

Guia Foca GNU/Linux

Capítulo 6 - Gerenciadores de Partida (boot loaders)

Gerenciadores de Partida são programas que carregam um sistema operacional e/ou permitem escolher qual será iniciado. Normalmente estes programas são gravados no *setor de boot* (inicialização) da partição ativa ou no *master boot record* (MBR) do disco rígido.

Este capítulo explica o funcionamento de cada um dos principais gerenciadores de partida usados no GNU/Linux, em que situações é recomendado seu uso, as características, como configura-lo e alguns exemplos de configuração.

6.1 LILO

O LILO (*Linux Loader*) é sem dúvida o gerenciador de partida padrão para quem deseja iniciar o GNU/Linux através do disco rígido. Ele permite selecionar qual sistema operacional será iniciado (caso você possua mais de um) e funciona tanto em discos rígidos *IDE* como *SCSI*.

A seleção de qual sistema operacional e a passagem de parâmetros ao kernel pode ser feita automaticamente ou usando o aviso de boot : do LILO.

6.1.1 Criando o arquivo de configuração do LILO

Os dados para a criação do novo *setor de boot* que armazenará o gerenciador de partida são lidos do arquivo `/etc/lilo.conf`. Este arquivo pode ser criado em qualquer editor de textos (como o `ae` ou `vi`). Normalmente ele é criado durante a instalação de sua distribuição GNU/Linux mas por algum motivo pode ser preciso modifica-lo ou personaliza-lo (para incluir novos sistemas operacionais, mensagens, alterar o tempo de espera para a partida automática, etc).

O arquivo `/etc/lilo.conf` é dividido em duas seções: *Geral* e *Imagens*. A seção *Geral* vem no início do arquivo e contém opções que serão usadas na inicialização do LILO e parâmetros que serão passados ao kernel. A seção *Imagens* contém opções específicas identificando qual a partição que contém o sistema operacional, como será montado inicialmente o sistema de arquivos, tabela de partição, o arquivo que será carregado na memória para inicializar o sistema, etc. Abaixo um modelo do arquivo `/etc/lilo.conf` para sistemas que só possuem o GNU/Linux instalado:

```
boot=/dev/hda1
```

```
compact
```

```
install=text
```

```
map=/boot/map
```

```
vga=normal
```

```
delay=20
```

```
lba32
```

```
image=/vmlinuz
```

```
root=/dev/hda1
```

```
label=Linux
```

```
read-only
```

Para criar um novo gerenciador de partida através do arquivo `/etc/lilo.conf`, execute o comando `lilo`. No exemplo acima, o gerenciador de partida será instalado em `/dev/hda1` (veja [Identificação de discos e partições em sistemas Linux, Seção 5.12](#)), utilizará um setor de boot compacto (`compact`), modo de vídeo VGA normal (80x25), esperará 2 segundos antes de processar automaticamente a primeira seção `image=` e carregará o kernel `/vmlinuz` de `/dev/hda1`. Para detalhes sobre opções que podem ser usadas neste arquivo veja [Opções usadas no LILO, Seção 6.1.2](#).

Para mostrar o aviso de `boot :`, você deverá ligar as teclas Caps Lock ou Scrool lock na partida ou pressionar a tecla Shift durante os dois segundos de pausa. Outro método é incluir a opção `prompt` na seção *global* para que o aviso de `boot :` seja mostrado automaticamente após carregar o Lilo.

Abaixo uma configuração para computadores com mais de um sistema operacional (Usando GNU/Linux e DOS):

```
boot=/dev/hda1
```

```
compact
```

```
lba32
```

```
install=menu
```

```
map=/boot/map
```

```
vga=normal
```

```
delay=20
```

```
prompt
```

```
image=/vmlinuz
```

```
root=/dev/hda1
```

```
label=linux
```

```
read-only
```

```
other=/dev/hda2
```

```
table=/dev/hda
```

```
label=dos
```

O exemplo acima é idêntico ao anterior, o que foi acrescentado foi a opção `prompt` na seção *geral* (para que seja mostrado imediatamente o aviso de `boot`: no momento em que o LILO for carregado), e incluída uma imagem de disco DOS localizado em `/dev/hda2`. No momento da inicialização é mostrada a mensagem `boot`: e caso seja digitado `DOS` e pressionado ENTER, o sistema iniciará o DOS. Caso a tecla Enter seja pressionada sem especificar a imagem, a primeira será carregada (neste caso o GNU/Linux).

Você pode substituir a palavra `GNU/Linux` da opção `label` por o número 1 e `DOS` por 2, desta forma o número pode ser digitado para iniciar o sistema operacional. Isto é muito útil para construir um menu usando a opção `message`. Para detalhes veja [Opções usadas no LILO, Seção 6.1.2](#).

A seção *Geral* vem do início do arquivo até a palavra `delay=20`. A partir do primeiro aparecimento da palavra `image`, `other` ou `range`, tudo o que vier abaixo será interpretado como imagens de inicialização.

Por padrão, a imagem carregada é a especificada por `default=` ou a primeira que aparece no arquivo (caso `default=` não seja especificado). Para carregar o outro sistema (o DOS), digite o nome da imagem de disco no aviso de `boot`: (especificada em `label=`) que será carregada. Você também pode passar parâmetros manualmente ao kernel digitando o nome da imagem de disco e uma opção do kernel ou através do arquivo `/etc/lilo.conf` (veja [Opções usadas no LILO, Seção 6.1.2](#)).

