

系统引导器

内容提要

1. 理解GRUB的作用
2. 掌握GRUB三种操作界面的使用方法
3. 熟悉GRUB的配置文件及其常用命令

启动引导器是保存在硬盘引导扇区中，负责引导硬盘分区中指定操作系统启动的程序代码。不同的操作系统会使用不同的启动引导器程序来引导自己的系统启动，操作系统的安装程序通常会在安装过程中将启动引导器安装到硬盘的主引导扇区（MBR）或硬盘分区的引导扇区中。

Linux系统一般可以采用LILO和GRUB两种启动引导程序，LILO由于功能相对简单，目前已经很少被Linux发行版采用了，CentOS 5系统中使用了GRUB作为缺省的启动引导器。

GRUB 功能简介

GRUB (Grand Unified Boot Loader)是功能强大的启动引导器，不仅可以对各种发行版本的Linux系统进行引导，也能够正常引导PC机上的其他常见的操作系统。

GRUB启动引导器有许多优秀的功能特性：

- GRUB提供了真正的命令行交互界面，用户能够灵活的使用各种命令引导操作系统和收集系统信息。与LILO和其他的启动引导器相比，GRUB提供了更丰富的功能
- GRUB支持使用配置文件对启动菜单的设置进行永久性保存，通过修改GRUB的配置文件，管理员可以自行定义系统启动菜单的功能
- GRUB提供了多系统启动的支持，除了可以引导各种版本的Linux操作，还可以引导硬盘中的DOS和Windows系统，实现了真正的多系统启动管理

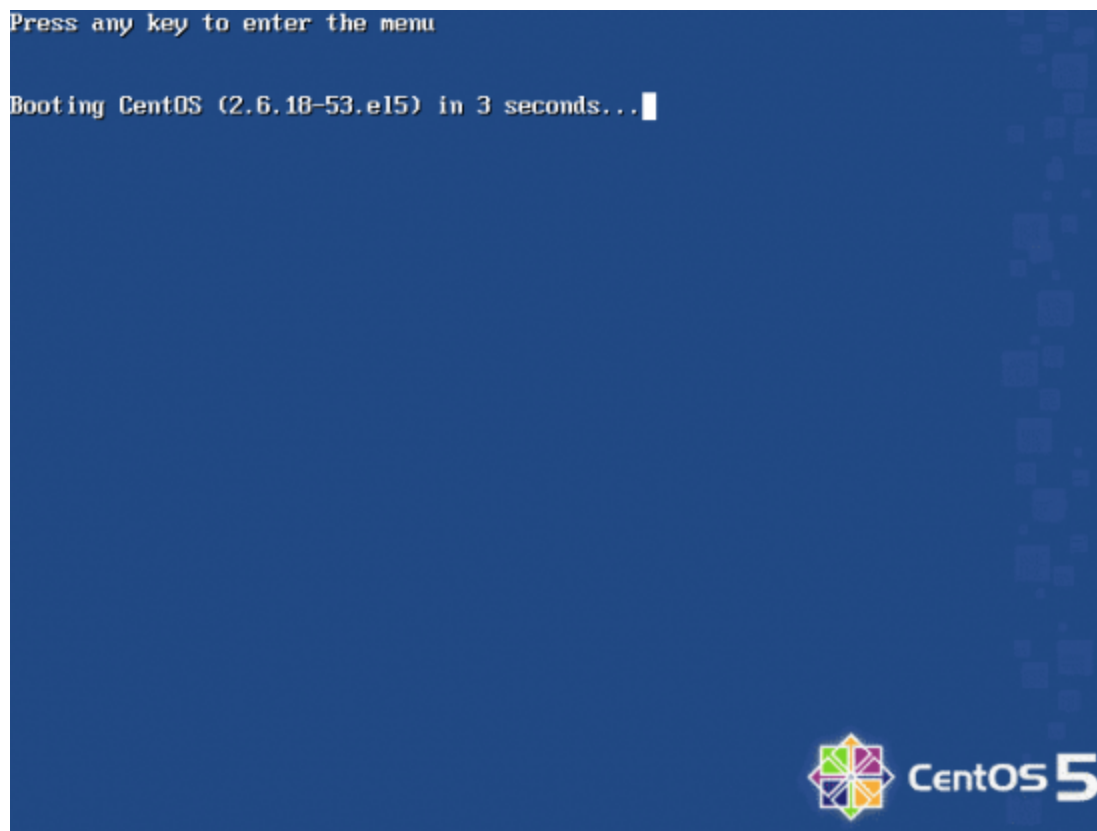
由于GRUB的功能强大，需要管理员掌握比其他启动引导器更多的交互命令和配置选项，对于Linux的初学者可能会感觉到有些“复杂”。

