内核编译、更改相关

前置条件

1. 下载安装VMware虚拟机 https://www.vmware.com/cn/products/workstation-pro.html
2. 下载安装ubuntu镜像 https://ubuntu.com/
3. 下载安装顺手的C IDE，如VS Code等 https://code.visualstudio.com/

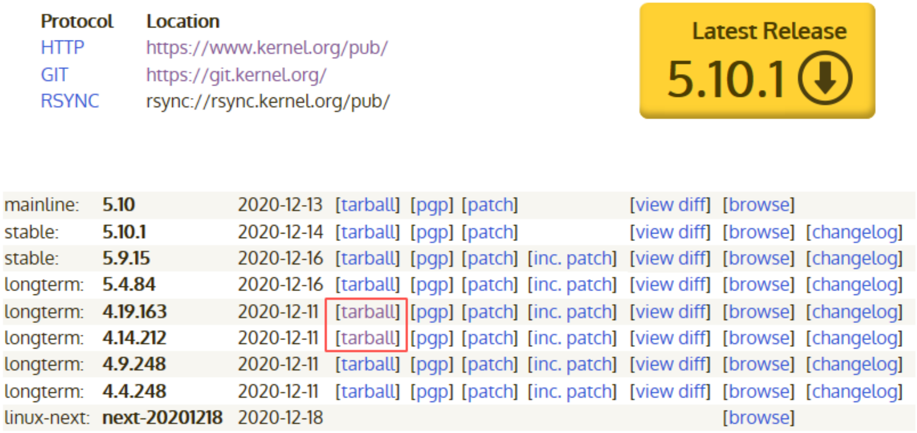
帮助：

安装虚拟机教程

注：虚拟机容量需要大一点，至少50G，否则编译时会报磁盘容量不足错误

通用流程

1. 下载需修改版本下载内核源码：www.kernel.org，本项目目前基于4.14，之后可能会迁移至4.19。



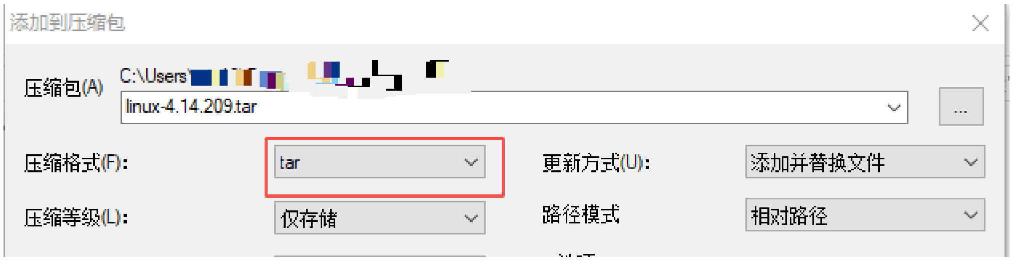
Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

mainline:	5.10	2020-12-13	[tarball]	[pgp]	[patch]		[view diff]	[browse]	
stable:	5.10.1	2020-12-14	[tarball]	[pgp]	[patch]		[view diff]	[browse]	[changelog]
stable:	5.9.15	2020-12-16	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.4.84	2020-12-16	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.19.163	2020-12-11	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.14.212	2020-12-11	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.9.248	2020-12-11	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.4.248	2020-12-11	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	next-20201218	2020-12-18						[browse]	

2. 解压至指定路径，根据需求修改源码
3. 将修改后的源码打包压缩成.tar.gz文件

   windows系统打包方式：

   a. 下载7zip，安装后直接在你想要打包的文件上点右键菜单，会有一个7-ZIP的子菜单栏
   b. 选“7-ZIP” -> “添加到压缩档案”，在弹出来的窗口里有个“压缩格式”的选项，里面并没有tar.gz格式，但里面有一个Tar格式，第一步就是要先压成tar格式。