O LILO pode inicializar o seguintes tipos de imagens:

- Imagens do kernel de um arquivo. Normalmente usado para iniciar o GNU/Linux pelo disco rígido e especificado pelo parâmetro `image=`.
- Imagens do kernel de um dispositivo de bloco (como um disquete). Neste caso o número de setores a serem lidos devem ser especificados na forma `PRIMEIRO-ÚLTIMO` ou `PRIMEIRO+NÚMERO de setores a serem lidos`. É necessário especificar o parâmetro `image=` e `range=`, por exemplo:

```
image=/dev/fd0
```

```
range=1+512
```

Todas as opções do kernel podem ser usadas na inicialização por dispositivo.

- O setor de boot de outro sistema operacional (como o DOS, OS/2, etc). O setor de partida é armazenado junto com a tabela de partição no arquivo `/boot/map`. É necessário especificar o parâmetro `OTHER=dispositivo` ou `OTHER=arquivo` e a inicialização através de um setor de partida possui algumas opções especiais como o `TABLE=` (para especificar a tabela de partição) e o `MAP-DRIVE=` (identificação da unidade de discos pelo sistema operacional). Veja o exemplo desta configuração abaixo:
- `other=/dev/hda2`
- `table=/dev/hda`

- `label=DOS`
- `map-drive=0x80`
- `to = 0x81`
- `map-drive=0x81`
- `to = 0x80`

Observações:

- Caso o gerenciador de partida seja instalado no MBR do disco rígido (`boot=/dev/hda`), o setor de boot do antigo sistema operacional será substituído, retire uma cópia do setor de boot para um disquete usando o comando `dd if=/dev/hda of=/floppy/mbr bs=512 count=1` no GNU/Linux para salvar o setor de boot em um disquete e `dd if=/floppy/mbr of=/dev/hda bs=446 count=1` para restaurá-lo. No DOS você pode usar o comando `fdisk /mbr` para criar um novo Master Boot Record.
- Após qualquer modificação no arquivo `/etc/lilo.conf`, o comando `lilo` deverá ser novamente executado para atualizar o setor de partida do disco rígido. Isto também é válido caso o kernel seja atualizado ou a partição que contém a imagem do kernel desfragmentada.
- A limitação de 1024 cilindros do Lilo não existe mais a partir da versão 21.4.3 (recomendada, por conter muitas correções) e superiores.
- A reinstalação, formatação de sistemas DOS e Windows pode substituir o setor de partida do HD e assim o gerenciador de partida, tornando impossível a inicialização do GNU/Linux. Antes de reinstalar o DOS ou Windows, verifique se possui um disquete de partida do GNU/Linux. Para gerar um novo boot loader, coloque o disquete na unidade e após o aviso `boot:` ser mostrado, digite `linux root=/dev/hda1` (no lugar de `/dev/hda1` você coloca a partição raiz do GNU/Linux), o sistema iniciará. Dentro do GNU/Linux, digite o comando `lilo` para gerar um novo setor de partida.

Agora reinicie o computador, tudo voltará ao normal.

6.1.2 Opções usadas no LILO

Esta seção traz opções úteis usadas no arquivo `lilo.conf` com explicações sobre o que cada uma faz. As opções estão divididas em duas partes: As usadas na seção *Global* e as da seção *Imagens* do arquivo `lilo.conf`.

Global

- `backup=[arquivo/dispositivo]` - Copia o setor de partida original para o arquivo ou dispositivo especificado.
- `boot=dispositivo` - Define o nome do dispositivo onde será gravado o setor de partida do LILO (normalmente é usada a partição ativa ou o Master Boot Record - MBR). Caso não seja especificado, o dispositivo montado como a partição raiz será usado.
- `compact` - Tenta agrupar requisições de leitura para setores seguintes ao sendo lido. Isto reduz o tempo de inicialização e deixa o mapa menor. É normalmente recomendado em disquetes.

- `default=imagem` - Usa a imagem especificada como padrão ao invés da primeira encontrada no arquivo `lilo.conf`.
 - `delay=[num]` - Permite ajustar o número de segundos (em décimos de segundos) que o gerenciador de partida deve aguardar para carregar a primeira imagem de disco (ou a especificada por `default=`). Esta pausa lhe permite selecionar que sistema operacional será carregado.
 - `install=interface` - Especifica que interface será usada para exibição de menu com as opções de inicialização ao utilizador. As seguintes opções são permitidas:
 - `text` - Exibe uma mensagem de texto (exibida através do parâmetro `message=`) na tela. Esta é a recomendada para terminais.
 - `menu` - Exibe um menu que lhe permite selecionar através de uma interface de menu a opção de inicialização. Esta é a padrão.
 - `bmp` - Exibe um bitmap gráfico com a resolução de 640x480 com 16 ou 256 cores.
 - `lba32` - Permite que o LILO quebre o limite de 1024 cilindros do disco rígido, inicializando o GNU/Linux em um cilindro acima deste através do acesso . Note que isto requer compatibilidade com o BIOS, mais especificamente que tenha suporte a chamadas int 0x13 e AH=0x42. É recomendado o seu uso.
 - `map=arquivo-mapa` - Especifica a localização do arquivo de mapa (`.map`). Se não for especificado, `/boot/map` é usado.
 - `message=arquivo` - Especifica um arquivo que contém uma mensagem que será mostrada antes do aviso de `boot :`. Nenhuma mensagem é mostrada até que seja pressionada a tecla `Shift` após mostrar a palavra `LILO`. O tamanho da mensagem deve ser no máximo 65535 bytes. O arquivo de mapa deve ser novamente criado caso a mensagem seja retirada ou modificada. Na mensagem, o caracter `FF` (`CTRL+L`) limpa a tela.
 - `nowarn` - Não mostra mensagens de alerta.
 - `password=senha` - Permite proteger todas as imagens de disco com uma única senha. Caso a senha esteja incorreta, o LILO é novamente carregado.
 - `prompt` - Mostra imediatamente o aviso de `boot :` ao invés de mostrar somente quando a tecla `Shift` é pressionada.
 - `verbose=[num]` - Ativa mensagens sobre o processamento do LILO. Os números podem ser especificados de 1 a 5, quanto maior o número, maior a quantidade de detalhes mostrados.
 - `timeout=[num]` - Ajusta o tempo máximo de espera (em décimos de segundos) de digitação no teclado. Se nenhuma tecla é pressionada no tempo especificado, a primeira imagem é automaticamente carregada. Igualmente a digitação de senha é interrompida se o utilizador estiver inativo por este período.
- Adicionalmente as opções de imagem do kernel `append`, `ramdisk`, `read-only`, `read-write`, `root` e `vga` podem ser especificadas na seção *global*. Opções por Imagem
- As opções por imagem iniciam com uma das seguintes opções: `image=`, `other=` ou `range=`. Opções usadas por cada imagem:
- `table=dispositivo` - Indica o dispositivo que contém a tabela de partição para aquele dispositivo. Necessário apenas para imagens especificadas por `other=`.
 - `unsafe` - Não acessa o setor de boot no momento da criação do mapa. Isto desativa algumas checagens, como a checagem da tabela de partição. `unsafe` e `table=` são incompatíveis.
 - `label=[nome]` - Permite especificar um nome para a imagem. Este nome será usado na linha `boot :` para inicializar o sistema.

