

# Project0: Compiling Linux Kernel

518021911193, 刘昊林

## 1 准备阶段

下载 VMware Workstation Pro 软件，下载 Ubuntu 操作系统镜像文件 ubuntu-16.04.6-desktop-amd64.iso。然后在 VMware 中安装一个新的虚拟机。

打开虚拟机，在终端中输入 `uname -a` 命令，可以看到当前 Ubuntu 系统中自带的 Linux 内核版本为 Linux Ubuntu 4.15.0-88-generic。具体内容如下图所示：

```
psyduckliu@ubuntu:~$ uname -a
Linux ubuntu 4.15.0-88-generic #88-16.04.1-Ubuntu SMP Wed Feb 12 04:19:15 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
psyduckliu@ubuntu:~$ /usr/src$ ls
bash: /usr/src: No such file or directory
psyduckliu@ubuntu:~$ apt-cache search linux-source
linux-source - Linux kernel source with Ubuntu patches
linux-source-4.4.0 - Linux kernel source for version 4.4.0 with Ubuntu patches
linux-source-4.10.0 - Linux kernel source for version 4.10.0 with Ubuntu patches
linux-source-4.11.0 - Linux kernel source for version 4.11.0 with Ubuntu patches
linux-source-4.13.0 - Linux kernel source for version 4.13.0 with Ubuntu patches
linux-source-4.15.0 - Linux kernel source for version 4.15.0 with Ubuntu patches
linux-source-4.8.0 - Linux kernel source for version 4.8.0 with Ubuntu patches
```

图 1: 查看当前内核信息

然后登陆 [www.kernel.org](http://www.kernel.org) 下载 Linux 内核源码，我这里下载的是 4.19.108 版内核，下载完成后对压缩包进行解压。这里我使用的是 `sudo tar -xavf linux-4.19.108.tar.xz -C /usr/src` 命令，将解压好的文件直接送入 `/usr/src` 文件夹下。并且在解压完成后还可以使用 `cd /usr/src` 命令进入该文件夹，然后使用 `ls` 命令就可以看到已经解压后的 `linux-4.19.108` 文件。具体内容如下图所示：


Protocol	Location	Latest Stable Kernel:	
HTTP	<a href="https://www.kernel.org/pub/">https://www.kernel.org/pub/</a>	5.5.8	
GIT	<a href="https://git.kernel.org/">https://git.kernel.org/</a>		
RSYNC	<a href="rsync://rsync.kernel.org/pub/">rsync://rsync.kernel.org/pub/</a>		
mainline:	5.6-rc5	2020-03-09	<a href="#">[tarball]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a> <a href="#">[view diff]</a> <a href="#">[browse]</a>
stable:	5.5.8	2020-03-05	<a href="#">[tarball]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a> <a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	5.4.24	2020-03-05	<a href="#">[tarball]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a> <a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	4.19.108	2020-03-05	<a href="#">[tarball]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a> <a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	4.14.172	2020-02-28	<a href="#">[tarball]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a> <a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	4.9.215	2020-02-28	<a href="#">[tarball]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a> <a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	4.4.215	2020-02-28	<a href="#">[tarball]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a> <a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
longterm:	3.16.82	2020-02-11	<a href="#">[tarball]</a> <a href="#">[pgp]</a> <a href="#">[patch]</a> <a href="#">[inc. patch]</a> <a href="#">[view diff]</a> <a href="#">[browse]</a> <a href="#">[changelog]</a>
linux-next:	next-20200306	2020-03-06	<a href="#">[browse]</a>