添加到压缩包

压缩包(A): C:\Users[redacted]
linux-4.14.209.tar

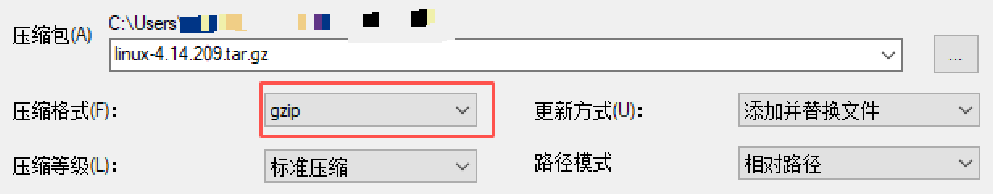
压缩格式(F): tar

更新方式(U): 添加并替换文件

压缩等级(L): 仅存储

路径模式 相对路径

c. 第二步，再在这个tar文件上面点右键，选“7-ZIP” -> “添加到压缩档案”，这时候弹出的窗口里再看“压缩格式选项”，选择“GZip”格式后确定，最后结果就是一个新的tar.gz格式的文件。



压缩包(A) C:\Users\[redacted]
linux-4.14.209.tar.gz

压缩格式(F): gzip
更新方式(U): 添加并替换文件
压缩等级(L): 标准压缩
路径模式 相对路径

4. 将打包后的.tar文件移动至虚拟机上，路径：/usr/src

   方法：scp远程复制、虚拟机硬盘映射方式、安装虚拟机工具拖拽、使用rz命令

注意：windows文件系统不区分大小写，linux文件系统区分，linux中有些大小写不同的文件在windows中会丢，所以只传修改的文件就好，不要传整个linux压缩包，也可直接用vim在linux上改代码。

5. 切换用户至root：

```bash
sudo su
```

6. 解压内核源码

```bash
tar -zxvf linux-4.14.209.tar.gz
```

7. 安装必要服务

```bash
sudo apt-get update
sudo apt-get install gcc
sudo apt-get install openssh-server
sudo apt-get install libncurses5-dev
sudo apt-get install bison
sudo apt-get install flex
sudo apt-get install vim
sudo apt-get install libssl-dev
sudo apt-get install libelf-dev
sudo apt-get install git fakeroot build-essential ncurses-dev xz-utils libssl-de
```

- 下载有问题，可以连一下学校wifi或者vpn试试。

8. 升级gcc

- gcc版本不宜太低，也不宜太高，最好在7/8。

Bash

```bash
gcc -v             #查看gcc的版本
sudo apt-get install gcc-7
ls /usr/bin/gcc*   #查看当前系统中安装的所有的gcc版本
#给各版本设置优先级，最后一位数字越大，优先级越高，会自动使用值最大的做当前版本
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 100
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-9 50
sudo update-alternatives --config gcc    #选择不同版本的gcc
```