- `alias=[nome]` - Apelido para a imagem de disco. É como um segundo `label`.
 - `optional` - Ignora a imagem caso não estiver disponível no momento da criação do mapa. É útil para especificar kernels que não estão sempre presentes no sistema.
 - `password=senha` - Protege a imagem atual com a senha. Caso a senha esteja incorreta, o setor de partida do Lilo é novamente carregado.
 - `restricted` - A senha somente é pedida para iniciar a imagem se o sistema for iniciado no modo `single`.
- Também podem ser usados parâmetros de inicialização do kernel no arquivo `/etc/lilo.conf`, veja a seção [Parâmetros de inicialização passados ao kernel, Seção 6.3](#) para maiores detalhes.
-

6.1.3 Um exemplo do arquivo de configuração `lilo.conf`

Abaixo um exemplo do arquivo `/etc/lilo.conf` que poderá ser usado em instalações GNU/Linux com o DOS.

```
boot=/dev/hda1 #Instala o LILO em /dev/hda1

compact

install=menu

map=/boot/map

message=/etc/lilo.message #mensagem que será mostrada na tela

default=1 #Carrega a Imagem especificada por label=1 como padrão

vga=normal #usa o modo de video 80x25 ao iniciar o Linux

delay=20 #aguarda 2 segundos antes de iniciar a imagem padrão

lba32 #permite quebrar o limite de 1024 cilindros na inicialização

prompt #mostra o aviso de "boot:" logo que o LILO é carregado

image=/vmlinuz #especifica o arquivo que contém a primeira imagem

root=/dev/hda1 #partição onde a imagem acima esta localizada

label=1 #identificação da imagem de disco

read-only #monta inicialmente como somente leitura

password=12345 #Usa a senha 12345

restricted #somente quando iniciar com o parâmetro single

other=/dev/hda2 #especifica outro sistema que será carregado
```

```
table=/dev/hda #a tabela de partição dele está em /dev/hda
```

```
label=2 #identificação desta imagem de disco
```

```
password=12345 #pede a senha antes de iniciar este sistema
```

Você pode usar o exemplo acima como base para construir sua própria configuração personalizada do `/etc/lilo.conf` mas não se esqueça de modificar as tabelas de partições para seu sistema. Se você usa o Windows NT 4.0, Windows NT 5.0 (Windows 2000) ou o OS/2, recomendo ler `ODOS+Windows+OS/2-HOWTO`.

Após criar seu arquivo `/etc/lilo.conf`, execute o comando `lilo` e se tudo ocorrer bem, o LILO será instalado.

6.2 GRUB

(Os detalhes contidos na seção sobre o GRUB, foram integralmente desenvolvidos por Alexandre Costa alebyte@bol.com.br como contribuição ao guia FOCA GNU/Linux.)

O GRUB (*Grand Unified Boot Loader*) é mais uma alternativa como gerenciador de boot e apresenta alguns recursos extras com relação as outras opções disponíveis. Ele é flexível, funcional e poderoso, podendo inicializar sistemas operacionais como o Windows (9x, ME, NT, 2000 e XP), Dos, Linux, GNU Hurd, *BSD, OS/2 e etc. Podemos destacar também o suporte aos sistemas de arquivos ext2 (Linux), ext3 e reiserfs (novos sistemas de arquivos journaling do Linux), FAT16 e FAT32 (Win 9x/ME), FFS (Fast File System usado no *BSD), minix (MINIX OS) e etc.

Por utilizar o padrão Multiboot ele é capaz de carregar diversas imagens de boot e módulos. Por esse motivo ele é o único gerenciador de inicialização capaz de carregar o conjunto de servidores do GNU Hurd. O GRUB também permite buscar imagens do kernel pela rede, por cabo seriais, suporta discos rígidos IDE e SCSI, detecta toda a memória RAM disponível no sistema, tem interface voltada para linha de comandos ou menus de escolha, além de suportar sistemas sem discos e terminais remotos.

Como possui inúmeros recursos, será apresentada sua utilização básica, ficando como sugestão ao leitor procurar se aprofundar mais em suas possibilidades de uso e configuração.

