浙江大学实验报告

课程名称：操作系统分析及实验　　实验类型：综合型/设计性

实验项目名称：实验 1 重建 Linux 内核

学生姓名：卢佳盈　专业：计算机科学与技术　学号：3180103570

电子邮件地址：ljy28501@163.com　　手机：18868703211

实验日期：2020 年 11 月 5 日

一、实验目的

学习重新编译 Linux 内核，理解、掌握 Linux 内核和发行版本的区别。

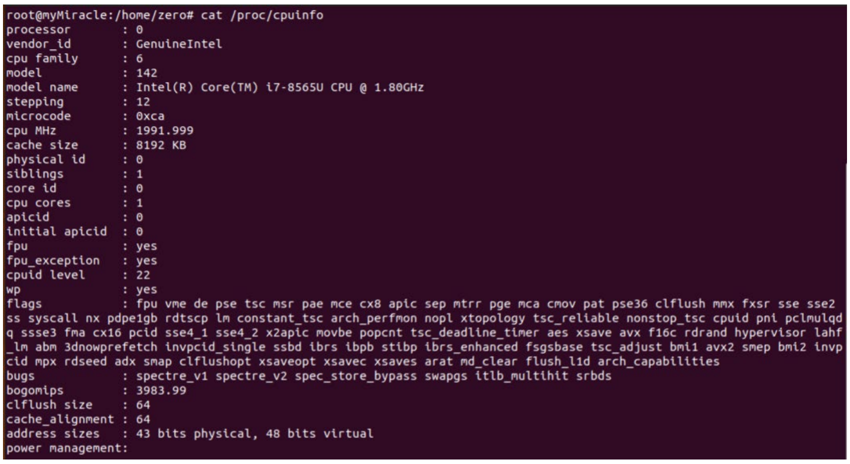
二、实验内容

重新编译内核是一件比你想像的还要简单的事情，它甚至不需要你对内核有任何的了解，只要你具备一些基本的 Linux 操作系统的知识就可以进行。

本次实验，要求你在 RedHat Fedora Core 5 的 Linux 系统里，下载并重新编译其内核源代码；然后，配置 GNU 的启动引导工具 grub，成功运行你刚刚编译成功的 Linux 内核。

三、主要仪器设备（必填）

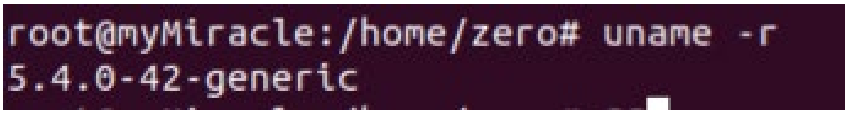
计算机配置：命令为 #cat /proc/cpuinfo



```
root@myMiracle:/home/zero# cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 142
model name      : Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz
stepping        : 12
microcode       : 0xca
cpu MHz         : 1991.999
cache size      : 8192 KB
physical id     : 0
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 0
initial apicid  : 0
fpu             : yes
fpu_exception   : yes
cpuid level     : 22
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2
ss syscall nx pdpe1gb rdtscp lm constant_tsc arch_perfmon nopl xtopology tsc_reliable nonstop_tsc cpuid pni pclmulqd
q ssse3 fma cx16 pcid sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand hypervisor lahf
_lm abm 3dnowprefetch invpcid_single ssbd ibrs ibpb stibp ibrs_enhanced fsgsbase tsc_adjust bmi1 avx2 smep bmi2 invp
cid mpx rdseed adx smap clflushopt xsaveopt xsavec xsaves arat md_clear flush_l1d arch_capabilities
bugs            : spectre_v1 spectre_v2 spec_store_bypass swapgs itlb_multihit srbds
bogomips        : 3983.99
clflush size    : 64
cache_alignment : 64
address sizes   : 43 bits physical, 48 bits virtual
power management:
```

LINUX 版本：ubuntu 20.04.1

原 KERNEL 版本：命令为#uname -r，版本号为 5.4.0-42-generic



```
root@myMiracle:/home/zero# uname -r
5.4.0-42-generic
```

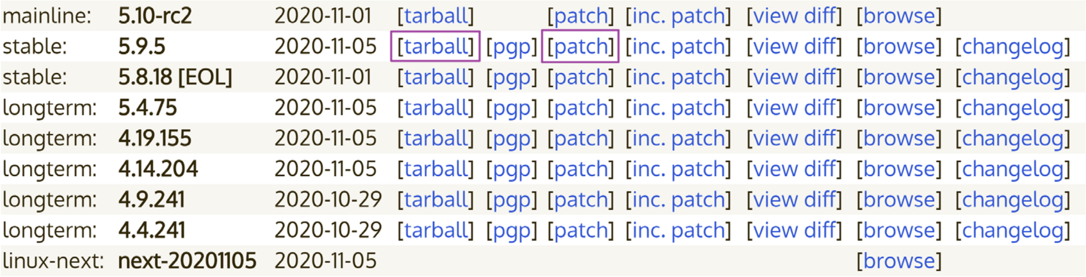
四、操作方法和实验步骤

1．查找并且下载一份内核源代码

在 Linux 的官方网站：www.kernel.org ，下载内核版本 5.9.5。

-linux-5.9.5.tar.xz

-patch-5.9.5.xz



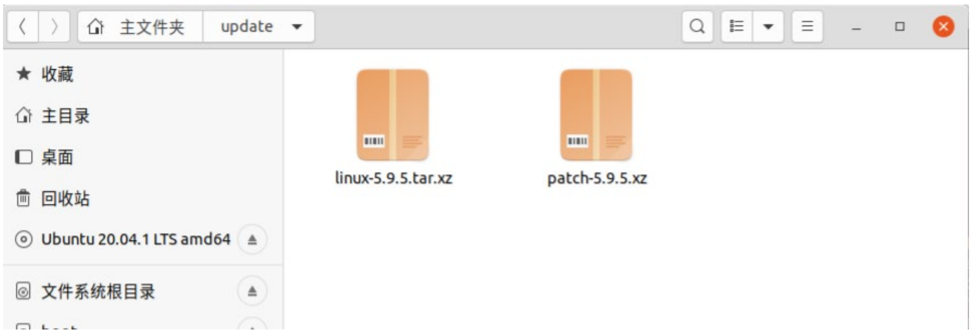
mainline:	5.10-rc2	2020-11-01	[tarball]		[patch]	[inc. patch]	[view diff]	[browse]	
stable:	5.9.5	2020-11-05	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
stable:	5.8.18 [EOL]	2020-11-01	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	5.4.75	2020-11-05	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.19.155	2020-11-05	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.14.204	2020-11-05	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.9.241	2020-10-29	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.4.241	2020-10-29	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	next-20201105	2020-11-05						[browse]	