GRUB 的操作界面

在安装有 CentOS 系统的主机启动过程中，管理员可以通过GRUB交互界面对当前的启动过程进行干预，以实现不同于缺省启动的效果。

GRUB 的启动界面

在CentOS系统启动过程中，会出现如图所示的 GRUB 启动延迟画面。

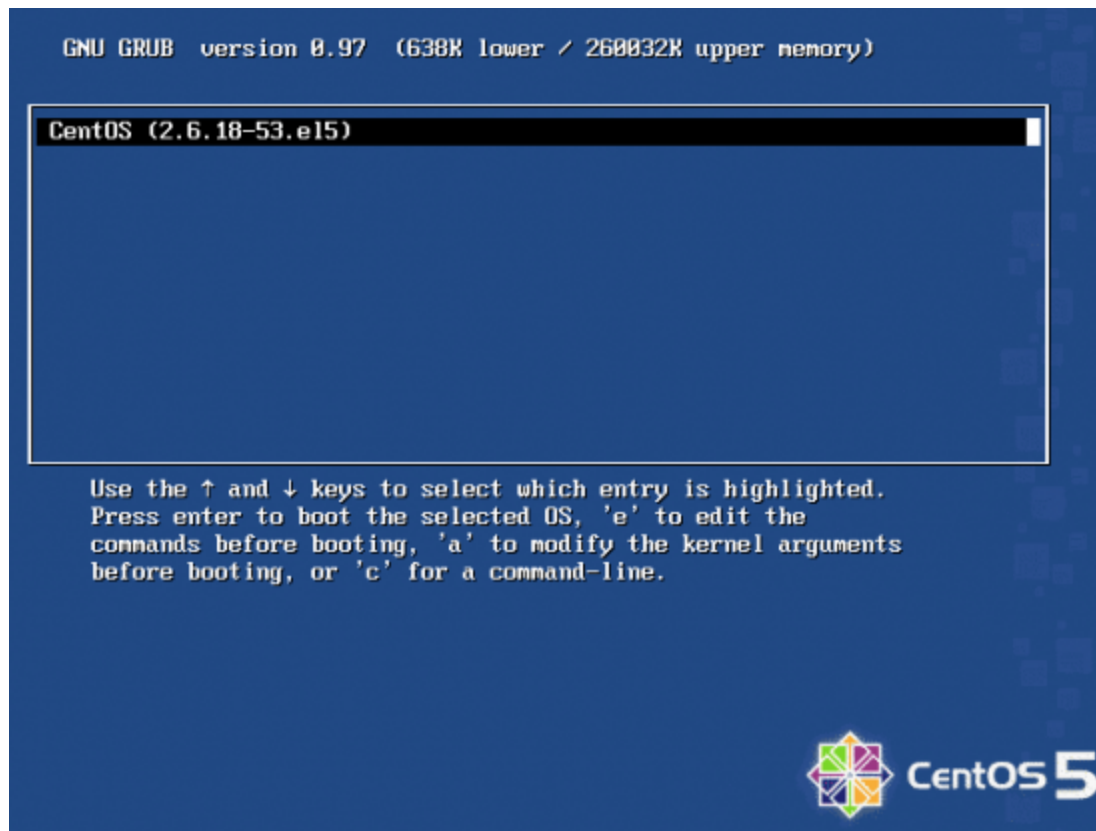


GRUB的隐藏菜单延时界面

该界面默认会持续显示5秒钟的时间，如果用户没有任何按键操作，5 秒钟后 GRUB 