6.2.1 Como o GRUB trabalha com discos e partições

O GRUB trabalha com uma notação diferente para apontar discos e partições sendo necessário algumas explicações antes de prosseguir. Veja a tabela comparativa:

No Linux No GRUB

`/dev/hda` (`hd0`)

`/dev/hda1` (`hd0,0`)

```
/dev/hda2 (hd0,1)

/dev/hdb (hd1)

/dev/hdb1 (hd1,0)

/dev/hdb2 (hd1,1)

/dev/sda (hd0) # Disco SCSI ID 0

/dev/sda1 (hd0,0) # Disco SCSI ID 0, partição 1

/dev/sda2 (hd0,1) # Disco SCSI ID 0, partição 2

/dev/sdb (hd1) # Disco SCSI ID 1

/dev/sdb1 (hd1,0) # Disco SCSI ID 1, partição 1

/dev/sdb2 (hd1,1) # Disco SCSI ID 1, partição 2

/dev/fd0 (fd0)
```

OBS: Os discos *IDE* e *SCSI* são referenciados ambos como (hd?) pelo GRUB. Não há distinção entre os discos e de modo geral a identificação de unidades IDE é menor do que qualquer tipo de drive SCSI, salvo se você alterar a sequência de inicialização (boot) na BIOS.

Para saber como o Linux trabalha com partições veja [Identificação de discos e partições em sistemas Linux, Seção 5.12](#).

6.2.2 Instalando o GRUB

A instalação do GRUB ao contrário da instalação do LILO ([LILO, Seção 6.1](#)), só precisa ser executada uma única vez. Caso seja necessária alguma mudança como por exemplo adicionar uma nova imagem, esta pode ser feita apenas editando o arquivo de configuração `menu.lst`.

6.2.2.1 No MBR

Um método simples de adicionar o GRUB para gerenciar seu MBR (*Master Boot Record*) é rodando o seguinte comando (como superutilizador):

```
# /sbin/grub-install /dev/hda
```

Este comando grava o GRUB no MBR do primeiro disco e cria o diretório `/boot/grub` onde estarão os arquivos necessários para o seu funcionamento. Neste ponto o GRUB já está instalado e quando você reiniciar seu computador irá se deparar com uma linha de comandos, onde terá que carregar a imagem do kernel manualmente. Mais adiante será explorada a utilização desta linha de comando que é muito eficiente.

Provavelmente você achará mais interessante copiar o arquivo de configuração de exemplos do GRUB e otimizá-lo às suas necessidades. Note que isto não exclui a possibilidade de utilizar a linha de comando, apenas cria uma interface de menus onde você pode configurar várias opções de boot de uma forma organizada, automatizada e funcional. Copie este arquivo para o diretório `/boot/grub` com o seguinte comando:

```
# cp /usr/share/doc/grub/examples/menu.lst /boot/grub
```

Por ser um arquivo de exemplos será necessário otimizá-lo de acordo com suas necessidades, o que será abordado mais a frente.

6.2.3 No disco flexível (somente linha de comando)

Quando criamos um disquete de partida, este funcionará em um sistema qualquer, podendo utilizar este disquete em várias máquinas diferentes ou em uma máquina em que tenha tido algum problema com o GRUB no MBR. Coloque um disquete virgem e digite os seguintes comandos:

```
# dd if=/usr/lib/grub/i386-pc/stage1 of=/dev/fd0 count=1
```

```
# dd if=/usr/lib/grub/i386-pc/stage2 of=/dev/fd0 seek=1
```

Estes comandos permitem que seja apresentada a linha de comando do grub quando este disco for utilizado para boot.

6.2.4 No disco flexível (com interface de menu)

Quando foi criado o disquete de partida anteriormente, este só nos permitia utilizar a linha de comando sendo necessário carregar o `menu.lst` pelo disco rígido (o qual deve estar presente). Em alguns casos este disco satisfaz as necessidades básicas mas pode haver um momento em que você deseje ter um disquete que funcione com vários sistema e não dependa de um disco fixo.

Digite os seguintes comandos:

```
# mke2fs /dev/fd0
```

```
# mount /dev/fd0 /floppy -t ext2
```

```
# mkdir /floppy/grub
```

```
# cp /usr/lib/grub/i386-pc/stage[12] /floppy/grub
```

```
# cp /usr/share/doc/grub/examples/menu.lst /floppy/grub
```

```
# umount /floppy
```

```
# /sbin/grub
```

Este último comando disponibiliza a linha de comando do GRUB. Digite os seguintes comandos:

```
grub> install (fd0)/grub/stage1 d (fd0) (fd0)/grub/stage2 p (fd0)/grub/menu.lst  
  
grub> quit
```

Neste momento o disquete está pronto. Note que o `menu.lst` que foi copiado para ele é um arquivo de exemplo, sendo necessário que você o configure de acordo com suas necessidades.

6.2.5 Opções do arquivo de configuração

Esta seção descreve o arquivo `menu.lst` com explicações sobre as opções mais usadas. Este arquivo é dividido em parâmetros Globais, que afetam o arquivo todo e parâmetros que só tem efeito para as imagens do sistema que será carregado. Algumas opções podem ser passadas para o kernel do Linux no momento do boot, algumas delas também serão detalhadas.

Parâmetros Globais

- `timeout` = Define um tempo (em segundos) de espera. Se nenhuma tecla for pressionada, carrega a imagem padrão.
- `default` = Define qual será a opção padrão que deve ser automaticamente selecionada quando nenhuma outra for especificada em um tempo definido por `timeout`.
- `fallback` = Caso ocorra algum erro inesperado e a opção padrão não possa ser carregada, este parâmetro define qual a outra opção deve ser utilizada.
- `color` = Permite que você escolha as cores usadas no menu de boot.
- `password` = Permite que você especifique uma senha. Está será solicitada sempre que houver necessidade de realizar uma função que não seja carregar as imagens disponíveis, como por exemplo acessar a linha de comandos do GRUB. Você pode utilizar também o parâmetro `password` para esconder um arquivo que contenha outras configurações, como um `menu.lst` secreto. O arquivo pode ter um nome qualquer.
- Ex.: `password = senha (hd0,0)/boot/grub/secret.conf`

Você pode ter várias entradas do parâmetro "password" em um mesmo arquivo sendo que uma delas é usada para bloquear o acesso as imagens/linha de comandos e as outras usadas para carregar arquivos de opções do GRUB. Quando você digitar `p` para entrar com a senha, você pode digitar a senha que protege as imagens/linha de comandos ou a que é utilizada para carregar os arquivos de opções.

