

 **Bluemix** Desenvolva na nuvem em um clique![Comece seu trial gratuito](#)

developerWorks Brasil Itens Técnicos Linux Biblioteca técnica

Aprenda Linux, 101: Gerenciadores de boot

Apresentando GRUB e LILO

Saiba como selecionar e configurar um gerenciador de boot para o seu sistema Linux®. Use o material deste artigo para se preparar para o exame LPI 101 de certificação de administrador de sistema Linux ou simplesmente por diversão.

Ian Shields trabalha em vários dos projetos Linux para a zona Linux do developerWorks. Ele é um programador senior da IBM em Research Triangle Park, NC. Ele iniciou na IBM em Canberra, Austrália, como um Engenheiro de Sistemas em 1973 e, desde então, trabalhou em sistemas de comunicações e computação disseminada em Montreal, no Canadá, e em RTP na Carolina do Norte, nos Estados Unidos. Ele possui diversas patentes. Sua graduação é em matemática pura e filosofia na Australian National University. Ele possui mestrado em ciências e é doutor em ciências da computação na Universidade do Estado da Carolina do Norte.

07/Out/2011

Visão geral

Neste artigo, aprenda como projetar um layout de particionamento de disco para um sistema Linux. Aprenda a:

- Alocar sistemas de arquivos e espaço de troca para separar partições ou discos

- Padronizar o projeto para o uso pretendido do sistema

- Assegurar que o sistema possa ser inicializado

Este artigo discute o processo de boot do PC e dois importantes loaders de boot usados em Linux—LILO e GRUB—e ajuda você a se preparar para o Objetivo 102.2 no Tópico 102 do exame 101 do Linux Professional Institute's Junior Level Administration (LPIC-1). Também o apresentamos ao GRUB 2, embora não seja exigido para os objetivos do exame atual. O objetivo tem peso 2.

Pré-requisitos

Para tirar o máximo dos artigos desta série, é necessário ter conhecimento básico de Linux e possuir um sistema Linux funcional em que seja possível praticar os comandos aqui abordados. Você também deve estar familiarizado com o layout da unidade de disco rígido, conforme discutido no artigo "[Aprenda Linux, 101: Layout do disco rígido](#)". Algumas vezes, é possível que versões diferentes de determinado programa formatem a saída de modo diferente, assim, pode ser que seus resultados não sejam exatamente iguais às listas e figuras exibidas neste documento.

Sobre esta série

Esta série de artigos destina-se a elucidar as tarefas de administração de sistema Linux. Também é possível usar os materiais nestes artigos como preparação para os exames [Linux Professional Institute Certification nível 1 \(LPIC-1\)](#).

Consulte a nossa [série roteiros](#) para obter a descrição e o link para cada um dos artigos nesta série. O roteiro está em andamento e reflete os últimos objetivos (abril de 2009) dos exames LPIC-1: à medida que formos concluindo estes artigos, eles serão adicionados ao roteiro. Enquanto isso, é possível achar versões anteriores do mesmo material, que dão suporte aos objetivos do LPIC-1 anteriores a abril de 2009, em nossos [tutoriais de preparação para o exame de certificação LPI](#).

Observação: As imagens deste artigo são capturas de tela retiradas de um processo de boot anterior. Se você estiver lendo este artigo usando um leitor de tela, pode considerar útil ter os arquivos de configuração correspondentes disponíveis para referência; faça o download deles na seção [Download](#), mais adiante neste artigo.

Visão geral do processo de boot

Antes de falarmos de LILO e GRUB, vamos rever como um PC é iniciado ou *inicializado*. O código denominado *BIOS* (de *Basic Input Output Service*) é armazenado em memória não volátil, como ROM, EEPROM ou memória flash. Quando o PC é ligado ou reinicializado, esse código é executado. Em geral, ele realiza um autoteste inicial (POST) para verificar a máquina. Por fim, ele carrega o primeiro setor a partir do registro de boot mestre (MBR) na unidade de boot.

Conforme discutido no artigo "[Learn Linux, 101: Layout do disco rígido](#)", o MBR também tem a tabela de partições, para que a quantidade de código executável no MBR seja inferior a 512 bytes, o que não é muito código. Observe que cada disco, mesmo um flexível, possui código executável em seu MBR, mesmo se o código for apenas suficiente para apresentar uma mensagem, como "Disco não inicializável na unidade A:". Esse código, carregado pelo BIOS a partir desse primeiro setor, é chamado de *loader de boot de primeiro estágio* ou de *loader de boot de estágio 1*.

O MBR de unidade de disco rígido padrão usado pelos sistemas operacionais MS DOS, PC DOS e Windows® verifica a tabela de partições para localizar uma partição primária na unidade de boot marcada como *ativa*, carrega o primeiro setor a partir dessa partição e passa o controle para o começo do código carregado. Essa nova porção de código é também conhecida como o *registro de boot de partição*. Esse registro é na verdade outro loader de boot de estágio 1, mas esse tem inteligência suficiente apenas para carregar um conjunto de blocos da partição. O código neste novo conjunto de blocos é chamado de *loader de boot de estágio 2*. Conforme usado por MS-DOS e PC-DOS, o loader de estágio 2 continua diretamente para carregar o restante do sistema operacional. Esse é o modo como o seu sistema operacional entra em operação pela sua autoinicialização até estar funcionando normalmente.

Isso funciona bem para um sistema com um único sistema operacional. O que acontecerá se você quiser vários sistemas operacionais, digamos, OS/2, Windows XP e três distribuições Linux diferentes? Você *poderá* usar algum programa (como o DOS FDISK) para alterar a partição ativa e reinicializar. Isso é incômodo. Além do mais, um disco pode ter apenas quatro partições principais, e o MBR padrão pode ter apenas uma partição primária ativa; não se pode fazer o boot a partir de uma partição lógica. Mas, nosso exemplo hipotético citou cinco sistemas operacionais, e cada um precisa de uma partição. Opa!

A solução está em usar algum código especial que permita que um usuário escolha em qual sistema operacional será feito o boot. Exemplos incluem:

Loadlin

Um programa executável DOS invocado de um sistema DOS em execução para fazer o boot de uma

Entre em contato com Ian

Ian é um dos nossos autores mais populares e produtivos. Pesquise [todos os artigos do Ian](#) no developerWorks. Confira [o perfil de Ian](#) e entre em contato com ele, com outros autores e leitores no My developerWorks.

partição Linux. Isso era popular quando a configuração de um sistema multiboot era um processo complexo e arriscado.

OS/2 Boot Manager

Um programa que é instalado em uma pequena partição dedicada. A partição foi marcada como ativa, e o processo de boot de MBR padrão iniciou o OS/2 Boot Manager, que apresenta um menu permitindo escolher qual sistema operacional inicializar.

Um loader de boot inteligente

Um programa que pode residir em uma partição de sistema operacional e é invocado pelo registro de boot da partição de uma partição ativa ou pelo registro de boot mestre. Exemplos incluem:

BootMagic, parte do Norton PartitionMagic

LILO, o LOader do Linux

GRUB, o loader GRand Unified Boot (agora chamado de GRUB Legacy)

GRUB 2, um novo loader de boot que acaba de começar a aparecer em distribuições comuns

Evidentemente, se você puder obter o controle do sistema para um programa que tenha mais de 512 bytes de código para realizar essa tarefa, não será muito difícil permitir o boot de partições lógicas ou de partições que não estejam na unidade de boot. Todas essas soluções permitem essas possibilidades, porque podem carregar um registro de boot de uma partição arbitrária ou porque eles têm algum entendimento de qual arquivo ou quais arquivos carregar para iniciar o processo de boot.

Carregamento em cadeia

Quando um gerenciador de boot obtém o controle, uma das coisas que ele pode carregar é outro gerenciador de boot. Isso é chamado *carregamento em cadeia*, e ocorre mais frequentemente quando o gerenciador de boot que está localizado no registro de boot mestre (MBR) carrega o loader de boot que está no registro de boot de partição. Isso quase sempre é feito quando um loader de boot Linux é solicitado a inicializar uma partição Windows ou DOS, mas também pode ocorrer quando o loader de boot Linux de um sistema é solicitado a carregar o loader de boot de outro sistema. Veja a seguir exemplos que mostram quando é possível usar isso para vários sistemas Linux.

Loaders de boot Linux

Daqui em diante, nos concentraremos em LILO e GRUB, já que esses são os loaders de boot incluídos na maioria das distribuições Linux. LILO já existe há algum tempo. GRUB é mais recente. O GRUB original agora se tornou *GRUB Legacy*, e o GRUB 2 está sendo desenvolvido sob o patrocínio da Free Software Foundation (consulte [Recursos](#) para obter detalhes). Discutiremos o GRUB 2 brevemente, principalmente para que você tenha uma ideia das principais diferenças e de como o GRUB e o GRUB 2 podem coexistir. Para o restante do artigo, consideramos que GRUB significa GRUB Legacy, a menos que o contexto especificamente indique o GRUB 2. Também está disponível uma nova versão do LILO, chamada *ELILO* (que é projetada para sistemas de boot que usam *Extensible Firmware Interface* ou *EFI* da Intel, em vez de BIOS). Consulte [Recursos](#) para obter informações adicionais sobre GRUB 2 e ELILO.

O processo de instalação de sua distribuição provavelmente dará a você uma opção de qual loader de boot configurar. Tanto o GRUB como o LILO funcionarão com a maioria dos discos modernos, embora

algumas distribuições, como Fedora, não forneçam mais o LILO. Lembre-se de que a tecnologia de disco tem avançado rapidamente, portanto, você deve sempre ter certeza de que o loader de boot, a distribuição Linux escolhida (ou outro sistema operacional) e o BIOS de sistema funcionarão com o seu disco novo em folha. Caso contrário, poderá ocorrer perda de dados. Da mesma forma, se você estiver adicionando uma nova distribuição a um sistema existente, pode ser necessário certificar-se de que você tenha o LILO ou o GRUB mais recente em seu MBR. Também será necessária uma versão recente o bastante do GRUB ou do LILO, case você planeje fazer o boot a partir de um disco LVM ou RAID.