会按照缺省的启动项设置引导系统启动；如果在该界面显示过程中用户有任何的键盘按键动作，将显示 GRUB 的启动菜单。管理员如果需要进入 GRUB 交互界面，需要在上图所示的界面中按任意键。

GRUB 的启动菜单

CentOS 安装后缺省的 GRUB 启动菜单如图所示。



GRUB启动菜单

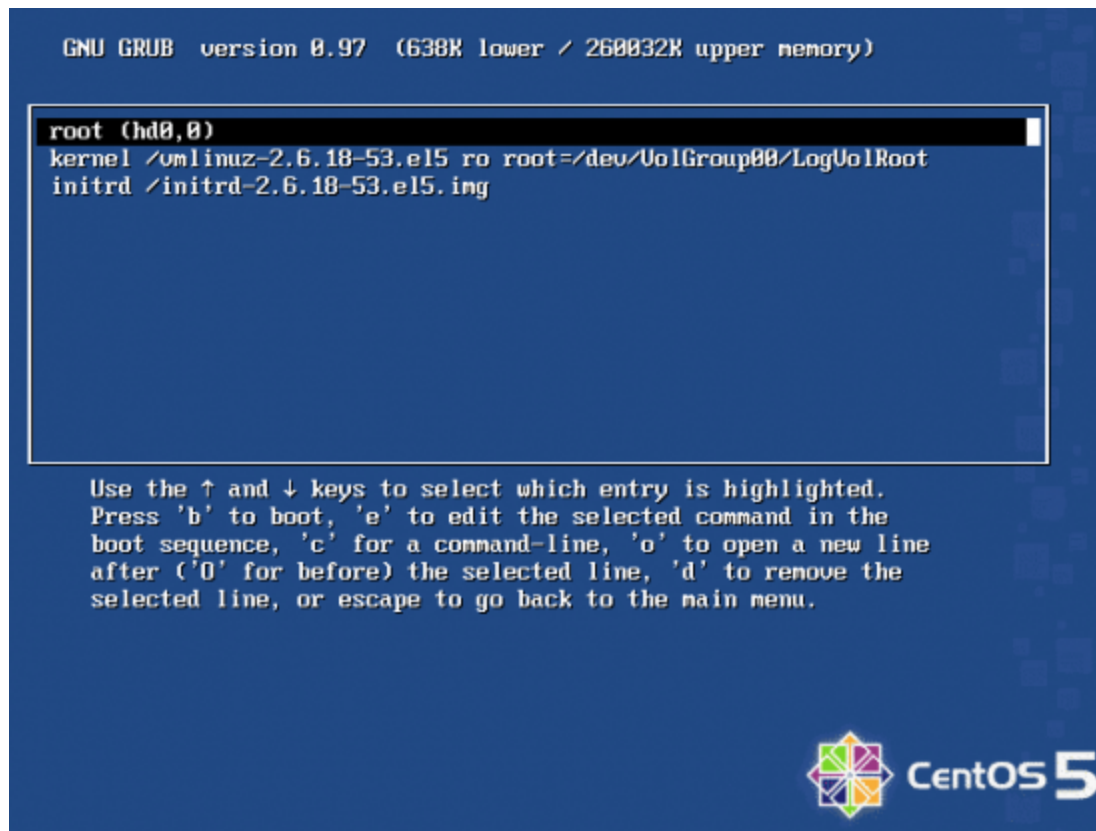
在该界面中只存在一个启动菜单项，用户可以使用相应的操作按键对启动菜单进行操作，下表说明了 GRUB 启动菜单的按键。