图 2: 选择下载的 Linux 内核版本

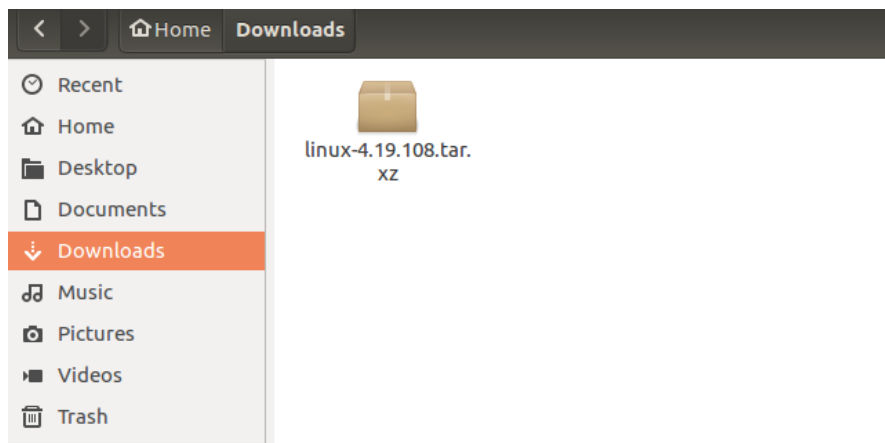


图 3: 下载好的 Linux 源码压缩包

```
psyduckliu@ubuntu:~/Downloads$ sudo tar -xavf linux-4.19.108.tar.xz -C /usr/src
```

图 4: 解压源码压缩包

```
psyduckliu@ubuntu:~/Downloads$ cd /usr/src
psyduckliu@ubuntu:/usr/src$ ls
linux-4.19.108          linux-headers-4.15.0-88
linux-headers-4.15.0-45 linux-headers-4.15.0-88-generic
linux-headers-4.15.0-45-generic open-vm-tools-10.0.7
```

图 5: 进入 /usr/src 文件夹检查解压结果

## 2 安装编译内核需要的一些程序

首先输入 `sudo apt-get upgrade` 命令，更新软件来源，否则可能会出现安装失败的情况。再依次输入以下命令，来下载必要程序：

```
sudo apt-get install libncurses5-dev openssl libssl-dev
```

```
sudo apt-get install build-essential openssl
```

```
sudo apt-get install pkg-config
```

```
sudo apt-get install libc6-dev
```

```
sudo apt-get install bison
```

```
sudo apt-get install flex
```

```
sudo apt-get install libelf-dev
```

```
sudo apt-get install zlibc minizip
```

```
sudo apt-get install libidn11-dev libidn11
```

具体内容如下图所示 (只选择了一部分程序的下载截图)：

```

psyduckliu@ubuntu: /usr/src$ sudo apt-get install gcc make libncurses5-dev openssl libssl-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
gcc is already the newest version (4:5.3.1-1ubuntu1).
make is already the newest version (4.1-6).
openssl is already the newest version (1.0.2g-1ubuntu4.15).
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
Suggested packages:
  ncurses-doc
The following NEW packages will be installed:
  libncurses5-dev libssl-dev libssl-doc libtinfo-dev zlib1g-dev
0 upgraded, 5 newly installed, 0 to remove and 2 not upgraded.
Need to get 2,840 kB of archives.
After this operation, 12.0 MB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 libtinfo-dev amd64 6.0+20160213-1ubuntu1 [77.4 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial/main amd64 libncurses5-dev amd64 6.0+20160213-1ubuntu1 [175 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 zlib1g-dev amd64 1:1.2.8.dfsg-2ubuntu4.3 [167 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 libssl-dev amd64 1.0.2g-1ubuntu4.15 [1,344 kB]

```

图 6: 下载必要程序 (1)

```

psyduckliu@ubuntu: /usr/src$ sudo apt-get install build-essential
Reading package lists... Done
Building dependency tree
Reading state information... Done
build-essential is already the newest version (12.1ubuntu2).
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
psyduckliu@ubuntu: /usr/src$ sudo apt-get install pkg-config
Reading package lists... Done
Building dependency tree
Reading state information... Done
pkg-config is already the newest version (0.29.1-0ubuntu1).
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.
psyduckliu@ubuntu: /usr/src$ sudo apt-get install libc6-dev
Reading package lists... Done
Building dependency tree
Reading state information... Done
libc6-dev is already the newest version (2.23-0ubuntu11).
The following package was automatically installed and is no longer required:
  snapd-login-service
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 2 not upgraded.