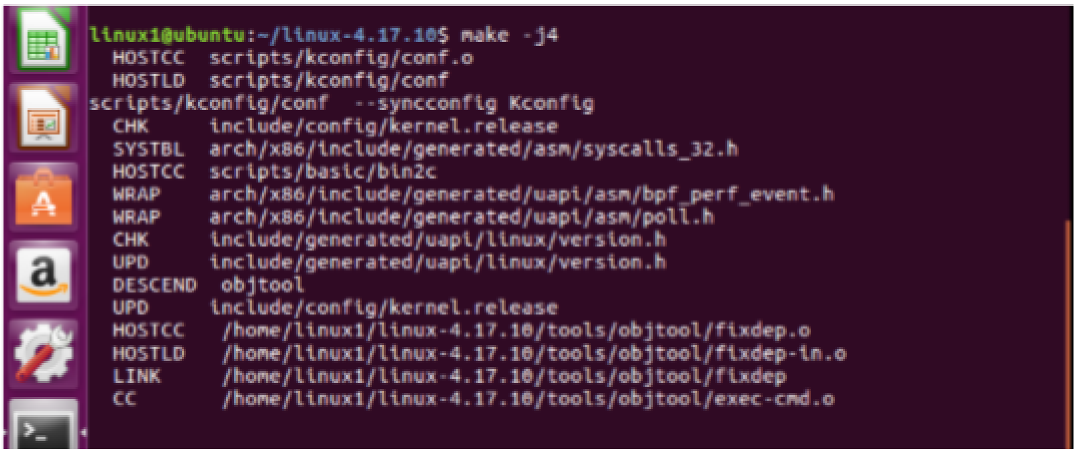
9. 拷贝配置文件到/boot下，并编译内核

查看 cat /sys/kernel/
security/lsm

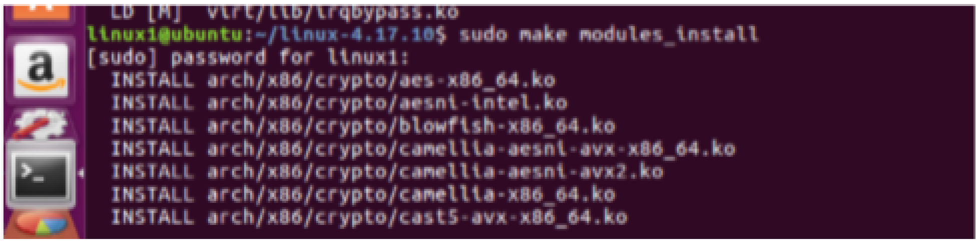
Bash

```bash
cp /boot/config-`uname -r`  ./.config
make menuconfig      #使用默认选项就好，save后exit
make -j4             #耗时会比较长，需要关注是否有error
make modules_install
make install
```

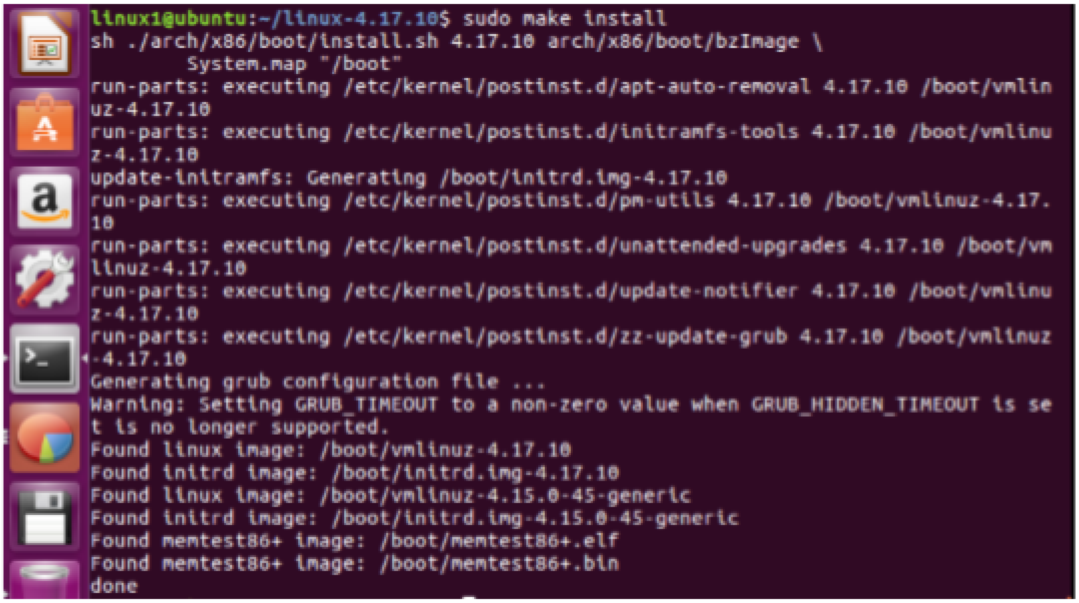
- 编译过程中有失败的情况，基本上都是因为缺包，apt-get install一下就好
- 编译过程中会有一些warning，主要是因为gcc太新了，可以忽略。https://bbs.archlinux.org/viewtopic.php?id=236890
- 编译过程中可能会出现错误，若中途出错在重新执行 make 之前首先要执行 make clean 指令清除之前产生的文件。



```
linux1@ubuntu:~/linux-4.17.10$ make -j4
  HOSTCC  scripts/kconfig/conf.o
  HOSTLD  scripts/kconfig/conf
scripts/kconfig/conf  --syncconfig Kconfig
  CHK     include/config/kernel.release
  SYSTBL  arch/x86/include/generated/asm/syscalls_32.h
  HOSTCC  scripts/basic/bin2c
  WRAP    arch/x86/include/generated/uapi/asm/bpf_perf_event.h
  WRAP    arch/x86/include/generated/uapi/asm/poll.h
  CHK     include/generated/uapi/linux/version.h
  UPD     include/generated/uapi/linux/version.h
  DESCEND  objtool
  UPD     include/config/kernel.release
  HOSTCC   /home/linux1/linux-4.17.10/tools/objtool/fixdep.o
  HOSTLD   /home/linux1/linux-4.17.10/tools/objtool/fixdep-in.o
  LINK     /home/linux1/linux-4.17.10/tools/objtool/fixdep
  CC       /home/linux1/linux-4.17.10/tools/objtool/exec-cmd.o
```



```
  LD [M]  virt/lib/irqbypass.ko
linux1@ubuntu:~/linux-4.17.10$ sudo make modules_install
[sudo] password for linux1:
  INSTALL arch/x86/crypto/aes-x86_64.ko
  INSTALL arch/x86/crypto/aesni-intel.ko
  INSTALL arch/x86/crypto/blowfish-x86_64.ko
  INSTALL arch/x86/crypto/camellia-aesni-avx-x86_64.ko
  INSTALL arch/x86/crypto/camellia-aesni-avx2.ko
  INSTALL arch/x86/crypto/camellia-x86_64.ko
  INSTALL arch/x86/crypto/cast5-avx-x86_64.ko
```