在 root 文件夹下新建 update 文件夹，并将刚刚下载的两个文件移入

#cd /root

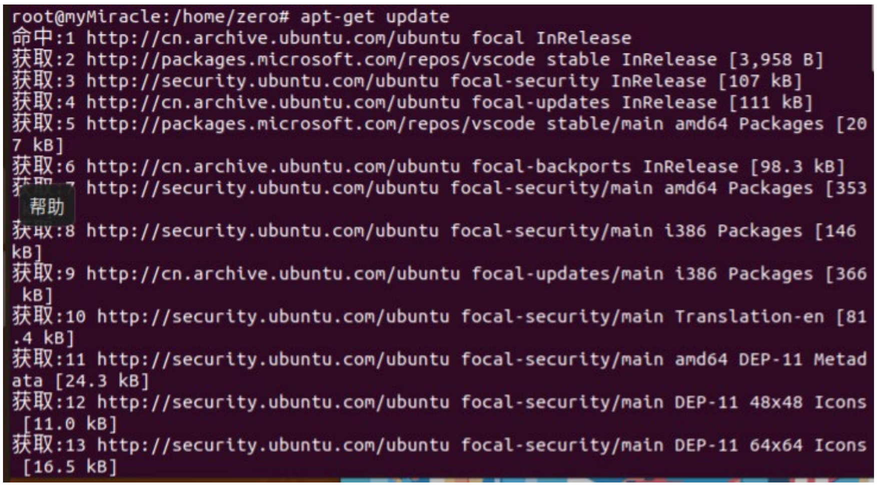
#mkdir update

#sudo nautilus



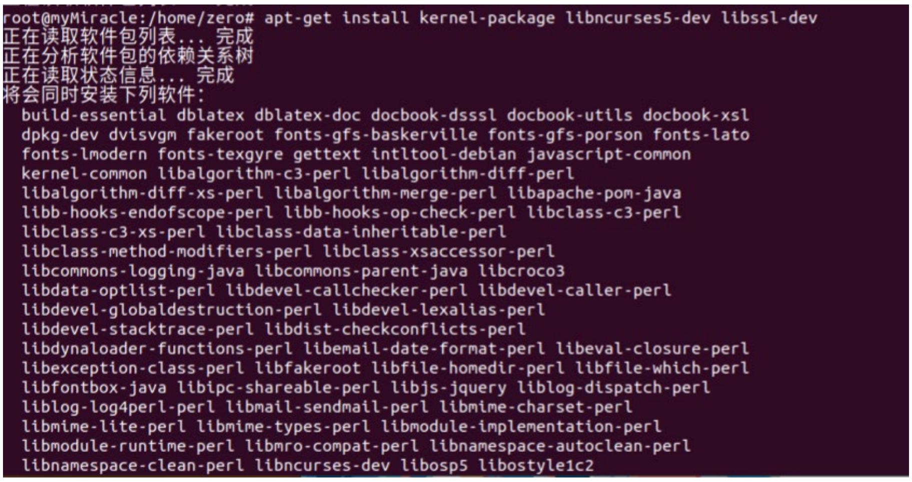
2．部署内核源代码

1）执行#apt-get update 来更新软件源



```
root@myMiracle:/home/zero# apt-get update
命中:1 http://cn.archive.ubuntu.com/ubuntu focal InRelease
获取:2 http://packages.microsoft.com/repos/vscode stable InRelease [3,958 B]
获取:3 http://security.ubuntu.com/ubuntu focal-security InRelease [107 kB]
获取:4 http://cn.archive.ubuntu.com/ubuntu focal-updates InRelease [111 kB]
获取:5 http://packages.microsoft.com/repos/vscode stable/main amd64 Packages [20
7 kB]
获取:6 http://cn.archive.ubuntu.com/ubuntu focal-backports InRelease [98.3 kB]
获取:7 http://security.ubuntu.com/ubuntu focal-security/main amd64 Packages [353
获取:8 http://security.ubuntu.com/ubuntu focal-security/main i386 Packages [146
kB]
获取:9 http://cn.archive.ubuntu.com/ubuntu focal-updates/main i386 Packages [366
 kB]
获取:10 http://security.ubuntu.com/ubuntu focal-security/main Translation-en [81
.4 kB]
获取:11 http://security.ubuntu.com/ubuntu focal-security/main amd64 DEP-11 Metad
ata [24.3 kB]
获取:12 http://security.ubuntu.com/ubuntu focal-security/main DEP-11 48x48 Icons
 [11.0 kB]
获取:13 http://security.ubuntu.com/ubuntu focal-security/main DEP-11 64x64 Icons
 [16.5 kB]
```

2）执行#apt-get install kernel-package libncurses5-dev libssl-dev 安装需要的环境



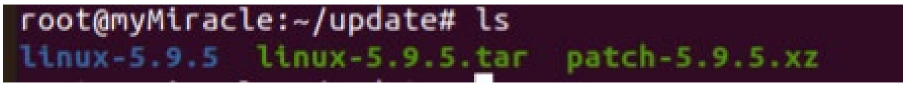
```
root@myMiracle:/home/zero# apt-get install kernel-package libncurses5-dev libssl-dev
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
将会同时安装下列软件:
  build-essential dblatex dblatex-doc docbook-dsssl docbook-utils docbook-xsl
  dpkg-dev dvisvgm fakeroot fonts-gfs-baskerville fonts-gfs-porson fonts-lato
  fonts-lmodern fonts-texgyre gettext intltool-debian javascript-common
  kernel-common libalgorithm-c3-perl libalgorithm-diff-perl
  libalgorithm-diff-xs-perl libalgorithm-merge-perl libapache-pom-java
  libb-hooks-endofscope-perl libb-hooks-op-check-perl libclass-c3-perl
  libclass-c3-xs-perl libclass-data-inheritable-perl
  libclass-method-modifiers-perl libclass-xsaccessor-perl
  libcommons-logging-java libcommons-parent-java libcroco3
  libdata-optlist-perl libdevel-callchecker-perl libdevel-caller-perl
  libdevel-globaldestruction-perl libdevel-lexalias-perl
  libdevel-stacktrace-perl libdist-checkconflicts-perl
  libdynaloader-functions-perl libemail-date-format-perl libeval-closure-perl
  libexception-class-perl libfakeroot libfile-homedir-perl libfile-which-perl
  libfontbox-java libipc-shareable-perl libjs-jquery liblog-dispatch-perl
  liblog-log4perl-perl libmail-sendmail-perl libmime-charset-perl
  libmime-lite-perl libmime-types-perl libmodule-implementation-perl
  libmodule-runtime-perl libmro-compat-perl libnamespace-autoclean-perl
  libnamespace-clean-perl libncurses-dev libosp5 libostyle1c2
```

c）在工作目录下解压缩内核并执行

# xz -d linux-5.9.5.tar.xz

# tar xvf linux-5.9.1.tar

完成后工作目录如下：



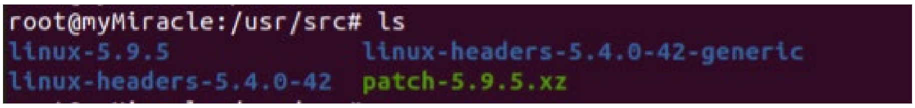
```
root@myMiracle:~/update# ls
linux-5.9.5  linux-5.9.5.tar  patch-5.9.5.xz
```

d）把工作目录下的 linux-5.9.5 和补丁 patch-5.9.5.xz 复制到/user/src 文件夹

#cp linux-5.9.5 /usr/src -rf

#cp patch-5.9.5.xz /usr/src

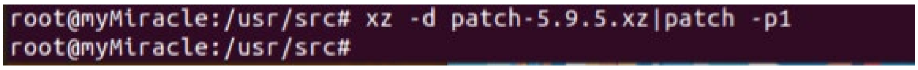
#cd /usr/src



```
root@myMiracle:/usr/src# ls
linux-5.9.5            linux-headers-5.4.0-42-generic
linux-headers-5.4.0-42  patch-5.9.5.xz
```

e）打内核补丁

xz -d patch-5.9.1.xz | patch -p1



```
root@myMiracle:/usr/src# xz -d patch-5.9.5.xz|patch -p1
root@myMiracle:/usr/src#
```

f）创建符号链接，从而可以通过/usr/src/linux 访问它

ln -s /usr/src/linux-5.9.1/ linux

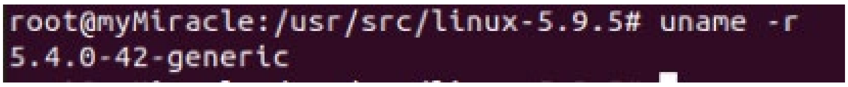
3．配置内核

在编译内核前，一般来说都需要对内核进行相应的配置。配置是精确控制新内核功能的机会。配置过程也控制哪些需编译到内核的二进制映像中(在启动时被载入)，哪些是需要时才装入的内核模块（module）。

第一次编译的话，有必要将内核源代码树置于一种完整和一致的状态。因此，我们推荐执行命令 make mrproper。它将清除目录下所有配置文件和先前生成核心时产生的.o 文件：