```

图 7: 下载必要程序 (2)

### 3 编译内核前的一些准备

在终端中依次输入以下三条命令：

```
sudo make mrproper
```

```
sudo make clean
```

```
sudo make menuconfig
```

在这其中，`sudo make mrproper` 是为了清除编译过程中产生的所有中间文件；`sudo make clean` 是为了清除上一次产生的编译中间文件；`sudo make menuconfig` 是为了对内核选项进行配置，在这里我直接选择了 `exit` 退出，保留默认设置。

具体内容如下图所示：

```
psyducktliu@ubuntu: /usr/src/linux-4.19.108$ sudo make mrproper
psyducktliu@ubuntu: /usr/src/linux-4.19.108$ sudo make clean
psyducktliu@ubuntu: /usr/src/linux-4.19.108$ sudo make menuconfig
HOSTCC scripts/basic/fixdep
UPD scripts/kconfig/.mconf.cfg
HOSTCC scripts/kconfig/mconf.o
YACC scripts/kconfig/zconf.tab.c
LEX scripts/kconfig/zconf.lex.c
HOSTCC scripts/kconfig/zconf.tab.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
HOSTLD scripts/kconfig/mconf
scripts/kconfig/mconf Kconfig
#
# using defaults found in /boot/config-4.15.0-88-generic
#
/boot/config-4.15.0-88-generic:897:warning: symbol value 'm' invalid for HOTPLUG_PCI_SHPC
/boot/config-4.15.0-88-generic:1151:warning: symbol value 'm' invalid for NF_NAT_REDIRECT
/boot/config-4.15.0-88-generic:1154:warning: symbol value 'm' invalid for NF_TABLES_INET
/boot/config-4.15.0-88-generic:1155:warning: symbol value 'm' invalid for NF_TABLES_NETDEV
/boot/config-4.15.0-88-generic:1338:warning: symbol value 'm' invalid for NF_TABLES_IPV4
/boot/config-4.15.0-88-generic:1343:warning: symbol value 'm' invalid for NF_TABLES_ARP
/boot/config-4.15.0-88-generic:1350:warning: symbol value 'm' invalid for NF_NAT_MASQUERADE_IPV4
/boot/config-4.15.0-88-generic:1385:warning: symbol value 'm' invalid for NF_TABLES_IPV6
/boot/config-4.15.0-88-generic:1397:warning: symbol value 'm' invalid for NF_NAT_MASQUERADE_IPV6
/boot/config-4.15.0-88-generic:1423:warning: symbol value 'm' invalid for NF_TABLES_BRIDGE
/boot/config-4.15.0-88-generic:4003:warning: symbol value 'm' invalid for HW_RANDOM_TPM
/boot/config-4.15.0-88-generic:4949:warning: symbol value 'm' invalid for LIRC
/boot/config-4.15.0-88-generic:6175:warning: symbol value 'm' invalid for SND_SOC_INTEL_SST_TOPLEVEL
/boot/config-4.15.0-88-generic:6180:warning: symbol value 'm' invalid for SND_SOC_INTEL_MACH
configuration written to .config

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

图 8: 依次输入三条命令

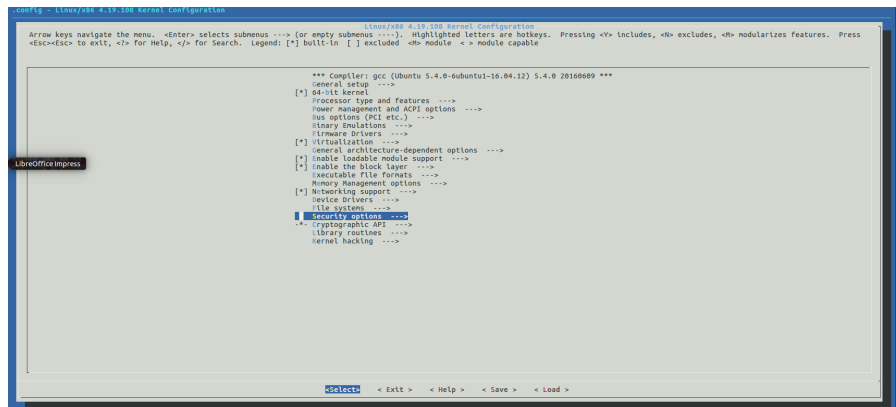


图 9: sudo make menuconfig 的配置界面

## 4 编译内核

在终端中输入命令：sudo make -j2，我在这里采用了 2 个线程并行编译的方法，来加快编译速度，但是编译内核的过程还是花费了 3 个小时。并且在编译过程中还出现了 gcc: internal compiler error: Killed (program cc1plus)”这样的错误。经过在 CSDN 上的搜索了解，我知道这个错误出现的原因是内存不足，所以我把虚拟机关机，把内存从原来的 2GB 调整到了 4GB，再打开虚拟机，就可以顺利地继续进行编译了。

具体内容如下图所示：