按键	说明
↑↓	使用上下箭头键，在启动菜单项间进行移动
enter	输入回车键启动当前的菜单项
e	选择“e”键编辑当前的启动菜单项
a	选择“a”键追加内核的启动参数
c	选择“c”键进入GRUB的命令行方式

在 GRUB 启动菜单界面中输入“e”键可以进入菜单项编辑界面。

GRUB 启动菜单项

进入到GRUB启动菜单项的编辑界面后，会显示该启动菜单项所包括的启动配置行，如图所示。



GRUB 启动项编辑界面

在该界面中输入相应的按键可以对配置行内容进行相应的操作，下表说明了 GRUB 菜单项编辑界面的按键。

按键	说明
↑↓	使用上下箭头键，选择菜单项中的行
b	选择“b”键启动当前的菜单项
e	选择“e”键编辑当前选中的行
c	选择“c”键进入GRUB的命令行方式
o	选择“o”键在当前行后面插入一行
O	选择“O”键当前行前面插入一行
d	选择“d”键删除当前行
Esc	选择“Esc”键返回 GRUB 启动菜单界面，取消对当前启动菜单项所做的任何修改

使用上下方向键选择需要编辑的配置行后，选择“e”键可进入配置行的编辑界面，如图所示。

```
[ Minimal BASH-like line editing is supported. For the first word, TAB
lists possible command completions. Anywhere else TAB lists the possible
completions of a device/filename. ESC at any time cancels. ENTER
at any time accepts your changes.]

grub edit> kernel /vmlinuz-2.6.18-53.el5 ro root=/dev/UolGroup00/LogUolRoot
```



启动项配置行编辑

配置行编辑界面属于行编辑界面，支持左右方向键、退格键和删除键等基本行编辑操作。在完成配置行编辑内容后，输入“回车键”将确认已进行的编辑并返回菜单项编辑界面。

在返回菜单项编辑界面后，如果已经完成了对所有配置行内容的修改，输入“b”键将按照当前的配置内容引导系统启动。

在 GRUB 交互界面对启动菜单项进行的修改只会作用于本次启动，并不会将已修改的内容自动保存到系统中，因此如果需要永久性的更改启动项的配置，需要在 GRUB 的配置文件中进行修改。

GRUB 命令行界面

GRUB 交互界面除了可以提供友好的菜单界面供用户选择配置，还提供了功能全面的命令行界面供熟悉命令的管理员对 GRUB 启动项进行配置管理。在 GRUB 启动菜单或启动菜单项配置界面中输入“c”键将进入 GRUB 的命令行界面，如图所示。

```
GNU GRUB version 0.97 (638K lower / 268832K upper memory)

[ Minimal BASH-like line editing is supported. For the first word, TAB
  lists possible command completions. Anywhere else TAB lists the possible
  completions of a device/filename. ESC at any time exits.]

grub> help
background RRGGBB                blocklist FILE
boot                               cat FILE
chainloader [--force] FILE        clear
color NORMAL [HIGHLIGHT]          configfile FILE
displayapm                        displaymen
find FILENAME                     foreground RRGGBB
geometry DRIVE [CYLINDER HEAD SECTOR] [ halt [--no-apm]
help [--all] [PATTERN ...]        hide PARTITION
initrd FILE [ARG ...]             kernel [--no-men-option] [--type=TYPE]
makeactive                        map TO_DRIVE FROM_DRIVE
ndScript                          module FILE [ARG ...]
modulenounzip FILE [ARG ...]      pager [FLAG]
partnew PART TYPE START LEN       parttype PART TYPE
reboot                            root [DEVICE [HDBIAS]]
rootnoverify [DEVICE [HDBIAS]]    serial [--unit=UNIT] [--port=PORT] [--
setkey [TO_KEY FROM_KEY]          setup [--prefix=DIR] [--stage2=STAGE2_
splashimage FILE                  terminal [--dumb] [--no-echo] [--no-ed
terminfo [--name=NAME --cursor-address testobe MODE
unhide PARTITION                  uppermen KBYTES
vbeprobe [MODE]

grub> 
```

