China-pub.com
下载

第10章　Linux内核重构

大多数UNIX环境的基础是内核的可配置项， Red Hat Linux也不例外。Red Hat Linux的系统内核修改，可以通过实时地直接配置或通过对操作环境的完全重构来完成。本章将重点介绍内核构造环境和生成可操作内核所需的方法。

Linux的内核与大多数商业 UNIX有着明显的区别，因为其内核的全部源代码是公开的，而且都包括在大多数主要的 Linux发布中。如果用户非常关心某一部分 Linux的实现细节，可以阅读该部分的源码。如果用户对系统有某种特殊需求，要求修改内核，则可以通过给内核应用增加某个补丁来构造特定版本的 Linux。

这并不是说构造内核是管理 Linux系统的必要步骤。本书提到的所有应用程序对内核都没有任何特殊的要求，安装者安装在系统中的通用内核已包含了所需的一切。为了能够正常的工作，较旧版本的 Linux通常都需要定制好的内核（尤其是在有内核模块之前），然而近来对 Linux的改进排除了很多对定制过程的需求。没有任何理由，却坚持要定制内核，这样的 Linux系统管理员已经落后于时代。然而有时却需要对内核进行重新配置。

虽然Red Hat Linux 6.0版本发布不久以后，人们就发现2.2.5内核在安全性上容易受到攻击，但是，任何基于2.2系列内核的Linux都可以进行升级，以使其成为一个有补丁的内核。该升级可以通过一套复杂的包括 RPM的步骤，或通过构造一个新的内核来实现。这两种方法将在下面作简单介绍。

对于针对Internet开发的服务器，构造定制内核比对工作站更重要。当然对服务器定制内核也更容易，因为服务器不需要声卡或其他可能增加内核复杂性的硬件设备。

Linux内核通常存贮在系统根目录下名为 vmlinuz的文件中。然而Red Hat Linux 6.0将内核以另一个名字（例如 vmlinuz-2.2.5-15）存放在/boot目录中，内核在系统启动时装载。当系统启动时，内核将检测系统并配置基本的硬件设置，然后调用 init程序以继续启动进程（启动的顺序本书没有介绍）。虽然系统管理员可以随意装载或删除内核的某一部分，但切换到一个完全不同的内核映像的唯一方法是重启系统。

本书的内容主要面向i386体系结构，但是读者应该清楚，对 Alpha和Sparc体系结构而言，内核的准备和启动与i386有一些细微的差别，这里就不再介绍了。

10.1　安装内核源代码

Linux内核是用C语言编写的，需要在开发环境中有 C编译器。在本书所附光盘的 /linux目录下有一个内核源代码（版本 2.2.12）的拷贝，该拷贝是利用 bzip2压缩工具压缩的。在缺省的情况下，系统不安装bzip2工具，但是Red Hat Linux 6.0的标准发布中包含一个安装bzip2所需的RPM软件包。若用户对安装这种软件包不了解，那么就不要准备构造新的内核，可以先阅读第1章中关于RPM选择的安装信息。

在构造内核的时候，很容易造成一些错误，使得系统无法引导。内核构造对缺乏经验的人而言是不适宜的，而在安装的时候创建一张紧急引导盘尤为重要。对于缺乏经验的系统管

China-pub.com 下载

理员来说，内核构造可能引起系统无法启动—问题在于内核何时无法启动。

假设本书附带的光盘已安装在了 /cdrom目录下，则用下面的命令序列可以将内核源代码解包：

```
cd /usr/src
rm -f linux
bzip2 -cd /cdrom/linux/linux-2.2.12.tar.bz2 | tar xvf -
chown -R root.root linux
cd linux
```

一旦安装了内核2.2.12的源代码，用户应该在互联网上随时查找更新的内核版本，两个常用的关于Linux内核的站点是http://www.kernel.org和http://www.kernelnotes.org。如果发现可用的新内核版本，应该将它下载下来，并将补丁文件应用到内核源代码中，使之升级。下面的命令可以使补丁得以应用：

```
cd /usr/src/linux; bzip2 -cd~/patch-2.2.x.bz2 | patch -pl
```