Os loaders de estágio 2 usados em LILO e GRUB permitem escolher quais sistemas operacionais ou versões carregar dentre inúmeras opções. Entretanto, LILO e GRUB diferem bastante no fato de que uma mudança no sistema requer o uso de um comando para recriar a configuração de boot LILO sempre que você atualizar um kernel ou efetuar outras alterações específicas no seu sistema, ao passo que GRUB pode fazer isso por meio de um arquivo de texto de configuração editável. O GRUB 2 também requer uma reconstrução de um arquivo de configuração que geralmente está armazenado em /etc.

LILO

LILO, ou Linux LOader, é um dos loader de boot Linux mais comuns. O LILO pode ser instalado no MBR da unidade de disco rígido inicializável ou no registro de boot de partição de uma partição. Também é possível instalá-lo em dispositivos removíveis, como disquetes, CDs ou chaves USB. Será uma boa ideia praticar em um disco flexível ou chave USB se ainda não estiver familiarizado com LILO, como em nossos exemplos.

Durante a instalação do Linux, normalmente você especificará LILO ou GRUB como gerenciador de boot. Se escolher GRUB, LILO não poderá estar instalado. Caso ainda não tenha instalado o pacote, será preciso instalá-lo. Também é recomendável instalar o pacote lilo-doc para obter a documentação adicional e mais exemplos. Daqui em diante, consideraremos que você já tenha o pacote LILO instalado. Consulte o [roteiro da nossa série](#) para ver os artigos sobre gerenciamento de pacotes, caso precise de ajuda.

Configure o LILO usando um arquivo de configuração, que geralmente é /etc/lilo.conf. É possível usar o comando `liloconfig` (normalmente disponível em /usr/sbin) para gerar um arquivo de configuração inicial e, em seguida, editá-lo conforme necessário. O arquivo de configuração na Listagem 1 foi gerado dessa forma. O arquivo é razoavelmente bem anotado, e as man pages para lilo e lilo.conf fornecerão ajuda adicional. Este é um típico arquivo de configuração LILO que pode ser usado em um sistema de boot duplo com Windows e um ou mais sistemas Linux.

Listagem 1. Exemplo de /etc/lilo.conf

```
# Originally generated by liloconfig - modified by Ian Shields

# This allows booting from any partition on disks with more than 1024
# cylinders.
lba32

# Specifies the boot device (floppy)
boot=/dev/fd0

# Specifies the device that should be mounted as root.
# If the special name CURRENT is used, the root device is set to the
# device on which the root file system is currently mounted. If the root
# has been changed with -r , the respective device is used. If the
# variable ROOT is omitted, the root device setting contained in the
# kernel image is used. It can be changed with the rdev program.
root=/dev/sda7
```

```

# Bitmap configuration for /boot/coffee.bmp
bitmap=/boot/coffee.bmp
bmp-colors=12,,11,15,,8
bmp-table=385p,100p,1,10
bmp-timer=38,2,13,1

# Enables map compaction:
# Tries to merge read requests for adjacent sectors into a single
# read request. This drastically reduces load time and keeps the map
# smaller. Using COMPACT is especially recommended when booting from a
# floppy disk.
compact

# Install the specified file as the new boot sector.
# LILO supports built in boot sectors, you only need
# to specify the type, choose one from 'text', 'menu' or 'bitmap'.
# new: install=bmp      old: install=/boot/boot-bmp.b
# new: install=text    old: install=/boot/boot-text.b
# new: install=menu    old: install=/boot/boot-menu.b or boot.b
# default: 'menu' is default, unless you have a bitmap= line
# Note: install=bmp must be used to see the bitmap menu.
# install=menu
install=bmp

# Specifies the number of _tenths_ of a second LILO should
# wait before booting the first image. LILO
# doesn't wait if DELAY is omitted or if DELAY is set to zero.
# delay=20

# Prompt to use certain image. If prompt is specified without timeout,
# boot will not take place unless you hit RETURN. Timeout is in tenths of
# a second.
prompt
timeout=200

# Enable large memory mode.
large-memory

# Specifies the location of the map file. If MAP is
# omitted, a file /boot/map is used.
map=/boot/map

# Specifies the VGA text mode that should be selected when
# booting. The following values are recognized (case is ignored):
# NORMAL select normal 80x25 text mode.
# EXTENDED select 80x50 text mode. The word EXTENDED can be
# abbreviated to EXT.
# ASK stop and ask for user input (at boot time).
# <number> use the corresponding text mode. A list of available modes
# can be obtained by booting with vga=ask and pressing [Enter].
vga=normal

# Defines non-standard parameters for the specified disk.
#disk=/dev/sda
# bios=0x80

# If you are using removable USB drivers (with mass-storage)
# you will need to tell LILO to not use these devices even
# if defined in /etc/fstab and referenced in /proc/partitions.
# Adjust these lines to your devices:
#
# disk=/dev/sda inaccessible
# disk=/dev/sdb inaccessible

# These images were automatically added. You may need to edit something.

image=/boot/vmlinuz-2.6.31-14-generic
label="Lin 2.6.31-14"
initrd=/boot/initrd.img-2.6.31-14-generic
read-only

image=/boot/vmlinuz-2.6.31-20-generic
label="Lin 2.6.31-20"
initrd=/boot/initrd.img-2.6.31-20-generic
read-only

image=/boot/memtest86+.bin
label="Memory Test+"
read-only

# If you have another OS on this machine (say DOS),
# you can boot if by uncommenting the following lines
# (Of course, change /dev/sda1 to wherever your DOS partition is.)
other=/dev/sda6
label="Fedora 8"

other=/dev/sda1
label="Windows XP"

```

Testaremos nossa configuração criando um disquete de boot, então especificamos `boot=/dev/fd0`, ao lado da parte superior do arquivo. Também aumentamos o tempo limite de prompt para 20 segundos (200 incrementos de um décimo de segundo) e acrescentamos uma segunda "outra" entrada ao

carregamento em cadeia do loader de boot na partição Fedora 8 (/dev/sda6). O LILO permite apenas um comando `root` em um arquivo de configuração, por isso pode inicializar várias imagens a partir da partição `root`, mas precisa fazer o carregamento em cadeia para outro loader de boot a fim de inicializar uma imagem de uma instalação diferente, como Fedora 8 (/dev/sda6) ou Windows XP (/dev/sda1) em nosso exemplo.

Use o comando `lilo`, localizado em `/sbin/lilo`, para escrever um registro de boot de estágio 1 e criar um arquivo de mapa (/boot/map) usando informações de configuração, como no exemplo acima, normalmente localizadas em `/etc/lilo.conf`. Existem alguns usos auxiliares que mencionaremos depois.

Agora, se formatarmos um disco flexível com um sistema de arquivos Linux como `ext2`, podemos executar o comando `lilo` (/sbin/lilo) para criar um disco flexível inicializável. Nossa saída é mostrada na Listagem 2. Observe que o comando `lilo` tem cinco níveis de detalhamento. Especifique um `-v` extra para cada nível. Esse exemplo foi criado usando `lilo -v -v`.

Listagem 2. Criando um disco flexível inicializável com lilo

```
LILO version 22.8, Copyright (C) 1992-1998 Werner Almesberger
Development beyond version 21 Copyright (C) 1999-2006 John Coffman
Released 19-Feb-2007, and compiled at 10:52:38 on Aug 25 2009
Running Linux kernel 2.6.31-14-generic on i686
Ubuntu

raid_setup returns offset = 00000000 ndisk = 0
  BIOS VolumeID Device
Reading boot sector from /dev/fd0
pf_hard_disk_scan: ndevs=1
  0800 54085408 /dev/sda
device codes (user assigned pf) = 0
device codes (user assigned) = 0
device codes (BIOS assigned) = 1
device codes (canonical) = 1
mode = 0x03, columns = 80, rows = 25, page = 0
Using BITMAP secondary loader
Calling map_insert_data
Secondary loader: 19 sectors (0x3800 dataend).
Warning: The boot sector and map file are on different disks.
bios_boot = 0x00 bios_map = 0x80 map==boot = 0 map S/N: 54085408
Mapping bitmap file /boot/coffee.bmp
Calling map_insert_file
Compaction removed 592 BIOS calls.
Bitmap: 603 sectors.
BIOS data check was okay on the last boot

Boot image: /boot/vmlinuz-2.6.31-14-generic
Setup length is 26 sectors.
Compaction removed 7452 BIOS calls.
Mapped 7601 sectors.
Mapping RAM disk /boot/initrd.img-2.6.31-14-generic
Compaction removed 14696 BIOS calls.
RAM disk: 14930 sectors.
Added Lin_2.6.31-14 *

Boot image: /boot/vmlinuz-2.6.31-20-generic
Setup length is 26 sectors.
Compaction removed 7468 BIOS calls.
Mapped 7617 sectors.
Mapping RAM disk /boot/initrd.img-2.6.31-20-generic
Compaction removed 14704 BIOS calls.
RAM disk: 14938 sectors.
Added Lin_2.6.31-20

Boot image: /boot/memtest86+.bin
Setup length is 4 sectors.
Compaction removed 243 BIOS calls.
Mapped 254 sectors.
Added Memory_Test+

Boot other: /dev/sda6, loader CHAIN
Pseudo partition start: 43198848
Compaction removed 0 BIOS calls.
Mapped 6 (4+1+1) sectors.
Added Fedora_8

Boot other: /dev/sda1, on /dev/sda, loader CHAIN
Compaction removed 0 BIOS calls.
Mapped 6 (4+1+1) sectors.
Added Windows_XP

  BIOS VolumeID Device
  80 54085408 0800
Writing boot sector.
```

```

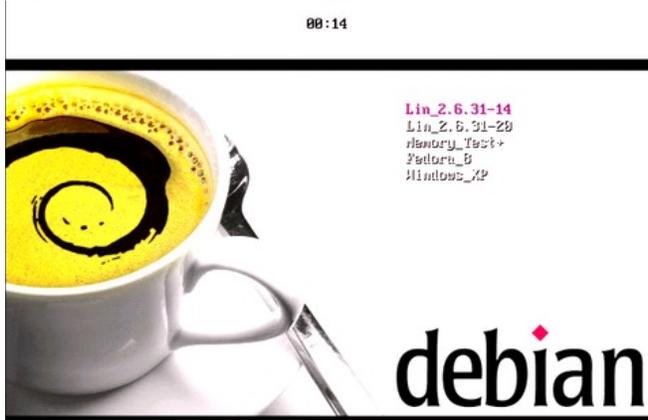
/boot/boot.0200 exists - no boot sector backup copy made.
Map file size: 336896 bytes.
RAID device mask 0x0000
One warning was issued.