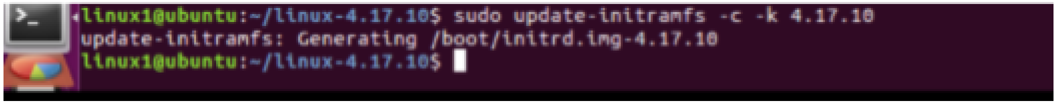


```
linux1@ubuntu:~/linux-4.17.10$ sudo make install
sh ./arch/x86/boot/install.sh 4.17.10 arch/x86/boot/bzImage \
        System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.17.10 /boot/vmlin
uz-4.17.10
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.17.10 /boot/vmlinu
z-4.17.10
update-initramfs: Generating /boot/initrd.img-4.17.10
run-parts: executing /etc/kernel/postinst.d/pm-utils 4.17.10 /boot/vmlinuz-4.17.
10
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 4.17.10 /boot/vm
linuz-4.17.10
run-parts: executing /etc/kernel/postinst.d/update-notifier 4.17.10 /boot/vmlinu
z-4.17.10
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 4.17.10 /boot/vmlinuz
-4.17.10
Generating grub configuration file ...
Warning: Setting GRUB_TIMEOUT to a non-zero value when GRUB_HIDDEN_TIMEOUT is se
t is no longer supported.
Found linux image: /boot/vmlinuz-4.17.10
Found initrd image: /boot/initrd.img-4.17.10
Found linux image: /boot/vmlinuz-4.15.0-45-generic
Found initrd image: /boot/initrd.img-4.15.0-45-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
```

10. 将内核作为引导

Bash

```bash
sudo cp arch/x86/boot/bzImage /boot/vmlinuz-4.14.201
sudo cp System.map /boot/System.map-4.14.201
cd /lib/modules/4.14.201/
sudo update-initramfs -c -k 4.14.201     #制作启动镜像
```