- `hiddenmenu` = Está opção faz com que o menu de opções não seja mostrado e de boot na imagem especificada por "default" depois de expirado o tempo definido em `timeout`. O utilizador pode requisitar o menu com as opções pressionando a tecla <ESC> antes que o tempo definido em `timeout` expire.

Parâmetros que afetam apenas as imagens

- `title` = Define um texto que será apresentado no menu de boot para identificar o sistema a ser inicializado.
- `root` = Determina qual a partição raiz do sistema a ser inicializada.

- `rootnoverify` = Idêntica a opção `root`, mas não tenta montar a partição-alvo, o que é necessário para alguns sistemas como Dos e Windows.
- `kernel` = Nesta opção você informa qual o kernel vai ser inicializado. Você pode passar parâmetros diretamente para o kernel também.
- Ex.: `kernel (hd0,0)/boot/vmlinuz-2.4.16 vga=6`
- `module` = Faz com que algum módulo necessário para o boot seja carregado. Lembre-se que estes não são módulos do kernel (módulos de som, rede, etc.) e sim módulos necessários ao boot de alguns sistemas, como por exemplo o GNU Hurd.
- `lock` = Quando você quiser controlar se uma pessoa pode iniciar um sistema que esteja listado nas opções do menu de boot, você pode utilizar esta opção que faz com que a senha especificada com o comando "password" seja solicitada no momento em que se tentar carregar a imagem em questão.
- `pause` = Emite uma mensagem na tela e espera uma tecla ser pressionada.
- `makeactive` = Torna a partição ativa. Este comando está limitado a partições primárias dos discos.
- `chainloader` = Alguns sistemas como o Windows ou Dos armazenam seu próprio gerenciador de boot no início da partição em que ele está instalado. Para efetuar o boot destes sistemas através do GRUB, você precisa pedir para que o gerenciador de boot de tal sistema seja carregado e faça seu trabalho, dando o boot.
- `hide` e `unhide` = Esconde e mostra partição respectivamente. Estas duas opções são necessárias quando houver mais de uma versão do Dos ou Windows na máquina em partições diferentes, já que estes sistemas detectam automaticamente a partição e quase sempre o fazem de modo errado. Suponha o Windows na primeira partição primária (hd0,0) e o Dos na segunda partição primária (hd0,1). Quando quisermos carregar estes sistemas devemos proceder da seguinte maneira:
- `title Windows`
- `hide (hd0,1)`
- `unhide (hd0,0)`
- `rootnoverify (hd0,0)`
- `chainloader +1`
- `makeactive`
- `title Dos`
- `hide (hd0,0)`
- `unhide (hd0,1)`
- `rootnoverify (hd0,1)`
- `chainloader +1`

- `makeactive`
- `map` = Alguns sistemas não permitem ser inicializados quando não estão no primeiro disco (Dos, Win 9x, etc.). Para resolver esta e outras situações o GRUB tem um comando que permite enganar tal sistema mapeando as unidades de disco do modo como lhe for mais conveniente.

Imagine que você tenha o primeiro disco (hd0) com o GNU/Linux instalado e em um outro disco (hd1) com o Windows/Dos instalado. O Windows/Dos não permitem serem inicializados desta forma e como solução você poderia usar a seguinte entrada no arquivo de configurações do GRUB:

```
title Windows

unhide (hd1,0)

rootnoverify (hd1,0)

chainloader +1

map (hd1) (hd0)

makeactive
```

Isso faz com que o disco (hd1), onde esta o Windows/Dos, seja apresentado a este sistema como (hd0) "enganado" o mesmo e possibilitando o boot.

Parâmetros enviados diretamente ao kernel

Pode ser necessário passar alguns parâmetros para o kernel no momento do boot. Para maiores informações ver a seção [Parâmetros de inicialização passados ao kernel, Seção 6.3](#). Você pode passar os parâmetros da seguinte maneira:

```
# Exemplo de entrada no 'menu.lst'.

title Linux 2.4.16

root (hd0,0)

kernel (hd0,0)/boot/vmlinuz-2.4.16 vga=6 mem=512M ramdisk=0
```

Neste exemplo, a linha com o comando "kernel" é usada para indicar qual imagem deve ser carregada. As opções que seguem (vga, mem e ramdisk) são parâmetros que devem ser passados diretamente ao kernel do sistema a ser carregado.