```

Agora, temos nosso disquete LILO inicializável. Se LILO encontrar um erro, talvez você veja uma mensagem de erro, e o setor de boot não será gravado. A mensagem dará uma ideia sobre o que precisa ser corrigido.

Se usarmos o disquete que acabamos de criar e reinicializarmos a máquina, veremos um prompt semelhante ao da Figura 1. Observe o cronômetro de contagem regressiva na parte superior da tela, que está fazendo uma contagem regressiva iniciada em 20 segundos (200 décimos de segundo). Se nenhuma opção for feita dentro de 20 segundos, a opção destacada (a primeira, ou Lin_2.6.31-14 neste caso) é automaticamente inicializada.

Figura 1. Tela de boot do LILO



É possível usar as teclas de seta para mover-se para cima e para baixo na lista de seleção. A Figura 2 mostra como selecionar o kernel Lin_2.6.31-20 mais recente, que é a segunda entrada. Observe que o valor do cronômetro não é mais exibido. Pressione **Enter** para fazer o boot deste kernel.

Figura 2. Selecionando outro destino de boot com LILO



Quando tiver testado seu disquete de boot, altere a entrada `boot=/dev/fd0` no seu arquivo `lilo.conf` para instalar o LILO no MBR ou em um registro de boot de partição. Por exemplo, `boot=/dev/sda` instalará o

LILO no registro de boot mestre de sua primeira unidade de disco rígido.

Agora, você já teve uma introdução ao LILO e ao seu arquivo de configuração. É possível substituir algumas opções de configuração da linha de comando `lilo`. Você encontrará mais informações na man page do `lilo` usando o comando `man lilo` ou `man lilo.conf`. Você encontrará informações ainda mais abrangentes no guia do usuário PostScript instalado com o pacote `lilo` ou `lilo-doc`. Isso deve ser instalado no seu diretório de documentação, mas o local exato pode variar de acordo com o sistema. A única forma de localizar o arquivo é filtrar a lista de pacotes por meio de `grep`. A Listagem 3 mostra isso para o sistema Ubuntu 9.10 baseado em Debian que estamos usando neste exemplo.

Listagem 3. Localizando o guia do usuário LILO com dpkg

```
ian@pinguino:~$ dpkg -L lilo-doc | grep "ps"
/usr/share/doc/lilo-doc/user.ps.gz
/usr/share/doc/lilo-doc/tech.ps.gz
```

O comando correspondente em um sistema baseado em RPM como o Fedora seria: `rpm -ql lilo-doc | grep "ps"`

Comandos auxiliares LILO

O LILO possui vários comandos auxiliares:

lilo -q

Exibe informações do arquivo de mapa.

lilo -R

Define o `lilo` para fazer boot automaticamente o sistema especificado somente no próximo reboot. Isso é muito conveniente para o reboot automático de sistemas remotos.

lilo -l

Exibe informações sobre o caminho de um kernel.

lilo -u

Desinstala o `lilo` e restaura o registro de boot anterior.

Quando LILO inicializar um sistema Linux, você poderá fornecer parâmetros adicionais no momento do boot. Por exemplo, se a inicialização gráfica não estiver funcionando, você poderá fazer o boot no modo 3 ou no modo de usuário único para recuperação. Se você selecionar uma entrada e pressionar a tecla **TAB**, verá uma janela de terminal que foi pré-preenchida com o nome da sua etiqueta, como mostra a Figura 3. Qualquer texto que você digitar depois do nome da etiqueta será passado ao kernel. Por exemplo, em nosso exemplo, você selecionaria o sistema Ubuntu mais recente simplesmente movendo-se uma linha para baixo e, em seguida, pressionando **TAB**. Se você não selecionar um nome primeiro, pode ser necessário digitar o nome completo especificado na etiqueta. É possível ver por que as etiquetas que são mais sucintas que nossos exemplos podem ser uma boa ideia: digitar `Lin_2.6.31.20` não é fácil.

Figura 3. Adicionando parâmetros de kernel com LILO

```
Lin_2.6.31-14      Lin_2.6.31-20      Memory_Test+      Fedora_8
Windows_XP
boot: Lin_2.6.31-20 single_
```

Na Figura 3, adicionamos "single" aos parâmetros de boot para indicar que queremos inicializar no modo de usuário único para fazer algum tipo de recuperação do sistema. Consulte a seção [Recuperação](#) para obter mais informações sobre como usar parâmetros de boot.

Lembre-se também de que com o LILO, é **necessário** executar o comando `lilo` sempre que você atualizar o arquivo de configuração (`/etc/lilo.conf`). Execute também o comando `lilo` se adicionar, mover ou remover partições ou efetuar qualquer outra alteração que possa invalidar o loader de boot gerado.

GRUB

GRUB, ou o loader GRand Unifood Boot, é o outro dos dois loaders de boot Linux mais comuns. Assim como LILO, GRUB pode ser instalado no MBR da unidade de disco rígido inicializável ou no registro de boot de partição de uma partição. Também pode ser instalado em dispositivos removíveis, como discos flexíveis, CDs ou chaves USB. Será uma boa ideia praticar em um disco flexível ou chave USB se ainda não estiver familiarizado com GRUB, como em nossos exemplos.

GRUB, ou GNU GRUB, é agora desenvolvido sob o patrocínio da Free Software Foundation. Uma nova versão, GRUB 2, está em desenvolvimento e começa a ser fornecida em distribuições Linux, para que as versões GRUB 0.9x originais sejam agora conhecidas como Grub Legacy.

Durante a instalação do Linux, normalmente você especificará LILO ou GRUB como gerenciador de boot. Se escolher LILO, não será possível ter GRUB instalado. Caso não tenha instalado o GRUB, será preciso instalar seu pacote. Daqui em diante, consideraremos que você já tenha o pacote GRUB instalado. Consulte o [roteiro da nossa série](#) para ver os artigos sobre gerenciamento de pacote, caso precise de ajuda.

O GRUB (Legacy) tem um arquivo de configuração que geralmente é armazenado em `/boot/grub/grub.conf`. Se o seu sistema de arquivos suportar links simbólicos, como a maior parte dos sistemas de arquivos Linux, você provavelmente terá `/boot/grub/menu.lst` como link simbólico para `/boot/grub/grub.conf`.

O comando `grub` (`/sbin/grub` ou, em alguns sistemas, `/usr/sbin/grub`) é um shell pequeno mas razoavelmente poderoso que suporta vários comandos para instalar o GRUB, sistemas de boot, localizar e exibir arquivos de configuração e tarefas similares. Esse shell compartilha muitos códigos com o loader de boot GRUB de segundo estágio, portanto, ele é útil para conhecer GRUB sem precisar inicializar para um ambiente GRUB de segundo estágio. O GRUB estágio 2 é executado no *modo de menu* para permitir que você escolha um sistema operacional em um menu ou no *modo de comando*, onde você especifica comandos individuais para carregar um sistema. Há também diversos outros comandos, como `grub-install`, que usa o shell `grub` e ajuda a automatizar tarefas como instalação de GRUB.

A Listagem 4 mostra parte de um arquivo de configuração de GRUB. Enquanto você o observa, lembre-se de algo importante: o GRUB, pelo menos o GRUB Legacy, tem uma contagem de unidades, partições e outras coisas que precisam ser contadas que é iniciada em 0 em vez de 1. Observe ainda que a primeira linha do kernel é muito longa. Indicamos isso com uma barra invertida (`\`) para mostrar que a linha foi quebrada para fins de publicação.

Listagem 4. Exemplo de configuração do GRUB `/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 do not have a /boot partition. This means that
#           all kernel and initrd paths are relative to /, eg.
#           root (hd0,5)
#           kernel /boot/vmlinuz-version ro root=/dev/sda6
#           initrd /boot/initrd-version.img
#boot=/dev/sda6
```