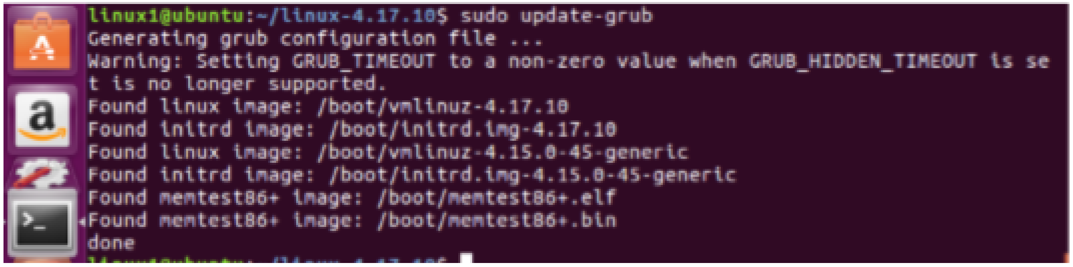


```
linux1@ubuntu:~/linux-4.17.10$ sudo update-initramfs -c -k 4.17.10
update-initramfs: Generating /boot/initrd.img-4.17.10
linux1@ubuntu:~/linux-4.17.10$ 
```

11. 更新grub

Bash

```bash
sudo update-grub        #自动查找新内核并更新grub
```



```
linux1@ubuntu:~/linux-4.17.10$ sudo update-grub
Generating grub configuration file ...
Warning: Setting GRUB_TIMEOUT to a non-zero value when GRUB_HIDDEN_TIMEOUT is se
t is no longer supported.
Found linux image: /boot/vmlinuz-4.17.10
Found initrd image: /boot/initrd.img-4.17.10
Found linux image: /boot/vmlinuz-4.15.0-45-generic
Found initrd image: /boot/initrd.img-4.15.0-45-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
```