6.2.6 Um exemplo de arquivo de configuração

```
# Exemplo de arquivo de configuração do GRUB.
```

```
# Note que você pode usar o caracter '#' para fazer comentários.

# Se após 30 segundos nenhuma tecla for pressionada, carrega a imagem padrão.

timeout 30

# Define a primeira imagem como padrão.

default 0

# Caso a imagem padrão não funcione carrega a imagem definida aqui.

fallback 1

# Define as cores que serão usadas no menu.

color light-cyan/black white/blue

# Permite utilizar uma senha.

password minha-senha-secreta

password minha-senha (hd0,0)/boot/grub/secret.conf

# Para boot com o GNU/Hurd

title GNU/Hurd

root (hd0,0)

kernel /boot/gnumach.gz root=hd0s1

module /boot/serverboot.gz

# Para boot com o GNU/Linux

title Linux 2.4.16

# Pedir a senha configurada em "password" antes de carregar esta imagem.

lock

root (hd0,0)

# Atente as opções passadas diretamente para o kernel (vga, mem, etc.).

kernel (hd0,0)/boot/vmlinuz-2.4.16 vga=6 mem=512M ramdisk=0

# Para boot com o Mach (obtendo o kernel de um disquete)

title Utah Mach4 multiboot
```

```
root (hd0,2)

pause Insira o disquete agora!!!

kernel (fd0)/boot/kernel root=hd0s3

module (fd0)/boot/bootstrap

# Para boot com FreeBSD

title FreeBSD 3.4

root (hd0,2,a)

kernel /boot/loader

# Para boot com OS/2

title OS/2

root (hd0,1)

makeactive

chainloader +1

chainloader /boot/chain.os2

# Para boot com Windows 9x, ME, NT, 2000, XP.

title Windows 9x, ME, NT, 2000, XP

unhide (hd0,0)

rootnoverify (hd0,0)

chainloader +1

makeactive

# Para instalar o GRUB no disco rígido.

title = Instala o GRUB no disco rígido

root = (hd0,0)

setup = (hd0)

# Muda as cores.

title Mudar as cores
```

6.2.7 Usando a linha de comandos do GRUB

O GRUB possui inúmeros recursos, mas com certeza um dos mais importantes e que merece destaque é sua linha de comandos. A maioria dos comandos usados no arquivo de configuração `menu.lst` são válidos aqui e muitos outros estão disponíveis. Uma breve apresentação da linha de comandos será dada, ficando por conta do leitor se aprofundar o quanto achar necessário em sua flexibilidade.

Quando o GRUB é inicializado você pode se deparar com sua linha de comandos ou se possuir o arquivo `menu.lst` configurado, um menu de escolha. Mesmo usando os menus de escolha você pode utilizar a linha de comandos, bastando para isso seguir as instruções no rodapé da tela onde o GRUB nos informa que podemos digitar `e` para editar as entradas de boot ou `c` para ter acesso a linha de comandos (lembre-se que pressionar `<ESC>` faz com que você volte aos menus de escolha).

Caso a opção `password` tenha sido especificada no arquivo `menu.lst`, será necessário antes de acessar as outras opções (que estarão desabilitadas) pressionar `p` e entrar com a senha correta.

Agora, com acesso a linha de comandos, você pode verificar os comandos disponíveis pressionando duas vezes a tecla `<TAB>`. Note que você também pode utilizar esta tecla para completar nomes de comandos bem como parâmetros de alguns comandos.

Alguns comandos disponíveis:

- `cat` = Este comando permite verificar o conteúdo de um arquivo qualquer, o qual deve estar gravado em um dispositivo ligado a sua máquina. Embora seja um recurso útil, nenhuma permissão de acesso é verificada e qualquer pessoa que tenha acesso a linha de comandos do GRUB pode listar o conteúdo de arquivos importantes. Para contornar este problema o parâmetro `password` é utilizado no arquivo `menu.lst` e faz com que uma senha seja solicitada antes de liberar o acesso a linha de comandos. Não esqueça que ainda é possível utilizar um disquete com o GRUB para dar boot na máquina o que permite usar a linha de comandos pelo disquete.
- Ex.: `grub> cat (hd0,0)/etc/passwd`
- `cmp` = Este comando é utilizado para comparar dois arquivos.
- Ex.: `grub> cmp (hd0,0)/arquivo1 (hd0,0)/arquivo2`
- `configfile` = Carrega um arquivo de configuração do GRUB.
- Ex.: `grub> configfile (hd0,0)/boot/grub/menu.lst`
- `displayapm` = Mostra informações sobre APM.
- `displaymem` = Mostra informações sobre a memória RAM.
- `find` = Permite encontrar um arquivo. A saída deste comando disponibiliza o nome completo do caminho para o arquivo e a partição onde o mesmo está localizado.

- `Ex.: grub> find stage1`
- `geometry` = Mostra informações sobre a geometria reconhecida de seu drive e permite que você escolha a geometria desejada caso esta esteja sendo reconhecida erroneamente.
- `help` = `help "comando"` para ver a ajuda.
- `Ex.: help color`
- `install` = Instala o GRUB, embora não seja recomendado o uso deste comando diretamente, pois é possível esquecer ou trocar facilmente um parâmetro e sobrescrever a tabela de partições de seu disco.
- `Ex.: install (fd0)/grub/stage1 d (fd0) (fd0)/grub/stage2 p (fd0)/grub/menu.lst`
- `setup` = Você pode usar este comando para instalar o GRUB. Note que sua sintaxe é menos complexa do que a usada em `install`.
- `Ex.:`
- `grub> root = (hd0,0)`
- `grub> setup = (hd0)`
- `quit` = Abandona a linha de comandos do GRUB.
- `reboot` = Reinicia o computador.
- `boot` = Efetua o boot. Suponha o Linux instalado em `(hd0,0)`, podemos passar os seguintes comandos na linha de comandos para efetuar o boot de uma imagem do GNU/Linux:
- `grub> root (hd0,0)`
- `grub> kernel (hd0,0)/boot/vmlinuz-2.4.16 vga=6`
- `grub> boot`

Muitos outros comandos estão disponíveis tanto na linha de comandos do GRUB quanto no arquivo de configuração `menu.lst`. Estes comandos adicionais podem ser necessários apenas para algumas pessoas e por isso não serão explicados.

6.2.8 Removendo o GRUB do MBR

Não existe a necessidade de se remover o GRUB do MBR pois não há utilização para o mesmo vazio. Para substituir o GRUB do MBR é necessário apenas que outro gerenciador de boot escreva algo nele. Você pode seguir o procedimento de instalação do LILO para escrever algo no MBR ou usar o comando `fdisk /mbr` do DOS.