#make mrproper

用 uname –r 命令查看当前的内核版本号，得知为 5.4.0-42-generic

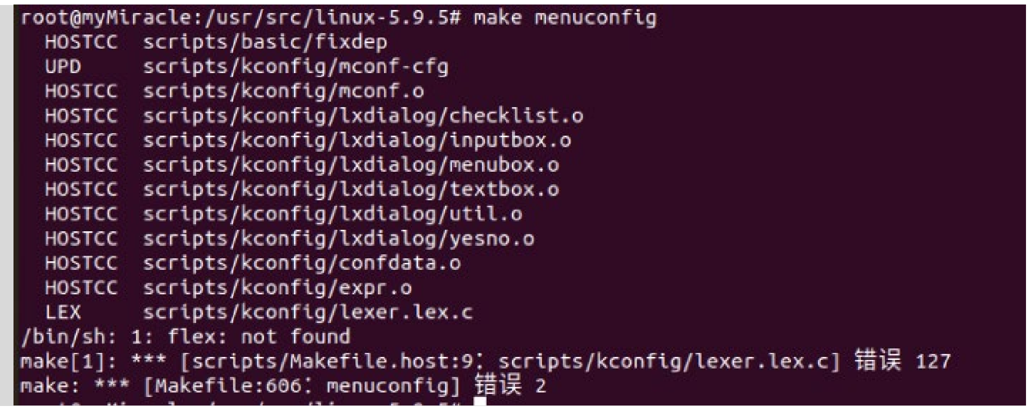


```
root@myMiracle:/usr/src/linux-5.9.5# uname -r
5.4.0-42-generic
```

把/usr/src/linux-headers-5.4.0-42-generic 里面的.config 文件复制到 linux-5.9.5 文件夹中:

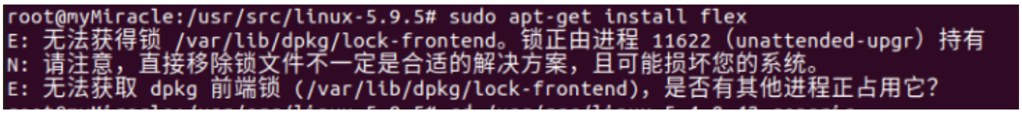
# cp /usr/src/linux-headers-5.4.0-42-generic/.config /usr/src/linux-5.9.5/

执行 make menuconfig



```
root@myMiracle:/usr/src/linux-5.9.5# make menuconfig
  HOSTCC  scripts/basic/fixdep
  UPD     scripts/kconfig/mconf-cfg
  HOSTCC  scripts/kconfig/mconf.o
  HOSTCC  scripts/kconfig/lxdialog/checklist.o
  HOSTCC  scripts/kconfig/lxdialog/inputbox.o
  HOSTCC  scripts/kconfig/lxdialog/menubox.o
  HOSTCC  scripts/kconfig/lxdialog/textbox.o
  HOSTCC  scripts/kconfig/lxdialog/util.o
  HOSTCC  scripts/kconfig/lxdialog/yesno.o
  HOSTCC  scripts/kconfig/confdata.o
  HOSTCC  scripts/kconfig/expr.o
  LEX     scripts/kconfig/lexer.lex.c
/bin/sh: 1: flex: not found
make[1]: *** [scripts/Makefile.host:9: scripts/kconfig/lexer.lex.c] 错误 127
make: *** [Makefile:606: menuconfig] 错误 2
```

产生报错 error127 与 error2，使用 sudo apt-get install flex 进行安装 flex，发生 lock-frontend 锁错误



```
root@myMiracle:/usr/src/linux-5.9.5# sudo apt-get install flex
E: 无法获得锁 /var/lib/dpkg/lock-frontend。锁正由进程 11622（unattended-upgr）持有
N: 请注意，直接移除锁文件不一定是合适的解决方案，且可能损坏您的系统。
E: 无法获取 dpkg 前端锁 (/var/lib/dpkg/lock-frontend)，是否有其他进程正占用它？
```

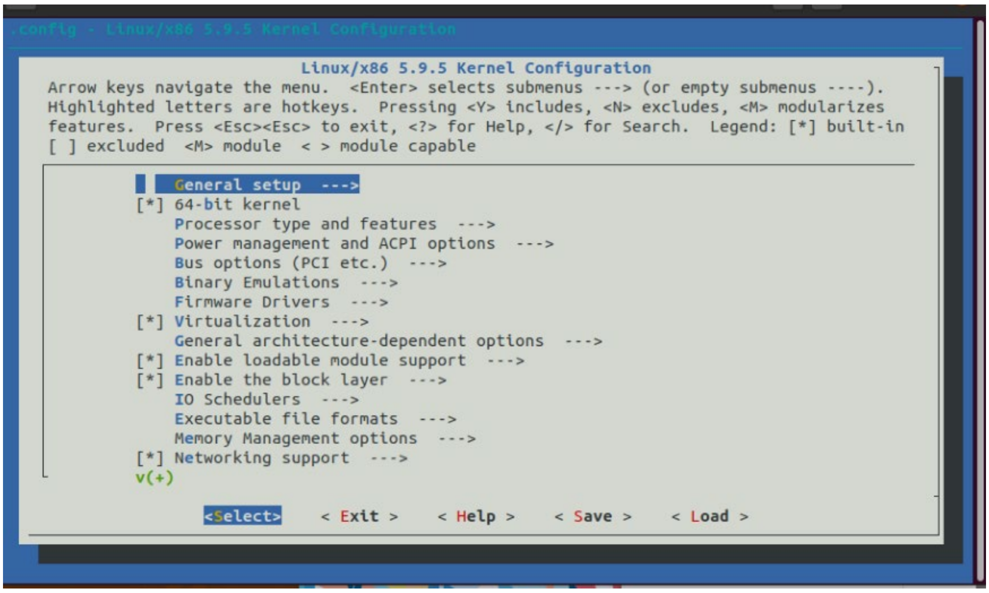
使用如下命令杀死该锁

#sudo rm /var/lib/dpkg/lock-frontend

#sudo rm /var/cache/apt/archives/lock

#sudo rm /var/lib/dpkg/lock

完成后重新下载 flex，并执行#sudo apt-get install flex

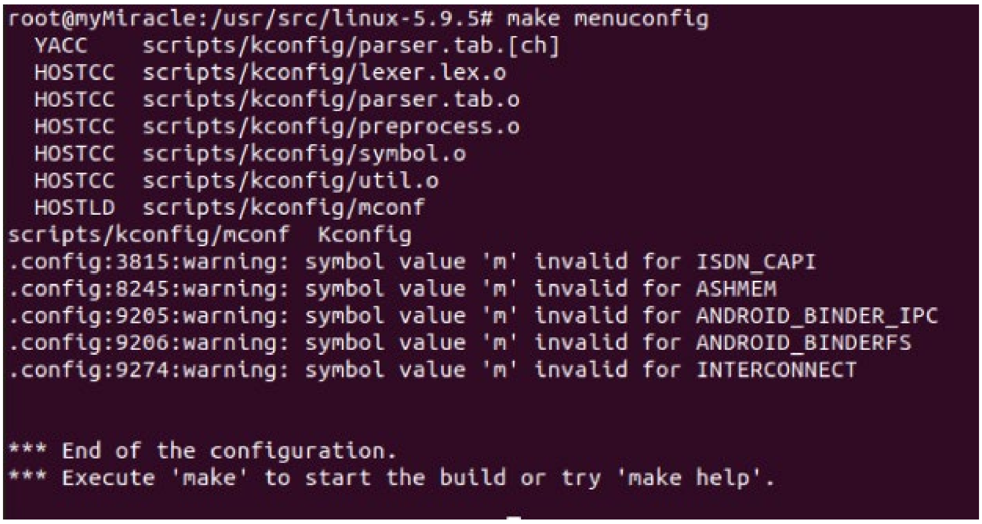


```
.config - Linux/x86 5.9.5 Kernel Configuration
                     Linux/x86 5.9.5 Kernel Configuration
 Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus ----).
 Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
 features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in
 [ ] excluded  <M> module  < > module capable

            General setup  --->
        [*] 64-bit kernel
            Processor type and features  --->
            Power management and ACPI options  --->
            Bus options (PCI etc.)  --->
            Binary Emulations  --->
            Firmware Drivers  --->
        [*] Virtualization  --->
            General architecture-dependent options  --->
        [*] Enable loadable module support  --->
        [*] Enable the block layer  --->
            IO Schedulers  --->
            Executable file formats  --->
            Memory Management options  --->
        [*] Networking support  --->
        v(+)

         <Select>    < Exit >    < Help >    < Save >    < Load >
```