用FTP访问kernel.org要根据本国使用的语言来进行。例如在美国的系统管理员应使用ftp.us.kernel.org地址，并在 /pub/linux/kernel/v2.2目录下载内核补丁。英国用户使用地址ftp.uk.kernel.org，加拿大用户使用ftp.ca.kernel.org等。

注意 没有必要下载整个内核的源代码包。只要使用补丁就够了，它们很小，并且可以完成同样的工作（使源代码升级到新版本）。

在标准的Red Hat Linux发布中，也有可用的RPM格式的内核源代码，但是我们不推荐使用RPM的Linux内核。RPM版本并没有包含完整的源代码，而且在该版本上增加源代码补丁通常会导致无法正常工作的错误。

“make config”概述

我们对使用make config命令来配置内核的用户有如下值得考虑的建议——“不要这样做”。make config已逐渐被更简单的make menuconfig命令所取代，该命令是最灵活的内核构造配置工具。make xconfig也是很有用的命令，但是它需要 XWindow环境，所以本书不对其进行介绍。

在Shell提示下键入下面的命令可以配置Linux内核构造选项：

```
make menuconfig
```

这时清单10-1所示的菜单将显示出来。

每一个菜单选项都是由内核配置参数的一个主要子集组成的。在这些子选择中，各独立的选项可以开启或关闭。大多数选项启用后，都将把一部分附加的代码编译到内核中，这使得二进制内核映像增大。

大多数选项都有两种构造方式。它们可以直接编译到二进制内核映像中，也可以构造为模块（系统管理员可以随意地将它装载内核中，或者将其从内核中删除）。有一些配置选项不能构造为模块，而有的选项只能作为模块来使用；但大多数的配置选项可以按这两种方式使用。

清单10-1 Linux内核v2.2.12配置

```
*****************************************************************
************************* Main Menu *****************************************
* Arrow keys navigate the menu.  <Enter> selects submenus --->.        *
```

China-pub.com
下载


```
* Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes,*
* <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help.   *
* Legend: [*] built-in  [ ] excluded  <M> module  < > module capable   *
*************************************************************************
* *             Code maturity level options  --->                     * *
* *             Processor type and features  --->                     * *
* *             Loadable module support  --->                         * *
* *             General setup  --->                                   * *
* *             Plug and Play support  --->                           * *
* *             Block devices  --->                                   * *
* *             Networking options  --->                              * *
* *             SCSI support  --->                                    * *
* *             Network device support  --->                          * *
* *             Amateur Radio support  --->                           * *
* *             IrDA subsystem support  --->                          * *
* *             ISDN subsystem  --->                                  * *
* *             Old CD-ROM drivers (not SCSI, not IDE)  --->          * *
* *             Character devices  --->                               * *
* *             Filesystems  --->                                     * *
* *             Console drivers  --->                                 * *
* *             Sound  --->                                           * *
* *             Kernel hacking  --->                                  * *
* *             ---                                                   * *
* *             Load an Alternate Configuration File                  * *
* *             Save Configuration to an Alternate File               * *
*************************************************************************
*************************************************************************
*                   <Select>    < Exit >    < Help >                    *
*************************************************************************
```

当然，用户有时候是别无选择的。包括针对SCSI（Small Computer Systems Interface，小型计算机系统接口）和IDE（Integrated Drive Electronics，集成电路设备）驱动程序的存储选项。若内核是从SCSI驱动器中装载的，那么内核中SCSI控制器的驱动程序必须可用，否则系统将因为无法操作SCSI硬件，而无法启动。对于SCSI系统，准备内核的最佳方法是将SCSI控制器的驱动程序直接编译到内核映像之中。然而，对于安装过程所提供的通用内核而言，这并不是一种合理的解决方案；在通用内核中，SCSI驱动程序是作为一个模块来提供的，因为将驱动程序直接编译入内核将使得内核映像变得相当大。若模块存贮在SCSI硬件驱动器中，就出现了“先有鸡，还是先有蛋”的问题。通用内核使用initial ramdisk来解决这一问题。initrd映像拥有所有SCSI模块的拷贝，它由LILO（或其他装载工具）装载，并在内核启动时可用。但无论如何，initrd并不方便，所以最好使用不依赖于它的定制内核。

