

编译 Linux 内核

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

本次实验指导是在 RedHat Fedora Core 5 的 Linux 系统里，下载并重新编译其内核源代码（版本号 KERNEL-2.6.15-1.2054）；然后，配置 GNU 的启动引导工具 grub，成功运行编译成功的 Linux 内核。

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。

许多同学使用 Ubuntu 发行版本来做实验，在 Ubuntu 下编译内核与本实验指导是有区别的。Ubuntu 下做本实验可以参考“课件”。

鼓励同学们使用最新内核版本完成本实验！

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。

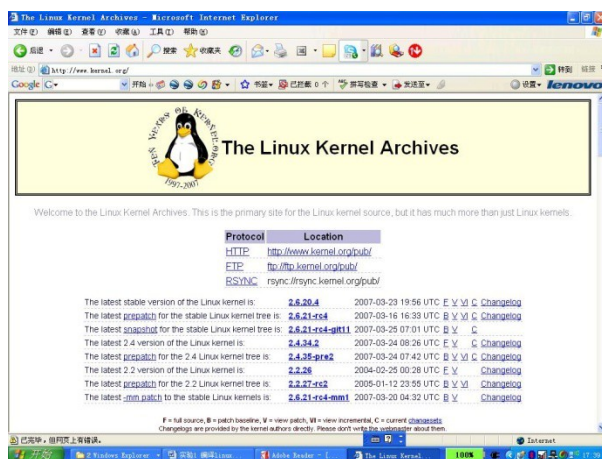


图 1 Linux 的官方网站

Fedora Core 5 并不附带内核源代码，第一步首先想办法获取合适版本的 Linux 内核代码。通过命令

```
# uname -r  
2.6.15-1.2054_FC5
```

这就是说，RedHat Fedora Core 5 采用的内核版本是 2.6.15-1.2054_FC5。但是，官方网站 <http://www.kernel.org/pub/linux/kernel/> 找不到对应版本。请别着急，既然它是 RedHat 发布的，RedHat 的官方网站总有吧。浏览 <http://download.fedora.redhat.com/pub/fedora/linux/core/5/source/SRPMS/>，我们发现果然有文件 [kernel-2.6.15-1.2054_FC5.src.rpm](#)，这个 rpm 文件就是 2.6.15-1.2054_FC5 版的内核源代码了。下载后保存。需要说明的是，其实还有许多网站保存此文件的；有时候以 ISO 压缩包的形式出现，文件名为 kernel-2.6.15-1.2054_FC5.src.iso。

2. 部署内核源代码

此过程比较机械、枯燥，因而容易出错。请严格按下述步骤来操作。

首先，解开 rpm 包，放在 /usr/src/redhat。使用操作序列：

```
# rpm -Uvh kernel-2.6.15-1.2054_FC5.src.rpm  
# cd /usr/src/redhat/SPECS  
# rpmbuild -bp --target $(uname -m) kernel-2.6.spec
```

这里，命令 `uname -m` 检测 CPU 型号。作者的主机是 i686。假如执行 shell 命令：

```
# ls /usr/src/redhat/BUILD/kernel-2.6.15/  
Config.mk linux-2.6.15.i686 vanilla xen xen-vanilla
```

可见，Linux 内核源代码已经在 /usr/src/redhat/BUILD/kernel-2.6.15/linux-2.6.15.i686 下面了。作者习惯了 RedHat 过去的部署，还是希望通过路径 /usr/src/linux 去访问它。这只要建一个符号链接：

```
# cd /usr/src  
# ln -s ./redhat/BUILD/kernel-2.6.15/linux-2.6.15.i686/ linux
```

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)。

```
# cd /usr/src/linux
```

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
```

```
# cp configs/kernel-2.6.15-i686.config .config
```

```
cp: overwrite '.config'? y
```

当前目录下的 Makefile 有一项内容：

```
EXTRAVERSION = -prep
```

因为版本号已经变成 2.6.15-1.2054_FC5 了，所以，使用任何一种文本编辑工具，将它换成：
`EXTRAVERSION = -1.2054_FC5`

然后：