依次选择 load→OK→Save→OK→EXIT→EXIT，得到更新的.config 文件。



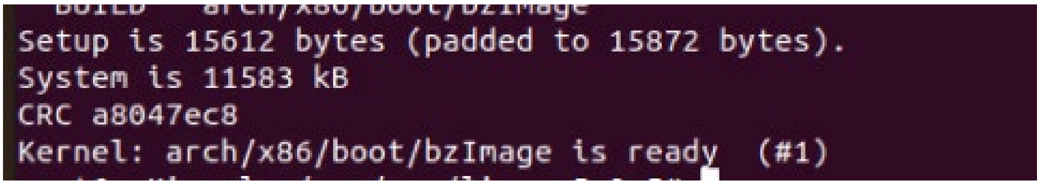
```
root@myMiracle:/usr/src/linux-5.9.5# make menuconfig
  YACC    scripts/kconfig/parser.tab.[ch]
  HOSTCC  scripts/kconfig/lexer.lex.o
  HOSTCC  scripts/kconfig/parser.tab.o
  HOSTCC  scripts/kconfig/preprocess.o
  HOSTCC  scripts/kconfig/symbol.o
  HOSTCC  scripts/kconfig/util.o
  HOSTLD  scripts/kconfig/mconf
scripts/kconfig/mconf  Kconfig
.config:3815:warning: symbol value 'm' invalid for ISDN_CAPI
.config:8245:warning: symbol value 'm' invalid for ASHMEM
.config:9205:warning: symbol value 'm' invalid for ANDROID_BINDER_IPC
.config:9206:warning: symbol value 'm' invalid for ANDROID_BINDERFS
.config:9274:warning: symbol value 'm' invalid for INTERCONNECT


*** End of the configuration.
*** Execute 'make' to start the build or try 'make help'.
```

4．编译内核和模块

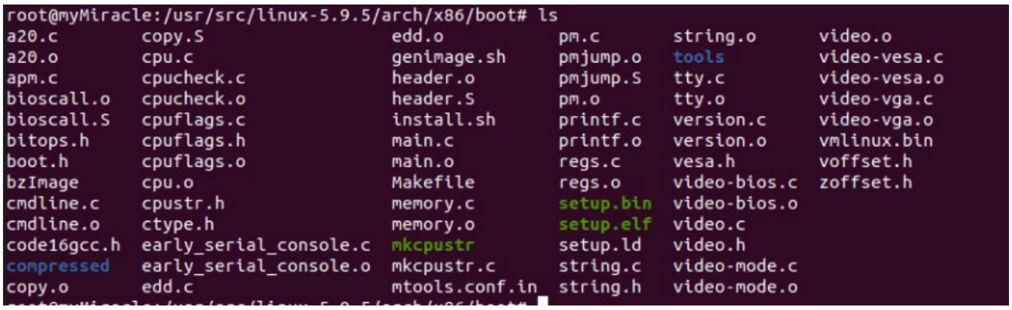
使用#make　bzImage　-jN 命令编译内核，其中虚拟机主机是 2 核 CPU

#make　bzImage　-j4



```
  BUILD   arch/x86/boot/bzImage
Setup is 15612 bytes (padded to 15872 bytes).
System is 11583 kB
CRC a8047ec8
Kernel: arch/x86/boot/bzImage is ready  (#1)
```

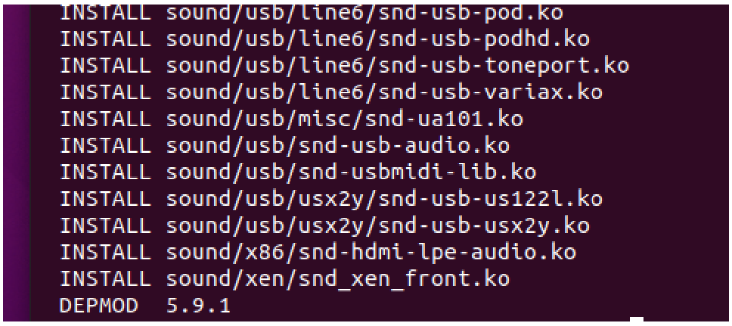
完成后产生的内核文件 bzImage 的位置在/usr/src/linux/arch/x86/boot 目录下，用户的 CPU 是 Intel x86 型的，内核源代码放在/usr/src/linux 目录下。



```
root@myMiracle:/usr/src/linux-5.9.5/arch/x86/boot# ls
a20.c        copy.S                  edd.o           pm.c       string.o       video.o
a20.o        cpu.c                   genimage.sh     pmjump.o   tools          video-vesa.c
apm.c        cpucheck.c              header.o        pmjump.S   tty.c          video-vesa.o
bioscall.o   cpucheck.o              header.S        pm.o       tty.o          video-vga.c
bioscall.S   cpuflags.c              install.sh      printf.c   version.c      video-vga.o
bitops.h     cpuflags.h              main.c          printf.o   version.o      vmlinux.bin
boot.h       cpuflags.o              main.o          regs.c     vesa.h         voffset.h
bzImage      cpu.o                   Makefile        regs.o     video-bios.c   zoffset.h
cmdline.c    cpustr.h                memory.c        setup.bin  video-bios.o
cmdline.o    ctype.h                 memory.o        setup.elf  video.c
code16gcc.h  early_serial_console.c  mkcpustr        setup.ld   video.h
compressed   early_serial_console.o  mkcpustr.c      string.c   video-mode.c
copy.o       edd.c                   mtools.conf.in  string.h   video-mode.o
```

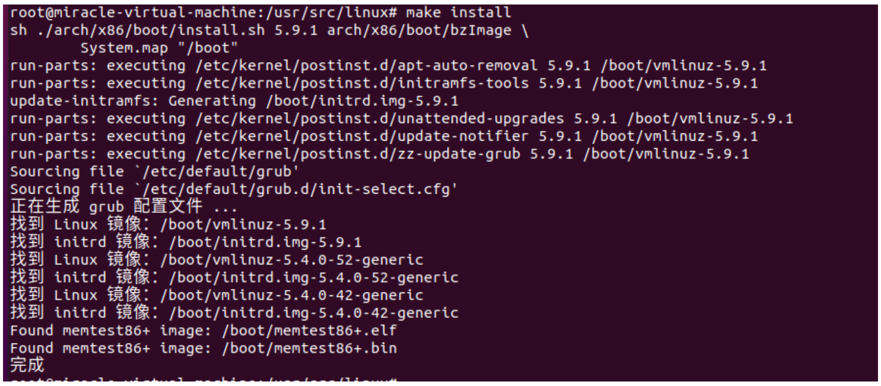
如果选择了可加载模块，编译完内核后，要对需要的模块进行编译：#make modules -j8

安装模块：#make modules_install



```
  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
  DEPMOD  5.9.1
```

安装内核：#make install

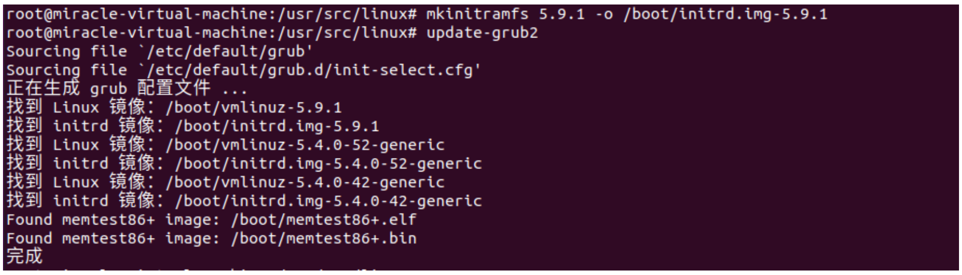


```
root@miracle-virtual-machine:/usr/src/linux# make install
sh ./arch/x86/boot/install.sh 5.9.1 arch/x86/boot/bzImage \
        System.map "/boot"
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 5.9.1 /boot/vmlinuz-5.9.1
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 5.9.1 /boot/vmlinuz-5.9.1
update-initramfs: Generating /boot/initrd.img-5.9.1
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 5.9.1 /boot/vmlinuz-5.9.1
run-parts: executing /etc/kernel/postinst.d/update-notifier 5.9.1 /boot/vmlinuz-5.9.1
run-parts: executing /etc/kernel/postinst.d/zz-update-grub 5.9.1 /boot/vmlinuz-5.9.1
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
正在生成 grub 配置文件 ...
找到 Linux 镜像： /boot/vmlinuz-5.9.1
找到 initrd 镜像： /boot/initrd.img-5.9.1
找到 Linux 镜像： /boot/vmlinuz-5.4.0-52-generic
找到 initrd 镜像： /boot/initrd.img-5.4.0-52-generic
找到 Linux 镜像： /boot/vmlinuz-5.4.0-42-generic
找到 initrd 镜像： /boot/initrd.img-5.4.0-42-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
完成
```