GRUB 命令行界面

GRUB 命令行界面的提示符是“grub>”，在该提示符下可输入 GRUB 支持的交互命令，并可回车后执行命令。

下表列出了可以在 GRUB 命令行操作界面下使用的常用命令及其功能。

命令	功能
boot	引导先前已经被指定并载入的操作系统或链式装载程序。
chainloader	将指定的文件作为一个链式装载程序载入，将启动控制权软交给另外的启动引导器。为了获取在一个指定分区第一扇区内的文件，使用+1 作为文件名。
displaymem	显示当前内存的使用情况，这个信息是基于 BIOS 的。这个命令有助于确定系统在引导前有多少内存。
initrd	使用户能够指定一个在引导时可用的初始 RAM 盘。当内核为了完全引导而需要某些模块时，这是必需的。
install p	安装 GRUB 到系统的主引导记录。这个命令允许系统重启时出现 GRUB 接口。如果命令被执行，那么除了 GRUB 信息之外的其他用于引导其他操作系统的信息都将丢失。
kernel	当使用直接载入方式引导操作系统时，kernel 命令指定内核从 GRUB 的根文件系统中载入。其后可以跟 options 选项使在内核载入时传给内核。

GRUB 命令行界面提供了方便友好的命令行交互方式，其主要特点包括：

- 提供在线帮助命令“help”，并且可以获得每条命令的详细帮助。
- 可使用左右方向键编辑行命令
- 可使用上下方向键滚动历史命令
- 可使用“Tab”键补全命令和路径

由以上的特性可以看出，GRUB 的命令行界面功能非常类似于 Bash，可以作为一个用于系统引导的小型 Shell 环境。

在 GRUB 的命令行界面中，不仅可以使 help 命令显示支持的命令列表，还可以将某个命令作为 help 命令的参数来获得关于该命令的具体帮助信息。例如：

```
grub> help kernel
kernel: kernel [--no-mem-option] [--type=TYPE] FILE [ARG ...]
  Attempt to load the primary boot image from FILE. The rest of the
  line is passed verbatim as the "kernel command line". Any
  modules must be reloaded after using this command. The option
  --type is used to suggest what type of kernel to be loaded. TYPE
  must be either of "netbsd", "freebsd", "openbsd", "linux",
  "biglinux" and "multiboot". The option --no-mem-option tells GRUB
  not to pass a Linux's mem option automatically.
```

GRUB 命令行环境中的 `clear`、`cat`、`reboot`、`halt` 等命令与 `Bash` 中的同名命令可实现相同的功能。

GRUB 的配置文件

GRUB 的启动菜单是完全按照 GRUB 的配置文件内容显示和执行的，下面来学习 GRUB 的配置文件。

配置文件的位置和名称

GRUB 启动引导器的相关文件被缺省保存在 `/boot/grub` 目录中，其中 `menu.lst` 文件是 GRUB 的主配置文件，GRUB 在启动时就是根据该文件的配置内容显示启动菜单的。

```
# ls /boot/grub/
device.map      grub.conf      minix_stagel_5  stage2
e2fs_stagel_5   iso9660_stagel_5 reiserfs_stagel_5 ufs2_stagel_5
fat_stagel_5    jfs_stagel_5    splash.xpm.gz   vstafs_stagel_5
ffs_stagel_5    menu.lst        stagel          xfs_stagel_5
```

在 Red Hat 系列的 Linux 系统中（包括 CentOS）更习惯于使用文件名 `grub.conf` 来作为 GRUB 的配置文件。为了保持与其他系统的兼容性和文件的一致性，CentOS 分别在 `/boot/grub` 和 `/etc` 两个目录中建立了名为 `grub.conf` 的符号链接文件指向真正的配置文件 `/boot/grub/menu.lst`。

```
# cd /boot/grub
# ls -l menu.lst grub.conf
-rw----- 1 root root 595 Mar 31 18:06 grub.conf
lrwxrwxrwx 1 root root 11 Mar 31 18:06 menu.lst -> ./grub.conf
# ls -l /etc/grub.conf
lrwxrwxrwx 1 root root 22 Mar 31 18:06 /etc/grub.conf -> ../boot/grub/grub.conf
```

