794feb14 ro recovery nomodeset
echo '载入初始化内存盘...'
initrd /boot/initrd.img-4.4.0-142-generic
}
menuentry 'Ubuntu 0.000000 112020 Linux3.19.1' --
class ubuntu --class gnu-linux --class gnu --class os
$menuentry_id_option 'gnulinux-3.19.1-advanced-
bcd3c0f-1386-41de-9087-161d794feb14' {
recordfail
load_video
gfxmode $linux_gfx_mode
insmod gzio
insmod part_msdos
insmod ext2
set root='hd0,msdos1'
```

GRUB 是多启动规范的实现，其允许用户可以在计算机内同时拥有多个操作系统内核，并在计算机启动时选择希望运行的操作系统内核，同时 GRUB 可用于向这些内核传递启动参数。

(8) 重启效果展示