5．应用 grub 配置启动文件

#mkinitramfs 5.9.1 -o /boot/initrd.img-5.9.1

#update-grub2

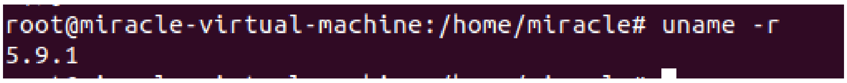


```
root@miracle-virtual-machine:/usr/src/linux# mkinitramfs 5.9.1 -o /boot/initrd.img-5.9.1
root@miracle-virtual-machine:/usr/src/linux# update-grub2
Sourcing file `/etc/default/grub'
Sourcing file `/etc/default/grub.d/init-select.cfg'
正在生成 grub 配置文件 ...
找到 Linux 镜像： /boot/vmlinuz-5.9.1
找到 initrd 镜像： /boot/initrd.img-5.9.1
找到 Linux 镜像： /boot/vmlinuz-5.4.0-52-generic
找到 initrd 镜像： /boot/initrd.img-5.4.0-52-generic
找到 Linux 镜像： /boot/vmlinuz-5.4.0-42-generic
找到 initrd 镜像： /boot/initrd.img-5.4.0-42-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
完成
```

update-grub2 命令会帮我们自动修改 grub。

完成编译内核 bzImage (vmlinuz-5.9.1)，放到了指定位置/boot；initrd.img-5.9.1、System.map-5.9.1 等辅助资源，同样放到了/boot；我们也配置了 grub。

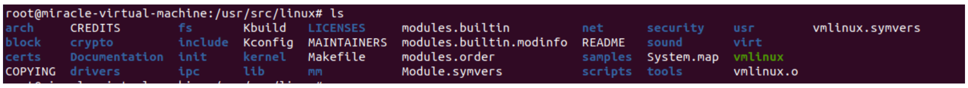
重启主机，发现主机内核版本已经被修改：



```
root@miracle-virtual-machine:/home/miracle# uname -r
5.9.1
```

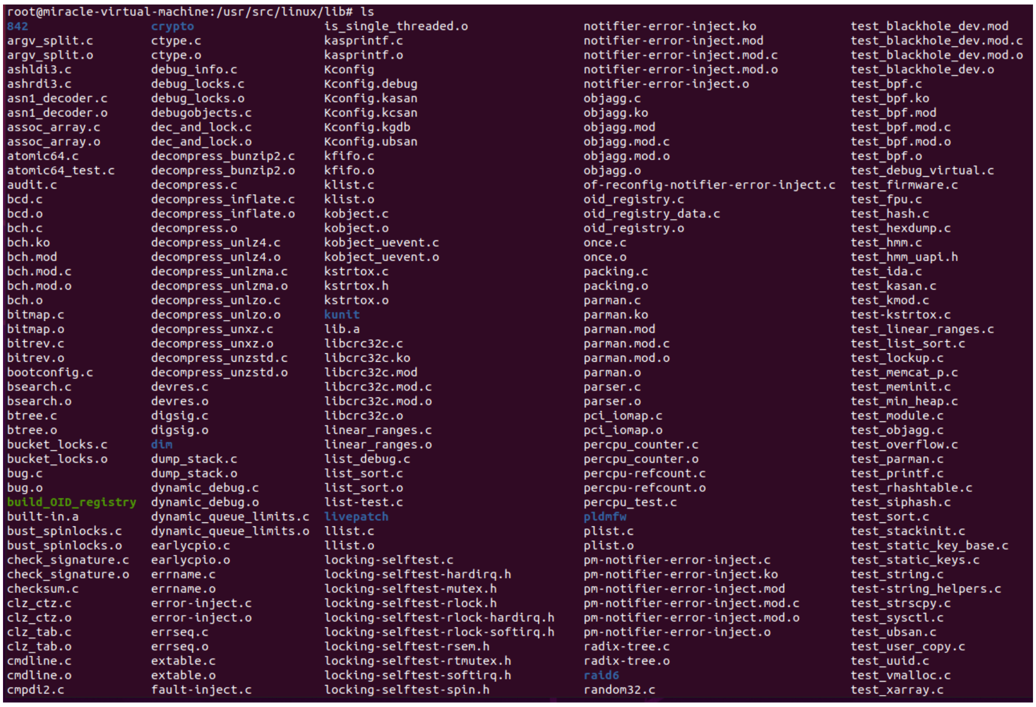
五、实验结果和分析

内核压缩包：

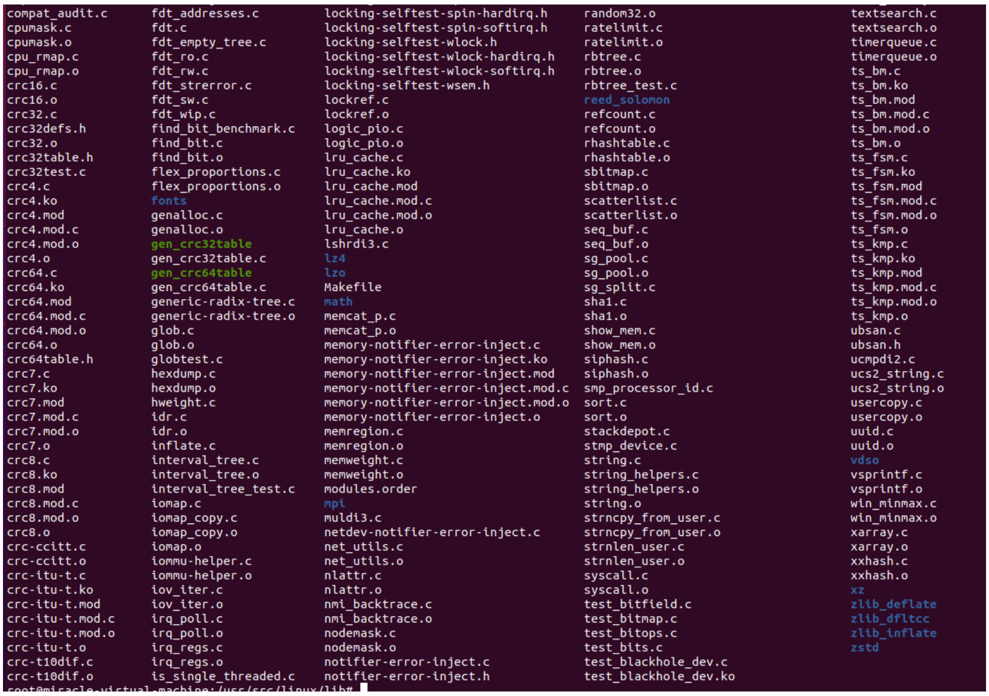


```
root@miracle-virtual-machine:/usr/src/linux# ls
arch     CREDITS        fs       Kbuild   LICENSES     modules.builtin          net      security    usr        vmlinux.symvers
block    crypto         include  Kconfig  MAINTAINERS  modules.builtin.modinfo  README   sound       virt
certs    Documentation  init     kernel   Makefile     modules.order            samples  System.map  vmlinux
COPYING  drivers        ipc      lib      mm           Module.symvers           scripts  tools       vmlinux.o
```