12. 修改/etc/default/grub文件

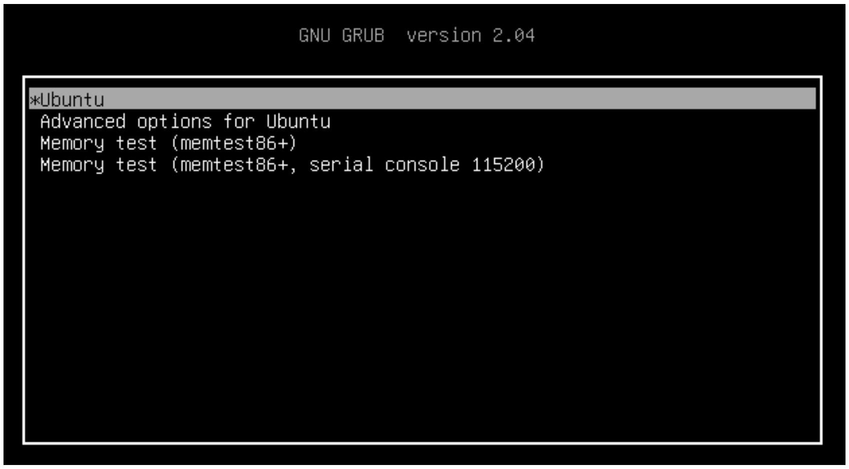
sudo vi /etc/default/grub

Bash

```bash
#将GRUB_HIDDEN_TIMEOUT=0注释掉
#将GRUB_TIMEOUT值改成=10
```

- 修改之后都要update-grub一下

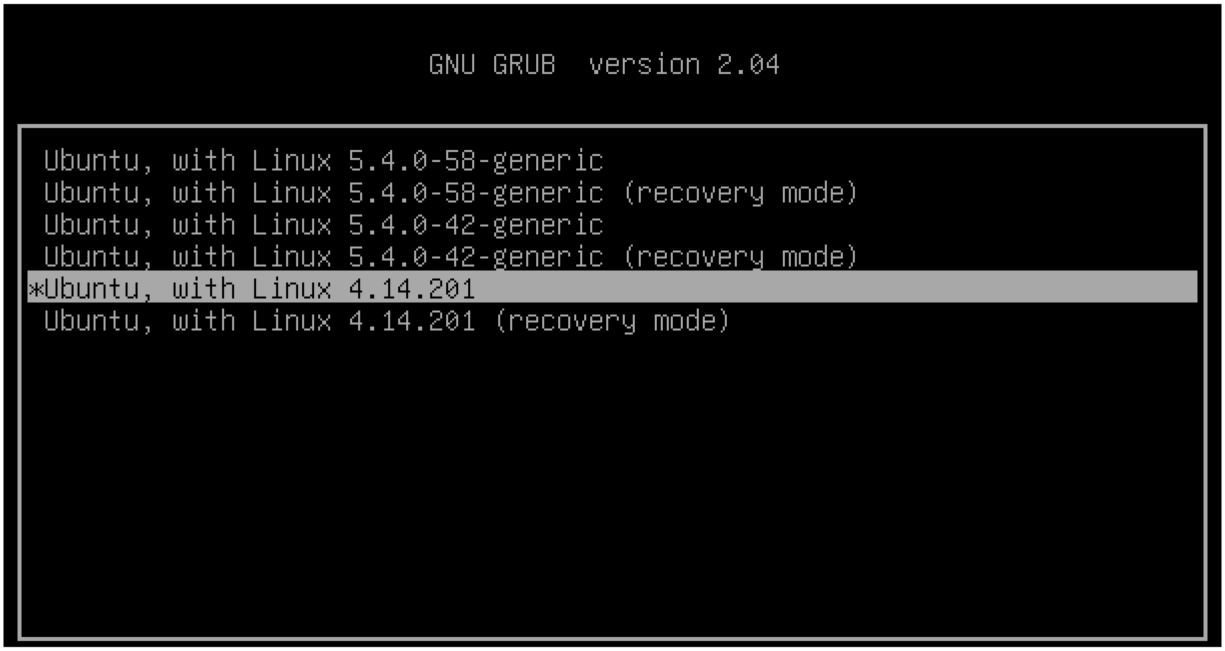
13. 重启计算机，选择 Advanced options for Ubuntu；



```
GNU GRUB  version 2.04

*Ubuntu
 Advanced options for Ubuntu
 Memory test (memtest86+)
 Memory test (memtest86+, serial console 115200)
```

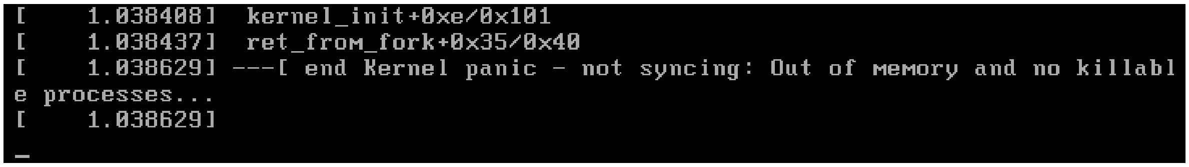
14. 选择新安装的内核



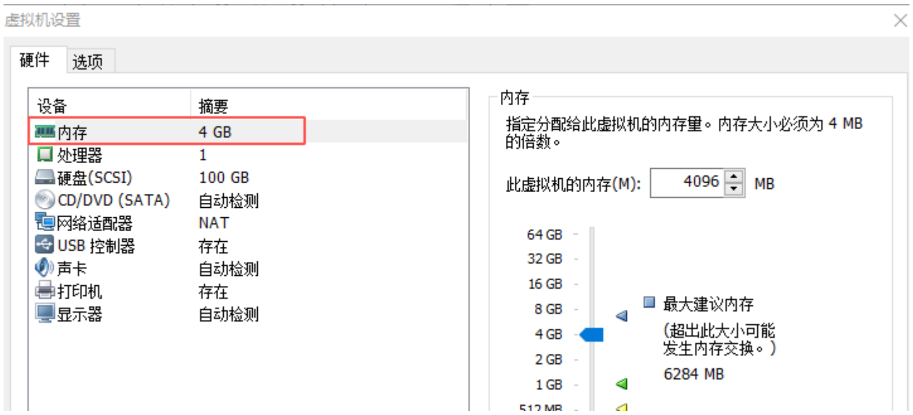
```
                    GNU GRUB  version 2.04

 Ubuntu, with Linux 5.4.0-58-generic
 Ubuntu, with Linux 5.4.0-58-generic (recovery mode)
 Ubuntu, with Linux 5.4.0-42-generic
 Ubuntu, with Linux 5.4.0-42-generic (recovery mode)
*Ubuntu, with Linux 4.14.201
 Ubuntu, with Linux 4.14.201 (recovery mode)
```