除了不能访问硬件驱动器以外，其他一些配置错误也可能导致系统无法启动。例如，提供ext2fs（Second Extended File System，次要扩展文件系统）所引起的故障就很容易引起系统启动失败，因为ext2fs是目前Red Hat Linux使用的本地文件系统（在Linux 2.4内核中，可以使用XFS文件系统来解决这个问题）。将不正确的设备配置选项作为参数传递给内核，通常会导致启动后立即死机。用户经常会遇到可能引发错误的事件，关键在于能够很快地找到应付的方法（可以依赖于LILO和紧急引导盘）。

在构造内核之前，用户应该决定模块的构成方法。当构造用于很多机器上的内核时，用户可能会觉得不使用模块，而是将所有需要的硬件驱动程序直接编译到内核中更加方便。然

China-pub.com 下载

后，用户可以将内核作为一个单一的文件拷贝到远程系统中，再运行 LILO，并用新的内核环境重新启动系统。如果用户没有采用静态内核的方法，那么最好将 /etc/cron.d/kmod文件中的单个crontab表项注释掉。否则，cron将每小时产生一次关于模块操作故障的消息。无论如何，从系统管理员的角度来看，静态内核比使用模块的内核更方便。虽然它对内核二进制映像的大小有限制，但是一些特征是必须作为模块编译的。如果环境要求，应使用模块，用户会发现比较好使用。

在开始内核配置之前，最好先检查一下 l smod命令的输出。l smod命令将列出在当前活跃的内核映像中已装载的所有模块。该列表将确认计划中的内核特征是否忽略了列出的硬件驱动程序。显然，如果使用静态内核，则 l smod命令将失败。

此处所讲述的内容并不覆盖所有的内核配置选项。要了解特定领域的问题，系统管理员必须构造内核并用该内核启动，然后调试并检查是否所有需要的功能都已可用。如果有些重要的模块不再正常工作（如网络路由，光盘驱动器等），那么，可能是因为没有将所有必要的模块构造到内核中。

下面是make menuconfig中选项的简要描述（在菜单的所有选项中，用户随时可以使用help）：

1. 代码成熟度级别选项

该菜单允许将实验性的驱动程序和不完整的代码包含在构造的内核中。除非用户不担心进入一种混乱的状态，否则就应该将此开关关闭。然而基于内核的 NFS服务（现在Red Hat Linux支持NFS服务）被划分在试验性代码一类，为了提供对 NFS服务的支持，必须将该选项打开。

2. 处理器类型和特征

选择将构造的内核的处理器类型。若通过内核重构所获得的内核将用于各类机器，则至少应包括通用的类型。例如，若处理器缺乏“ 数字协处理器 ”（386、486sx和NexGen处理器不包含协处理器，但所有 Pentium芯片包含协处理器），则用户应启用“ math emulation ”。若系统有多个处理器，则要启用对称多处理器（ Symmetric Multi-Processing，SMP）选项。否则，应该将此选项关闭。

3. 可装载模块支持

若开启该选项，则附加的内核代码将存贮在 /lib/modules目录中，而支持模块的内核特征可以作为动态模块装载。若使用了模块，则内核模块装载工具被启用（这样，旧的 kerneld模块装载工具就不再需要了）。若不使用该模块，则用户应将 /etc/cron.d/kmod中的表项注释掉，否则cron进程将每小时产生一个关于模块操作故障的状态消息。

4. 通用设置

基本网络、PCI总线控制（在大多数现在的 PC机中，PCI是硬件的标准配置）、程序二进制代码格式以及系统V的IPC在此选项中进行配置。系统管理员应将此处的大部分选项置为 on。进程间通信（ Inter-Process Communication，IPC）被大多数的应用程序所使用，尤其是Apache Web服务器。

5. 即插即用支持

用户应尽量避免使用即插即用硬件设备，但如果用户在系统中有即插即用设备，则应启用此选项。

China-pub.com
下载

6. 块设备

应该启用IDE磁盘和CD-ROM支持，即使系统中只有SCSI设备，因为IDE设备既便宜而且种类丰富。如果可以确认IDE芯片组，则应该在此处专门设置（如果不必要，则应关闭对有问题的CMD640和RZ1000的检测）。若内核还要使用可刻写的CD-ROM，则打开回送设备选项十分有用。此外，mkinitrd命令也需回送设备，因此无论构造什么样的内核，最好都包括对回送设备的支持。