lib 文件夹下：



```
root@miracle-virtual-machine:/usr/src/linux/lib# ls
842                 crypto                  is_single_threaded.o                notifier-error-inject.ko             test_blackhole_dev.mod
argv_split.c        ctype.c                 kasprintf.c                         notifier-error-inject.mod            test_blackhole_dev.mod.c
argv_split.o        ctype.o                 kasprintf.o                         notifier-error-inject.mod.c          test_blackhole_dev.mod.o
ashldi3.c           debug_info.c            Kconfig                             notifier-error-inject.mod.o          test_blackhole_dev.o
ashrdi3.c           debug_locks.c           Kconfig.debug                       notifier-error-inject.o              test_bpf.c
asn1_decoder.c      debug_locks.o           Kconfig.kasan                       objagg.c                             test_bpf.ko
asn1_decoder.o      debugobjects.c          Kconfig.kcsan                       objagg.ko                            test_bpf.mod
assoc_array.c       dec_and_lock.c          Kconfig.kgdb                        objagg.mod                           test_bpf.mod.c
assoc_array.o       dec_and_lock.o          Kconfig.ubsan                       objagg.mod.c                         test_bpf.mod.o
atomic64.c          decompress_bunzip2.c    kfifo.c                             objagg.mod.o                         test_bpf.o
atomic64_test.c     decompress_bunzip2.o    kfifo.o                             objagg.o                             test_debug_virtual.c
audit.c             decompress.c            klist.c                             of-reconfig-notifier-error-inject.c  test_firmware.c
bcd.c               decompress_inflate.c    klist.o                             oid_registry.c                       test_fpu.c
bcd.o               decompress_inflate.o    kobject.c                           oid_registry_data.c                  test_hash.c
bch.c               decompress.o            kobject.o                           oid_registry.o                       test_hexdump.c
bch.ko              decompress_unlz4.c      kobject_uevent.c                    once.c                               test_hmm.c
bch.mod             decompress_unlz4.o      kobject_uevent.o                    once.o                               test_hmm_uapi.h
bch.mod.c           decompress_unlzma.c     kstrtox.c                           packing.c                            test_ida.c
bch.mod.o           decompress_unlzma.o     kstrtox.h                           packing.o                            test_kasan.c
bch.o               decompress_unlzo.c      kstrtox.o                           parman.c                             test_kmod.c
bitmap.c            decompress_unlzo.o      kunit                               parman.ko                            test-kstrtox.c
bitmap.o            decompress_unxz.c       lib.a                               parman.mod                           test_linear_ranges.c
bitrev.c            decompress_unxz.o       libcrc32c.c                         parman.mod.c                         test_list_sort.c
bitrev.o            decompress_unzstd.c     libcrc32c.ko                        parman.mod.o                         test_lockup.c
bootconfig.c        decompress_unzstd.o     libcrc32c.mod                       parman.o                             test_memcat_p.c
bsearch.c           devres.c                libcrc32c.mod.c                     parser.c                             test_meminit.c
bsearch.o           devres.o                libcrc32c.mod.o                     parser.o                             test_min_heap.c
btree.c             digsig.c                libcrc32c.o                         pci_iomap.c                          test_module.c
btree.o             digsig.o                linear_ranges.c                     pci_iomap.o                          test_objagg.c
bucket_locks.c      dim                     linear_ranges.o                     percpu_counter.c                     test_overflow.c
bucket_locks.o      dump_stack.c            list_debug.c                        percpu_counter.o                     test_parman.c
bug.c               dump_stack.o            list_sort.c                         percpu-refcount.c                    test_printf.c
bug.o               dynamic_debug.c         list_sort.o                         percpu-refcount.o                    test_rhashtable.c
build_OID_registry  dynamic_debug.o         list-test.c                         percpu_test.c                        test_siphash.c
built-in.a          dynamic_queue_limits.c  livepatch                           pldmfw                               test_sort.c
bust_spinlocks.c    dynamic_queue_limits.o  llist.c                             plist.c                              test_stackinit.c
bust_spinlocks.o    earlycpio.c             llist.o                             plist.o                              test_static_key_base.c
check_signature.c   earlycpio.o             locking-selftest.c                  pm-notifier-error-inject.c           test_static_keys.c
check_signature.o   errname.c               locking-selftest-hardirq.h          pm-notifier-error-inject.ko          test_string.c
checksum.c          errname.o               locking-selftest-mutex.h            pm-notifier-error-inject.mod         test-string_helpers.c
clz_ctz.c           error-inject.c          locking-selftest-rlock.h            pm-notifier-error-inject.mod.c       test_strscpy.c
clz_ctz.o           error-inject.o          locking-selftest-rlock-hardirq.h    pm-notifier-error-inject.mod.o       test_sysctl.c
clz_tab.c           errseq.c                locking-selftest-rlock-softirq.h    pm-notifier-error-inject.o           test_ubsan.c
clz_tab.o           errseq.o                locking-selftest-rsem.h             radix-tree.c                         test_user_copy.c
cmdline.c           extable.c               locking-selftest-rtmutex.h          radix-tree.o                         test_uuid.c
cmdline.o           extable.o               locking-selftest-softirq.h          raid6                                test_vmalloc.c
cmpdi2.c            fault-inject.c          locking-selftest-spin.h             random32.c                           test_xarray.c
```



```
compat_audit.c      fdt_addresses.c         locking-selftest-spin-hardirq.h     random32.o                           textsearch.c
cpumask.c           fdt.c                   locking-selftest-spin-softirq.h     ratelimit.c                          textsearch.o
cpumask.o           fdt_empty_tree.c        locking-selftest-wlock.h            ratelimit.o                          timerqueue.c
cpu_rmap.c          fdt_ro.c                locking-selftest-wlock-hardirq.h    rbtree.c                             timerqueue.o
cpu_rmap.o          fdt_rw.c                locking-selftest-wlock-softirq.h    rbtree.o                             ts_bm.c
crc16.c             fdt_strerror.c          locking-selftest-wsem.h             rbtree_test.c                        ts_bm.ko
crc16.o             fdt_sw.c                lockref.c                           reed_solomon                         ts_bm.mod
crc32.c             fdt_wip.c               lockref.o                           refcount.c                           ts_bm.mod.c
crc32defs.h         find_bit_benchmark.c    logic_pio.c                         refcount.o                           ts_bm.mod.o
crc32.o             find_bit.c              logic_pio.o                         rhashtable.c                         ts_bm.o
crc32table.h        find_bit.o              lru_cache.c                         rhashtable.o                         ts_fsm.c
crc32test.c         flex_proportions.c      lru_cache.ko                        sbitmap.c                            ts_fsm.ko
crc4.c              flex_proportions.o      lru_cache.mod                       sbitmap.o                            ts_fsm.mod
crc4.ko             fonts                   lru_cache.mod.c                     scatterlist.c                        ts_fsm.mod.c
crc4.mod            genalloc.c              lru_cache.mod.o                     scatterlist.o                        ts_fsm.mod.o
crc4.mod.c          genalloc.o              lru_cache.o                         seq_buf.c                            ts_fsm.o
crc4.mod.o          gen_crc32table          lshrdi3.c                           seq_buf.o                            ts_kmp.c
crc4.o              gen_crc32table.c        lz4                                 sg_pool.c                            ts_kmp.ko
crc64.c             gen_crc64table          lzo                                 sg_pool.o                            ts_kmp.mod
crc64.ko            gen_crc64table.c        Makefile                            sg_split.c                           ts_kmp.mod.c
crc64.mod           generic-radix-tree.c    math                                sha1.c                               ts_kmp.mod.o
crc64.mod.c         generic-radix-tree.o    memcat_p.c                          sha1.o                               ts_kmp.o
crc64.mod.o         glob.c                  memcat_p.o                          show_mem.c                           ubsan.c
crc64.o             glob.o                  memory-notifier-error-inject.c      show_mem.o                           ubsan.h
crc64table.h        globtest.c              memory-notifier-error-inject.ko     siphash.c                            ucmpdi2.c
crc7.c              hexdump.c               memory-notifier-error-inject.mod    siphash.o                            ucs2_string.c
crc7.ko             hexdump.o               memory-notifier-error-inject.mod.c  smp_processor_id.c                   ucs2_string.o
crc7.mod            hweight.c               memory-notifier-error-inject.mod.o  sort.c                               usercopy.c
crc7.mod.c          idr.c                   memory-notifier-error-inject.o      sort.o                               usercopy.o
crc7.mod.o          idr.o                   memregion.c                         stackdepot.c                         uuid.c
crc7.o              inflate.c               memregion.o                         stmp_device.c                        uuid.o
crc8.c              interval_tree.c         memweight.c                         string.c                             vdso
crc8.ko             interval_tree.o         memweight.o                         string_helpers.c                     vsprintf.c
crc8.mod            interval_tree_test.c    modules.order                       string_helpers.o                     vsprintf.o
crc8.mod.c          iomap.c                 mpi                                 string.o                             win_minmax.c
crc8.mod.o          iomap_copy.c            muldi3.c                            strncpy_from_user.c                  win_minmax.o
crc8.o              iomap_copy.o            netdev-notifier-error-inject.c      strncpy_from_user.o                  xarray.c
crc-ccitt.c         iomap.o                 net_utils.c                         strnlen_user.c                       xarray.o
crc-ccitt.o         iommu-helper.c          net_utils.o                         strnlen_user.o                       xxhash.c
crc-itu-t.c         iommu-helper.o          nlattr.c                            syscall.c                            xxhash.o
crc-itu-t.ko        iov_iter.c              nlattr.o                            syscall.o                            xz
crc-itu-t.mod       iov_iter.o              nmi_backtrace.c                     test_bitfield.c                      zlib_deflate
crc-itu-t.mod.c     irq_poll.c              nmi_backtrace.o                     test_bitmap.c                        zlib_dfltcc
crc-itu-t.mod.o     irq_poll.o              nodemask.c                          test_bitops.c                        zlib_inflate
crc-itu-t.o         irq_regs.c              nodemask.o                          test_bits.c                          zstd
crc-t10dif.c        irq_regs.o              notifier-error-inject.c             test_blackhole_dev.c
crc-t10dif.o        is_single_threaded.c    notifier-error-inject.h             test_blackhole_dev.ko
root@miracle-virtual-machine:/usr/src/linux/lib# 
```