- 可能会因内存过小而失败，可把内存调大。



```
[    1.038408]  kernel_init+0xe/0x101
[    1.038437]  ret_from_fork+0x35/0x40
[    1.038629] ---[ end Kernel panic - not syncing: Out of memory and no killabl
e processes...
[    1.038629]
_
```



虚拟机设置

设备	摘要
内存	4 GB
处理器	1
硬盘(SCSI)	100 GB
CD/DVD (SATA)	自动检测
网络适配器	NAT
USB 控制器	存在
声卡	自动检测
打印机	存在
显示器	自动检测

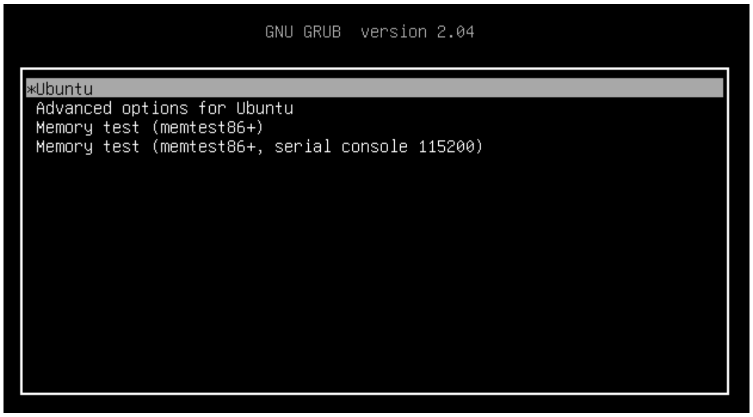
内存

指定分配给此虚拟机的内存量。内存大小必须为 4 MB 的倍数。

此虚拟机的内存(M): 4096 MB

■ 最大建议内存
(超出此大小可能发生内存交换。)
6284 MB

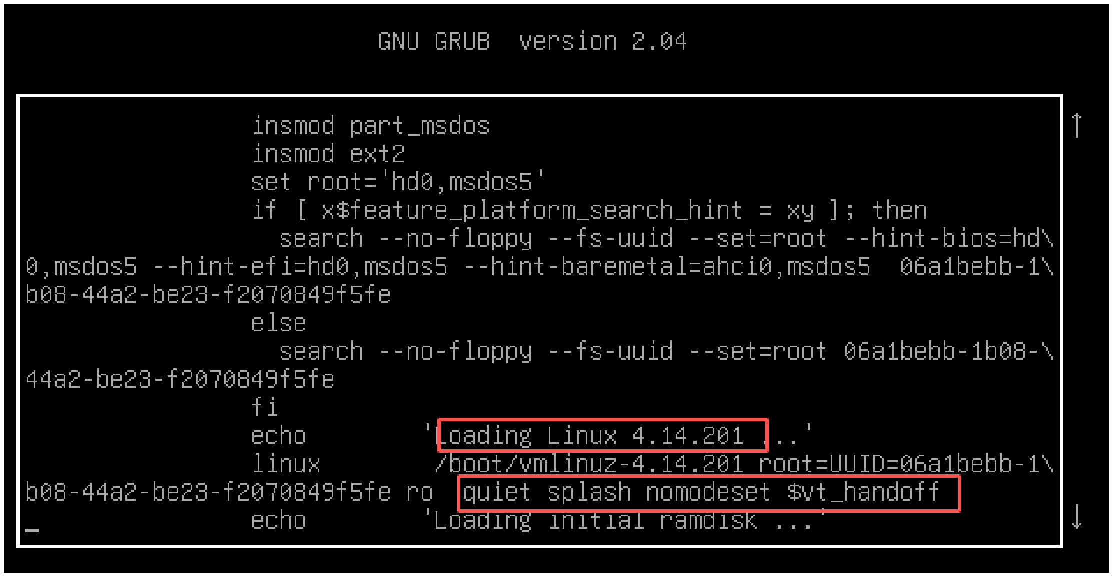
- 可能会出现长时间卡在紫屏的问题，是显卡的问题



```
                GNU GRUB  version 2.04

*Ubuntu
 Advanced options for Ubuntu
 Memory test (memtest86+)
 Memory test (memtest86+, serial console 115200)
```