因此，无论直接编辑 `menu.lst` 文件，还是编辑相应目录中的 `grub.conf` 文件都可以达到修改 GRUB 配置的目的。

CentOS 5 中 GRUB 的缺省配置

CentOS 5 服务器方式缺省安装后，文件 `menu.lst` 已经进行了缺省的配置。

```
## 执行 cat 命令后显示 menu.lst 文件的缺省配置
# cat /boot/grub/menu.lst
```

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE: You have a /boot partition. This means that
#           all kernel and initrd paths are relative to /boot/, eg.
#           root (hd0,0)
#           kernel /vmlinuz-version ro root=/dev/VolGroup00/LogVolRoot
#           initrd /initrd-version.img
#boot=/dev/hda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
```

```
hiddenmenu
title CentOS (2.6.18-53.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-53.el5 ro root=/dev/VolGroup00/LogVolRoot
    initrd /initrd-2.6.18-53.el5.img
```

menu.lst 文件中的内容可以分为三类：

- 注释行通常放在文件的首部，以“#”开始，起到说明的作用；由于文件中的注释行在GRUB执行时会被忽略，因此也可以将暂时不使用的配置行加入“#”到行首作为注释，以使其失效而不必急于删除该行。
- 全局配置行通常放在文件首部注释行的后面，其作用范围是整个GRUB运行环境（包括所有菜单项）；不包括在title命令区域内的所有行都被视为全局配置行。
- 菜单项配置行以title命令开始，直到下一个title命令或文件末尾结束，其包括的所有配置行都作用于同一个启动项，该启动项由title命令的参数命名；配置文件中可以具有多个菜单项配置，用于多系统启动的配置。

menu.lst 的配置命令

下表列出了一些常用的可在grub.conf中使用的配置命令。

配置命令	功能
default <菜单项序号>	设置启动菜单的默认启动项,菜单项序号参数其值从“0”开始，即“0”代表第1个菜单项，“1”代表第2个菜单项等等。
timeout <延时秒数>	设置 GRUB 菜单的启动延时（以秒为单位），即经过该延时后 GRUB 将自动启动默认的菜单项。当 timeout 命令设置的延时秒数为负数或不设置 timeout 命令时，GRUB 启动菜单将取消启动延时功能，用户需手工选择要启动的菜单项。
title <标题>	设置用来装载一个操作系统的一组特定命令的标题，标题将在 GRUB 启动菜单中显示。
splashimage <图像文件全路名>	指定某文件作为启动菜单的背景图像，该文件必须是 xpm 格式的图像，可以是 gz 压缩文件。
root <GRUB 的设备名称>	设置 GRUB 的根分区（root），通常是文件系统的 boot 分区，即 Linux 内核所在的分区。
rootnoverify <GRUB 的设备名称>	做 root 命令同样的事情，只是不挂装分区（用于启动 Windows 系统）。
kernel	用 kernel 命令后跟 Linux 内核文件作为参数，加载 Linux 的内核文件。
initrd	用 initrd 命令加载初始 RAM 磁盘镜像文件。
map	映射磁盘或分区在 BIOS 中的标志符，用于启动在第二个硬盘中的 Windows。
makeactive	设置根设备为活动分区（针对于Windows不能从非活动分区启动）。
chainloader	使用 chainloader 命令调用Windows分区的启动引导器引导 Windows 系统。

GRUB 配置文件举例

选择启动不同的 Linux 内核

要完成对 Linux 操作系统的引导至少要有title、root、kernel、initrd 四条命令。对于不同的 Linux 操作系统（或同一系统的不同内核版本），需要根据内核文件实际的位置和名称进行启动行的配置，即title、root、kernel和initrd 四条命令的命令参数需要具体设定。下面是一个更新系统内核后的 GRUB 配置例子，可以选择启动不同的内核。