grub.conf 文件内容如下：

```
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#

### BEGIN /etc/grub.d/00_header ###
if [ -s $prefix/grubenv ]; then
```

```
  set have_grubenv=true
  load_env
fi
if [ "${initrdfail}" = 2 ]; then
    set initrdfail=
elif [ "${initrdfail}" = 1 ]; then
    set next_entry="${prev_entry}"
    set prev_entry=
    save_env prev_entry
    if [ "${next_entry}" ]; then
        set initrdfail=2
    fi
fi
if [ "${next_entry}" ] ; then
    set default="${next_entry}"
    set next_entry=
    save_env next_entry
    set boot_once=true
else
    set default="0"
fi

if [ x"${feature_menuentry_id}" = xy ]; then
  menuentry_id_option="--id"
else
  menuentry_id_option=""
fi

export menuentry_id_option

if [ "${prev_saved_entry}" ]; then
  set saved_entry="${prev_saved_entry}"
  save_env saved_entry
  set prev_saved_entry=
  save_env prev_saved_entry
  set boot_once=true
fi

function savedefault {
  if [ -z "${boot_once}" ]; then
    saved_entry="${chosen}"
    save_env saved_entry
  fi
}
```

```
    function initrdfail {
        if [ -n "${have_grubenv}" ]; then if [ -n "${partuuid}" ]; then
          if [ -z "${initrdfail}" ]; then
            set initrdfail=1
            if [ -n "${boot_once}" ]; then
              set prev_entry="${default}"
              save_env prev_entry
            fi
          fi
          save_env initrdfail
        fi; fi
    }
    function recordfail {
      set recordfail=1
      if [ -n "${have_grubenv}" ]; then if [ -z "${boot_once}" ]; then save_env recordfail; fi; fi
    }
    function load_video {
      if [ x$feature_all_video_module = xy ]; then
        insmod all_video
      else
        insmod efi_gop
        insmod efi_uga
        insmod ieee1275_fb
        insmod vbe
        insmod vga
        insmod video_bochs
        insmod video_cirrus
      fi
    }

    if [ x$feature_default_font_path = xy ] ; then
       font=unicode
    else
    insmod part_msdos
    insmod ext2
    set root='hd0,msdos5'
    if [ x$feature_platform_search_hint = xy ]; then
      search  --no-floppy  --fs-uuid  --set=root  --hint-bios=hd0,msdos5  --hint-efi=hd0,msdos5
--hint-baremetal=ahci0,msdos5    6b150898-d356-4343-a456-81b423e8511a
    else
      search --no-floppy --fs-uuid --set=root 6b150898-d356-4343-a456-81b423e8511a
    fi
        font="/usr/share/grub/unicode.pf2"
    fi
```

```
if loadfont $font ; then
  set gfxmode=auto
  load_video
  insmod gfxterm
  set locale_dir=$prefix/locale
  set lang=zh_CN
  insmod gettext
fi
terminal_output gfxterm
if [ "${recordfail}" = 1 ] ; then
  set timeout=30
else
  if [ x$feature_timeout_style = xy ] ; then
    set timeout_style=hidden
    set timeout=0
  # Fallback hidden-timeout code in case the timeout_style feature is
  # unavailable.
  elif sleep --interruptible 0 ; then
    set timeout=0
  fi
fi
### END /etc/grub.d/00_header ###

### BEGIN /etc/grub.d/05_debian_theme ###
set menu_color_normal=white/black
set menu_color_highlight=black/light-gray
### END /etc/grub.d/05_debian_theme ###

### BEGIN /etc/grub.d/10_linux ###
function gfxmode {
    set gfxpayload="${1}"
    if [ "${1}" = "keep" ]; then
        set vt_handoff=vt.handoff=7
    else
        set vt_handoff=
    fi
}
if [ "${recordfail}" != 1 ]; then
  if [ -e ${prefix}/gfxblacklist.txt ]; then
    if hwmatch ${prefix}/gfxblacklist.txt 3; then
      if [ ${match} = 0 ]; then
        set linux_gfx_mode=keep
      else
```

```
        set linux_gfx_mode=text
      fi
    else
      set linux_gfx_mode=text
    fi
  else
    set linux_gfx_mode=keep
  fi
else
  set linux_gfx_mode=text
fi
export linux_gfx_mode
menuentry 'Ubuntu' --class ubuntu --class gnu-linux --class gnu --class os $menuentry_id_option 'gnulinux-simple-6b150898-d356-4343-a456-81b423e8511a' {
    recordfail
    load_video
    gfxmode $linux_gfx_mode
    insmod gzio
    if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
    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=hd0,msdos5 --hint-efi=hd0,msdos5 --hint-baremetal=ahci0,msdos5  6b150898-d356-4343-a456-81b423e8511a
    else
      search --no-floppy --fs-uuid --set=root 6b150898-d356-4343-a456-81b423e8511a
    fi
    linux    /boot/vmlinuz-5.9.1 root=UUID=6b150898-d356-4343-a456-81b423e8511a ro  quiet splash $vt_handoff
    initrd    /boot/initrd.img-5.9.1
}
submenu 'Ubuntu 的高级选项' $menuentry_id_option 'gnulinux-advanced-6b150898-d356-4343-a456-81b423e8511a' {
    menuentry 'Ubuntu，Linux 5.9.1' --class ubuntu --class gnu-linux --class gnu --class os $menuentry_id_option 'gnulinux-5.9.1-advanced-6b150898-d356-4343-a456-81b423e8511a' {
        recordfail
        load_video
        gfxmode $linux_gfx_mode
        insmod gzio
        if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
        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=hd0,msdos5
--hint-efi=hd0,msdos5 --hint-baremetal=ahci0,msdos5
6b150898-d356-4343-a456-81b423e8511a
            else
              search --no-floppy --fs-uuid --set=root
6b150898-d356-4343-a456-81b423e8511a
            fi
            echo'载入 Linux 5.9.1 ...'
            linux    /boot/vmlinuz-5.9.1
root=UUID=6b150898-d356-4343-a456-81b423e8511a ro  quiet splash $vt_handoff
            echo'载入初始化内存盘...'
            initrd    /boot/initrd.img-5.9.1
        }
        menuentry 'Ubuntu, with Linux 5.9.1 (recovery mode)' --class ubuntu --class gnu-linux
--class gnu --class os $menuentry_id_option
'gnulinux-5.9.1-recovery-6b150898-d356-4343-a456-81b423e8511a' {
            recordfail
            load_video
            insmod gzio
            if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
            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=hd0,msdos5
--hint-efi=hd0,msdos5 --hint-baremetal=ahci0,msdos5
6b150898-d356-4343-a456-81b423e8511a
            else
              search --no-floppy --fs-uuid --set=root
6b150898-d356-4343-a456-81b423e8511a
            fi
            echo'载入 Linux 5.9.1 ...'
            linux    /boot/vmlinuz-5.9.1
root=UUID=6b150898-d356-4343-a456-81b423e8511a ro recovery nomodeset
            echo'载入初始化内存盘...'
            initrd    /boot/initrd.img-5.9.1
        }
        menuentry 'Ubuntu，Linux 5.4.0-52-generic' --class ubuntu --class gnu-linux --class gnu
--class os $menuentry_id_option
'gnulinux-5.4.0-52-generic-advanced-6b150898-d356-4343-a456-81b423e8511a' {
            recordfail
            load_video
            gfxmode $linux_gfx_mode
```

```
                insmod gzio
                if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
                insmod part_msdos
                insmod ext2
                set root='hd0,msdos5'
                if [ x$feature_platform_search_hint = xy ]; then
                  search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos5
--hint-efi=hd0,msdos5 --hint-baremetal=ahci0,msdos5
6b150898-d356-4343-a456-81b423e8511a
                else
                  search --no-floppy --fs-uuid --set=root
6b150898-d356-4343-a456-81b423e8511a
                fi
                echo '载入 Linux 5.4.0-52-generic ...'
                linux    /boot/vmlinuz-5.4.0-52-generic
root=UUID=6b150898-d356-4343-a456-81b423e8511a ro  quiet splash $vt_handoff
                echo '载入初始化内存盘...'
                initrd   /boot/initrd.img-5.4.0-52-generic
        }
        menuentry 'Ubuntu, with Linux 5.4.0-52-generic (recovery mode)' --class ubuntu --class
gnu-linux --class gnu --class os $menuentry_id_option
'gnulinux-5.4.0-52-generic-recovery-6b150898-d356-4343-a456-81b423e8511a' {
                recordfail
                load_video
                insmod gzio
                if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
                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=hd0,msdos5
--hint-efi=hd0,msdos5 --hint-baremetal=ahci0,msdos5
6b150898-d356-4343-a456-81b423e8511a
                else
                  search --no-floppy --fs-uuid --set=root
6b150898-d356-4343-a456-81b423e8511a
                fi
                echo '载入 Linux 5.4.0-52-generic ...'
                linux    /boot/vmlinuz-5.4.0-52-generic
root=UUID=6b150898-d356-4343-a456-81b423e8511a ro recovery nomodeset
                echo '载入初始化内存盘...'
                initrd   /boot/initrd.img-5.4.0-52-generic
        }
        menuentry 'Ubuntu，Linux 5.4.0-42-generic' --class ubuntu --class gnu-linux --class gnu
```

```
--class os $menuentry_id_option
'gnulinux-5.4.0-42-generic-advanced-6b150898-d356-4343-a456-81b423e8511a' {
		recordfail
		load_video
		gfxmode $linux_gfx_mode
		insmod gzio
		if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
		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=hd0,msdos5
--hint-efi=hd0,msdos5 --hint-baremetal=ahci0,msdos5
6b150898-d356-4343-a456-81b423e8511a
		else
		  search --no-floppy --fs-uuid --set=root
6b150898-d356-4343-a456-81b423e8511a
		fi
		echo	'载入 Linux 5.4.0-42-generic ...'
		linux	/boot/vmlinuz-5.4.0-42-generic
root=UUID=6b150898-d356-4343-a456-81b423e8511a ro  quiet splash $vt_handoff
		echo	'载入初始化内存盘...'
		initrd	/boot/initrd.img-5.4.0-42-generic
	}
	menuentry 'Ubuntu, with Linux 5.4.0-42-generic (recovery mode)' --class ubuntu --class
gnu-linux --class gnu --class os $menuentry_id_option
'gnulinux-5.4.0-42-generic-recovery-6b150898-d356-4343-a456-81b423e8511a' {
		recordfail
		load_video
		insmod gzio
		if [ x$grub_platform = xxen ]; then insmod xzio; insmod lzopio; fi
		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=hd0,msdos5
--hint-efi=hd0,msdos5 --hint-baremetal=ahci0,msdos5
6b150898-d356-4343-a456-81b423e8511a
		else
		  search --no-floppy --fs-uuid --set=root
6b150898-d356-4343-a456-81b423e8511a
		fi
		echo	'载入 Linux 5.4.0-42-generic ...'
		linux	/boot/vmlinuz-5.4.0-42-generic
root=UUID=6b150898-d356-4343-a456-81b423e8511a ro recovery nomodeset
            echo'载入初始化内存盘...'
            initrd      /boot/initrd.img-5.4.0-42-generic
        }
    }

    ### END /etc/grub.d/10_linux ###

    ### BEGIN /etc/grub.d/10_linux_zfs ###
    ### END /etc/grub.d/10_linux_zfs ###

    ### BEGIN /etc/grub.d/20_linux_xen ###

    ### END /etc/grub.d/20_linux_xen ###

    ### BEGIN /etc/grub.d/20_memtest86+ ###
    menuentry 'Memory test (memtest86+)' {
        insmod part_msdos
        insmod ext2
        set root='hd0,msdos5'
        if [ x$feature_platform_search_hint = xy ]; then
          search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos5 --hint-efi=hd0,msdos5
--hint-baremetal=ahci0,msdos5  6b150898-d356-4343-a456-81b423e8511a
        else
          search --no-floppy --fs-uuid --set=root 6b150898-d356-4343-a456-81b423e8511a
        fi
        knetbsd   /boot/memtest86+.elf
    }
    menuentry 'Memory test (memtest86+, serial console 115200)' {
        insmod part_msdos
        insmod ext2
        set root='hd0,msdos5'
        if [ x$feature_platform_search_hint = xy ]; then
          search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos5 --hint-efi=hd0,msdos5
--hint-baremetal=ahci0,msdos5  6b150898-d356-4343-a456-81b423e8511a
        else
          search --no-floppy --fs-uuid --set=root 6b150898-d356-4343-a456-81b423e8511a
        fi
        linux16   /boot/memtest86+.bin console=ttyS0,115200n8
    }
    ### END /etc/grub.d/20_memtest86+ ###

    ### BEGIN /etc/grub.d/30_os-prober ###
    ### END /etc/grub.d/30_os-prober ###
    ### BEGIN /etc/grub.d/30_uefi-firmware ###
    ### END /etc/grub.d/30_uefi-firmware ###

    ### BEGIN /etc/grub.d/40_custom ###
    # This file provides an easy way to add custom menu entries.  Simply type the
    # menu entries you want to add after this comment.  Be careful not to change
    # the 'exec tail' line above.
    ### END /etc/grub.d/40_custom ###

    ### BEGIN /etc/grub.d/41_custom ###
    if [ -f  ${config_directory}/custom.cfg ]; then
      source ${config_directory}/custom.cfg
    elif [ -z "${config_directory}" -a -f  $prefix/custom.cfg ]; then
      source $prefix/custom.cfg;
    fi
    ### END /etc/grub.d/41_custom ###
```