选中第二个Advanced options for Ubuntu，按e，在对应的linux 版本splash后加nomodeset，按ctrl+x，选择要启动版本即可



```
                     GNU GRUB  version 2.04

                insmod part_msdos
                insmod ext2
                set root='hd0,msdos5'
                if [ x$feature_platform_search_hint = xy ]; then
                  search --no-floppy --fs-uuid --set=root --hint-bios=hd\
0,msdos5 --hint-efi=hd0,msdos5 --hint-baremetal=ahci0,msdos5  06a1bebb-1\
b08-44a2-be23-f2070849f5fe
                else
                  search --no-floppy --fs-uuid --set=root 06a1bebb-1b08-\
44a2-be23-f2070849f5fe
                fi
                echo        'Loading Linux 4.14.201 ...'
                linux        /boot/vmlinuz-4.14.201 root=UUID=06a1bebb-1\
b08-44a2-be23-f2070849f5fe ro  quiet splash nomodeset $vt_handoff
                echo        'Loading initial ramdisk ...'
```

问题解决

E:Could not get lock /var/lib/apt/lists/lock

Bash

```bash
ps -e | grep apt    #查看占用锁的进程
sudo killall apt    #把相关的进程都杀掉
sudo killall apt-get
sudo killall synaptic
```

下载source问题

Bash

```bash
sudo cp /etc/apt/sources.list /etc/apt/sources.list.bak    #备份一下
sudo vim /etc/apt/sources.list   #修改
```

清空/etc/apt/sources.list，将其替换为

1. 阿里源

Bash

```bash
deb http://mirrors.aliyun.com/ubuntu/ groovy main restricted universe multiverse
#deb-src http://mirrors.aliyun.com/ubuntu/ groovy main restricted universe multi
deb http://mirrors.aliyun.com/ubuntu/ groovy-security main restricted universe mu
#deb-src http://mirrors.aliyun.com/ubuntu/ groovy-security main restricted unive
deb http://mirrors.aliyun.com/ubuntu/ groovy-updates main restricted universe mu
#deb-src http://mirrors.aliyun.com/ubuntu/ groovy-updates main restricted univer
deb http://mirrors.aliyun.com/ubuntu/ groovy-backports main restricted universe
#deb-src http://mirrors.aliyun.com/ubuntu/ groovy-backports main restricted univ
# 预发布软件源，不建议启用
#deb http://mirrors.aliyun.com/ubuntu/ groovy-proposed main restricted universe
#deb-src http://mirrors.aliyun.com/ubuntu/ groovy-proposed main restricted unive
```

2. 清华源

Bash

```bash
# 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/ groovy main restricted universe
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ groovy main restricted un
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ groovy-updates main restricted
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ groovy-updates main restr
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ groovy-backports main restricte
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ groovy-backports main res
deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ groovy-security main restricted
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ groovy-security main rest
# 预发布软件源，不建议启用
# deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ groovy-proposed main restrict
# deb-src https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ groovy-proposed main rest
```

3. 官网源

Bash

```bash
deb http://cn.archive.ubuntu.com/ubuntu/ groovy main universe restricted multive
deb-src http://cn.archive.ubuntu.com/ubuntu/ groovy main universe restricted mul
deb http://security.ubuntu.com/ubuntu/ groovy-security main universe restricted
deb-src http://cn.archive.ubuntu.com/ubuntu/ groovy-security main universe restr
deb http://cn.archive.ubuntu.com/ubuntu/ groovy-updates main universe restricted
deb-src http://cn.archive.ubuntu.com/ubuntu/ groovy-updates main universe restric
```

修改之后，需使其生效。

Bash

```bash
apt-get update
apt-get upgrade
```