```

syduckliu@ubuntu:/usr/src/linux-4.19.108$ sudo make -j2
SYSTBL arch/x86/include/generated/asm/syscalls_32.h
UPD include/config/kernel.release
WRAP arch/x86/include/generated/uapi/asm/bpf_perf_event.h
WRAP arch/x86/include/generated/uapi/asm/poll.h
UPD include/generated/uapi/linux/version.h
UPD include/generated/utsrelease.h
DESCEND objtool
HOSTCC /usr/src/linux-4.19.108/tools/objtool/fixdep.o
HOSTLD /usr/src/linux-4.19.108/tools/objtool/fixdep-in.o
LINK /usr/src/linux-4.19.108/tools/objtool/fixdep
CC /usr/src/linux-4.19.108/tools/objtool/exec-cmd.o
SYSHDR arch/x86/include/generated/asm/unistd_32_ia32.h
SYSHDR arch/x86/include/generated/asm/unistd_64_x32.h
SYSTBL arch/x86/include/generated/asm/syscalls_64.h
CC /usr/src/linux-4.19.108/tools/objtool/help.o
CC /usr/src/linux-4.19.108/tools/objtool/pager.o
CC /usr/src/linux-4.19.108/tools/objtool/parse-options.o
CC /usr/src/linux-4.19.108/tools/objtool/run-command.o

```

图 10: sudo make -j2, 开始编译

```

LD [M] sound/usb/line6/snd-usb-line6.ko
CC sound/usb/line6/snd-usb-pod.mod.o
LD [M] sound/usb/line6/snd-usb-pod.ko
CC sound/usb/line6/snd-usb-podhd.mod.o
LD [M] sound/usb/line6/snd-usb-podhd.ko
CC sound/usb/line6/snd-usb-toneport.mod.o
LD [M] sound/usb/line6/snd-usb-toneport.ko
CC sound/usb/line6/snd-usb-variak.mod.o
LD [M] sound/usb/line6/snd-usb-variak.ko
CC sound/usb/misc/snd-ua101.mod.o
LD [M] sound/usb/misc/snd-ua101.ko
CC sound/usb/snd-usb-audio.mod.o
LD [M] sound/usb/snd-usb-audio.ko
CC sound/usb/snd-usbmidi-lib.mod.o
LD [M] sound/usb/snd-usbmidi-lib.ko
CC sound/usb/usx2y/snd-usb-us122l.mod.o
LD [M] sound/usb/usx2y/snd-usb-us122l.ko
CC sound/usb/usx2y/snd-usb-usx2y.mod.o
LD [M] sound/usb/usx2y/snd-usb-usx2y.ko
CC sound/x86/snd-hdmi-lpe-audio.mod.o
LD [M] sound/x86/snd-hdmi-lpe-audio.ko
CC virt/lib/irqbypass.mod.o
LD [M] virt/lib/irqbypass.ko
syduckliu@ubuntu:/usr/src/linux-4.19.108$

```

图 11: 编译结束

## 5 安装内核模块和内核

在终端中依次输入以下两条命令：

```
sudo make modules_install
```

```
sudo make install
```

其中 `sudo make modules_install` 安用于安装内核模块；`sudo make install` 用于安装内核。都安装好之后，再输入 `reboot` 命令，重启虚拟机，就安装完成了。具体内容如下图所示：

```
psyduckliu@ubuntu:/usr/src/linux-4.19.108$ sudo make modules_install
```

图 12: sudo make modules\_install, 开始安装内核模块

```
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-variak.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 virt/lib/irqbypass.ko
DEPMOD 4.19.108
psyduckliu@ubuntu:/usr/src/linux-4.19.108$
```

图 13: 安装内核模块结束

```
psyduckliu@ubuntu:/usr/src/linux-4.19.108$ sudo make install
[cmd] password for psyduckliu:
sh ./arch/x86/boot/install.sh 4.19.108 arch/x86/boot/bzImage \
    System.map /boot
run-parts: executing /etc/kernel/postinst.d/apt-auto-removal 4.19.108 /boot/vmlinuz-4.19.108
run-parts: executing /etc/kernel/postinst.d/dmcc 4.19.108 /boot/vmlinuz-4.19.108
* dmcc: running auto installation service for kernel 4.19.108
Kernel preparation unnecessary for this kernel. Skipping...

