# Linux内核重建

# 实验目的

学习重建Linux内核，掌握Linux内核和发行版本的区别。

# 实验内容

在linux操作系统环境下重建内核。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

本实验指导的Linux操作系统发行版本为Ubuntu 14.10，Linux内核版本为3.17.1。主机操作系统为Windows 7，虚拟机软件为VMware Play 6.0 Windows。

教材中的实验指导是在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）。

# 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。

本实验指导认为你已经学会Linux系统的安装，并已初步掌握Linux环境命令行界面的基本命令的使用。

#### 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。在这里你可以找到所有的内核版本。

在www.kernel.org下载 3.17.1版本:linux-3.17.1.tar.xz

#### 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，把linux-3.17.1.tar.xz包放在主目录下，解开linux-3.17.1.tar.xz包：

**tar –xvf linux-3.17.1.tar.xz**

解压出来的内核代码存放在~/linux-3.17.1目录下。为了方便操作及一致性，可以通过路径~/linux去访问它，这只要建一个符号链接：

**ln -s ~/linux-3.17.1/ ~/linux**

#### 配置内核

**第1次编译内核的准备：**

在ubuntu环境下，用命令make menuconfig对内核进行配置时，需要用终端模式下的字符菜单支持软件包libncurses5-dev，因此你是第一次重建内核，需要下载并安装该软件包，下载并安装命令如下：

**sudo apt-get install libncurses5-dev**

若上面这一步提示错误信息，则输入下面的命令sudo apt-get –f install ，建立库依赖关系。

**查看README文件：**

在你进行这项工作之前，不妨先看一看~/linux-3.17.1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。

**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）。

**cd ~/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**

为了与正在运行的操作系统内核的运行环境匹配，可以先把当前已配置好的文件复制到当前目录下，新的文件名为.config文件：

**cp /boot/config-`uname -r` .config**

这里，命令`uname –r`得到当前内核版本号。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**

**或 make –j2 使用两个进程编译。**

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2小时以上）。完成后产生的内核文件bzImage的位置在~/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~/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：

**sudo make modules**

**sudo make modules\_install**

#### 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:

**sudo make install**

#### 了解grub配置启动文件

如果使用grub启动Linux，启动文件名为/boot/grub/grub.cfg(不同的发行版本该文件名可能不一样)，这个文件是自动生成的，不要随意去修改这个文件的内容，否则，系统在下次启动会产生问题。

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.cfg。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！

**sudo reboot**

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。这些临时文件是你花费2个多小时辛辛苦苦建立起来的，在实验3和实验4还要用到这些临时文件。不到有十分的把握，建议不要轻易删除它们,否则你下一次编译还需要花这么多时间。

要缩短编译时间，可以使用 CCache (<http://ccache.samba.org/>)，可以加快 make clean 之后重新编译速度，有兴趣的同学可以试试。

# 实验问答题：

1. Linux内核目录下有一个.config文件，请说明这个文件的作用？
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？
3. 浏览/boot目录，你一定发现了System.map-3.17.1文件，以及initrd.img-3.17.1文件。如果打开/boot/grub目录下面的grub.cfg文件，不难发现命令：

initrd /boot/ initrd.img-3.17.1

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. **整个实验过程的截图；**
3. 实验过程中遇到的问题及解决方法等；
4. **回答实验问答题；**
5. 心得体会。





