6.2.9 Como obter informações mais detalhadas

Para obter informações mais detalhadas sobre o GRUB é recomendado o site oficial do mesmo, o qual está disponível apenas na língua inglesa. Os seguintes sites foram utilizados na pesquisa:

- Site oficial do GRUB: <http://www.gnu.org/software/grub/>
 - Site Debian-br (<http://debian-br.cipsga.org.br/>), na parte de suporte, documentação, "Como usar o GRUB: Um guia rápido para usar o GRUB, feito por Vitor Silva Souza e Gustavo Noronha Silva".
-

6.3 Parâmetros de inicialização passados ao kernel

Abaixo algumas das opções mais usadas para passar parâmetros de inicialização de hardware/características ao kernel.

- `append=string` - Passa os parâmetros especificados ao kernel. É extremamente útil para passar parâmetros de hardwares que podem ter problemas na hora da detecção ou para parâmetros que precisam ser passados constantemente ao kernel através do aviso `boot:`.
Exemplo: `append="mem=32m"`
- `ramdisk=tamanho` - Especifica o tamanho do disco RAM que será criado. Caso for igual a zero, nenhum disco RAM será criado. Se não for especificado, o tamanho do disco RAM usado na imagem de inicialização do kernel será usada.
- `read-only` - Especifica que o sistema de arquivos raiz deverá ser montado como somente leitura. Normalmente o sistema de inicialização remonta o sistema de arquivos como leitura/gravação.
- `read-write` - Especifica que o sistema de arquivos raiz deverá ser montado como leitura e gravação.
- `root=dispositivo` - Especifica o dispositivo que será montado como raiz. Se a palavra `current` é usada, o dispositivo atual será montado como raiz.
- `vga=modo` - Especifica o modo de vídeo texto que será usado durante a inicialização.
- `normal` - Usa o modo 80x25 (80 colunas por 25 linhas)
- `extended` (ou `ext`) - Usa o modo de texto 80x50
- `ask` - Pergunta que modo de vídeo usar na inicialização. Os modos de vídeo podem ser obtidos pressionando-se `enter` quando o sistema perguntar o modo de vídeo.

Uma lista mais detalhada de parâmetros de inicialização pode ser obtida no documento `Boot-prompt-howto` (veja [Documentos HOWTO's, Seção 31.8](#)).

6.4 LOADLIN

É um gerenciador de partida que permite iniciar o GNU/Linux a partir do DOS. A vantagem do uso do Loadlin é não ser preciso reiniciar o computador para se entrar no GNU/Linux. Ele funciona carregando o kernel (copiado para a partição DOS) para a memória e inicializando o GNU/Linux.

Outro motivo pelo qual é muito usado é quando o GNU/Linux não tem suporte a um certo tipo de dispositivo, mas este tem seu suporte no DOS ou Windows e funciona corretamente com eles.

O truque é o seguinte: Você inicia normalmente pelo DOS e após seu dispositivo ser configurado corretamente pelo driver do DOS e funcionando corretamente, você executa o Loadlin e o GNU/Linux assim poderá usa-lo.

Muitos usam o comando `Loadlin` dentro do arquivo `autoexec.bat` para iniciar o GNU/Linux automaticamente após o dispositivo ser configurado pelo DOS.

ATENÇÃO!!! Não execute o `Loadlin` dentro do Windows.

6.4.1 Opções do LOADLIN

Abaixo a lista de opções que podem ser usadas com o programa `LOADLIN` (note que todas são usadas no DOS):

`loadlin [imagem_kernel] [argumentos] [opções]`

- `imagem_kernel` - Arquivo que contém o kernel.
 - `root=dispositivo` - Especifica o dispositivo que contém o sistema de arquivos raiz. É especificado de acordo com a identificação de dispositivos no GNU/Linux (`/dev/hda1`, `/dev/hdb1`, etc).
 - `ro` - Diz ao kernel para montar inicialmente o sistema de arquivos raiz como somente leitura. Os scripts de inicialização normalmente modificam o sistema de arquivos para leitura e gravação após sua checagem.
 - `rw` - Diz ao kernel para montar inicialmente o sistema de arquivos raiz como leitura e gravação.
 - `initrd=[NUM]` - Define o tamanho do disco RAM usado no sistema.
 - `-v` - Mostra detalhes sobre mensagens e configuração
 - `-t` - Modo de teste, tudo é feito menos a inicialização do GNU/Linux.
 - `-d arquivo` - Mesma função de `-t`, mas envia a saída para o arquivo
 - `-txmode` - Altera o modo de vídeo para 80x25 antes de inicializar o kernel.
 - `-diskreset` - Após carregar a imagem do kernel, reseta todos os discos rígidos antes de inicializar o GNU/Linux.
-

6.4.2 Exemplo de inicialização com o LOADLIN

Abaixo você encontra um exemplo do comando `loadlin` que poderá ser usado em sua instalação GNU/Linux (precisando apenas ajustar a localização da partição raiz do GNU/Linux de acordo com seu sistema).

```
C:\> LOADLIN vmlinuz root=/dev/hda1 ro
```

```
| | |
```

```
| | +- Montar como somente leitura
```

```
| |
```

```
| +- Partição raiz
```

```
|
```

```
+- Nome do kernel copiado para o DOS
```

6.5 syslinux

Outro gerenciador de partida que funciona somente com sistemas de arquivos DOS. A principal diferença do `syslinux` em relação ao `LOADLIN` é que foi feito especialmente para funcionar em disquetes formatados no DOS, facilitando a instalação do GNU/Linux e para a criação de disquetes de recuperação ou de inicialização. Um disquete gerado pelo `syslinux` é lido sem problemas pelo DOS/Windows.

```
syslinux [-s] [dispositivo]
```

A opção `-s` instala no disquete uma versão segura, lenta e estúpida do `syslinux`. Isto é necessário para algumas BIOS problemáticas.