```

default=1
timeout=10
splashimage=(hd0,5)/boot/grub/splash.xpm.gz
#hiddenmenu
password --md5 $1$RW1VW/$4XGAKlxB7/GJk0u047Srx1
title Upgrade to Fedora 11 (Leonidas)
    kernel /boot/upgrade/vmlinuz preupgrade \
    repo=hd:/var/cache/yum/preupgrade stage2=\
    hd:UUID=8b4c62e7-2022-4288-8995-5eda92cd149b:/boot/upgrade/install.img \
    ks=hd:UUID=8b4c62e7-2022-4288-8995-5eda92cd149b:/boot/upgrade/ks.cfg
    initrd /boot/upgrade/initrd.img
title Fedora (2.6.26.8-57.fc8)
    root (hd0,5)
    kernel /boot/vmlinuz-2.6.26.8-57.fc8 ro root=LABEL=FEDORA8 rhgb quiet
    initrd /boot/initrd-2.6.26.8-57.fc8.img
title Fedora (2.6.26.6-49.fc8)
    root (hd0,5)
    kernel /boot/vmlinuz-2.6.26.6-49.fc8 ro root=LABEL=FEDORA8 rhgb quiet
    initrd /boot/initrd-2.6.26.6-49.fc8.img
title GRUB Menu
    rootnoverify (hd0,1)
    chainloader +1
title Windows
    rootnoverify (hd0,0)
    chainloader +1

```

Como ocorre com o arquivo de configuração do LILO, o primeiro conjunto de opções controla a forma pela qual o GRUB opera. Para GRUB, elas são chamadas de *comandos de menu* e devem aparecer antes de outros comandos. As seções restantes oferecem opções por imagem para os sistemas operacionais que queremos permitir que o GRUB seja inicializado. Observe que "title" é considerado um comando de menu. Cada instância do título é seguida por um ou mais comandos de entrada de menu ou gerais. Nosso exemplo LILO foi um exemplo básico típico para um sistema de boot duplo com sistemas Windows e dois Linux. Esse exemplo é proveniente do mesmo sistema que usamos antes. De fato, é o menu de boot que é inicializado em cadeia a partir da entrada "Fedora 8" no menu do LILO. Você reconhecerá muitos dos mesmos tipos de elementos que ocorrem nos arquivos de configuração LILO e GRUB. Talvez goste de pensar no que seria necessário para alterar se tiver adicionado os sistemas operacionais extras aqui ao exemplo LILO anterior.

Os comandos de menu que se aplicam a todas as outras seções em nosso exemplo são:

#

Qualquer linha que comece com um **#** é um comentário e é ignorada pelo GRUB. Esse arquivo de configuração particular foi originalmente gerado por anaconda, o instalador de Red Hat. Você provavelmente encontrará comentários adicionados ao seu arquivo de configuração GRUB se instalar GRUB ao instalar o Linux. Os comentários em geral servirão como um auxílio para o programa de atualização do sistema, para que sua configuração GRUB possa ser mantida atual com kernels atualizados. Preste atenção a quaisquer marcadores existentes com essa finalidade, caso você mesmo edite a configuração.

default

Especifique qual sistema carregar se o usuário não fizer a escolha dentro do tempo limite. Em nosso exemplo, default=2 significa carregar a entrada **third**. Lembre-se de que o GRUB inicia a contagem em 0 e não em 1. Se não for especificado, o padrão é fazer o boot da primeira entrada, a entrada número 0.

timeout

Especifica o tempo limite em segundos antes de inicializar a entrada padrão. Observe que o LILO usa décimos de segundo para os tempos limite, enquanto o GRUB usa segundos inteiros.

splashimage

Especifica o segundo plano ou a *splashimage*, que será exibida com o menu de boot. GRUB Legacy se

refere à primeira unidade de disco rígido como (hd0) e à primeira partição dessa unidade como (hd0,0), portanto, a especificação de splashimage=(hd0,6)/boot/grub/splash.xpm.gz indica o uso do arquivo /boot/grub/splash.xpm.gz localizado na partição 7 da primeira unidade de disco rígido. Lembre-se de contar a partir de 0. Observe também que a imagem é um arquivo XPM compactado com gzip. O suporte para splashimage é uma correção que pode ou não estar inclusa em sua distribuição.

password

Especifica uma senha que deve ser inserida para que um usuário possa desbloquear o menu e editar uma linha de configuração ou inserir comandos GRUB. Assim como LILO, a senha pode estar em texto não criptografado. GRUB também permite que senhas sejam armazenadas como uma compilação MD5, como em nosso exemplo. Isso é um pouco mais seguro, e a maioria dos administradores definirá uma senha. Sem uma senha, um usuário tem acesso completo à linha de comando de GRUB.

Nosso exemplo mostra dois kernels Fedora 8 e uma opção de atualização para o Fedora 11, além do Windows XP e uma opção "GRUB Menu" carregada em cadeia. Os comandos usados nessas seções são:

title

É um título descrito mostrado como o item de menu quando Grub é inicializado. Use as teclas de seta para mover-se para cima e para baixo pela lista de título e, em seguida, pressione a tecla **Enter** para selecionar uma entrada em particular.

root

Especifica a partição que será inicializada. Assim como splashimage, lembre-se de que a contagem inicia em 0, portanto, o primeiro sistema Red Hat especificado como root (hd0,6) está na verdade na partição 7 da primeira unidade de disco rígido (/dev/hda7 nesse caso), ao passo que o primeiro sistema Ubuntu especificado como root (hd1,10) está na segunda unidade de disco rígido (/dev/hdb11). O GRUB tentará montar essa partição para verificá-la e fornecerá os valores para o sistema operacional inicializado, em alguns casos.

kernel

Especifica a imagem do kernel a ser carregada e quaisquer parâmetros do kernel que sejam necessários. Isso é igual a uma combinação de comandos de imagem e anexação LILO. Neste exemplo, temos dois diferentes kernels Fedora 8, além do kernel de atualização.

initrd

É o nome do *disco RAM inicial*, que contém módulos de que o kernel precisa para que os seus sistemas de arquivos sejam montados.

savedefault

Não é usado neste exemplo. Se o comando de menu default=saved for especificado, e o comando savedefault for especificado para um sistema operacional, o boot desse sistema operacional fará com que ele se torne o padrão até outro sistema operacional com savedefault especificado ser inicializado. Neste exemplo, a especificação default=2 substituirá qualquer padrão salvo.

boot

É um parâmetro opcional que instrui o GRUB a fazer o boot do sistema operacional selecionado. Essa é a ação padrão quando todos os comandos de uma seleção tiverem sido processados.

lock

Não é usado neste exemplo. Isso não irá fazer o boot da entrada especificada até que a senha seja

inserida. Se isso for usado, você deverá também especificar uma senha nas opções iniciais; caso contrário, um usuário poderá editar sua opção de bloqueio e fazer o boot do sistema ou adicionar "single" a uma das outras entradas. É possível especificar uma senha diferente para entradas individuais, se você desejar.

rootnoverify

É semelhante ao root, exceto pelo fato de que o GRUB não tenta montar o sistema de arquivos nem verificar seus parâmetros. Isso é geralmente usado para sistemas de arquivos como NTFS, que não têm o suporte de GRUB. Também é possível usar isso se você desejar que o GRUB carregue o registro de boot mestre em uma unidade de disco rígido, por exemplo, para acessar um arquivo de configuração diferente, ou para recarregar o loader de boot anterior.

chainloader

Especifica que outro arquivo será carregado como arquivo de estágio 1. O valor "+1" é equivalente a 0+1, o que significa carregar um setor que comece em 0, ou seja, carregar o primeiro setor do dispositivo especificado por root ou rootnoverify.

Você agora tem alguma ideia do que pode encontrar em um arquivo /boot/grub/grub.conf (ou /boot/grub/menu.lst) típico. Há muitos outros comandos GRUB para fornecer um controle amplo sobre o processo de boot, bem como ajudar com a instalação de GRUB e outras tarefas. É possível saber mais sobre isso no manual do GRUB, que deve estar disponível no seu sistema por meio do comando `info grub`.

Agora que temos um arquivo de configuração GRUB, precisamos criar um disquete de boot para testá-lo. A forma mais simples de fazer isso é usar o comando `grub-install`, como mostrado na Listagem 5. Se estiver instalado o GRUB em um disquete ou em uma partição, será preciso desmontar o dispositivo primeiro. Isso não se aplicará se você estiver instalando o GRUB no MBR de uma unidade de disco rígido, já que você monta apenas partições (/dev/sda1, /dev/sda2, etc.), e não toda a unidade de disco rígido (/dev/sda).

Listagem 5. Instalando GRUB em um disco flexível

```
[root@pinguino ~]# grub-install /dev/fd0
Installation finished. No error reported.
This is the contents of the device map /boot/grub/device.map.
Check if this is correct or not. If any of the lines is incorrect,
fix it and re-run the script 'grub-install'.

(fd0) /dev/fd0
(hd0) /dev/sda
```

Observação: É possível também usar o nome de dispositivo GRUB (fd0) em vez de /dev/fd0, mas se isso for feito, será preciso colocá-lo entre aspas para evitar a interpretação dos parênteses pelo shell. Por exemplo: `grub-install '(fd0)'`

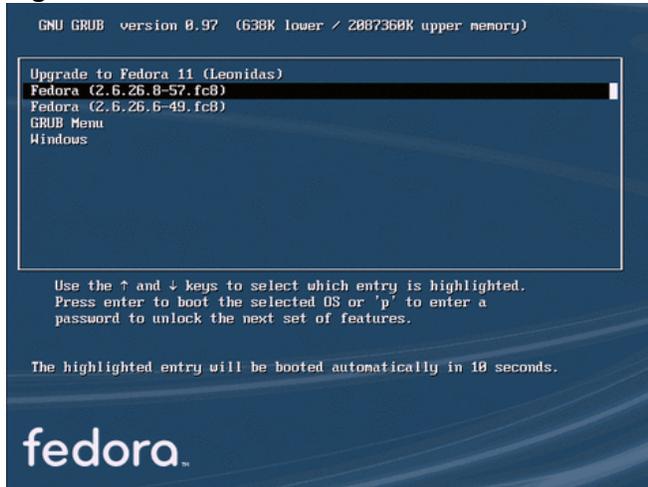
O mapa de dispositivo mostra a você como o GRUB fará a correspondência da visualização interna dos discos (fd0, hd0, hd1) com a visualização de Linux (/dev/fd0, /dev/sda, /dev/sdb). Em um sistema com um ou dois discos rígidos IDE e talvez uma unidade de disquete, isso provavelmente estará correto. Se um mapa de dispositivo já existir, GRUB o reutilizará sem realizar análise. Caso tenha adicionado recentemente uma nova unidade e queira forçar a geração de um novo mapa de dispositivo, adicione a opção `--recheck` ao comando `grub-install`. Às vezes, será possível ver uma mensagem como aquela mostrada na Listagem 6. Nesse caso também, será necessário adicionar a opção `--recheck` ao comando `grub` para forçar o GRUB a analisar os dispositivos disponíveis e reconstruir um mapa de dispositivo.

Listagem 6. Instalando o GRUB em um disco flexível

```
[root@pinguino ~]# grub-install /dev/fd0
/dev/fd0 does not have any corresponding BIOS drive.
```

Caso tenha começado com um disquete vazio e agora você o monte, descobrirá que ele ainda aparenta estar vazio. O que aconteceu é que GRUB gravou um loader de estágio 1 personalizado no primeiro setor do disco. Isso não aparece no sistema de arquivos. Esse loader de estágio 1 carregará o estágio 2 e o arquivo de configuração a partir da unidade de disco rígido. Tente inicializar o disquete e você verá muito pouca atividade de EO antes de o menu ser exibido. Assim como ocorre com o LILO, use as setas para cima e para baixo para selecionar diferentes destinos de boot.

Figura 4. Inicializando usando o GRUB



Quando tiver testado o seu disquete de boot, você estará pronto para instalar o GRUB no MBR de sua unidade de disco rígido. Para a primeira unidade de disco rígido IDE, você usaria:

```
grub-install /dev/sda
```

ou

```
grub-install '(hd0)'
```

Para instalar a opção no registro de boot de partição da partição 11, use:

```
grub-install /dev/sda11
```

ou

```
grub-install '(hd0,10)'
```

Lembre-se de que o GRUB faz a numeração a partir de 0.

Assim como o LILO, é possível adicionar parâmetros de boot. De fato, é possível editar qualquer um dos comandos na configuração do GRUB ou até mesmo carregar outro arquivo de configuração no momento do boot. Se você tiver usado uma senha, será necessário inserir o comando `p` e, em seguida, digitar sua senha. Depois disso, selecione uma entrada e, em seguida insira o comando `e` para editá-la. Selecione a linha que deseja editar e insira `e` novamente. Edite a linha digitando novos caracteres ou apagando os caracteres existentes e, em seguida, pressione **Enter** para retornar ao menu. Insira o comando `b` para fazer o boot do seu sistema com os valores modificados. A Figura 5 mostra a fase de edição do comando desse processo, em que você adiciona o parâmetro "single" à instrução do kernel para fazer o boot no modo de único usuário.

Figura 5. Inicializando no modo de usuário único

Atualizações do sistema

A maior parte das distribuições fornece ferramentas para atualizar o sistema. Essas ferramentas geralmente identificam o loader de boot em uso atualizam o arquivo de configuração automaticamente. Se você criar seu próprio kernel personalizado ou preferir usar um arquivo de configuração com um local ou nome não padrão, você mesmo precisará atualizar o arquivo de configuração.

Se usar LILO, você **deverá** executar o comando `lilo` sempre que atualizar o arquivo de configuração ou efetuar alterações, como adicionar uma unidade de disco rígido ou excluir uma partição.

Caso o GRUB seja usado, será possível editar o arquivo `/boot/grub/grub.conf` para efetuar as suas alterações, e o loader de estágio 2 GRUB lerá o arquivo quando você reinicializar. Normalmente, não é preciso reinstalar GRUB só porque um novo kernel é adicionado. Entretanto, se uma partição for movida ou unidades forem adicionadas, poderá haver a necessidade de reinstalar GRUB. Lembre-se de que o loader de estágio 1 é muito pequeno, portanto, ele simplesmente tem uma lista de endereços em bloco para o loader de estágio 2. Mova a partição e a mudança endereçada, para que o estágio 1 não consiga mais localizar o estágio 2. Abordaremos algumas estratégias de recuperação e também discutiremos os loaders de estágio 1,5 de GRUB na sequência.

Se você executar mais de um sistema Linux em diferentes partições, considere [usar uma partição de boot](#), conforme descrito a seguir.

Recuperação

Agora vamos ver algumas coisas que podem dar errado com a configuração de boot preparada cuidadosamente, em particular quando você instalar e inicializar vários sistemas operacionais. A primeira coisa a se lembrar é resistir à tentação inicial de pânico. A recuperação em geral precisa de apenas algumas etapas. Estas estratégias devem ajudá-lo a superar vários tipos de crises.

Qualquer pessoa com acesso físico a uma máquina tem muito poder. Da mesma forma, qualquer pessoa que tenha acesso a uma linha de comando de GRUB também tem acesso aos arquivos do seu sistema sem o benefício de qualquer propriedade ou outras precauções de segurança fornecidas por um sistema em execução. Mantenha esses pontos em mente ao selecionar o loader de boot. A escolha entre LILO e

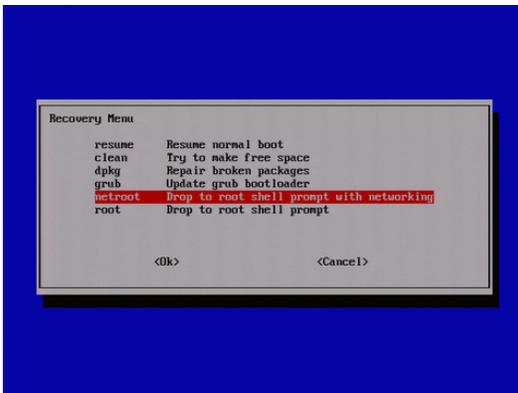
GRUB é, principalmente, uma questão de preferência pessoal. Escolha o loader que melhor se adequar às suas necessidades específicas e ao seu estilo de trabalho.

Outra instalação destrói seu MBR

Pode ocorrer que você instale outro sistema operacional e, inadvertidamente, substitua seu MBR. Alguns sistemas, como DOS e Windows, sempre instalam seu próprio MBR. Em geral, é muito fácil se recuperar dessa situação. Se criar o hábito de criar um disquete de boot toda vez que executar o `lilo` ou reinstalar o GRUB, você ficará tranquilo. Simplesmente faça o boot do sistema Linux pelo disquete e execute novamente o `lilo` ou o `grub-install`.

Se não tiver um disquete de boot, mas ainda tiver praticamente qualquer distribuição Linux disponível, você poderá em geral inicializar a mídia de instalação Linux em um modo de recuperação. Quando isso for feito, o sistema de arquivos root na unidade de disco rígido será montado em um ponto de recuperação estranho ou não será mais montado. É possível usar o comando `chroot` para tornar esse ponto de montagem singular seu diretório root (`/`). Em seguida, execute o `lilo` ou o `grub-install` para criar um novo disquete de boot ou reinstalar o MBR. Prefiro criar um disquete e usá-lo para fazer o boot, verificando antes se tudo está funcionando, e regravar o MBR, mas talvez você seja mais corajoso que eu. Inicializar um CD ou DVD no modo de resgate pode oferecer um menu de opções ou simplesmente levá-lo ao modo de usuário único, em que você tem a propriedade de administrador. Se você tentar inicializar o sistema Ubuntu 9.10 que usamos para os exemplos do LILO no modo de usuário único, verá um menu como o da Figura 6. A opção destacada leva a um shell root com rede.

Figura 6. Menu de recuperação do Ubuntu 9.10



Um menu como esse é uma opção comum com mídia de resgate. Para nosso exemplo de resgate, não usamos um CD, mas fazemos o boot do sistema Ubuntu 9.19 em modo de usuário único, conforme descrito acima na seção sobre LILO. Em seguida, passaremos pelas etapas típicas necessárias para criar um disquete de boot para outro sistema, neste caso o sistema Fedora 8 em `/dev/sda6`. Independentemente de você fazer o boot a partir de um CD de resgate ou de outro sistema operacional no mesmo sistema, as etapas são semelhantes, embora algumas (como montar a imagem a ser resgatada) já possam ter sido feitas para você.

Ao final, quando atingirmos um `prompt root`, é recomendável fazer algumas ou todas as coisas mostradas na Listagem 7. Para nosso exemplo, criamos um ponto de montagem (`/mnt/sysimage`),

montamos /dev/sda6, montamos alguns sistemas de arquivos especiais no novo root e, em seguida, executamos o comando chroot no sistema montado. O comando cat /etc/issue mostra o efeito do comando chroot, que torna o diretório de destino o novo root do sistema de arquivos, pelo menos até sairmos do ambiente chroot com Ctrl-d ou exit.

A Listagem 7 mostra um exemplo no mesmo ambiente que usamos para nossos exemplos de configuração anteriores. Pense nisso como uma janela de terminal com você conectado como root. Em outras palavras, tome muito cuidado com o que grava na unidade de disco rígido.

Listagem 7. Usando um sistema de resgate e o comando chroot

```

root@pinguino:~# mkdir /mnt/sysimage
root@pinguino:~# mount /dev/sda6 /mnt/sysimage
root@pinguino:~# cd /mnt/sysimage
root@pinguino:~# mount -t proc none proc/
root@pinguino:~# mount -t sysfs none sys/
root@pinguino:~# mount -o bind /dev dev/
root@pinguino:~# cat /etc/issue
Ubuntu 9.10 \n \l

root@pinguino:~# chroot /mnt/sysimage
root@pinguino:~#
sh-3.00# chroot /mnt/sysimage
[root@oinguine /]# cat /etc/issue
Fedora release 8 (Werewolf)
Kernel \r on an \m

[root@oinguine /]# grub-install /dev/fd0
Installation finished. No error reported.
This is the contents of the device map /boot/grub/device.map.
Check if this is correct or not. If any of the lines is incorrect,
fix it and re-run the script `grub-install'.

(fd0) /dev/fd0
(hd0) /dev/sda
[root@oinguine /]#

```

Depois que você tiver seu disquete inicializável, pressione Ctrl-d para sair do ambiente chroot e, em seguida, reinicialize seu sistema, lembrando-se de remover qualquer CD de instalação ou mídia de recuperação que tenha sido usado. Se não tiver um CD ou DVD de instalação à mão, haverá muitos CDs de LINUX de recuperação e ativos disponíveis on-line e alguns disquetes ou memory keys USB também. Consulte [Recursos](#) para obter detalhes.

Embora esteja além do escopo deste artigo, saiba que é possível fazer com que seu MBR inicialize um sistema Windows 2000 ou Windows XP e instale o LILO ou o GRUB em um registro de boot de partição. O programa de boot ntldr pode também carregar em cadeia outros setores de boot, embora a configuração possa ser um pouco complicada. Será preciso copiar o setor de boot para uma partição do Windows (geralmente usando o comando dd) e modificar o arquivo boot.ini oculto para realizar esse trabalho.

Você moveu uma partição

Se tiver movido uma partição e esquecido de sua configuração de boot, você terá um problema temporário. Em geral, LILO ou GRUB se recusam a carregar. LILO provavelmente imprimirá um 'L', indicando que o estágio 1 foi carregado, e depois parará. GRUB apresentará uma mensagem de erro. O que ocorreu aqui é que o loader de estágio 1, contendo uma lista de setores a carregar para se chegar ao loader de estágio 2, talvez carregue os setores a partir dos endereços que ele tem, mas os setores não terão mais a assinatura de estágio 2. Se você tiver criado um disquete de boot usando os métodos descritos anteriormente, lembre-se de que todo o lilo ou o grub-install colocado no disquete era um único setor de boot, portanto, seu disquete de boot provavelmente não ajudará. Como no exemplo anterior, provavelmente será preciso inicializar algum tipo de ambiente de resgate e criar novamente o

seu disquete de boot com LILO ou GRUB. Em seguida, reinicialize, verifique o seu sistema e reinstale o loader de boot no MBR.

Talvez você tenha percebido que nossos exemplos de configuração usaram algumas tabelas e (Universally Unique IDs) UUIDs para partições. Por exemplo:

```
kernel /boot/vmlinuz-2.6.26.8-57.fc8 ro root=LABEL=FEDORA8 rhgb quiet
ou
kernel /boot/upgrade/vmlinuz preupgrade repo=hd::/var/cache/yum/preupgrade
stage2=\
hd:UUID=8b4c62e7-2022-4288-8995-5eda92cd149b:/boot/upgrade/install.img \
ks=hd:UUID=8b4c62e7-2022-4288-8995-5eda92cd149b:/boot/upgrade/ks.cfg
```

Frequentemente, uso UUIDs de etiquetas como essas para ajudar a evitar problemas quando movo partições. É preciso ainda atualizar o arquivo de configuração GRUB ou LILO e executar novamente o `lilo`, mas não é necessário atualizar `/etc/fstab` também. Isso será particularmente conveniente se você criar uma imagem de partição em um sistema e restaurá-la em um local diferente em outro sistema. Também é útil se você fizer o boot a partir de uma unidade, como uma unidade USB, que nem sempre pode estar conectada no mesmo local.

Usando uma partição de boot

Outra abordagem para recuperação, ou talvez para evitar a recuperação, é usar uma partição separada para boot. Isso é particularmente útil se você tiver um sistema de testes com várias distribuições que podem ser reconstruídas frequentemente. A partição de boot não precisa ser muito grande, talvez 100 MB ou mais. Coloque essa partição em algum lugar de onde não possa ser movida e onde seu número de partição não possa ser movido pela adição ou remoção de outra partição. Em um ambiente misto de Windows e Linux, `/dev/sda2` (ou `/dev/hda2`, dependendo de como seus discos estão etiquetados) frequentemente é uma boa opção para sua partição de boot. A Listagem 8 mostra as entradas na partição de boot pequena (`/dev/sda2`) que é realmente carregada pelo registro de boot mestre no sistema que usamos como exemplo.

Listagem 8. Usando uma partição de boot pequena com GRUB

```
# menu.lst - See: grub(8), info grub, update-grub(8)
#
#       grub-install(8), grub-floppy(8),
#       grub-md5-crypt, /usr/share/doc/grub
#       and /usr/share/doc/grub-doc/.
## default num
# Set the default entry to the entry number NUM. Numbering starts from 0, and
# the entry number 0 is the default if the command is not used.
#
# You can specify 'saved' instead of a number. In this case, the default entry
# is the entry saved with the command 'savedefault'.
default      0

## timeout sec
# Set a timeout, in SEC seconds, before automatically booting the default entry
# (normally the first entry defined).
timeout      10

## password [--md5] passwd
# If used in the first section of a menu file, disable all interactive editing
# control (menu entry editor and command-line) and entries protected by the
# command 'lock'
# e.g. password topsecret
#       password --md5 $1$gLhU0/$aW78kHK1QfV3P2b2znUoe/
# password topsecret

title        Fedora 8 (sda6) Grub menu
root         (hd0,5)
chainloader  +1
boot

# Ubuntu 9.10 uses GRUB 2, so use kernel to drive GRUB 2 menu.
```

```
title          Ubuntu 9.10 (sda7)
root           (hd0,6)
kernel        /boot/grub/core.img

# This is a divider, added to separate the menu items below from the Debian
# ones.
title          Other operating systems:
root

# Windows XP
title          Windows NT/XP
root           (hd0,0)
savedefault
chainloader   +1
```

Outra razão para ter uma partição de boot surge quando sua partição root usa um sistema de arquivos não suportado pelo loader de boot. Por exemplo, é muito comum ter uma partição /boot com o formato ext2 ou ext3 quando a partição root (/) usa LVM, que não é suportado pelo GRUB Legacy.

Se você tiver várias distribuições no seu sistema, **não** compartilhe a partição /boot entre elas. Lembre-se de configurar LILO ou GRUB para inicializar da partição que posteriormente será montada como /boot. Lembre-se também de que os programas de atualização de uma distribuição em geral atualizarão a configuração de GRUB ou LILO para esse sistema. Em um ambiente com vários sistemas, é possível manter um com sua própria partição /boot como a principal e manualmente atualizar esse arquivo de configuração sempre que uma atualização de um de seus sistemas exigir isso. Outra abordagem é cada sistema instalar um loader de boot em seu próprio registro de boot de partição e fazer o sistema principal simplesmente carregar em cadeia os registros de boot de partição para os sistemas individuais, oferecendo-lhe um processo de menu em dois estágios.

Construindo um disquete ou CD-ROM de boot autocontido

Por fim, vamos observar mais de perto a configuração GRUB e ver como criar um disquete ou CD de boot independente que lhe apresentará um prompt de GRUB, não importa o que tenha acontecido com a unidade de disco rígido.

Lembre-se de toda aquela coisa sobre cilindros em discos rígidos. Embora você possa pensar em um cilindro como entidade fictícia com unidades modernas, muitos aspectos do seu sistema de arquivos não as esqueceu. Em particular, você descobrirá que partições usam um número integral de cilindros alinhados nos limites do cilindro. Em uma partição, muitos sistemas de arquivos também gerenciam o espaço em unidades de cilindros. Em muitos sistemas UNIX® e Linux, o layout do sistema de arquivos é armazenado como um *superbloco*, que é a primeira unidade de alocação no sistema de arquivos. Para sistemas de arquivos como ext2 ou ext3 e discos rígidos razoavelmente grandes, o espaço será dividido em várias seções, com uma cópia do superbloco no começo de cada seção. Isso ajudará na recuperação se você acidentalmente se atrapaçar nos limites de partição com um programa como fdisk.

Outro benefício da mentalidade de cilindro é que existe algum espaço no começo de um disco logo após o MBR. GRUB aproveita as vantagens disso integrando um loader de boot de estágio 1,5 a esse espaço ou em espaço similar ainda não utilizado em uma partição sempre que possível. O loader de estágio 1,5 entende o sistema de arquivos na partição que contém o estágio 2, para que fique de alguma forma mais protegido contra problemas associados a arquivos sendo movidos.

Tudo bem, mas como isso se relaciona a um disquete inicializável? Bem, um disquete não tem muito espaço ou muita noção de cilindros, por isso, se você desejar inicializar ambos os estágios 1 e 2 do GRUB a partir de um disquete, é necessário instalar o estágio 1 e, em seguida, copiar o estágio 2 para

os setores imediatamente posteriores ao setor de boot. A Listagem 9 mostra um exemplo de como fazer isso. Use um disquete vazio, já que esse processo destruirá dados. Copie os arquivos que vieram com a sua distribuição GRUB em vez dos que estão no diretório /boot/grub, já que /boot/grub/stage2 foi modificado para funcionar com as partições da sua unidade de disco rígido. Localize os arquivos originais stage1 e stage2 em um subdiretório de /usr/share/grub. Em nosso exemplo, eles estão em /usr/share/grub/i386-redhat.

Listagem 9. Criando um disquete de boot GRUB

```
[root@pinguino ~]# ls /usr/share/grub/
i386-redhat
[root@pinguino ~]# cd /usr/share/grub/i386-redhat/
[root@pinguino i386-redhat]# ls -l st*
-rw-r--r-- 1 root root 512 2008-05-28 12:05 stage1
-rw-r--r-- 1 root root 109212 2008-05-28 12:05 stage2
-rw-r--r-- 1 root root 109212 2008-05-28 12:05 stage2_eltorito
[root@pinguino i386-redhat]# dd if=stage1 of=/dev/fd0 bs=512 count=1
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.29007 s, 1.8 kB/s
[root@pinguino i386-redhat]# dd if=stage2 of=/dev/fd0 bs=512 seek=1
213+1 records in
213+1 records out
109212 bytes (109 kB) copied, 7.4094 s, 14.7 kB/s

[root@lyrebird root]# ls /usr/share/grub
i386-redhat
[root@lyrebird root]# cd /usr/share/grub/i386-redhat
[root@lyrebird i386-redhat]# ls -l st*
-rw-r--r-- 1 root root 512 Aug 3 2004 stage1
-rw-r--r-- 1 root root 104092 Aug 3 2004 stage2
[root@lyrebird i386-redhat]# dd if=stage1 of=/dev/fd0 bs=512 count=1
1+0 records in
1+0 records out
[root@lyrebird i386-redhat]# dd if=stage2 of=/dev/fd0 bs=512 seek=1
203+1 records in
203+1 records out
```

Se você tiver começado com um disquete MS-DOS formatado antes de fazer isso e agora tentar montar o disquete, o comando de montagem gerará um erro. A cópia de stage2 logo após o setor de boot do disquete (seek=1) destruiu o sistema de arquivos do disquete.

A criação de um CD inicializável GRUB é um pouco semelhante, embora você prepare a imagem do CD na sua unidade de disco rígido. Será necessário um diretório temporário, digamos grubcd, com subdiretórios boot e boot/grub. Em seguida, é necessário copiar o arquivo stage2_eltorito dos seus arquivos de distribuição GRUB para o subdiretório grub que você acabou de criar. Em seguida, use `genisoimage` para criar um arquivo de imagem inicializável .iso que possa ser gravado no CD com sua ferramenta de gravação favorita. A Listagem 10 mostra como criar a imagem do CD, como grubcd.iso.

Listagem 10. Criando uma imagem de CD inicializável GRUB

```
[ian@pinguino ~]# mkdir -p grubcd/boot/grub
[ian@pinguino ~]# cp /usr/share/grub/i386-redhat/stage2_eltorito grubcd/boot/grub
[ian@pinguino ~]# genisoimage -R -b boot/grub/stage2_eltorito -no-emul-boot \
> -boot-load-size 4 -boot-info-table -o grubcd.iso grubcd
I: -input-charset not specified, using utf-8 (detected in locale settings)
Size of boot image is 4 sectors -> No emulation
Total translation table size: 2048
Total rockridge attributes bytes: 760
Total directory bytes: 4576
Path table size(bytes): 34
Max brk space used 0
233 extents written (0 MB)
```

É possível inicializar o CD ou disquete em um PC arbitrário; não é necessário que seja um computador com sistema Linux. Se você fizer o boot agora desse disquete, observará o atraso enquanto ele carrega o estágio 2 do disquete. De forma semelhante, se você fizer o boot do CD, ele carregará o shell GRUB a partir do CD. Ao inicializar qualquer um deles, você receberá um prompt de boot de GRUB. Pressione a

tecla Tab para ver uma lista de comandos disponíveis. Tente `help commandname` para obter ajuda sobre o comando chamado *commandname*. A Listagem 11 mostra a linha de comando de GRUB.

Listagem 11. A linha de comando de GRUB