```
# 设置默认启动第一个启动项
default=0
# 设置 GRUB 菜单的启动延时为 5 秒
timeout=5
# 指定启动菜单的背景图像位于 /boot/grub/splash. xpm. gz
splashimage=(hd0,0)/grub/splash. xpm. gz
# 设置 GRUB 菜单为隐藏
hiddenmenu
```



```
# title 命令开始一个菜单项的配置
title CentOS (2.6.18-53.1.13.el5)
    # 设置 GRUB 的根分区 (即 /boot 分区)
    root (hd0,0)
    # 设置 Linux 内核及参数
    kernel /vmlinuz-2.6.18-53.1.13.el5 ro root=/dev/VolGroup00/LogVolRoot
    # 设置初始 RAM 磁盘镜像文件
    initrd /initrd-2.6.18-53.1.13.el5.img

title CentOS (2.6.18-53.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-53.el5 ro root=/dev/VolGroup00/LogVolRoot
    initrd /initrd-2.6.18-53.el5.img
```

具有 Windows 的多系统启动配置

虽然在本书的 CentOS 服务器方式安装中，只安装了唯一的 Linux 操作系统，但是功能强大的 GRUB 启动引导器完全可以实现 Linux 和 Windows 共存的多系统启动。

由于 GRUB 不能够直接加载 Windows 系统的核心文件进行引导，而是通过调用 Windows 操作系统自身的启动引导器（通常为 Windows 所在分区的引导扇区），所以对于 Windows 系统的引导需使用与 Linux 系统引导不同的命令。Windows 的启动菜单配置经常会使用 map、root、makeactive、chainloader 等命令。

1、Linux 和 Windows 分别安装在两块硬盘的情况

在下面的实例中系统中的第一块硬盘中已经按照本书的步骤安装了 CentOS 系统，现在又添加了系统中的第二块硬盘，在该硬盘的第一个主分区中已经安装有能够在本机正常运行的 Windows 系统，剩下的任务就是在 menu.lst 配置文件中添加对 Windows 系统的引导配置内容。

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-53.el5)
    root (hd0,0)
    kernel /vmlinuz-2.6.18-53.el5 ro root=/dev/VolGroup00/LogVolRoot
    initrd /initrd-2.6.18-53.el5.img
title Windows
    map (hd0) (hd1)
    map (hd1) (hd0)
    # 设置真实的 Windows 分区为 GRUB 的根设备。
    rootnoverify (hd1,0)
    # 设置根设备为活动分区 (针对 Windows 不能从非活动分区启动)
    makeactive
    # 调用 Windows 分区的引导记录引导 Windows 系统启动
    chainloader +1
```

由于 Windows 系统自身只能安装到主硬盘（系统中的第一块硬盘）中的活动分区中进行引导启动，因此在配置行中需要使用两条 map 命令双向的映射两块硬盘在 BIOS 中的标志符（目的是为了迷惑 Windows 系统），并使用 makeactive 将 Windows 分区设置为活动分区，这样“chainloader +1”命令才能够使用 Windows 分区中的引导记录启动 Windows 系统。

2、Linux 和 Windows 安装在同一块硬盘的情况

下面是一个使用 GRUB 选择启动 Windows 或 CentOS 的 grub.conf 的另一个例子。在这个例子中，Windows 系统安装在硬盘的第一个主分区，Linux 的引导系统在磁盘的第一个逻辑分区上。

```
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-53.el5)
    root (hd0,5)
```

```
kernel /vmlinuz-2.6.18-53.el5 ro root=/dev/VolGroup00/LogVolRoot
initrd /initrd-2.6.18-53.el5.img
title Windows
rootnoverify (hd0,0)
chainloader +1
```

- 显示源文件
- 登录