7. 网络选项

主要的网络参数包括路由、防火墙和伪装，它们都在该菜单下设置（只有设置了“Network Firewalls”、“TCP/IP Networking”、“IP Firewalls”和“always defragment”，伪装设置选项才会出现）。

8. SCSI支持

此处提供对SCSI支持的配置。CD记录需要通用的SCSI支持。独立的SCSI适配器可以在菜单底部的SCSI底层驱动程序菜单中选择。若用户的根文件系统位于SCSI控制器上，那么应将控制器的驱动程序直接编译到内核中（不是作为模块），否则需要initrd映像。

9. 网络设备支持

用户可以在此处找到网络硬件的驱动程序，包括以太网、令牌环网，FDDI等。对PPP的支持也在这个菜单下配置。

10. 无线电爱好者支持

此处可以配置Linux对业余无线电爱好者的支持。

11. lrDA子系统支持

如果计算机系统中有无线电红外设备，那么就应该在此菜单中配置它们。

12. ISDN子系统

如果子系统中存在综合业务数据网适配器，则应在此菜单下进行配置。

13. 老式CD-ROM驱动程序

如果用户的系统中有老式的，不基于SCSI也不基于IDE接口的CD-ROM驱动器，那么用户最好去买一个新式的CD-ROM。IDE的CD-ROM只需要50美元。当然，如果系统真的需要提供对某种老式驱动器的支持，那么就在此菜单下进行配置。

14. 字符设备

最好不要修改此菜单下的控制台和虚拟终端选项。若系统中有支持多端口的串口板，那么应在此菜单下配置它的驱动程序。伪终端设备也应该在此处配置。伪终端是假定由telent、xterm或其他类终端设备连接的。缺省的设备最大数目为256，如果必要的话，可以将此限定值加大。此外，对串行总线鼠标PS/2和QIC磁带的支持也在此菜单下配置。

15. 文件系统

本菜单下最重要的表项是次要扩展文件系统支持；该文件系统是Red Hat Linux 6.0的本地文件系统。如果没有提供对它的支持，那么系统将无法启动。如果没有提供ISO9660文件系统支持，那么CD-ROM将不可能被读取。DOS文件系统通常需要FAT支持，另外再加上一种DOS文件系统类型（例如MSDOS或VFAT）。对/proc文件系统的支持也是需要的。如果没有这种支持，则将导致很多程序的终止。在网络文件系统菜单下可以配置对NFS客户和服务器的支持——NFS服务支持还需要Code Maturity菜单下试验性驱动程序的支持。

China-pub.com 下载

16. 控制台驱动程序

此菜单下包括基本VGA的驱动程序，以及新的帧缓冲区配置。

17. 声音

此处进行声卡的配置。

18. 内核裁剪

本菜单中的选项通常只有在调试Linux内核时才会使用。除非用户要主动修改内核源代码，否则最好不要设置此菜单中的选项。

19. 装载/保存可选择的配置文件

将内核的配置参数保存到一个文件中是完全可行的。该文件可以移到另一台机器上，并装载到一个独立的内核构造环境中， Red Hat Linux内核源代码包的内核配置存储在本书所附光盘的/kernel/redhat.config文件中，它可以用这个菜单选项装载。该配置需要 initrd进程，所以如果可能的话，就要改变SCSI配置以将SCSI控制器的驱动程序直接编译到内核中。

当所有选项都正确配置结束后，退出菜单。这时系统将提示是否放弃内核配置的改变。若全部正确则回答“ No ”。

10.2 编译内核

一旦内核配置完成以后，就要运行下列两条命令以清理源代码树和产生相关文件：

```
make dep
make clean
```

此时，产生新内核的一种方法是使用如下命令：

```
make zlilo
```

这个命令将产生 /vmlinuz（压缩的内核映像）和 /System.map（内核符号表）两个文件。因为Red Hat Linux将内核安装在/boot目录下，因此新的内核不会被引导。使用新内核的一种方法是修改LILO使其从/vmlinuz中引导。下面是一个实际的 /etc/lilo.conf文件的例子，该文件的配置内容实现了内核以“ Red Hat方式 ”引导。这个LILO配置文件来自IDE系统，因此不使用initrd。