```
GNU GRUB version 0.97 (640K lower / 3072K 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.]

grub>
Possible commands are: blocklist boot cat chainloader clear cmp color configfile
debug device displaym displaymem dump embed find fstest geometry halt help
hide impsprobe initrd install ioprobe kernel lock makeactive map md5crypt
module modulenounzip pager partnew parttype password pause quit read reboot
root rootnoverify savedefault serial setkey setup terminal terminfo testload testvbe
unhide uppermem vbeprobe

grub> help rootnoverify
rootnoverify: rootnoverify [DEVICE [HDBIAS]]
  Similar to 'root', but don't attempt to mount the partition. This
  is useful for when an OS is outside of the area of the disk that
  GRUB can read, but setting the correct root device is still
  desired. Note that the items mentioned in 'root' which derived
  from attempting the mount will NOT work correctly.

grub> find /boot/grub/grub.conf
(hd0,1)
(hd0,5)

grub>
```

Neste exemplo, descobrimos que há arquivos de configuração GRUB em duas partições diferentes em nossa primeira unidade de disco rígido. Podemos carregar o menu GRUB de uma delas usando o comando `configfile (hd0,1)/boot/grub/grub.conf`

Isso carregará o menu a partir desse arquivo de configuração e poderemos fazer o boot no sistema a partir desse ponto. É possível explorar esses comandos de grub no manual GRUB. Tente digitar `info grub` em uma janela de terminal Linux para abrir o manual.

Um último ponto antes de deixarmos GRUB. Mencionamos que o arquivo de GRUB de estágio 2 destruiu o sistema de arquivos no disquete. Se quiser um disquete de recuperação de GRUB inicializável que carregue arquivos GRUB, incluindo um arquivo de configuração do disquete, você poderá fazer isso seguindo estas etapas:

1. Usar o comando `mkdosfs` para criar um sistema de arquivos DOS FAT no disquete e usar a opção `-R` a fim de reservar setores suficientes para o arquivo de estágio 2.
2. Montar o disquete
3. Criar um diretório `/boot/grub` no disquete
4. Copiar os arquivos `stage1`, `stage2` e `grub.conf` de GRUB para o diretório `boot/grub` no disquete. Copiar o arquivo de imagem inicial também se quiser um.
5. Editar seu arquivo `grub.conf` no disquete para que ele se refira ao arquivo inicial no disquete.
6. Desmontar o disquete
7. Usar o shell de comando `grub` para configurar o GRUB no disquete usando os comandos `root` e `setup` de GRUB.

Ilustramos isso na Listagem 12.

Listagem 12. Instalando o GRUB no disquete com um sistema de arquivos

```
[root@pinguino ~]# mkdosfs -R 220 /dev/fd0
mkdosfs 2.11 (12 Mar 2005)
[root@pinguino ~]# mkdir /mnt/floppy
[root@pinguino ~]# mount /dev/fd0 /mnt/floppy
[root@pinguino ~]# mkdir -p /mnt/floppy/boot/grub
[root@pinguino ~]# cp /boot/grub/stage1 /mnt/floppy/boot/grub
[root@pinguino ~]# cp /boot/grub/stage2 /mnt/floppy/boot/grub
```

```

[root@pinguino ~]# cp /boot/grub/splash* /mnt/floppy/boot/grub
[root@pinguino ~]# cp /boot/grub/grub.conf /mnt/floppy/boot/grub
[root@pinguino ~]# umount /dev/fd0
[root@pinguino ~]# grub
Probing devices to guess BIOS drives. This may take a long time.

GNU GRUB version 0.97 (640K lower / 3072K 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.]

grub> root (fd0)
Filesystem type is fat, using whole disk

grub> setup (fd0)
Checking if "/boot/grub/stage1" exists... yes
Checking if "/boot/grub/stage2" exists... yes
Checking if "/boot/grub/fat_stage1_5" exists... no
Running "install /boot/grub/stage1 (fd0) /boot/grub/stage2 p /boot/grub/grub.c
onf "... succeeded
Done.

```

Este último exemplo também ilustra o fato de que é possível executar o comando `grub` em um shell Linux. Essa é uma boa forma de aprender sobre o GRUB sem ter que reinicializar sua máquina o tempo todo. Será necessária propriedade de administrador para executar o `grub` e será possível fazer a maior parte das coisas que o GRUB faz, exceto realmente reinicializar o sistema.

GRUB 2

O GRUB 2 foi desenvolvido novamente a partir do zero para tornar-se significativamente mais modular e portátil. Ele é destinado a diferentes arquiteturas e métodos de boot e tem muitos novos recursos. Consulte [Recursos](#) para obter links para mais informações. Se você estiver acostumado ao GRUB e começar a usar o GRUB 2, descobrirá que ele é completamente diferente e, provavelmente, terá muitas surpresas.

Talvez a primeira coisa que você descubra sobre o GRUB 2 é que ele não é instalado como um loader de boot de partição. Embora eu tenha instruído o instalador do Ubuntu a instalar o GRUB na partição em que instalei o Ubuntu 9.10 em `(/dev/sda7)`, essa partição não é inicializável pelo carregamento em cadeia. O restante desta seção lhe oferecerá uma breve introdução ao GRUB 2 com informações suficientes para ajudá-lo a começar a usar o GRUB 2 de forma independente ou em conjunto com o GRUB Legacy.

A versão do GRUB 2 para PCs é distribuída no pacote `grub-pc`. Você verá diversos programas, normalmente em `/usr/bin` ou `/usr/sbin`, que fazem parte do pacote `grub-pc`, conforme mostrado na Listagem 13. No momento em que este artigo foi escrito, havia poucas man pages para esses programas. Tente executar os programas com a opção `--help` para obter mais informações, ou pesquise na Internet para obter ajuda adicional.

Listagem 13. Os executáveis do GRUB 2 em `/usr/bin` e `/usr/sbin`

```

ian@pinguino:~$ dpkg -L grub-pc|grep "bin/"
/usr/bin/grub-mkimage
/usr/bin/grub-mkrescue
/usr/sbin/grub-setup
/usr/sbin/update-grub2
/usr/sbin/update-grub
/usr/sbin/grub-install
/usr/sbin/upgrade-from-grub-legacy
/usr/sbin/grub-set-default
/usr/sbin/grub-reboot

```

O coração do GRUB 2 é formado por um kernel multiboot (`/boot/grub/core.img`) e um arquivo de configuração (`/boot/grub/grub.cfg`). Eles serão gerados para você se você executar `grub-install` e definir o destino como seu MBR (por exemplo: `grub-install /dev/sda`). Execute

`grub-install --help` conforme mostrado na Listagem 14 para ter uma ideia dos programas que são chamados para que todo o trabalho seja feito. Algumas coisas são semelhantes ao Grub Legacy, mas há vários itens novos, como `--modules`, `--grub-setup`, `--grub-mkimage`, e assim por diante.

Listagem 14. Ajuda do GRUB 2 para `grub-install`

```
ian@pinguino:~$ grub-install --help
Usage: grub-install [OPTION] install_device
Install GRUB on your drive.

-h, --help                print this message and exit
-v, --version             print the version information and exit
--modules=MODULES        pre-load specified modules MODULES
--root-directory=DIR     install GRUB images under the directory DIR
                        instead of the root directory
--grub-setup=FILE        use FILE as grub-setup
--grub-mkimage=FILE      use FILE as grub-mkimage
--grub-mkdevicemap=FILE  use FILE as grub-mkdevicemap
--grub-probe=FILE        use FILE as grub-probe
--no-floppy              do not probe any floppy drive
--recheck                probe a device map even if it already exists
--force                  install even if problems are detected
--disk-module=MODULE    disk module to use

INSTALL_DEVICE can be a GRUB device name or a system device filename.

grub-install copies GRUB images into the DIR/boot directory specified by
--root-directory, and uses grub-setup to install grub into the boot
sector.

Report bugs to <bug-grub@gnu.org>.
```

Então, se você executar `grub-install /dev/sda`, o processo construirá um arquivo de imagem principal para você, um arquivo de configuração e instalará o GRUB 2 em seu MBR. Se você não estiver pronto para se comprometer com o GRUB2 em toda a sua configuração, você mesmo pode construir essas partes e, em seguida, fazer o boot da imagem principal do GRUB 2 a partir do GRUB Legacy ou do LILO.

Construindo a imagem principal do GRUB 2

A forma mais fácil de construir um novo arquivo de imagem principal é executar o `grub-install` com a opção `--grub-setup=/bin/true`, que faz com que o `/bin/true` seja executado em vez do programa de configuração real do GRUB 2, como mostra a Listagem 15. Primeiro, removemos o `core.img` existente para mostrar que o arquivo é, de fato, gerado.

Listagem 15. Ajuda do GRUB 2 para `grub-install`

```
ian@pinguino:~$ sudo rm /boot/grub/core.img
ian@pinguino:~$ sudo grub-install --grub-setup=/bin/true /dev/sda
Installation finished. No error reported.
This is the contents of the device map /boot/grub/device.map.
Check if this is correct or not. If any of the lines is incorrect,
fix it and re-run the script `grub-install'.