6.5.1 Criando um disquete de inicialização com o syslinux

Siga os passos abaixo para criar um disquete de inicialização com o `syslinux`:

1. Formate o disquete no DOS ou com alguma ferramenta GNU/Linux que faça a formatação de disquetes para serem usados no DOS.
2. Copie um ou mais arquivos de `kernel` para o disquete
3. Digite `syslinux /dev/fd0` (lembre-se de usar a opção `-s` se tiver problemas de inicialização). Este comando modificará o setor de partida do disquete e gravará um arquivo chamado `LDLINUX.SYS` no diretório raiz do disquete.

Lembre-se: O disquete deve estar desmontado antes de usar o comando `syslinux`, caso o disquete estiver montado uma mensagem será mostrada e `osyslinux` abortado.

Por padrão é carregado o kernel de nome `GNU/Linux`. Este padrão pode ser modificado através do arquivo de configuração `SYSLINUX.CFG` que também é gravado no diretório raiz do disquete. Veja [O arquivo SYSLINUX.CFG, Seção 6.5.2](#) para detalhes.

Se as teclas Caps Lock ou Scrool Lock estiverem ligadas ou Shift, Alt forem pressionadas durante o carregamento do `syslinux`, o `syslinux` mostrará um aviso de `boot` : no estilo do `LILO`. O utilizador pode então digitar o nome do kernel seguido de qualquer parâmetro para inicializar o GNU/Linux.

6.5.2 O arquivo SYSLINUX.CFG

Este arquivo é criado no diretório raiz da unidade de disquete e contém as opções que serão usadas para modificar o funcionamento do `syslinux`. Abaixo a listagem de opções que podem ser especificadas neste arquivo:

`default [kernel] [opções]`

Indica o nome do kernel e as opções dele que serão usadas na inicialização, caso `syslinux` seja iniciado automaticamente. Caso não for especificada, o valor assumido será `linux auto` sem nenhuma opção de inicialização.

`append [opções]`

Passa uma ou mais opções ao kernel na inicialização. Elas serão adicionadas automaticamente para inicializações automáticas e manuais do `syslinux`.

`label [nome]`

`kernel [kernel]`

`append [opções]`

Nome que identificará o kernel no aviso de `boot :` (idêntica a opção `label=` do `LILO`). Se a imagem especificada por `nome` for selecionada, o kernel usado será o especificado pelo parâmetro `kernel` e as opções usadas por `append`.

Caso seja passado um hífen – ao parâmetro `append`, os parâmetros passados pelo `append` global serão anulados.

`implicit [valor]`

Se o `[valor]` for igual a 0, não carrega a imagem até que seja explicitamente especificada na opção `label`.

`timeout [tempo]`

Indica quanto tempo o `syslinux` aguardará antes de inicializar automaticamente (medido em 1/10 de segundos). Caso alguma tecla seja pressionada, a inicialização automática é interrompida. Para desativar esta característica, use 0 como `timeout`. O valor máximo é de 35996.

`font [nome]`

Especifica uma fonte (em formato `.psf`) que será usada para mostrar as mensagens do `syslinux` (após o aviso de copyright do programa). Ele carrega a fonte para a placa de vídeo, se a fonte conter uma tabela unicode, ela será ignorada. Somente funciona em placas EGA e VGA.

`kbdmap [mapa]`

Instala um simples mapa de teclado. O mapa de teclados usado é muito simples: somente remapeia códigos conhecidos pela BIOS, o que significa que somente teclas usadas no teclado padrão EUA serão usadas.

O utilitário `keytab-lilo.pl` da distribuição do `lilo` pode ser usado para criar tais mapas de teclado.

`prompt [valor]`

Se `[valor]` for igual a 1, mostra automaticamente o aviso de `boot :` assim que o `syslinux` for iniciado. Caso seja igual a 0, mostra o aviso de `boot :` somente se as teclas Shift ou Alt forem pressionadas ou Caps Lock e Scrool Lock estiverem ativadas.

`display [arquivo]`

Mostra o conteúdo do `[arquivo]` durante a inicialização do `syslinux`.

`F1 [arquivo]`

`F2 [arquivo]`

...

`F0 [arquivo]`

Especifica que arquivos serão mostrados quando as teclas de F1 até F10 forem pressionadas. Para detalhes, veja [Formatação dos arquivos de tela do syslinux, Seção 6.5.3](#).

6.5.3 Formatação dos arquivos de tela do syslinux

Os arquivos de texto que são mostrados na tela pelo `syslinux` podem ter suas cores modificadas usando parâmetros simples, isto causa um bom efeito de apresentação. Abaixo estão os códigos que podem ser usados para criar um arquivo texto que será exibido pelo `syslinux`:

`CTRL+L` - Limpa a tela (semelhante ao que o `clear` faz).

`CTRL+O[frente][fundo]` - Define a cor de frente e fundo, se somente uma cor for especificada, esta será assumida como frente.

Veja os valores para `[frente]` e `[fundo]` abaixo:

00 - preto 08 - cinza escuro

01 - azul escuro 09 - azul claro

02 - verde escuro 0a - verde claro

03 - ciano escuro 0b - ciano claro

04 - vermelho escuro 0c - vermelho claro

05 - purple escuro 0d - purple claro

06 - marrom 0e - amarelo

07 - cinza claro 0f - branco

`CTRL+Z` - Equivalente ao fim de arquivo no DOS

O código padrão usado é o 07. Escolhendo uma cor clara para o fundo (08-0f) resultará em uma cor piscante correspondente para a texto (00-07).