六、问题解答

1. Linux 内核目录下有一个.config 文件，请说明这个文件的作用？

   是 linux 内核的配置文件

2. 在 Linux 内核代码树中，很多子目录有 Makefile 文件和 Kconfig 文件，请分别解释这两个文件的作用？

   Kconfig 文件可以建立各层子目录之间的链接，Makefile 文件可以选择各个目录下的文件是否被编译，Kconfig 文件建立了整个编译关系（即.config 文件控制 Makefile 文件编译指定程序代码的过程）

3. 浏览/boot 目录，你一定发现了 System.map-4.6.0 文件，以及 initrd.img-4.6.0 文件。这两个文件分别起什么作用？你能否设计一个实验来验证你的判断？

   每当你编译一个新内核时，各种符号名的地址会变化。System.map 存储符号信息和地址的对应关系，随着每次内核的编译，就会产生一个新的 System.map 文件，并且需要用该文件取代原来的文件。initrd 的含义是 initialized ram disk，ramdisk 是用一部分内存模拟成磁盘，让操作系统访问。initrd.img 文件就是个 ramdisk 的映像文件。ram disk 是标准内核文件认识的设备(/dev/ram0)文件系统也是标准内核认识的文件系统。内核加载这个 ram disk 作为根文件系统并开始执行其中的某个文件来加载各种模块,服务等。再经过一些配置和运行后，就可以去物理磁盘加载真正的 root 分区了。

七、讨论、心得

本实验帮助我对于 linux 中断的指令更加熟悉，同时对于 linux 内核模块的编译过程有了更加清晰的认识。