(hd0) /dev/sda
ian@pinguino:~$ ls -l /boot/grub/core.img
-rw-r--r-- 1 root root 25105 2010-04-08 12:10 /boot/grub/core.img
```

Construindo o arquivo de configuração do GRUB 2

O arquivo de configuração do GRUB 2 geralmente é `/boot/grub/grub.cfg`. Diferentemente do GRUB Legacy, normalmente não é recomendável que você edite esse arquivo. É possível construí-lo usando o comando `grub-mkconfig` ou `update-grub`, que é um front-end para `grub-mkconfig`. Esses comandos procuram configurações gerais, como plano de fundo, tempos limite e assim por diante, em `/etc/default/grub`. Em seguida, os comandos executam executáveis de `/etc/grub.d/` para construir várias partes do arquivo de configuração, como o cabeçalho, uma seção da distribuição atual do Linux, seções de outros sistemas operacionais e suas próprias adições customizadas. Se for necessário customizar o menu do GRUB 2, é preciso adicionar suas alterações a um arquivo em `/etc/grub.d/`, como `40_custom`, ou adicionar seu próprio arquivo. Lembre-se de que ele precisa ser executável. Em seguida, execute

update-grub para gerar um novo arquivo /boot/grub/grub.cfg como mostra a Listagem 16.

Listagem 16. Construindo um arquivo de configuração do GRUB 2 com update-grub

```
ian@pinguino:~$ sudo update-grub
Generating grub.cfg ...
Found linux image: /boot/vmlinuz-2.6.31-20-generic
Found initrd image: /boot/initrd.img-2.6.31-20-generic
Found linux image: /boot/vmlinuz-2.6.31-14-generic
Found initrd image: /boot/initrd.img-2.6.31-14-generic
Found memtest86+ image: /boot/memtest86+.bin
Found GRUB boot loader on /dev/sda1
Found Fedora release 8 (Werewolf) on /dev/sda6
done
```

A Listagem 17 mostra algumas entradas do arquivo de configuração resultante. As linhas longas que foram quebradas para publicação foram indicadas usando uma barra invertida delimitadora (\). Observe que as sub-rotinas da entrada do menu se parecem um pouco mais com shell scripts do que os comandos simples sem lógica do GRUB Legacy. Outra importante mudança do GRUB Legacy é que a numeração da partição agora começa em 1, embora a numeração do disco ainda seja iniciada em 0. Por isso, /dev/sda7 é (hd0,7) no GRUB 2, onde seria (hd0,6) no GRUB Legacy.

Listagem 17. Arquivo de configuração parcial do GRUB 2

```
### BEGIN /etc/grub.d/05_debian_theme ###
set menu_color_normal=white/black
set menu_color_highlight=black/white
### END /etc/grub.d/05_debian_theme ###

### BEGIN /etc/grub.d/10_linux ###
menuentry "Ubuntu, Linux 2.6.31-20-generic" {
    recordfail=1
    if [ -n ${have_grubenv} ]; then save_env recordfail; fi
    set quiet=1
    insmod ext2
    set root=(hd0,7)
    search --no-floppy --fs-uuid --set 8954fa66-e11f-42dc-91f0-b4aa480fa103
    linux /boot/vmlinuz-2.6.31-20-generic \
        root=UUID=8954fa66-e11f-42dc-91f0-b4aa480fa103 ro quiet splash
    initrd /boot/initrd.img-2.6.31-20-generic
}
menuentry "Ubuntu, Linux 2.6.31-20-generic (recovery mode)" {
    recordfail=1
    if [ -n ${have_grubenv} ]; then save_env recordfail; fi
    insmod ext2
    set root=(hd0,7)
    search --no-floppy --fs-uuid --set 8954fa66-e11f-42dc-91f0-b4aa480fa103
    linux /boot/vmlinuz-2.6.31-20-generic
    root=UUID=8954fa66-e11f-42dc-91f0-b4aa480fa103 ro single
    initrd /boot/initrd.img-2.6.31-20-generic
}
```

Inicializando o GRUB 2 a partir do Grub Legacy

Mencionamos que não é possível carregar o GRUB 2 em cadeia a partir de um registro de boot de partição. No entanto, após construir sua imagem principal e o arquivo de configuração, é possível adicionar uma entrada ao arquivo grub.conf ou menu.lst do GRUB Legacy para fazer o boot da imagem principal do GRUB 2 que, em seguida, exibirá o menu do GRUB 2 que você construiu. A Listagem 18 mostra a entrada que já usamos na Listagem 8. Uma vez que o arquivo de configuração é para o GRUB Legacy, a entrada root especifica (hd0,6) para /dev/sda7.

Listagem 18. Entrada de configuração do Grub Legacy para inicializar a imagem principal do GRUB 2

```
# Ubuntu 9.10 uses GRUB 2, so use kernel to drive GRUB 2 menu.
title      Ubuntu 9.10 (sda7)
root       (hd0,6)
kernel     /boot/grub/core.img
```

Quando selecionar esta entrada no menu do seu GRUB Legacy, você verá um menu semelhante àquele na Figura 7, em que a primeira entrada é destacada e ela e a entrada seguinte são as duas ilustradas na Listagem 17.

Figura 7. Menu de exemplo do GRUB 2

```

GNU GRUB  version 1.97~beta4

Ubuntu, Linux 2.6.31-20-generic
Ubuntu, Linux 2.6.31-20-generic (recovery mode)
Ubuntu, Linux 2.6.31-14-generic
Ubuntu, Linux 2.6.31-14-generic (recovery mode)
Memory test (memtest86+)
Memory test (memtest86+, serial console 115200)
GRUB boot loader (on /dev/sda1)
Upgrade to Fedora 11 (Leonidas) (on /dev/sda6)
Fedora (2.6.26.8-57.fc8) (on /dev/sda6)
Fedora (2.6.26.6-49.fc8) (on /dev/sda6)

Use the + and - keys to select which entry is highlighted.
Press enter to boot the selected OS, 'e' to edit the
commands before booting or 'c' for a command-line.

```

Talvez você queira tentar instalar o GRUB 2 em um registro de boot de partição. Se você tentar, provavelmente verá uma saída semelhante à da Listagem 19.

Listagem 19. Tentando instalar o GRUB 2 em um registro de boot de partição

```

ian@pinguino:~$ sudo grub-install /dev/sda7
grub-setup: warn: Attempting to install GRUB to a partition instead of the MBR.
This is a BAD idea.
grub-setup: warn: Embedding is not possible. GRUB can only be installed in
this setup by using blocklists. However, blocklists are UNRELIABLE and its
use is discouraged.
Installation finished. No error reported.
This is the contents of the device map /boot/grub/device.map.
Check if this is correct or not. If any of the lines is incorrect,
fix it and re-run the script `grub-install'.

(hd0) /dev/sda

```

Isso explica por que minha tentativa de instalar o GRUB 2 na partição /dev/sda7 enquanto estava instalando o Ubuntu 9.10 falhou. Vale a pena observar que esta release do Ubuntu irá instalar o GRUB 2 para uma nova instalação, mas preservará o gerenciador de boot existente se você estiver atualizando uma instalação existente do Ubuntu.

Mídia de resgate para o GRUB 2

O GRUB 2 vem com um comando `grub-mkrescue` para ajudá-lo a criar um disquete ou imagem de CD de resgate. Consulte as páginas de ajuda ou as man pages para obter detalhes

Resumo

Agora, você aprendeu sobre os dois principais loaders de boot para sistemas Linux tradicionais, incluindo como recuperar-se de erros e problemas de boot. Você também teve uma introdução ao GRUB 2, que não está atualmente nos objetivos do LPI, mas está começando a aparecer na maior parte das distribuições Linux.

Download

Descrição	Nome	Tamanho
Configuration files used in this article	l-pic1-v3-102-2-samples.zip	8KB

Recursos

Aprender

Use o [Roteiro do developerWorks para LPIC-1](#) e localize os artigos do



Guias de capacitação
Se capacite através de diversos recursos de treinamento.

Programa IBM Champion

developerWorks que se destinam à preparação para a certificação LPIC-1, conforme os objetivos de abril de 2009.

No site [Programa LPIC](#), localize os objetivos detalhados, as listas de tarefas e as questões de amostra para os três níveis de certificação de administração de sistema Linux do Linux Professional Institute. Em particular, consulte os objetivos de abril de 2009 para o [exame LPI 101](#) e o [exame LPI 102](#). Sempre visite o site do Programa LPIC para obter os últimos objetivos.

Estude toda a [série de preparação para o exame LPI](#) no developerWorks a fim de conhecer os fundamentos do Linux e se preparar para a certificação de administrador do sistema com base nos objetivos para o exame LPI traçados antes de abril de 2009.

Aprenda sobre o GRUB Legacy e o GRUB 2 na [página inicial do GNU GRUB](#). GNU GRUB é um loader de boot multiboot derivado de GRUB, GRand Unified Bootloader.

Uma [Especificação Multiboot](#) de uma interface permite que loader de boot em conformidade com os padrões carregue qualquer sistema operacional em conformidade com os padrões.

Em "[Tarefas básicas para novos desenvolvedores Linux](#)" (developerWorks, março de 2005), descubra como abrir uma janela de terminal ou um prompt do shell, e muito mais.

O [Linux Documentation Project](#) apresenta uma variedade de documentos úteis, especialmente suas questões práticas.

Na [zona Linux do developerWorks](#), encontre centenas de [artigos e tutoriais](#), bem como downloads, fóruns de discussão e muitos outros recursos para desenvolvedores e administradores Linux.

Mantenha-se atualizado com os [eventos técnicos e webcasts do developerWorks](#) focados em uma variedade de produtos IBM e assuntos da indústria de TI.

Participe de uma [orientação gratuita do developerWorks Live!](#) para atualizar-se rapidamente sobre produtos e ferramentas da IBM, assim como tendências da indústria de TI.

Assista às [demos on-demand do developerWorks](#) que abrangem da instalação do produto e demos de configuração para iniciantes à funcionalidade avançada para desenvolvedores experientes.

Siga o [developerWorks no Twitter](#), ou inscreva-se em um feed dos [tweets do Linux no developerWorks](#).

Obter produtos e tecnologias

Faça o download do [System rescue CD-Rom](#), uma das diversas ferramentas disponíveis on-line para ajudá-lo a recuperar um sistema após um travamento.

[Avalie os produtos da IBM](#) da forma que melhor lhe convém: Faça o download de uma versão de teste de produto, experimente um produto on-line, use um produto em um ambiente de nuvem ou passe algumas horas



O programa reconhece contribuidores que estão ajudando a construir um Planeta Mais Inteligente.



Programa Global de Empreendedorismo da IBM

Faça parte do programa que busca por empreendedores que ajudam a modificar a maneira como o mundo funciona.

no [SOA Sandbox](#) aprendendo como implementar Arquitetura Orientada a Serviço de forma eficiente.

Discutir

[Participar do fórum de discussão.](#)

Envolve-se na [comunidade do My developerWorks](#). Entre em contato com outros usuários do developerWorks enquanto explora os blogs, fóruns, grupos e wikis dos desenvolvedores.