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.2.5-15
     label=linux
     root=/dev/hda5
     read-only
other=/dev/hda1
     label=dos
     table=/dev/hda
```

上述的/etc/lilo.conf文件可以被修改为引导/vmlinuz（如下面代码所示）。label描述了/boot目录下的内核，它改名成了 linuxold（label是在LILO提示时键入的关键字，以选择引导的操作系统）。这样，我们增加了一个以/vmlinuz描述的新内核映像stanza。

```
boot=/dev/hda
map=/boot/map
```

China-pub.com
下载


```
install=/boot/boot.b
prompt
timeout=50
image=/vmlinuz
      label=linux
      root=/dev/hda5
      read-only
image=/boot/vmlinuz-2.2.5-15
      label=linuxold
      root=/dev/hda5
      read-only
other=/dev/hda1
      label=dos
      table=/dev/hda
```

一旦该文件被重新配置后，运行/sbin/lilo以重新配置在新内核中的LILO启动管理器。

其他一些操作系统将内核映像放在完全分离的文件系统（如 /stand）中，在Red Hat Linux中应避免这样做。

为什么不应该使用make zlilo呢？有两个理由。第一，用户可以遵从Red Hat Linux 的习惯，将内核映像存储在boot目录下，第二， make zlilo可能运行失败，用户可能因此而埋怨内核映像太大了。

若内核映像太大，可以使用 make bzimage来利用bzip2压缩格式以进一步将内核压缩。然而，用户首先应确保所有不必要的驱动程序已从内核映像中删除。

在成功运行了make bzimage命令后，压缩后的内核拷贝将存贮在文件 /usr/src/linux/arch/i386/boot/bzimage中，同时System.map存贮在目录/usr/src/linux中，用户可以将这两个文件移动到希望的位置（根目录或者/boot目录，并按要求重新配置LILO）。

应该在所有的模块和 initrd（如果需要的话）都安装之后才重启系统。在启动系统并对新内核测试满意后，在/boot目录下的通用内核就可以被删除了。小心，不要删除 LILO需要的文件，不要改变/boot目录下的boot.d或map文件。

10.3 编译模块

若在内核配置中启用了模块，则模块的产生过程将与内核准备相分离。在 /usr/src/linux目录下键入如下命令：

```
make modules
```

在模块被编译后，可以用如下的命令安装：

```
make modules_install
```

这条命令将把/lib/modules下的模块安装到与内核版本同名的目录中。当把内核移动到另一个Linux系统中时，这个目录也必须被移动并放到目标系统的相同位置。

10.4 创建initrd

若用户使用的是SCSI系统，并且决定将SCSI适配器支持作为一个模块，那么就需要initrd。使用下面的命令可以产生initrd。

```
mkinitrd initrd 2.2.12
```

下面是说明使用initrd的/etc/lilo.conf文件的例子。注意Red Hat Linux安装时产生的通用内

China-pub.com 下载

核被保留了。

```
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/vmlinuz
      label=linux
      root=/dev/sda1
      initrd=/initrd
      read-only
image=/boot/vmlinuz-2.2.5-15smp
      label=linuxold
      root=/dev/sda1
      initrd=/boot/initrd-2.2.5-15smp.img
      read-only
image=/boot/vmlinuz-2.2.5-15
      label=linux-up
      root=/dev/sda1
      initrd=/boot/initrd-2.2.5-15.img
      read-only
```

10.5 使用勘误内核

如果用户不想在构造自己的内核时出现问题，那么可以使用 Red Hat邮寄出的勘误内核。本书发行时，Red Hat希望用户使用的最新内核版本是2.2.5，构造22。该版本取代了存在允许远程用户崩溃内核的2.2.5，构造15版本。

在前一个Linux发布中，Red Hat Linux的安装包括了一个为80386处理器编译的独立内核RPM。从Red Hat Linux 6.0开始，Red Hat Linux中就包含了针对各种型号的机器的几种内核。下面就是在勘误内核中的内核软件包：

kernel-2.2.5-22.i386.rpm：针对i386、i486和其他非Pentium系统。

kernel-2.2.5-22.i586.rpm：针对Pentium和其他类Pentium的CPU。

kernel-smp-2.2.5-22.i586.rpm：针对Pentium SMP结构。

kernel-2.2.5-22.i686.rpm：针对i686（包括PentiumII和Celeron）CPU。

kernel-smp-2.2.5-22.i686.rpm：针对i686的SMP结构。

Red Hat针对内核升级提供了一个很长的步骤表，但实际上不必那么复杂。在下面的例子中，假定有一个单处理器Pentium系统需要升级。键入如下的命令：