```
#make menuconfig
```

`make menuconfig` 是基于文本的选单式配置界面，作者一般使用这一配置命令。当然，其它的配置界面也不错啊，例如：

- `make xconfig`，使用 X Windows (Qt) 界面
- `make gconfig`，使用 X Windows (Gtk) 界面
- `make oldconfig`，使用文本界面，按照 `./config` 文件的内容取其缺省值
- `make silentoldconfig`，与上一个一样；不同的是，不再逐项提问了

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。

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

编译内核，就用：

```
#make
```

编译内核需要较长的时间，具体与机器的硬件条件及内核的配置等因素有关（作者采用 VMWare 虚拟机，需要约 50 分钟）。完成后产生的内核文件 bzImage 的位置在 /usr/src/linux/arch/i386/boot 目录下，当然这里假设用户的 CPU 是 Intel x86 型的，并且你将内核源代码放在 /usr/src/linux 目录下。

如果选择了可加载模块，编译完内核后，要对选择的模块进行编译。用下面的命令编译模块并安装到标准的模块目录中：

```
#make modules
```

```
#make modules_install
```

5. 了解 Linux 内核的启动

通常，Linux 在系统引导后从 /boot 目录下读取内核映像到内存中。因此我们如果想要使用自己编译的内核，就必须先将启动文件安装到 /boot 目录下。使用命令：

```
# make install
```

6. 了解 grub 配置启动文件

如果使用 grub 启动 Linux，启动文件为 /boot/grub/grub.conf 文件(**ubuntu 发行版本为 /boot/grub/grub.cfg**)，近几年发行的 Linux 发行版本，建议不要去编辑这个文件，这个文件是自动生成的。查看启动配置文件：

```
#cat /boot/grub/grub.conf
```

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You do not have a /boot partition. This means that
#         all kernel and initrd paths are relative to /, eg.
#         root (hd0,0)
#         kernel /boot/vmlinuz-version ro root=/dev/sda1
#         initrd /boot/initrd-version.img
#boot=/dev/sda1
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title Fedora Core (2.6.15-1.2054_FC5)
    root (hd0,0)
```

```
kernel /boot/vmlinuz-2.6.15-1.2054_FC5 ro root=LABEL=/
initrd /boot/initrd-2.6.15-1.2054_FC5.img
title Project One
root (hd0,0)
kernel /boot/bzImage ro root=LABEL=/
initrd /boot/initrd-2.6.15-1.2054_FC5.img
```

这里，以 ‘#’ 开头的，是注释行，grub 不会去解释、执行它。“default=0”命令表示，当用户没有在规定时间内响应时，将解释、执行第 1 个 “title” 选项。“timeout=5”规定用户有效响应时间是 5。“title”定义了启动菜单里面的菜单项，一旦选中了此菜单项，那么，只有这个 “title” 后跟的命令才被 grub 解释、执行了。“root”指定 “根设备” 的地址。通过 “kernel” 命令，要求 grub 从后跟的路径，装入操作系统内核。grub 将从 “initrd” 命令指定的路径，装入一个 ramdisk 文件，并设定一些必需的参数；Linux 操作系统启动时，需要这样的文件。

好了，我们已经编译了内核 bzImage，放到了指定位置 /boot；我们也配置了 /boot/grub/grub.conf。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。

在本实验完成后，不要清除编译时的临时文件，也既不要使用这个命令：

```
#make clean
```

实验思考

浏览 /boot 目录，你一定发现了 System.map-2.6.15-1.2054_FC5 文件，以及 initrd-2.6.15-1.2054_FC5.img 文件。如果打开 /boot/grub 目录下面的 grub.conf 文件，不难发现命令：

```
initrd /boot/initrd-2.6.15-1.2054_FC5.img
```

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？

撰写实验报告的要求

1. 按照实验报告模板格式撰写；
2. 整个实验过程的解图；
3. 编译内核成功后，您计算机上 grub.conf（或 grub.cfg）文件的内容；
4. 实验过程中遇到的问题及解决方法等；
5. 回答实验思考；

6. 心得体会。