Building module:
Cleaning build area....(bad exit status: 2)
make KERNELRELEASE=4.19.108 VMLINUX=4.19.108 MODULEBUILDDIR=/var/lib/dmcc/open-vn-tools/10.0.7/build -C vmmnet....(bad exit status: 2)
Error! Bad return status for module build on kernel: 4.19.108 (x86_64)
Consult /var/lib/dmcc/open-vn-tools/10.0.7/BuildMakeLog for more information.
run-parts: executing /etc/kernel/postinst.d/initramfs-tools 4.19.108 /boot/vmlinuz-4.19.108
update-initramfs: Generating /boot/initrd.img-4.19.108
run-parts: executing /etc/kernel/postinst.d/jm-utils 4.19.108 /boot/vmlinuz-4.19.108
run-parts: executing /etc/kernel/postinst.d/unattended-upgrades 4.19.108 /boot/vmlinuz-4.19.108
run-parts: executing /etc/kernel/postinst.d/update-notifier 4.19.108 /boot/vmlinuz-4.19.108
run-parts: executing /etc/kernel/postinst.d/zi-update-grub 4.19.108 /boot/vmlinuz-4.19.108
Generating grub configuration file ...
warning: Setting grub Timeout to a non-zero value when GRUB_HIDDEN_TIMEOUT is set is no longer supported.
Found linux image: /boot/vmlinuz-4.19.108
Found init image: /boot/initrd.img-4.19.108
Found linux image: /boot/vmlinuz-4.15.0-88-generic
Found init image: /boot/initrd.img-4.15.0-88-generic
Found memtest86+ image: /boot/memtest86+.elf
Found memtest86+ image: /boot/memtest86+.bin
done
```

图 14: sudo make install, 安装内核

## 6 检查新内核, 卸载旧内核

在终端输入 `uname -a` 命令, 就可以发现内核已经是新编译的 Linux Ubuntu 4.19.108 了。再输入 `sudo dpkg --get-selections | grep 'linux'` 命令, 就可以查找到所有的内核信息, 我们可以再输入 `sudo apt-get purge linux-image-4.15.0.88` 命令删除不再需要的内核。

具体内容如下图所示:

```

psyducklliu@ubuntu:~$ uname -a
Linux ubuntu 4.19.108 #1 SMP Wed Mar 11 14:22:15 CST 2020 x86_64 x86_64 x86_64 GNU/Linux
psyducklliu@ubuntu:~$ sudo dpkg --get-selections | grep 'linux'
[sudo] password for psyducklliu:
console-setup-linux                                install
libselinux1:amd64                                  install
linux-base                                          install
linux-firmware                                     install
linux-generic-hwe-16.04                            install
linux-headers-4.15.0-45                             install
linux-headers-4.15.0-45-generic                     install
linux-headers-4.15.0-88                             install
linux-headers-4.15.0-88-generic                     install
linux-headers-generic-hwe-16.04                     install
linux-image-4.15.0-45-generic                       deinstall
linux-image-4.15.0-88-generic                       install
linux-image-generic-hwe-16.04                       install
linux-libc-dev:amd64                                install
linux-modules-4.15.0-45-generic                     install
linux-modules-4.15.0-88-generic                     install
linux-modules-extra-4.15.0-45-generic                deinstall
linux-modules-extra-4.15.0-88-generic                install
linux-sound-base                                    install
pptp-linux                                          install
syslinux                                           install
syslinux-common                                    install
syslinux-legacy                                    install
util-linux                                          install
psyducklliu@ubuntu:~$ sudo apt-get purge linux-image-4.15.0-88

```

图 15: 检查新内核，卸载旧内核