```
/sbin/modprobe loop.o
rpm -ivh kernel-2.2.5-22.i586.rpm
/sbin/mkinitrd /boot/initrd-2.2.5-22.img 2.2.5-22
```

如果用户胆子足够大（甚至说是比较疯狂），那么就可以使用“ Uvh ”来取代原有的RPM。这样的操作将破坏系统中现有的内核，并使得 LILO无法启动系统，除非经过重新配置：

```
/sbin/modprobe loop.o
rpm -Uvh kernel-2.2.5-22.i586.rpm
/sbin/mkinitrd /boot/initrd-2.2.5-22.img 2.2.5-22
```

IDE用户用“ Uvh ”作选项可能比SCSI用户使用该选项更为实用。因为如果某种原因导致

China-pub.com
下载

某个回送设备不可用，那么mkinitrd将会出现故障。

在任何情况下，当用户试图使用上面两种代码段之一时，都应该确保已有一张启动软盘。如果启动盘遗失了，那么可以用如下的命令在升级之前创建一张启动盘：

```
mkbootdisk --device /dev/fd0 2.2.5-15
```

已经知道/dev/fd0对应PC上的A盘驱动器，并不证明已作好了升级的准备；还应该知道如果在RPM内核升级命令中使用了“ Uvh ”选项，那么启动盘上的loop.o内核模块将不可用（如果系统崩溃了）。当然，可以将该模块从更旧的 RPM中取出。最好不要使用“ Uvh ”选项，除非可以确保内核升级后的首次运行正确。还可以考虑在启动之前从旧内核中留下 loop.o的拷贝（该文件在/lib/modules目录中）。

上面的mkinitrd命令对IDE系统而言并不很重要，但是在 SCSI系统中却十分重要。如果在initrd中没有内核模块，那么系统将无法启动。

一旦新内核被安装后，系统就处于不稳定和不安全的状态中，应该立即重新配置 LILO。在内核升级和LILO重新配置之间进行系统重启，其后果可能不仅仅是一点点不方便，很可能是灾难性的。对 IDE系统的升级通常只需要简单地修改 /etc/lilo.conf文件中的内核映像名，如清单10-2所示。

清单10-2 为勘误内核修改的/etc/lilo.conf文件

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.2.5-22
      label=linux
      root=/dev/hda5
      read-only
other=/dev/hda1
      label=dos
      table=/dev/hda
```

如果没有使用“ Uvh ”选项，那么旧的内核可以作为一个备份设置下来：

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.2.5-22
      label=linux
      root=/dev/hda5
      read-only
image=/boot/vmlinuz-2.2.5-15
      label=linuxold
      root=/dev/hda5
      read-only
other=/dev/hda1
      label=dos
      table=/dev/hda
```

上面的这段代码可以使用户在 LILO提示时，键入 linux重引导新内核启动，而键入

China-pub.com 下载

Linuxold则启动旧的内核（除DOS表项以外）。

在SCSI系统中存在着一些细微的差别，因为需要设置 initrd，如下所示：

```
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.2.5-22
        label=linux
        root=/dev/sda1
        initrd=/boot/initrd-2.2.5-22.img
        read-only
image=/boot/vmlinuz-2.2.5-15
        label=linuxold
        root=/dev/sda1
        initrd=/boot/initrd-2.2.5-15.img
        read-only
```

一旦新的/etc/lilo.conf文件保存以后，用户千万不要忘了运行下面的命令：

```
/sbin/lilo
```

在重启系统之前，没有重新运行 LILO是引起Linux系统管理员恐惧的主要原因，它对 IDE和SCSI系统用户的影响是很相似的。

如果用户并没有被上面的介绍吓着的话，那么就可以准备开始改变内核了。希望能够很幸运地成功改变内核。记住，必须确保启动盘的完好！希望 LILO永远不出现故障。