Linux 内核重建

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。

实验内容

在 Linux 操作系统环境下重新编译内核。实验主要内容：

- 查找并且下载一份内核源代码
- 配置内核
- 编译内核和模块
- 配置启动文件

实验环境

说明:本实验指导基于老版本的 Ubuntu，实验环境、版本、命令仅作参考。本年度实验提交内容，要求使用新版本 Ubuntu 不低于 18.04（可选择 18.04，18.10，19.04），内核版本不低于 4.15，4.15 之后版本均可。

本实验指导的 Linux 操作系统发行版本为 ubuntu 16.04.9 LTS，Linux 内核版本为 4.13。

```
os@os:~$ cat /proc/version
Linux version 4.13.0-36-generic (buildd@lgw01-amd64-033) (gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.9)) #40~16.
04.1-Ubuntu SMP Fri Feb 16 23:25:58 UTC 2018
```

安装依赖包

sudo apt-get install git fakeroot build-essential ncurses-dev xz-utils libssl-dev bc flex libelf-dev bison

实验指导

Linux 是当今流行的操作系统之一。由于其源码的开放性，现代操作系统设计的思想和技术能够不断运用于它的新版本中。因此，读懂并修改 Linux 内核源代码无疑是学习操作系统设计技术的有效方法。

1. 查找并且下载一份内核源代码

Linux 受 GNU 通用公共许可证（GPL）保护，其内核源代码是完全开放的。现在很多 Linux 的网站都提供内核代码的下载。推荐你使用 Linux 的官方网站：http://www.kernel.org ，如图 1。在这里你可以找到所有的内核版本。

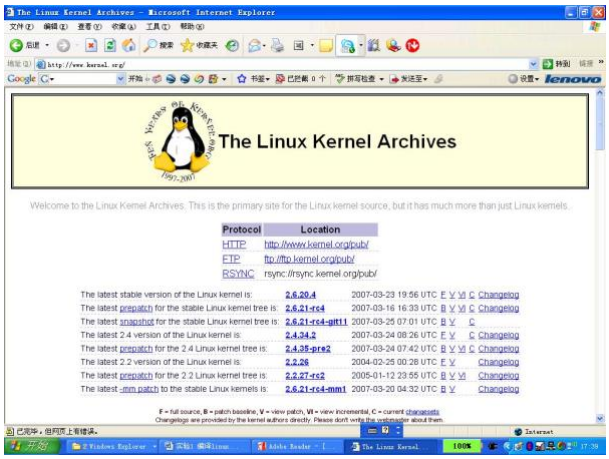


图 1　Linux 的官方网站

我们也可以通过 ubuntu 的源直接下载源代码(推荐方式)：

首先查找源代码

```
os@os:~$ 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
```

下载和当前系统使用的内核版本一样的源代码：

```
os@os:~$ sudo apt-get install linux-source-4.13.0
[sudo] password for os:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  libncurses-dev | ncurses-dev kernel-package libqt3-dev
The following NEW packages will be installed:
  linux-source-4.13.0
0 upgraded, 1 newly installed, 0 to remove and 184 not upgraded.
Need to get 127 MB of archives.
After this operation, 142 MB of additional disk space will be used.
```

2. 部署内核源代码

使用刚才步骤下载的源代码位于： /usr/src/linux-source-4.13.0

回到本地 home 目录，新建一个目录用来编译 linux 内核。

```
# cd
# mkdir build_kernel
# cd build_kernel
# cp /usr/src/linux-source-4.13.0.tar.bz2 .
# tar jxvf linux-source-4.13.0.tar.bz2
```

3. 配置内核

在你进行这项工作之前，不妨先看一看/usr/src/linux 目录下内核源代码自带的 README 文件。在这份文件中，对怎样进行内核的解压，配置，安装都进行了详细的讲解。不过，其介绍的步骤不完全符合我们的版本，所以还是以本书为准。

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）。

比较简单的方法是采用当前系统已经存在的配置文件。

```
# cp /usr/src/linux-headers-4.13.0-36-generic/.config linux-source-4.13.0
# cd linux-source-4.13.0
```

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

如果需要做一些个性化配置，可以再次进入：

```
# make menuconfig
```

进行配置时，大部分选项可以使用其缺省值，只有小部分需要根据用户不同的需要选择。例如，如果硬盘分区采用 ext2 文件系统（或 ext3 文件系统），则配置项应支持 ext2 文件系统（ext3 文件系统）。又例如，系统如果配有 SCSI 总线及设备，需要在配置中选择 SCSI 卡的支持。又例如，后续实验需要在一个文件上格式化一个文件系统并且安装到某目录，这需要 device drivers 之下的 block device 之下的 loopback device support 特征。

对每一个配置选项，用户有三种选择，它们分别代表的含义如下：

- “<*>”或“[*]” － 将该功能编译进内核
- “[ ]” － 不将该功能编译进内核
- “[M]” － 将该功能编译成可以在需要时动态插入到内核中的模块

将与核心其它部分关系较远且不经常使用的部分功能代码编译成为可加载模块，有利于减小内核的长度，减小内核消耗的内存，简化该功能相应的环境改变时对内核的影响。许多功能都可以这样处理，例如像上面提到的对 SCSI 卡的支持，等等。

4. 编译内核和模块,安装，生成启动文件

```
# make    -j4
# sudo make modules_install -j 4
# sudo make install -j 4
```

我们编译安装好了内核和模块后，生成可以 boot 的 initrd.img

```
# sudo update-initramfs -c -k 4.13.16
```

注意，这里的 **4.13.16 是内核具体版本。如何获得这个版本号呢？可以去 /lib/modules/ 目录看看我们新编译的内核模块被放在了哪一个目录下。比如**

```
os@os:~/build_kernel/linux-source-4.13.0$ ls -l /lib/modules/
total 8
drwxr-xr-x 5 root root 4096 Mar    1    2018 4.13.0-36-generic
drwxr-xr-x 3 root root 4096 Nov    7 03:03 4.13.16
```

我们可以看到，我们编译的内核在 4.13.16 目录下，因此版本号就是 4.13.16。

最后更新启动文件

```
#sudo update-grub
```

好了，我们已经编译了内核 ，放到了指定位置/boot，我们也配置了 grub。现在，请你重启主机系统，期待编译过的 Linux 操作系统内核正常运行！

做到这一步不容易。初次接触 Linux 的新手，恐怕没这么顺利。

最后，当你完成整个实验后，要清除编译时的临时文件，使用命令：

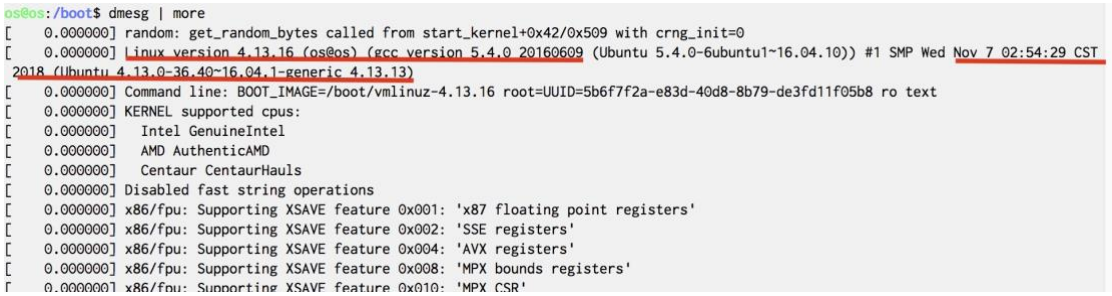
```
#make clean
```

注意，这些临时文件是你花费 1 小时左右的时间，辛辛苦苦建立起来的。不到有十分的把握，建议不要轻易删除它们。

5. 验证

如何验证当前运行的内核是你编译的呢？

方法 1： demsg | more



我们可以发现，内核是由 os@os （用户名@机器名）编译出来的。编译时间也显示是当前编译的内核

方法 2：

os@os:/boot$ uname -a

Linux os 4.13.16 #1 SMP Wed Nov 7 02:54:29 CST 2018 x86_64 x86_64 x86_64 GNU/Linux

同样通过编译时间能看出来内核是我们新编译的。

实验提交内容

说明:本实验指导基于老版本的 Ubuntu，实验环境、版本、命令仅作参考。本年度实验提交内容，要求使用新版本 Ubuntu 不低于 18.04（可选择 18.04，18.10，19.04），内核版本不低于 4.15，4.15 之后版本均可。

1. 提交你编译后的内核 dmesg | more 命令运行结果的截图，需要能明确显示出来你编译内核的时间和机器名，用户名（图片加框）
2. Linux 内核目录下有一个.config 文件，请说明这个文件的作用？
3. 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？
4. 浏览/boot 目录，你一定发现了 System.map-4.13.16 文件，以及 initrd.img-4.13.0 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？